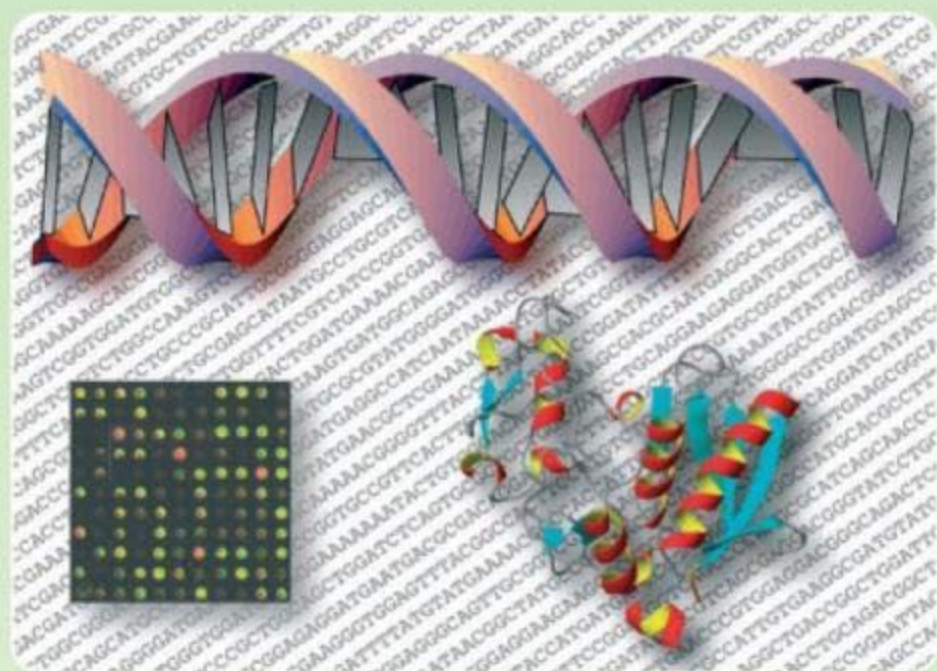


Frédéric Dardel and François Képès

Bioinformatics

Genomics and post-genomics

Translated by Noah Hardy



 **WILEY**

Bioinformatics

Genomics and post-genomics

Frédéric Dardel and François Képès

translated by
Noah Hardy

This work has been published with the help of the French Ministère de la
Culture – Centre national du livre



John Wiley & Sons, Ltd

Bioinformatics

Bioinformatics

Genomics and post-genomics

Frédéric Dardel and François Képès

translated by
Noah Hardy

This work has been published with the help of the French Ministère de la
Culture – Centre national du livre



John Wiley & Sons, Ltd

First published in French as *Bioinformatique. Génomique et post-génomique* © 2002 École Polytechnique

Translated into English by Noah Hardy

English language translation copyright © 2006 John Wiley & Sons Ltd
The Atrium, Southern Gate, Chichester,
West Sussex PO19 8SQ, England
Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk

Visit our Home Page on www.wileyeurope.com or www.wiley.com

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except under the terms of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London W1T 4LP, UK, without the permission in writing of the Publisher. Requests to the Publisher should be addressed to the Permissions Department, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, or emailed to permreq@wiley.co.uk, or faxed to (+44) 1243 770620.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The Publisher is not associated with any product or vendor mentioned in this book.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 6045 Freemont Blvd, Mississauga, Ontario, Canada L5R 413

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data

Bioinformatique. English

Bioinformatics : genomics and post-genomics / edited by Frédéric Dardel and François Képès; translated into English by Noah Hardy.

p. cm.

Includes bibliographical references and index.

ISBN-13: 978-0-470-02001-2 (cloth : alk. paper)

ISBN-10: 0-470-02001-6 (cloth : alk. paper)

1. Bioinformatics. 2. Genomics. I. Dardel, Frédéric. II. Képès, François. III. Title.

QH324.2.B558 2006

572.80285 – dc22

2006011225

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN-13 978-0-470-02001-2 ISBN 0-470-02001-6

Typeset in 10½/12½pt Sabon by SNP Best-set Typesetter Ltd., Hong Kong

Printed and bound in Great Britain by Antony Rowe Ltd., Chippenham, Wilts

This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

Contents

Preface to the French edition	vii
Preface to the English edition	ix
1 Genome sequencing	1
1.1 Automatic sequencing	1
1.2 Sequencing strategies	4
1.3 Fragmentation strategies	8
1.4 Sequence assembly	12
1.5 Filling gaps	14
1.6 Obstacles to reconstruction	16
1.7 Utilizing a complementary 'large' clone library	18
1.8 The first large-scale sequencing project: The <i>Haemophilus influenzae</i> genome	19
1.9 cDNA and EST	20
2 Sequence comparisons	25
2.1 Introduction: Comparison as a sequence prediction method	25
2.2 A sample molecule: the human androsterone receptor	26
2.3 Sequence homologies – functional homologies	27
2.4 Comparison matrices	28
2.5 The problem of insertions and deletions	33
2.6 Optimal alignment: the dynamic programming method	34
2.7 Fast heuristic methods	38
2.8 Sensitivity, specificity, and confidence level	46
2.9 Multiple alignments	50
3 Comparative genomics	61
3.1 General properties of genomes	61
3.2 Genome comparisons	67
3.3 Gene evolution and phylogeny: applications to annotation	75
4 Genetic information and biological sequences	85
4.1 Introduction: Coding levels	85
4.2 Genes and the genetic code	85
4.3 Expression signals	87

4.4	Specific sites	91
4.5	Sites located on DNA	91
4.6	Sites present on RNA	96
4.7	Pattern detection methods	96
5	Statistics and sequences	107
5.1	Introduction	107
5.2	Nucleotide base and amino acid distribution	107
5.3	The biological basis of codon bias	112
5.4	Using statistical bias for prediction	113
5.5	Modeling DNA sequences	116
5.6	Complex models	120
5.7	Sequencing errors and hidden Markov models	123
5.8	Hidden Markov processes: a general sequence analysis tool	127
5.9	The search for genes – a difficult art	127
6	Structure prediction	131
6.1	The structure of RNA	131
6.2	Properties of the RNA molecule	132
6.3	Secondary RNA structures	134
6.4	Thermodynamic stability of RNA structures	138
6.5	Finding the most stable structure	144
6.6	Validation of predicted secondary structures	149
6.7	Using chemical and enzymatic probing to analyze folding	150
6.8	Long-distance interactions and three-dimensional structure prediction	152
6.9	Protein structure	155
6.10	Secondary structure prediction	158
6.11	Three-dimensional modeling based on homologous protein structure	161
6.12	Predicting folding	166
7	Transcriptome and proteome: macromolecular networks	169
7.1	Introduction	169
7.2	Post-genomic methods	170
7.3	Macromolecular networks	182
7.4	Topology of macromolecular networks	193
7.5	Modularity and dynamics of macromolecular networks	199
7.6	Inference of regulatory networks	206
8	Simulation of biological processes in the genome context	211
8.1	Types of simulations	213
8.2	Prediction and explanation	213
8.3	Simulation of molecular networks	215
8.4	Generic post-genomic simulators	226
	Index	233

Preface to the French edition

This book is directly based on a course that we teach at the *École Polytechnique*. We thank all our colleagues and friends there who have made its existence and publication possible:

Sylvain Blanquet, Chairman of the Biology Department, who first thought of creating this interdisciplinary course some ten years ago, when such a project was highly innovative. He has followed and supported the development of the project in the context of the genomic revolution.

Jean-Marc Steyaert, our colleague at the Computer Science Department, where he initiated the teaching of bioinformatics, and in which he remains active. His critical attention, constant theoretical and methodological contributions, and increasing involvement in biological problematics, have contributed in an essential way to bringing this book about, as well as influencing its contents.

Philippe Dessen, who participated in some of the very early stages in the teaching of bioinformatics at the *École Polytechnique* while he was there; our contacts with him over the years have been invaluable.

Finally, going from the stage of course-notes to a published book would not have been possible without the decisive contributions of Jean-Paul Coard, of *Éditions de l'École Polytechnique*, as well of those of Jean-Claude Mathieu, Véronique Lecointe, Martine Maguer, and Frédéric Zantonio, involved in the technical production.

Frédéric Dardel
François Képès

Preface to the English edition

At the suggestion of Vincent Schachter, to whom we are very grateful, we decided to produce an English version of our book. We have worked closely with Noah Hardy, to ensure the accuracy of the translation, and have updated all chapters with new material where necessary. Chapter 8 was rewritten entirely in English. We hope that this edition will enable many more readers to enjoy our book. We would like to thank Joan Marsh and her colleagues at Wiley for their help in producing this edition.

Frédéric Dardel
François Képès

1

Genome sequencing

1.1 Automatic sequencing

The dideoxyribonucleotide method, developed during the early 1980s in England, at the Cambridge University laboratory of Fred Sanger, is today universally employed to sequence DNA fragments. It is based on the use of DNA polymerase to elongate a single strand of DNA, starting from a primer, utilizing another DNA strand as the template. The DNA polymerase elongates the strand in the presence of four deoxyribonucleotide monomers (dATP, dTTP, dGTP, and dCTP) and a dideoxyribonucleotide analog (ddNTP), which acts as the chain terminator (Figure 1.1). Specific incorporation of the analog by DNA polymerase yields a mixture of fragments that selectively terminate at positions corresponding to each nucleotide (As, in the example below).

The principle of the dideoxyribonucleotide ('dideoxy') method is illustrated in Figure 1.2. Four parallel reactions are carried out, one with each ddNTP, the DNA fragments obtained being separated by electrophoresis. A fluorescent tracer is used to identify fragments synthesized by the polymerase so as to distinguish them from template DNA. The tracer is attached to one extremity of the fragment, either at the 5'-end of the sequencing primer or at the 3'-end of the dideoxynucleotide terminator. Modern automatic sequencers utilize an *in situ* detection system during electrophoresis, in which a laser beam emitting in the fluorophore absorption spectrum is passed through the gel (Figure 1.3). A migrating DNA fragment in the path of the laser beam then emits a fluorescent signal detected by a photodiode on the other side of the gel. The signal is amplified and transmitted to a computer programmed with special software for analyzing it.

Under favorable conditions, this technique permits reading up to 1,000 nucleotides per sequenced fragment, and an average of **500 to 800 nucleotides during routine experiments**. Two dideoxy methodologies coexist at present: one employs a single photophore, and the other uses four, each with a distinct emission spectrum. In the first system, the four mixtures, corresponding to the four ddNTPs, are introduced into different electrophoresis gel wells. Analysis is

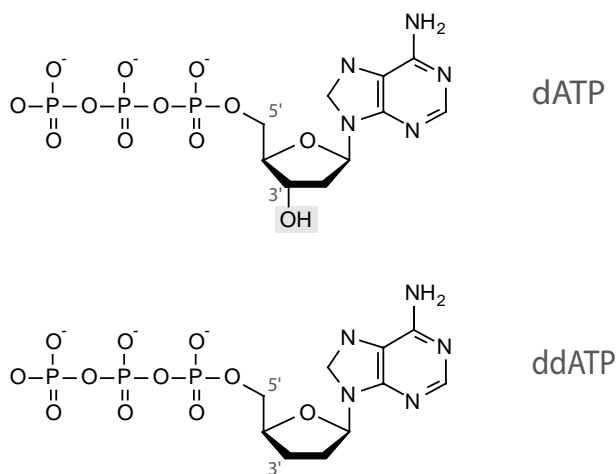


Figure 1.1 Dideoxynucleotide structure

Replacement of the 3'-OH group in the dideoxynucleotide (ddNTP) by a 3'-H group prevents formation of a phosphodiester link at its 3'-end. The modified nucleotides have a normal 5'-triphosphate side, thus may be incorporated into the chain by DNA polymerase. Since A-T and G-C pairing rules are followed during ddNTP incorporation, the ddATP will be incorporated wherever there is a T facing it on the template strand.

accomplished by comparing the migration rates of the fragments in the four resulting lanes.

In the second system, each of the four sequencing reactions uses a different fluorophore that modifies the corresponding ddNTP. After the four polymerization reactions have taken place, the resulting DNA fragments are mixed and introduced into the same gel well. Constituent nucleotides are identified according to the emission properties of the fluorescent tracers exposed to the laser beam using selective color filters, after which a single gel lane is analyzed.

The four-fluorophore technique is a bit more expensive, since it requires somewhat more varied chemistry. However, it has the advantage of being better adapted to high-throughput systems, since more samples are analyzed on the same gel. In the latest generation of automatic sequencers, the classical rectangular polyacrylamide gel is replaced by a reusable capillary tube, whereas the separation and detection principles remain unchanged. This technique reduces the time required for an experiment from several hours to a few minutes, also minimizing preparation time. In principle, the highest-performance multi-capillary machines can process up to 1,000 samples per day, equivalent to 0.5 Mbases of raw sequence per day per machine.

Massive high-throughput sequencing centers today often use several dozen such machines with robots that control the sequencing reactions automatically executing pipetting, mixing, and incubation steps, thereby minimizing the risk

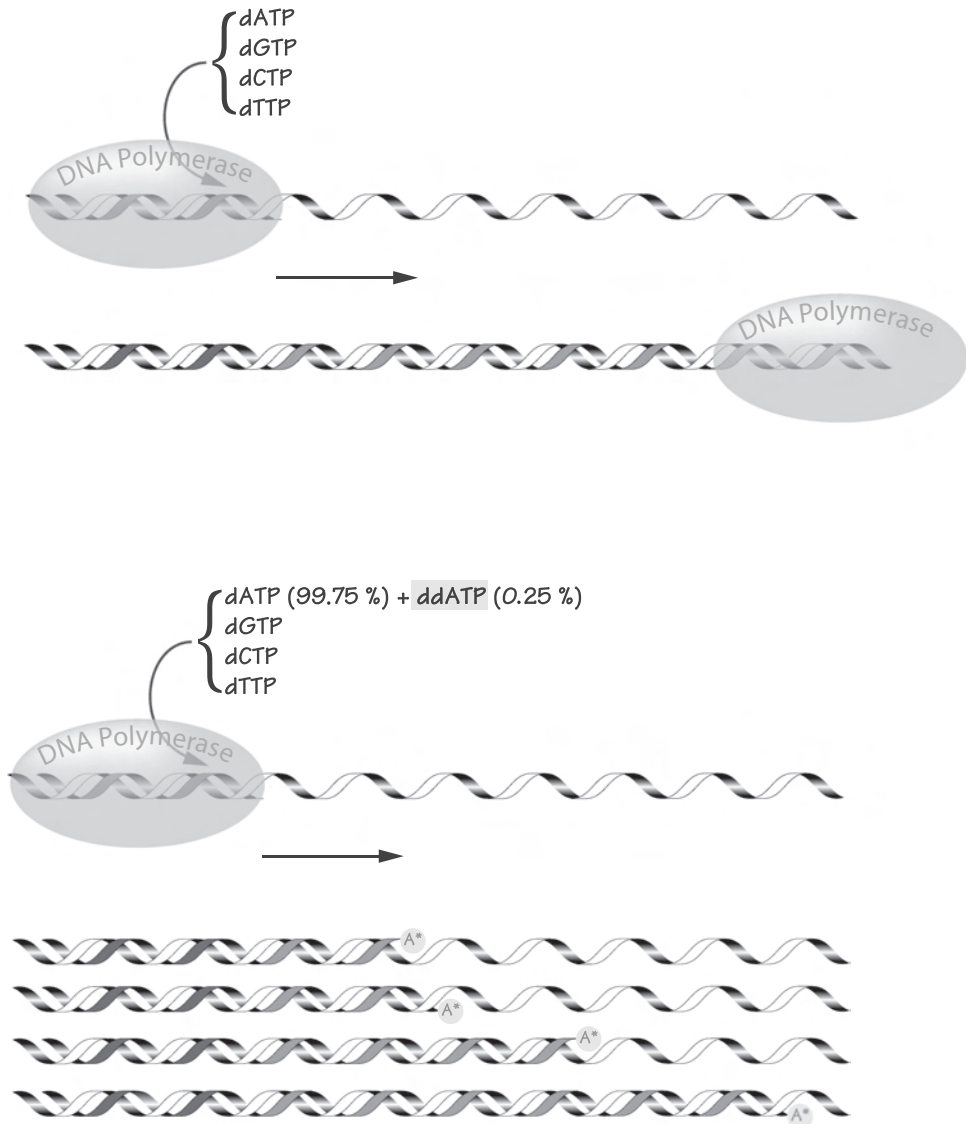


Figure 1.2 Principle of the Sanger sequencing method

In the presence of a template DNA strand and the four dNTPs, DNA polymerase can elongate a complementary DNA strand starting from an **oligonucleotide primer**, which hybridizes to the template strand. When a dideoxynucleotide is incorporated by the polymerase, it acts as a chain terminator, blocking further elongation. This incorporation is entirely random, proceeding at a rate that is a function of the ratio of the dideoxynucleotide concentration to that of the corresponding deoxynucleotide (here it is $[ddATP] / [dATP] = 1 / 400$).

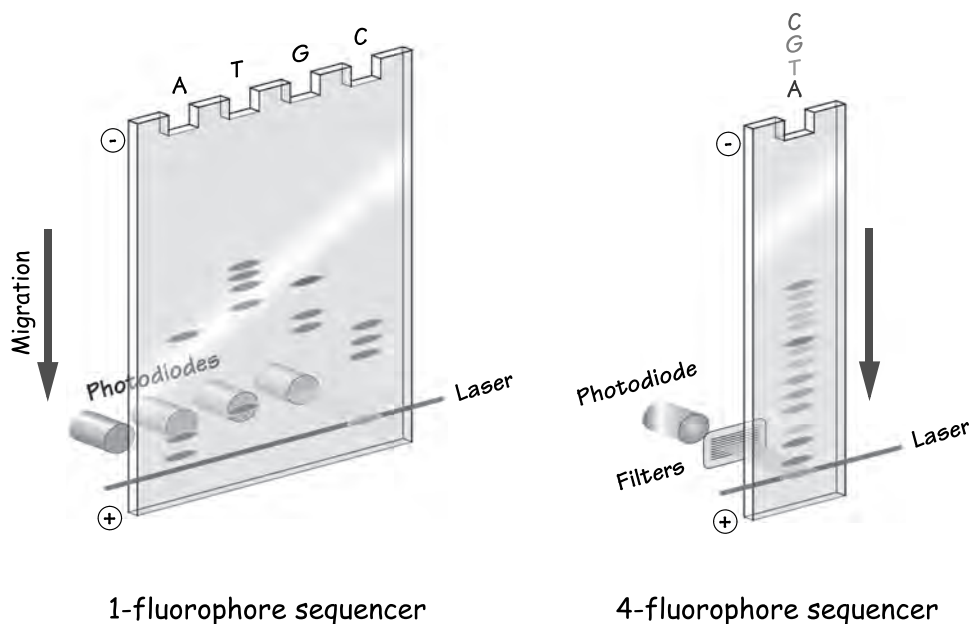


Figure 1.3 Automatic sequencing using 1- and 4-fluorophore sequencers
 Samples introduced into the wells (top) are separated by electrophoresis on a polyacrylamide-urea gel. The 5'-CAATCCGGATGTTT sequence is read from bottom to top.

of human error (see Figure 1.4). The preparation of DNA templates remains the most difficult step to automate, although significant progress has been made in this respect.

1.2 Sequencing strategies

The sequencing methodologies described above fail to address major difficulties that need to be considered when operating a large-scale sequencing program:

- Only DNA fragments of between 500 and 1,000 nucleotides may be sequenced;
- A sequencing primer that is complementary to the template is required for the DNA polymerase to begin synthesizing.

Fortunately, these two obstacles may be simultaneously overcome by fragmenting the DNA that is to be sequenced into segments of size compatible to

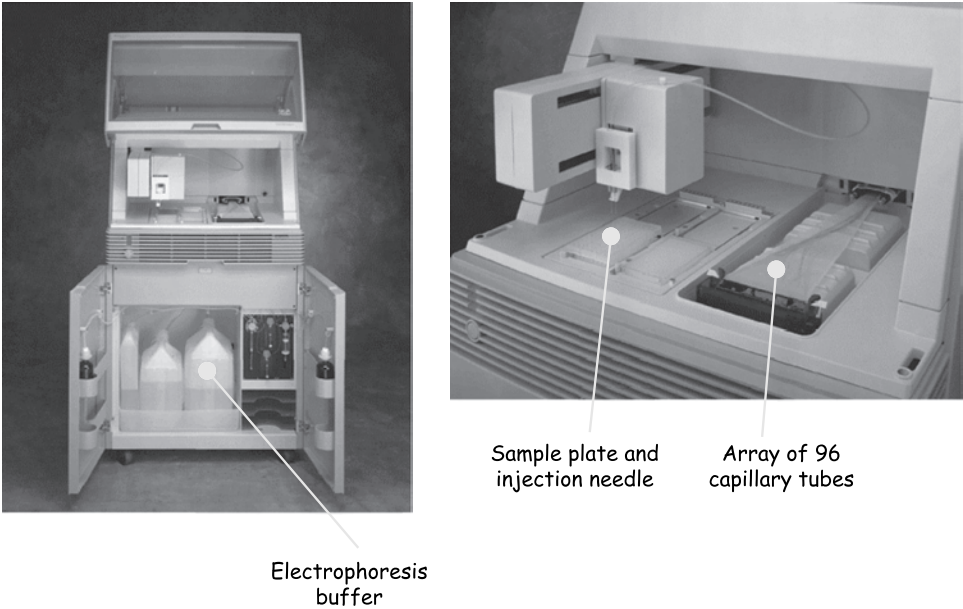


Figure 1.4 Advanced automatic multicapillary DNA sequencer for simultaneous sequencing of 96 samples. An automatic injection system executes several consecutive separations without manual intervention (®Applied Biosystems).

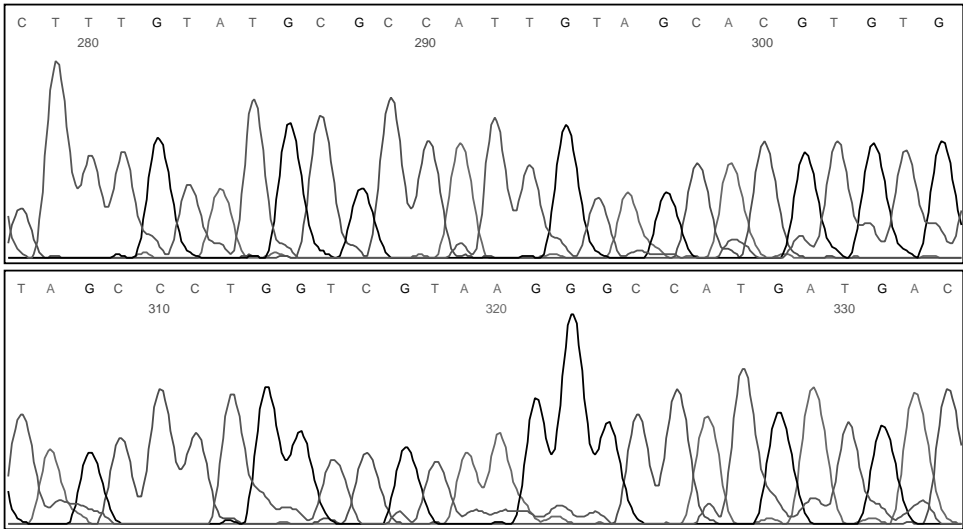


Figure 1.5 Example of a sequencing profile
Intensity of the signal detected by the photodiodes as a function of separation time. Each color is associated with one of the four separation reactions (A, G, C, and T).

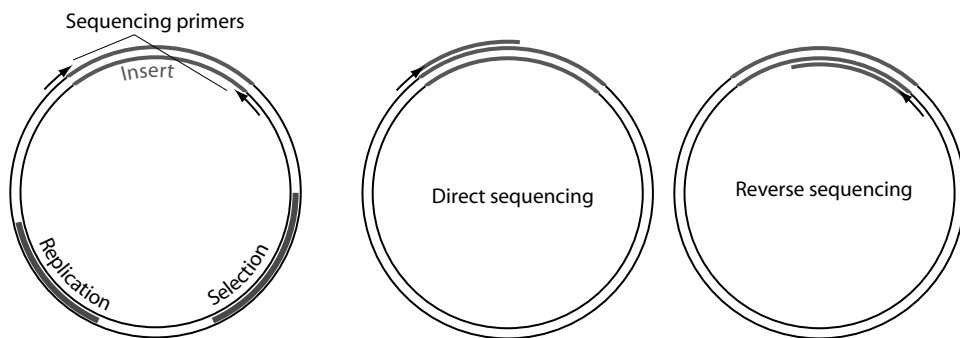


Figure 1.6 Sequencing in a vector starting with universal primers.

that of the sequencing system yield ($\sim 10^3$ base pairs) and by inserting them into an appropriate vector (plasmid or virus). The vector is selected according to several criteria:

- It must be able to replicate autonomously in a convenient host cell (usually *Escherichia coli*);
- It must bear one or several gene markers that permit selection of cells that contain it (antibiotic resistance, for example);
- Its nucleotide sequence must be known;
- It must contain restriction endonuclease sites that permit cloning by insertion of foreign DNA fragments.

In practice, small bacterial plasmids are generally used. The DNA to be sequenced is fragmented and ligated into the vector, which is then propagated in host cells. The clone cell lines (derived from a single initial cell by successive division), each containing a different recombinant vector with the same inserted DNA fragment are then isolated. A library of DNA fragments may thus be constituted by collecting a large number of these clone cell lines, and used for further study (see Figure 1.7).

In order to determine the DNA sequence of such a fragment, the corresponding cell line is cultured and its DNA extracted for sequencing by the dideoxynucleotide method. Since the nucleotide sequences located on each side of the vector clone site are known, they are used as the primer (see Figure 1.6). These primers are independent of the DNA inserted into the vector and may be used to sequence any fragment; they are therefore called *universal primers*. Because

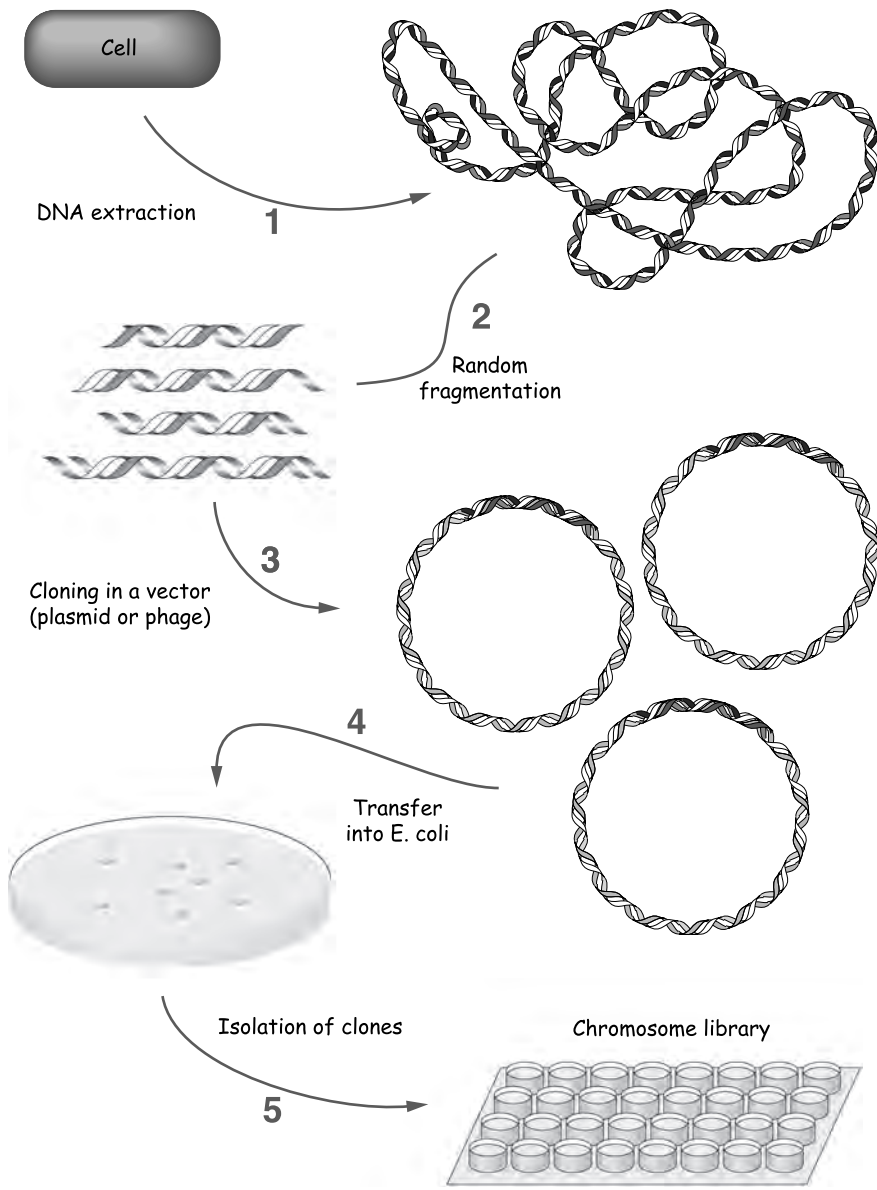


Figure 1.7 Strategy for constructing a DNA library.

such primers are constant, it is very easy to incorporate fluorescent tracers needed during oligonucleotide synthesis into them. The fluorescent primers thus produced may be used in most sequencing procedures.

1.3 Fragmentation strategies

In sequencing a long stretch of DNA – especially a complete genome – it is essential that it be cut into fragments of a size compatible with the sequencing technology. This poses two additional questions:

- Which cutting strategy should be employed?
- How can the complete sequence be reconstituted from the pieces?

These two questions are intricately related, since the reassembly method is sensitively dependent on how the fragmentation is accomplished. Two approaches have mainly been used: **random fragmentation** and **segmentation after mapping**.

Random fragmentation

In random fragmentation, the full length of the DNA to be sequenced is cut into small pieces of optimal sequencing size (~1,000 base pairs). A high cutting frequency (one site per 200–250 bp) restriction enzyme may be used for this purpose, under conditions of partial digestion (10–20 percent) in order to generate 1,000- to 2,000-bp fragments. Alternatively, ultrasound may be used to break the DNA into small pieces, since the mechanical constraints induced by ultrasonic vibrations in DNA in solution are sufficient to rupture the long phosphodiester chain.

The mechanical (ultrasound) method results in more random breaks than the enzymatic method but necessitates an additional step to repair the extremities of the DNA fragments produced, since breaks produced by ultrasound treatment do not occur at the same level in the two DNA strands. This may require paring the extended extremities of single strands, so that the resulting fragments may be inserted into blunt cloning sites in the sequencing vector.

The basic postulate of the random or *shotgun* method is that if enough clones are analyzed, the entire original DNA sequence will be covered. Assuming that fragmentation and cloning are really random processes and that the DNA sequence is sufficiently large compared with that of individual clones (which is generally the case for a full genome), the probability that a given DNA nucleotide studied **not be covered** by random sequencing is a Poisson distribution:

$$p_0 = e^{-N/L},$$

where N is the total number of nucleotides sequenced in the set of clones and L the total length of the DNA studied. N/L is the coverage rate, which is the rate of data redundancy. In order to obtain a 99 percent sequencing rate, that is, $p_0 = 0.01$, it is necessary to sequence a number of clones equal to 4.6 times ($\log 0.01 \approx -4.6$) the length of the DNA studied.

In the case of a genome or a very long DNA fragment, it is thus practically inevitable that gaps remain in the sequence, which must be filled using some approach other than the random *shotgun* method. It is also possible to statistically evaluate the length and average number of such gaps:

$$\text{Total length of gaps} = Le^{-N/L}$$

$$\text{Average length of each gap} = Ln/N$$

$$\text{Number of gaps} = N/ne^{-N/L},$$

where n is the average length of each fragment sequenced (~ 500 nucleotides). The following is an example of the results for a bacterial genome ($L \approx 10^6$ bp) and for the genome of a higher organism, such as a mammal or a plant ($L \approx 10^9$ bp) with a coverage rate of factor 6 (an average value for this type of project), which yields 99.75 percent of the sequenced nucleotides:

The random approach raises two important points:

- It is impossible to cover the entire genome without greatly increasing the number of clones to be sequenced; to be nearly certain of covering the entire bacterial genome in the above example would require that $p_0 \ll 10^{-6}$, at least 14 times the coverage rate. From the practical point of view, it is more economical to accept a coverage rate of between 4 and 6 and then fill the few dozen remaining gaps *ad hoc* (see Table 1.1 below).
- Assembly of the puzzle of the set of fragments may require systematic side-by-side comparison of all the sequences obtained. For k sequences, this

Table 1.1

	Bacteria (1 Mbp)	Mammals (1 Gbp)
Number of sequences	12,000	12,000,000
Number of remaining gaps	30	29,750
Average gap size	200	200

amounts to $k(k - 1)/2$, which is around 10^8 comparisons for a bacterial genome and 10^{14} for a mammalian genome. While today's computers can handle the first number of comparisons, the second number remains a formidable challenge.

Segmentation after mapping

The complexity of reconstruction becomes seriously difficult with very large genomes, which is why some groups have resorted to a two-level approach in such cases. The *shotgun* method is used only to reduce the number of clones required to cover the genome. A DNA library is then established in the following three types of vector, which can accept much larger fragments:

- **Cosmids** are hybrids of bacterial and bacteriophage plasmids grown in *E. coli*; they accept up to 30–40 kbp of inserted DNA.
- **BACs** (bacterial artificial chromosomes) are very large plasmids constructed starting from the replication origin of the bacterial chromosome. They can accept inserted DNA fragments of around 100–130 kbp.
- **YACs** (yeast artificial chromosomes) are analogs of the preceding two types of vectors, but derive from chromosomes of a lower eukaryote cell, such as brewer's yeast, *Saccharomyces cerevisiae*. YACs accept DNA fragments of the order of 1,000 kbp. Unlike the two other kinds of vector, which use *E. coli* as the host, YACs must be maintained in yeast cells.

These three types of vector can be employed to construct practically complete DNA libraries from given genomes, using a much smaller number of clones than plasmids intended for sequencing use. A human genome library consisting of 33,000 YACs that can accept inserted DNA fragments of average length around 1 Mbp has been constructed in a joint project between the Généthon (located in the Paris suburb of Evry) and the CEPH (*Centre d'étude du polymorphisme humain*, in Paris). The coverage rate of this clone library is thus practically 10 times the human genome, which consists of around 3.5×10^9 bp.

However, these large clone libraries cannot be directly exploited for sequencing purposes. A map must be constructed that positions the YAC, BAC, and cosmid vectors used in appropriate order on the chromosomes of the genome being studied. YAC positioning was accomplished in 1995 at Généthon/CEPH. The International Human Genome sequencing consortium later produced and mapped a library of 350,000 BACs, which provided similar tenfold coverage of the human genome. Since they bear smaller inserts, more BACs were required.

This map presents two advantages:

1. Used in conjunction with a genetic map, it allows isolation of genes associated with genetic diseases (Généthon's main goal) by localizing them on a given YAC.
2. It can be used as a framework in a sequencing program. Once YACs or BACs are positioned, they just have to be sequenced in an orderly manner, using the random strategy for each. The complexity of reconstruction is then analogous to that of a bacterial genome, since the construct size is on the order of one mega base pair. Considering the coverage rate (10×) of the YAC or BAC libraries, it is 'enough' to sequence a fraction of them to cover the entire human genome. This approach is a direct transposition of the classical 'divide and conquer' strategy used in computer science. For the human genome, the International Consortium sequenced 30,000 of the 350,000 BACs that made up its original library, enough to provide a 'tiling pathway' (see Figure 1.8) that completely covered the chromosomes. In the end, BACs were preferred to YACs, since they are smaller and easier to propagate in bacterial cells.

Mapping or positioning the primary library (cosmids, BACs, and YACs) is rather time-consuming. It is carried out using a combination of techniques, such

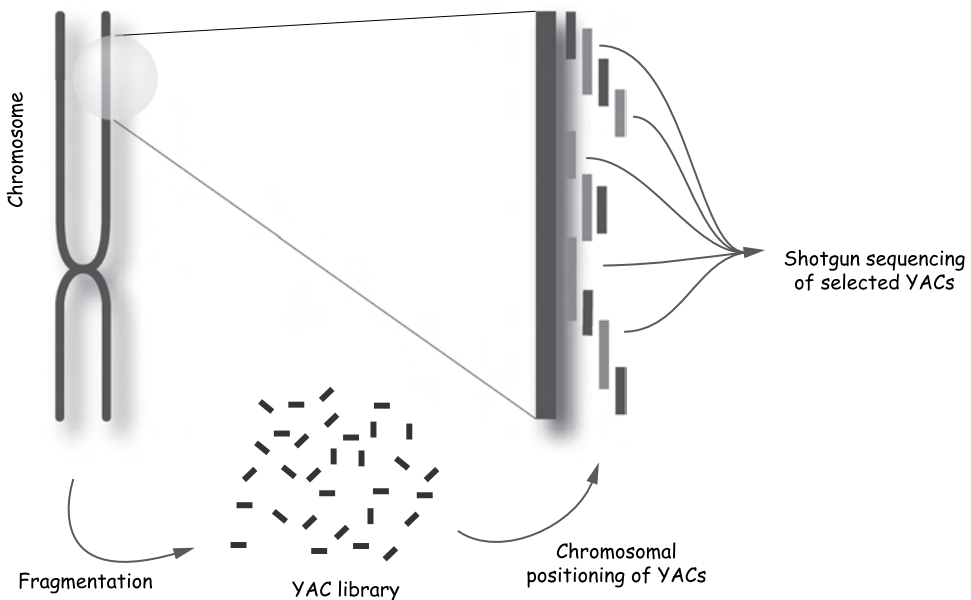


Figure 1.8 Two-step sequencing strategy with intermediate mapping.

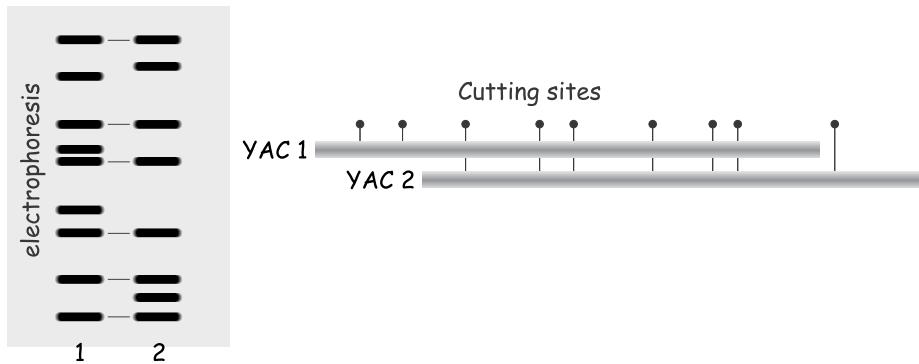


Figure 1.9 Example of the comparison of restriction enzyme digestion profiles of two overlapping YAC clones.

as comparing restriction enzyme digestion profiles, DNA-DNA hybridization, and genetic marker identification. For example, the restriction profiles of overlapping YACs contain common fragments. Systematic search for similarities among profiles throughout the entire YAC library identifies candidates.

It is then possible to verify the overlapping of two YACs by DNA-DNA hybridization. After the YAC① DNA restriction fragments have been separated by electrophoresis, they are transferred from the gel onto a membrane, where they are immobilized by cross-linking. The two strands are then separated by denaturing in an alkaline pH. The membrane is then incubated in the presence of the denatured YAC② DNA marked with a radioactive or fluorescent tracer. If the two YACs have common sequences, the marked YAC② DNA locally pairs with that of the YAC① DNA, in which case *hybridization* is said to have taken place between the two YACs. The tracer is seen to be fixed on the membrane, thereby revealing DNA fragments common to both YACs. If the restriction enzyme cutting profiles coincide accidentally and do not correspond to an overlap, there is no hybridization, thus no tracer fixation. Systematically conducted, this analysis results in a gene map.

1.4 Sequence assembly

The **sequenced fragment assembly method** requires that possible overlaps be identified first, so as to detect clones with common DNA sequences. If two clones are found to overlap, they are merged to form what is called a *contig*, a term designating a set of fragments connected to each other by overlapping sequences that are either identical or very similar (within the limits of sequenc-

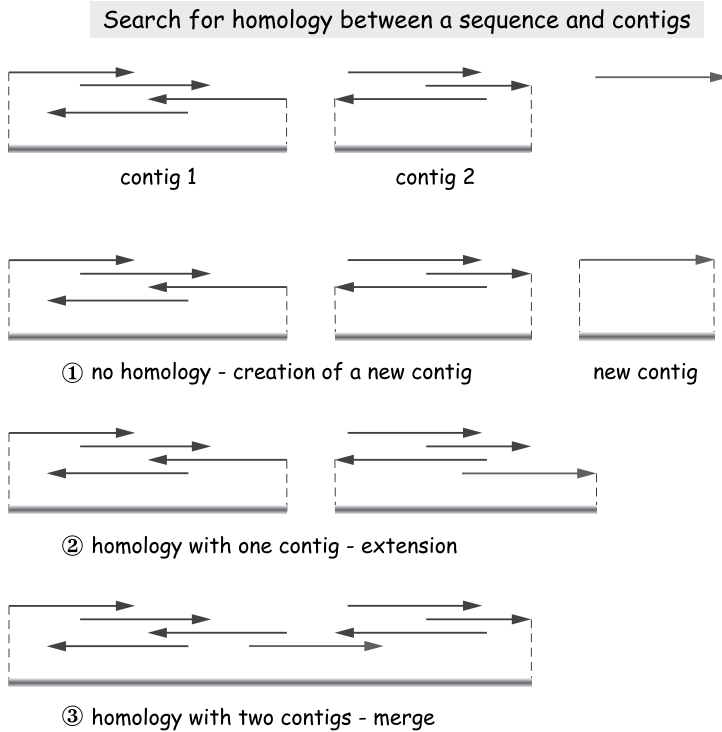


Figure 1.10 Iterative contig assembly
The arrows indicate the orientation of sequenced DNA strands.

ing error). The next step consists in step-by-step comparison of each new fragment with contigs that have already been identified (see Figure 1.10). This comparison must take into account the two possible relative orientations of the two sequences: If the fragment overlaps one contig, that contig is extended; otherwise a new contig consisting of this fragment alone is created. When a fragment simultaneously overlaps two contigs, both are fused with the fragment. At any time during a major sequencing project, the data correspond to a set of several contigs, whereas ideally, only one contig covering the whole sequence, remains at the end of the project.

While a contig is being assembled, a **consensus sequence** associated with it is defined according to the alignment of its constituent fragments. The consensus sequence compares the positions being read and checks their agreement, revealing any differences or ambiguity due to data errors (unread or incorrect nucleotide interpretation). Differences and interpretation ambiguities may be resolved by further data analysis and if necessary, by additional sequencing. This verification step is indispensable but time-consuming, since it is at least partly manual.

Overlap identification

Using the algorithms described in Chapter 2, overlaps may be located by 1×1 alignment of each new sequence added to already assembled contigs. If the alignment score is above a given threshold, the two sequences are considered to be overlaps. However, this ‘brute force’ method is costly in terms of calculation time, since the alignment algorithms are $O(nm)$, where n and m are the lengths of the two compared sequences. If k fragments are to be assembled, the algorithm is $O(k^2)$. This is prohibitive for very large genomes, where $k > 10^6$ base pairs. In order to simplify this problem, note that overlapping sequences are usually identical (or nearly so; except for a few rare errors) throughout the entire common region. In 1982, this observation led Roger Staden of Cambridge University to propose a more efficient strategy, which has since been improved. It consists in creating a table of 4^n n -uplets of possible nucleotides (n being of the order of 6 to 12). A list of fragments containing common n -uplets is compiled for each entry in the table, which is prepared in linear time $O(k)$. Two overlapping fragments will have a great number of common n -uplets, *i.e.*, all those that correspond to the common region. Applying this criterion, it is possible to identify candidate overlap fragments simply by looking up fragments that have several common n -uplets in the table. Overlapping may then be verified by applying a classical alignment method. This approach differs from the ‘brute force’ method in that the alignment algorithm is used only in cases in which overlapping is highly probable. The cost of this method is thus approximately a linear function of the number of gels to be analyzed.

The heuristic strategy developed by Staden can nevertheless be ineffective in certain cases, either owing to insufficient sequence data quality, resulting in failure to identify overlaps, or because the sequence analyzed contains several repetitions of a given motif, which can introduce contig fragment connection errors at the repetition sites. Several other methods avoid this obstacle by analyzing all possible overlaps according to criteria that permit evaluation of overlap quality (alignment scores). A graph of all possible connections among fragments is then drawn, in which the best pathway (‘minimal cost pathway’) is determined. However, although these methods (based on the Dijkstra algorithm) guarantee that the alignment obtained is optimum overall, they are considerably more costly in terms of calculation time.

1.5 Filling gaps

As discussed above, in large-scale, random ‘shotgun’ sequencing projects it is practically inevitable that DNA regions remain that are not covered by clones, which poses two questions:

- How to position contigs that are disconnected from each other, which amounts to ordering and orienting various contigs on the sequenced genome;
- How to complete the sequence of each gap.

This phase of the project can be rather laborious, since a combination of *ad hoc* methods must usually be mobilized in order to fill all the gaps. The three principal methods are integration of genetic map data; PCR across the gaps; and use of another DNA library, generally one containing larger fragments (cosmids, BACs, YACs, etc).

Integration of genetic maps

For certain model organisms, (*E. coli*, yeast, *Drosophila*, mouse, etc), genetic maps exist that indicate the relative positions of various chromosome loci. These maps, obtained by classical methods for measuring the co-transmission frequencies of several genetic markers, provide precise indications of the order of the associated genes, as well as a qualitative idea of the distances that separate them (measured in centimorgans by geneticists). For example, more than a thousand genetic loci of the colibacillus *E. coli* had been identified and mapped before systematic sequencing of its genome was undertaken. Some of these genes had been cloned and sequenced for specific purposes, as the need arose. If a gene associated with a known function and located in a genome is identified within a sequence, it is then possible to plot the site of the contig that contains it.

PCR amplification of missing regions

When the number of contigs is not too great, PCR amplification may be used to fill in some of their gaps. As may be seen in Table 1.1, the number of anticipated gaps in bacterial genomes is of the order of a few dozen. Oligonucleotides whose sequences match the 3'-extremities of the contigs obtained are then synthesized and chromosomal DNA amplification is attempted, using all pair combinations of the matching oligonucleotides as the primers. Such PCR amplification yields positive results only if the two primers match the sequences used on the complementary strands, which must be separated by no more than a few thousand nucleotides. In other words, it is possible to use PCR to amplify the corresponding chromosomal DNA segment when the extremities of the two contigs are separated by a gap of less than ~5 kbp. This method thus not only positions two neighboring contigs, but uses PCR to determine the sequence of a missing DNA fragment.

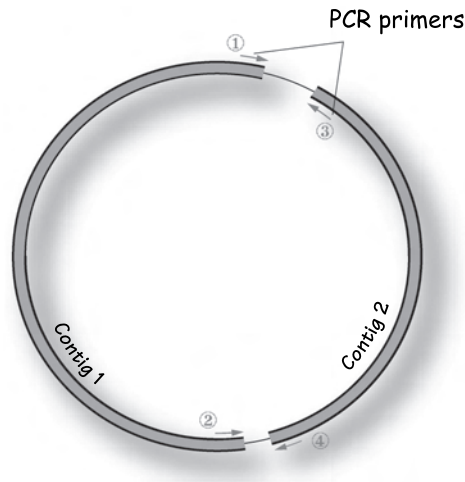


Figure 1.11 Simplified example of gap-filling by PCR, involving two contigs and two gaps on a circular chromosome. PCR amplifications of chromosomal DNA by primers 1 and 3 on one hand and primers 2 and 4 on the other yield a positive result, whereas the other combinations (1–4 and 2–3) fail. Thus, by sequencing the DNA fragments obtained with PCR, one can complete the project.

The PCR approach is very appealing, since it both solves the contig-ordering/*orientation* problem and determines the missing sequences. However, it runs into difficulty in several cases, especially when the gaps are too large for PCR to yield results. In other cases, the sequence repetitions at the ends of various contigs may lead to junction ambiguities, due to the appearance of false positives during PCR. Finally, when the number of gaps to be filled-in is too great, this method becomes difficult to implement. Indeed, if there are N contigs, therefore $2N$ PCR primers, $2N(N-1)$ PCR amplifications must be carried out. Consulting the data in Table 1.1, this amounts to between several hundred and a few thousand reactions for a bacterium, which may be carried out by existing automatic sequencers, but would be more than a billion PCR reactions for a mammalian genome, which remains unrealistic today.

1.6 Obstacles to reconstruction

Repetitive sequences

Most genomes contain some repetitive sequences. In bacteria and lower eukaryotes such as yeast, a very large fraction of chromosomal DNA is ‘useful’, that

is, corresponds to coding regions that are transcribed into RNA and translated into proteins. Nevertheless, certain sequences are repeated several times at different places in the genome. For example, identical multiple copies of the genes that code for ribosomal RNA are generally present in the genomes of such organisms over stretches of DNA several thousand base pairs long. Seven such copies are found in *E. coli*, and more than a hundred in brewer's yeast, *Saccharomyces cerevisiae*. Only a small fraction of the genomes of higher eukaryotes (around 2 percent for humans) is actually coding DNA, and numerous repetitive sequences, amounting to quite a significant portion of the genome, have been identified, the origins and exact functions of which remain obscure.

These repetitive sequences are a nuisance in genomic sequencing, since they considerably complicate the task of aligning and reconstructing the genome. A clone containing a repetitive sequence indicates potential alignments with other clones that contain copies of the repetitive sequence. Thus, in addition to authentic alignments with its real neighbors, spurious alignments are obtained. The inevitable ambiguities that result risk introducing assembly errors during reconstruction. These problems become all the more difficult as the repetitive sequences become long and increasingly similar.

'Unclonables'

Sequencing methods systematically require that DNA fragments be cloned in a sequencing vector. The recombinant vectors are then introduced into a host cell in order to constitute the DNA library to be sequenced. Owing to the toxicity of certain sequences to the host organism into which the vector is introduced, such exogenous sequence fragments sometimes cannot be stably maintained in the vector. For example, some DNA sequences can lead to the expression of an RNA segment or a protein that is toxic to the host cell, or that will titrate and sequester a factor essential to the host. The result is death of the cell bearing these vectors. Another possibility is that an inserted sequence will interfere with the vector's replication, in which case the vector can no longer be transmitted to the two daughter cells during cell division, therefore will not be perpetuated down the lineage of the host cell, rapidly disappearing in the succession of cell divisions. In both these cases, the corresponding genomic DNA fragment is not represented in the library, and is said to be 'unclonable'. The result of this technical difficulty is a sampling bias, which inevitably leads to a gap in the reconstruction of the contigs. However, if the gap is of reasonable size, it can be filled by PCR amplification, since it is possible to sequence the amplified fragment without the necessity of cloning it.

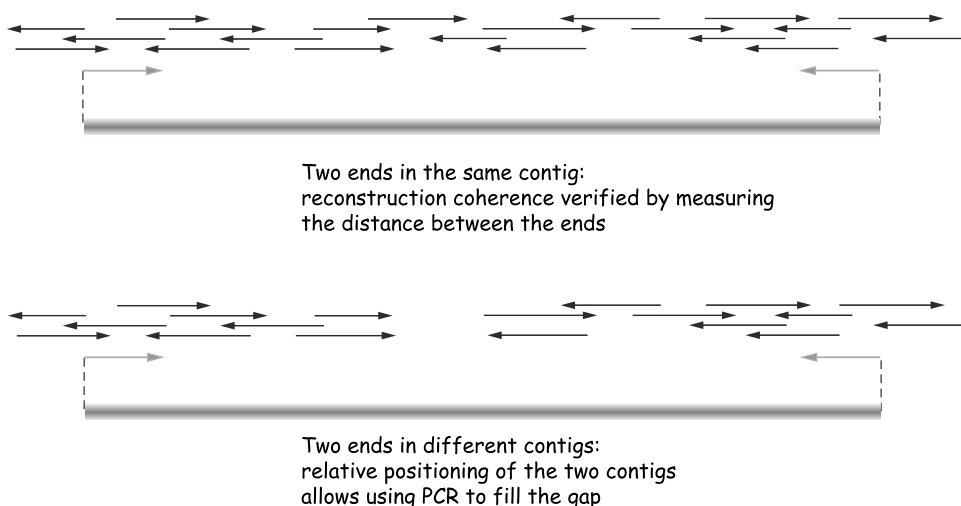


Figure 1.12 Utilization of the extreme ends of a 'large clone' to verify and guide the assembly of contigs and to fill gaps.

1.7 Utilizing a complementary 'large'¹ clone library

In order to verify the order of clones within contigs, and to help fill the remaining gaps with PCR, especially in cases of very large genomes, the need to resort to an additional library of larger DNA fragments (typically 20 to 100 kbp in cosmids and BACs) is practically unavoidable. Of course, in order to cover the whole genome, the number of clones required to constitute this library will be much smaller, since they bear DNA fragments that are 10 to 100 times larger than the library constructed in the sequencing vector.

The size of DNA fragments inserted into BAC and cosmid vectors may be very simply estimated by electrophoresis. The 500 nucleotides at each end of the large fragment are sequenced, and these two sequences are sought in the contigs obtained, referring to the first clone library. If the two ends are found in the same contig, it is possible to verify whether the distance between them in that contig is compatible with the size of the inserted DNA determined by electrophoresis. If the two sequences belong to two unconnected contigs, it is possible to position them with respect to each other, so as to estimate the size of the gap separating them, then to fill it.

By verifying the coherence of the distances between the ends of two DNA fragments inserted into a BAC or cosmid vector during assembly, this strategy

¹ This is an oversimplification. It is not the *clones* that are large, but the *DNA* fragments inserted into the vectors borne by the cloning lines that constitute the library.

eliminates most of the difficulties concerning repetitive sequences. The distances between the ends of large clones may even be directly introduced into the reconstruction algorithm in the form of constraints. This approach also considerably simplifies the problem of using PCR to fill gaps between contigs by reducing the number of primer combinations to be tried. Finally, in very unfavorable cases in which the gaps are too large to be accessible to PCR, the ‘large’ clone that covers the missing region may be utilized to try to sequence it directly, using its DNA as the template and synthesizing *ad hoc* its sequencing primers, which are complementary to the nucleotide sequences on either side of the gap. The only difference between this approach and classical sequencing is that instead of the universal primers described above, specific primers are used that allow sequencing to start at the edge of the missing zone rather than at the edge of the inserted sequence. The size of the gap may be progressively reduced until it is accessible by PCR.

1.8 The first large-scale sequencing project: The *Haemophilus influenzae* genome

Haemophilus influenzae is a small gram-negative bacterium whose natural host is the human upper respiratory tract, in which it can cause mucosal infections, otitis, sinusitis, and meningitis, particularly in young children. Its genome consists of a single circular chromosome 1.83 million base pairs long, which is relatively small compared with the genome of its cousin, *E. coli*, which exceeds 4.3 million base pairs. Its small genome and the therapeutic interest of this bacterium as a target made *H. influenzae* the candidate of choice for the first complete automated genome sequencing project, carried out in 1995 at TIGR (The Institute of Genome Research), in the United States.

The following is the comprehensive *H. influenzae* sequencing program undertaken at TIGR:

- Genomic DNA was mechanically cleaved by ultrasound and the ends of the fragments produced were ‘equalized’ by nuclease treatment.
- Fragments between 1.6 and 2.0 kbp long were selectively purified by electrophoresis and ligated into a plasmid bearing a resistance marker to ampicillin, an antibiotic in the penicillin family.
- The recombinant plasmids obtained were transformed in *E. coli* and the resulting colonies cultured in a selective medium with the antibiotic, then purified and isolated. The library thus obtained contained 19,687 *E. coli* clones, each bearing a recombinant vector that had incorporated a fragment of the *H. influenzae* genome.

- The double-stranded DNA of the 19,687 plasmids was prepared in 96-well plates, permitting 96 parallel purifications, using semi-robotic methods. The set of 24,304 sequences read represented 11.6 million nucleotides, a genome coverage rate greater than a factor of 6. The reconstruction was achieved using an automatic program that allowed identification of 42 contigs (connected sequence blocks) separated by 'gaps'.
- These gaps were filled by often laborious *ad hoc* methods: molecular hybridization, PCR, and recloning, starting from another DNA library that contained larger fragments (15–20 kbp vs. 1.6–2.0 for the library used during random sequencing). It was thereby possible to determine the complete sequence of 1,830,137 basepairs, with an error rate estimated to be around 1/10,000. The cost excluding overheads was estimated to be around \$0.50 per nucleotide in the final sequence, a total of \$0.9 million.

1.9 cDNA and EST

All the sequencing methods described above are meant for large DNA fragments, or even for entire genomes. Although they provide overall information, they are relatively cumbersome to implement. Only a small fraction of the very large genomes of higher eukaryotes actually codes for proteins (2–5 percent). In addition to the large non-coding regions that separate them, genes are often internally divided by a large number of introns, which are removed from the mature messenger RNA by RNA splicing. Thus, the coding part of the genome is dispersed and amounts to only a minor part of the whole genome. This is why some research groups, in addition to using the overall approach, have concentrated on the essential information contained in coding segments, which is easier to extract first.

The latter approach consists in isolating mature messenger RNA (*i.e.*, minus its introns), then synthesizing the complementary DNA (cDNA). The protocol is illustrated in Figure 1.13.

- Extracted and purified from tissues, mRNA bears a poly-A tail at the 3'-end, which permits it to be purified by chromatography on resins into which poly-T has been chemically cross-linked (the poly-T pairs with the poly-A).
- RNA/DNA reverse transcription is carried out using poly-T as the primer. In addition, the viral reverse transcriptase used for this purpose possesses the property of hydrolyzing the template DNA while polymerization² is proceeding.

² This specific hydrolysis activity of an RNA strand in an RNA:DNA heteroduplex is called *Ribonuclease H activity*.

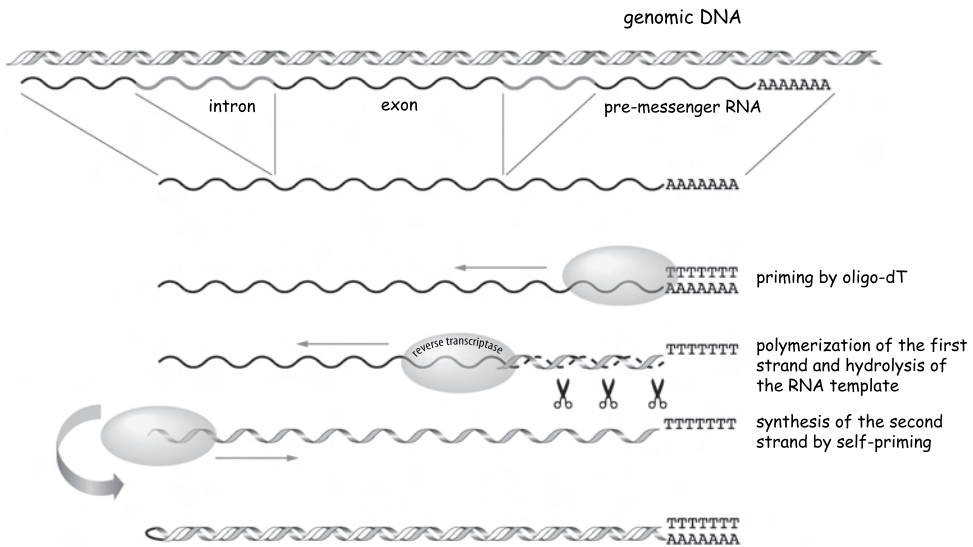


Figure 1.13 Construction of a cDNA segment starting from 3'-polyadenylated mRNA.

- When transcription of the first DNA strand has terminated, polymerization starts again in the other direction, this time using DNA as the template. The polymerization exploits a hairpin structure at the end of the first strand for this purpose, which permits priming the second strand.

The resulting double-stranded DNA is a copy of the messenger RNA known as cDNA (c for complementary to RNA). cDNA is different from genomic DNA in that it is devoid of introns. The interest in cloning and sequencing cDNAs is that it reveals which subset of genes is actually transcribed into mRNA in a given cell. It is thus possible to identify the transcription profiles of each cell type in a complex organism made up of differentiated cells that constitute specialized tissues and organs.

Another advantage of this approach is that, in conjunction with the genomic DNA sequence, it permits precise identification of intron and exon boundaries, thus of splicing sites. In addition, due to alternative splicing, and according to the tissue type and context, the same gene can give rise to several different messenger RNAs, therefore to several protein variants. The *transcriptome* (by homology with *genome*) is the set of all the messenger RNAs that can be transcribed from the chromosomes of a given cell. Transcriptome-related information that would be difficult to obtain from the genomic sequence alone is accessible by automatic cDNA sequencing, among other ways.

Two quite different strategies may be employed during cDNA construction and analysis: First, it is possible to attempt to obtain the longest cDNAs, in

order to cover the entire open reading-frame (ORF) that corresponds to the gene being studied. This requires considerable caution while extracting the messenger RNAs. Any degradation of the mRNA will result in the production of an incomplete cDNA. Obtaining a complete cDNA allows determination of the entire ORF sequence, from which the sequence of the corresponding protein may be deduced. Cloning the cDNA bearing the ORF in an appropriate genetic vector will then permit production of the recombinant protein in a heterologous prokaryote or eukaryote host. Since the cDNA is devoid of introns, it can be translated in any cell, independent of any splicing machinery.

Alternatively, the cDNA – even if incomplete – may be used to determine the sequences of its own ends, which constitute the ‘signatures’ allowing unequivocal identification of the corresponding gene. This strategy has been used in a massive and systematic way in attempting to identify all the genes transcribed in a given type of cell. These cDNA sequence fragments are called *ESTs* (Expressed Sequences Tags). To date, more than six million ESTs have been sequenced for the human, the most-studied organism, and the number of sequenced ESTs for eight other species exceeds 400,000: mouse, rat, cow, zebrafish, chicken, and frog in the animal kingdom; wheat and corn in the plant realm. In numerous cases, several ESTs correspond to the same gene, and it is possible to apply the reconstruction methods (contig assembly) described above to reconstitute the corresponding messenger RNA sequence.

Bibliography

- Adams M.D., *et al.* (1993). Rapid cDNA sequencing (expressed sequence tags) from a directionally cloned human infant brain cDNA library. *Nat Genet* **4**: 373–380.
- Adams M.D., *et al.* (2000). The genome sequence of *Drosophila melanogaster*. *Science* **287**: 2185–2195.
- Bonfield J.K., *et al.* (1995). A new DNA sequence assembly program. *Nucleic Acids Res* **23**: 4992–4999.
- Broder S., Venter J.C. (2000). Whole genomes: the foundation of new biology and medicine. *Curr Opin Biotechnol* **11**: 581–585.
- Cohen D., *et al.* (1993). A first-generation physical map of the human genome. *Nature* **366**: 698–701.
- Dear S., Staden R. (1991). A sequence assembly and editing program for efficient management of large projects. *Nucleic Acids Res* **19**: 3907–3911.
- Fleischmann R.D., *et al.* (1995). Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science* **269**: 496–512.
- International human genome sequencing consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature* **431**: 931–945.
- Kent W.J., Haussler D. (2001). Assembly of the working draft of the human genome with GigAssembler. *Genome Res* **11**: 1541–1548.

- Lander E.S., *et al.* (2001). Initial sequencing and analysis of the human genome. *Nature* **409**: 860–921.
- Lander E.S., Waterman M.S. (1988). Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics* **2**: 231–239.
- McPherson J.D., *et al.* (2001). A physical map of the human genome. *Nature* **409**: 934–941.
- Myers E.W., *et al.* (2000). A whole-genome assembly of *Drosophila*. *Science* **287**: 2196–2204.
- Olson M., *et al.* (1989). A common language for physical mapping of the human genome. *Science* **245**: 1434–1435.
- Osoegawa K., *et al.* (2001). A bacterial artificial chromosome library for sequencing the complete human genome. *Genome Res* **11**: 483–496.
- The *C. elegans* sequencing consortium (1998). Genome sequence of the nematode *C. elegans*: a platform for investigating biology. *Science* **282**: 2012–2018.
- Venter J.C., *et al.* (2001). The sequence of the human genome. *Science* **291**: 1304–1351.
- Venter J.C., *et al.* (1998). Shotgun sequencing of the human genome. *Science* **280**: 1540–1542.
- Weber J.L., Myers E.W. (1997). Human whole-genome shotgun sequencing. *Genome Res* **7**: 401–409.
- Weissenbach J., *et al.* (1992). A second-generation linkage map of the human genome. *Nature* **359**: 794–801.

2

Sequence comparisons

2.1 Introduction: Comparison as a sequence prediction method

From its inception, the science of biology has been concerned with comparisons of the various manifestations of life – cells, organs, and organisms. The study of nucleic acid and protein sequences is no exception to this rationale, and comparing their alignments has become an essential approach of the bioinformatician.

The accelerating growth of genome sequencing programs is generating a huge quantity of data. Information derived from genomic DNA sequences is used to predict regions that are transcribed into messenger RNA. The translation of open reading frames (ORFs) into transcribed sequences and comparison with cDNA fragments and expressed sequence tags (ESTs) permits deduction of the amino acids that constitute the proteins encoded by these genes. Bacterial genomes encode between 1,000 and 10,000 different proteins (slightly more than 4,000 for *Escherichia coli*), and eukaryote genomes between 5,000 and 50,000¹ (slightly more than 12,000 for *Drosophila*). These gene sequences must then be *annotated*; an attempt must be made to identify their functions in cells, which is easy in some cases, since a similar sequence from a related species may be located in a sequence database. For example, certain chimpanzee proteins are nearly identical to corresponding human proteins. However, proteins newly identified by genome sequencing often do not have obvious homologs among already known and characterized proteins. In such cases, further analysis must be undertaken in order to reveal faint resemblances, in the attempt to ascribe a function to a protein under study. Today, systematic comparison using computer-based analytic methods in conjunction with *a posteriori* interpretation by biologists plays an essential role in the decryption of genetic information.

¹ Excluding variants resulting from alternative RNA splicing and protein maturation.

2.2 A sample molecule: the human androsterone receptor

The nuclear receptor for the human steroid sex hormone androsterone will be used in this chapter as a sample protein sequence to illustrate questions and approaches. The function of the androsterone receptor protein is to stimulate transcription of certain genes in response to a hormone signal. Androsterone enters the target cell nucleus, where it binds to its receptor, part of which then binds to a specific DNA sequence located upstream from the transcription promoters of the genes to be stimulated. This in turn elicits RNA polymerase binding, thereby activating transcription. The amino acid sequence of that part of the androsterone receptor which binds to the DNA is given below (amino acids 550 through 620, of a total of 919 in the complete hormone protein).

⁵⁵⁰ . . . DYYFPPPQKTCLICGDEASGCHYGALTCGSCKVFFKRAAEGKQKYLCA
NDCTIDKRRKNCPSCRLRKCYE . . . ⁶²⁰

The three-dimensional (3D) structure of the corresponding protein domain bound to its DNA target is illustrated in Figure 2.1.

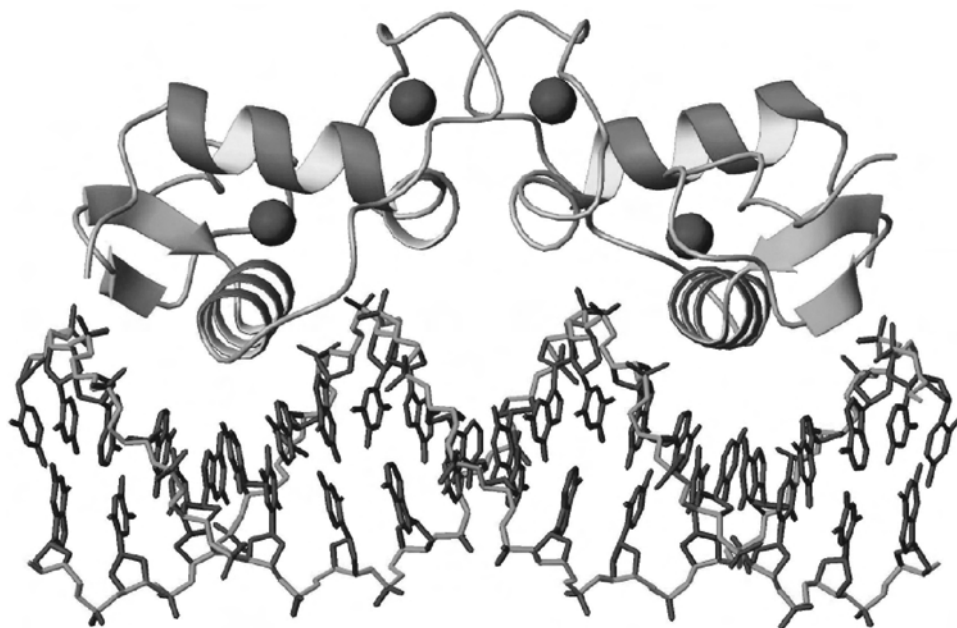


Figure 2.1 Structure of a steroid hormone nuclear receptor (dimer) bound to DNA.

2.3 Sequence homologies – functional homologies

The basic postulate of all biological sequence analysis is:

TWO MOLECULES OF RELATED FUNCTION
USUALLY HAVE SIMILAR SEQUENCES
-RECIPROCALLY-
TWO MOLECULES OF SIMILAR SEQUENCE
USUALLY HAVE RELATED FUNCTIONS

The above is true for proteins whose amino acids and sidechain chemical functions are conserved, especially those located in the active site. To a lesser degree, this is also true for DNA, where such regions as transcription promoters and binding sites for specific proteins generally present significant similarities.

The objective of the bioinformatician is to detect such similarities, using computer science methods to draw biological conclusions:

- If two molecules of known function resemble each other, it is reasonable to conclude that they at least partly share their mechanism of action.
- Similarity in the sequences of an unknown and a known protein suggests the function of the unknown protein.
- The stronger and more extensive the sequence resemblance, the more probable the protein function homology.
- Identifying the most highly conserved or most similar parts of a protein indicates important regions of the protein, thus the probable location of its active site.

In order to illustrate the nature of such similarities, Figure 2.2 presents three examples of protein sequences that are aligned with the same part of the androsterone receptor sequence:

In the first case (A), the homology is patent, since most amino acids are conserved. This resemblance is not very surprising, since it concerns the nuclear receptor for progesterone, another steroid sex hormone. These two proteins have quite analogous cell functions. In the second case, the resemblance is less clear, and the number of strictly conserved residues decreases. It was even necessary to introduce two gaps into the androsterone receptor sequence to continue aligning it with the sequence of the other receptor. Again however, the resemblance is significant, since the protein involved is also a nuclear receptor, whose ligand this time is thyroid hormone, which although not a steroid sex hormone, explains why the resemblance is more distant. The similarity observed

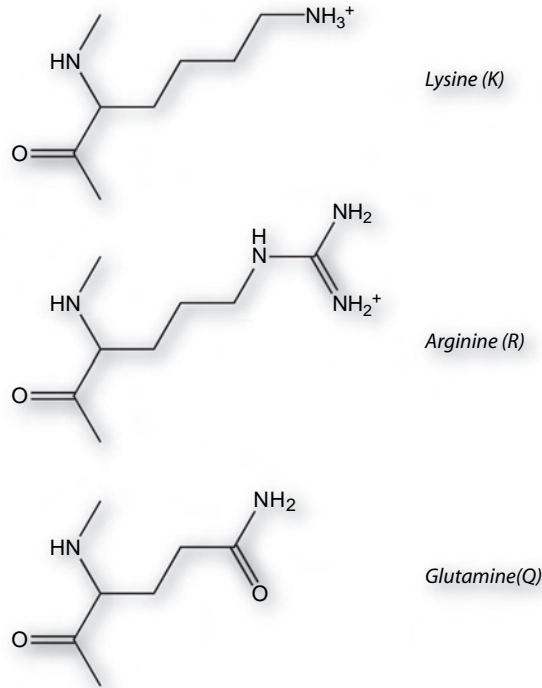


Figure 2.3 Example of homologous amino acids. All three bear -NH_2 or -NH_3 groups at the ends of straight carbon chains. Lysine and arginine are positively charged at physiological pH.

such as tryptophan (W) and cysteine (C), are relatively rare (~9 percent each for A and V and 1.5 percent for W and C.) Thus, conservation of a leucine is statistically less significant than conservation of a tryptophan in an alignment of two sequences.

- The chemical structures and functions of some amino acids, while not identical, are very similar (Figure 2.3), and therefore related. This must be taken into account when calculating the scores of two amino acids located opposite each other in a sequence alignment (as indicated by a plus sign in Figure 2.2).

Comparison tables, or *matrices*, are therefore used to attribute a score to an alignment of any two nucleotides or amino acids. For protein sequence alignments, 20×20 matrices M are used to evaluate all combinations of amino acid pairs. The value of the coefficient $M(a, b)$ indicates the quality of alignment between two amino acids a and b . It is then possible to calculate an overall score for the alignment of two sequences of length L (a_k and b_k represent the k^{th} amino acids of each sequence):

$$score = \sum_{k=1}^L M(a_k, b_k)$$

Calculation of the percentage sequence identity is a special case of this approach, in which the matrix M is reduced to the identity matrix. It is one of the most frequently used for DNA sequences.

	A	T	G	C
A	1	0	0	0
T	0	1	0	0
G	0	0	1	0
C	0	0	0	1

Several types of matrices have been proposed for protein sequences. Some are based on *a priori* estimations of physicochemical similarities. However, this is somewhat arbitrary and rarely used anymore. Others are based on the probability of mutation over evolution. For example, PAM (*Probability of Acceptable Mutation*) matrices assign a score based on the alignment of protein sequences with identical functions in related species. The frequency at which amino acid a is replaced by amino acid b in another species is noted. Knowing the evolutionary distance between the two species², it is possible to deduce the mutation probability $p(a \rightarrow b)$ per unit of time. Using the $M(a, b)$ matrix coefficient, the logarithm of this probability normalized by the frequencies of a and b is:

$$M(a, b) = \log \frac{p(a \rightarrow b)}{p(a) p(b)}$$

For example, if matrix M_{10} for an evolutionary period of ten million years is calculated using logarithms, it is possible to compute $M_{20} = M_{10}^2$ and $M_{100} = M_{10}^{10}$, for 20 and 100 million years, respectively, by simple matrix multiplication. Such PAM matrices have been widely used over the past twenty years. However, they also suffer from bias, owing to a choice of protein families that was initially too restrictive for use in calculating the probabilities $p(a \rightarrow b)$. These matrices have recently been replaced by others based on the alignment of blocks of highly conserved sequences known as a 'BLOSUM' (block substitution matrix.) These contiguous, gap-free blocks may be found in the sequence database (see Figure 2.4). This approach postulates that blocks correspond to highly conserved elements in the three-dimensional structure of the correspon-

² The period that has transpired since the two species diverged from a common ancestor. Among other criteria, estimation of the duration of this period is based on paleontological analysis.

YLCASRNDCTIDKFRKNCPCSLRKCIEAGMTLGAR	androsterone	human
YSCKYDSCCVIDKITRNQCQLCRFKKCIAGMAMDV	thyroid	human
YCCKFGRACEMDMYMRKQCQECRLKKCLAVGMRPECV	ecdysone	Drosophila
YTCHRDKNKVINKVTRNRCQYCRQLQKCFEVGMSKESV	retinoic acid	mouse
YSGRDNKDCITVDKQRNRNRCQYCRYQKCLATGMKREAV	retinoic acid	human
FTCPFNQDCKITKDNRRHCQACRLKRCVDIGMMKEFI	vitamin D3	chicken
YMCPTATQCTIDKNRRKSCQACRLRKCIEVGMKGGI	estrogen	pig
YLCAGRNDICVDKIRKNCPCACRLRKCQAGMVLGGR	glucocorticosteroid	rabbit
YTCESQSCCKIDKTQKRCPCFCRFQKCLTVGMRLAV	steroid	mouse

Figure 2.4 Example of a sequence block. The sequences (left) are all nuclear receptor fragments. Strictly conserved amino acids are shaded in gray. The receptor ligand and species from which it derives are indicated on the right.

ding proteins, which it has been possible to verify in cases of known 3D structure. The amino acid substitutions observed in these blocks thus indicate replacements that are ‘acceptable’ from the point of view of protein structure and function.

This may be verified in the left column of the block in Figure 2.4, which contains eight tyrosines (Y) and one phenylalanine (F). These two amino acids are very similar; both are aromatic, differing only by the presence of an additional hydroxyl (-OH) group on the tyrosine. It is possible to calculate the probabilities of the substitution of one amino acid for another by considering all unordered amino acid pair combinations that appear in the columns of the block. For example, the first column contains eight Y and one F, yielding eight (Y, F) pairs and twenty-eight (Y, Y) pairs. Compiling all the blocks and using all twenty amino acids, we obtain numerous occurrences of each of the 210 possible pairs. The $M(a, b) = M(b, a)$ term of the BLOSUM matrix is then calculated in a manner analogous to PAM matrices, taking the base 2 logarithm³ of the ratio of the frequency of the (a, b) pair to the frequencies of a and b, written as $p(a)$ and $p(b)$.

$$M(a, b) = \log_2 \frac{p(a, b)}{p(a) \cdot p(b)}$$

If $p(a, b) > p(a) \cdot p(b)$, the (a, b) substitution is overrepresented, indicating that amino acids a and b are readily interchangeable, thus probably homologs. The $M(a, b)$ term is therefore positive. If $M(a, b) = 0$, replacement of a by b is neutral, and if $M(a, b)$ is negative, the substitution is unfavorable.

Several BLOSUM matrices have been calculated using alignment blocks constructed according to more or less stringent identity criteria among the various sequences. In thorough analysis, the BLOSUM62 matrix, based on sequence

³ Since log base 2 is used, the coefficients of matrix M is expressed in *bits*.

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2	
S		4	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3	
T			5	-1	0	-2	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-2	
P				7	-1	-2	-2	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4	
A					4	0	-2	-2	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-3	
G						6	0	-1	-2	-2	-2	-2	-2	-3	-4	-4	-3	-3	-3	-2	
N							6	1	0	0	1	0	0	-2	-3	-3	-3	-3	-2	-4	
D								6	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4	
E									5	2	0	0	1	-2	-3	-3	-2	-3	-2	-2	
Q										5	0	1	1	0	-3	-2	-2	-3	-1	-2	
H											8	0	-1	-2	-3	-3	-3	-1	2	-2	
R												5	2	-1	-3	-2	-3	-3	-2	-3	
K													5	-1	-3	-2	-2	-3	-2	-3	
M														5	1	2	1	0	-1	-1	
I															4	2	3	0	-1	-3	
L																4	1	0	-1	-2	
V																	4	-1	-1	-3	
F																		6	3	1	
Y																			7	2	
W																					11

Figure 2.5 BLOSUM62 matrix calculated using over 2,000 sequence blocks. Matrix coefficients are multiplied by 2 and rounded-off to the nearest integer. The units are thus half-bits. Squares containing positive or null coefficients are shaded.

blocks of greater than 62 percent identity, yields the best empirical results and is the matrix most frequently used for searching data banks. It is presented in Figure 2.5, in which one can see that the diagonal coefficients, which correspond to the rarest residues (C and W), are the largest, indicating that their strict conservation is what is most significant.

By comparing PAM and BLOSUM matrix coefficients, it may be seen that while their general tendencies are similar, there are real differences between them. BLOSUM matrices are more tolerant of substitution among polar or hydrophilic amino acids, mostly found on the surfaces of proteins, and more stringent with respect to other types of modifications involving apolar or hydrophobic amino acids. The latter are usually present within the 3D folds of the protein, and their substitution often has major structural repercussions. BLOSUM matrices are thus more useful in identifying proteins whose 3D structures are homologous. It has been possible to verify this empirically, using

test-sequence sets corresponding to proteins of known structure. This is not surprising, since the blocks of aligned sequences used usually consist of helices and sheets, the structural elements of these proteins.

2.5 The problem of insertions and deletions

The matrices described in the preceding paragraph permit direct evaluation of the quality of the alignment of gap-free sequences. However, when comparing complete sequences of proteins or long DNA fragments, ‘jumps’ must often be introduced into one of the sequences being studied. This is referred to as an *insertion* or a *deletion* with respect to the reference sequence.

In proteins, insertions and deletions are generally found in loops located on the protein surface. In these regions, which are exposed to aqueous solvent, modification of the length of the peptide chain has little effect on the 3D structure of the protein, compared with insertions and deletions located within the folds of the protein.

Insertions and deletions must be taken into account in calculating the scores of the alignments that contain them. The simplest approach consists in extending the definition of matrices, adding coefficients for the alignment of a nucleotide or an amino acid with a gap (represented by a dash (‘-’) in the alignments, as seen in Figures 2.2 and 2.6). Thus new terms are introduced into the matrix, $M(a, -)$, corresponding to the cost of alignment of a gap ‘-’ with a residue a . Since there is no rigorous statistical method for defining the value (cost), it is chosen empirically. A value ‘ Δ ’ which is significantly lower than the lowest alignment score of the two amino acids or nucleotides, is generally used for all amino acids:

$$\forall a, M(a, -) = \Delta(\text{constant})$$

For example, with the BLOSUM62 matrix in Figure 2.5, a cost Δ of -6 to -10 is applied for the insertion of a gap, whereas the lowest score for the matrix is -4 . This extension of the notion of matrix score therefore permits quantitative evaluation of the alignment of two sequences, ‘weighing’ them in comparison with other alignments. A problem biologists have in analyzing two protein sequences is to find the *best* alignment; that is, the one whose score evaluated using the matrix is the highest.

ACGCGGCTATTCA	CGACGGACTTT	TAGACGCCCAGCGAGCGAC	reference sequence
ACGCGGCTATTCA	--CGACGGACTTT	TAGACGCCCAGCGAGCGAC	test sequence
	^^	^	
	deletion	insertion	

Figure 2.6 Examples of deletion and insertion in a test sequence.

2.6 Optimal alignment: the dynamic programming method

There are a great number of different ways to align two sequences of length n with insertions/deletions. If certain trivially equivalent alignments are eliminated, it is possible to express this number as:

$$A(n) = \binom{2n}{n}$$

This number, the central coefficient of the binomial, increases approximately as 2^{2n} . For two sequences of 20 nucleotides or amino acids, this already amounts to 137 billion combinations. It is therefore unthinkable to use an approach of the ‘brute force’ type, which would consist in testing all of them. Instead, a very general algorithmic method known as *dynamic programming* is used, which finds the optimal alignment in time $O(n^2)$. Let there be two sequences S_1 and S_2 of respective lengths m and n . The principle of the dynamic programming method is to progressively construct optimal alignments of longer and longer sub-sequences of S_1 and S_2 , using the results obtained with the shorter sub-sequences. For this purpose, a table T of dimensions $m \times n$ is constructed. The value of cell $T(i, j)$ indicates the score of the best alignment of the i first S_1 amino acids with the j first S_2 amino acids. The table is constructed recursively. Beginning by initializing at $T(0, 0)$, which corresponds to an empty alignment (zero), the table is filled in progressively from left to right and from top to bottom.

Three types of alignment should be considered when filling in the $T(i, j)$ cell, that is, the best score for sub-sequences of lengths i and j :

- Those that align $S_1(i)$ with $S_2(j)$. In this case, the score of the alignment of $S_1(i)$ with $S_2(j)$ is added to the best score obtained for sequences of length $i - 1$ and $j - 1$: $T(i - 1, j - 1) + M(S_1(i), S_2(j))$.
- Those that align $S_1(i)$ with a gap. In this case, the score of the alignment of $S_1(i)$ with a gap or with constant Δ (above) is added to the best score obtained for sequences of length $i - 1$ and j : $T(i - 1, j) + \Delta$.

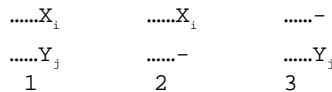


Figure 2.7 Possible types of alignment for sub-sequences of lengths i, j .

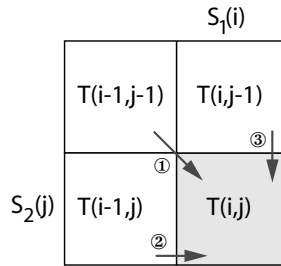


Figure 2.8 The three terms used in calculating $T(i, j)$ determine a matrix path.

- Those that align $S_2(j)$ with a gap. In this case, the score of the alignment of $S_2(j)$ with a gap or with constant Δ (above) is added to the best score obtained for sequences of length i and $j - 1$: $T(i, j - 1) + \Delta$.

To calculate $T(i, j)$, it suffices to take the maximum of these three scores:

$$T(i, j) = \max \begin{cases} T(i-1, j-1) + M(S_1(i), S_2(j)) & \text{①} \\ T(i-1, j) + \Delta & \text{②} \\ T(i, j-1) + \Delta & \text{③} \end{cases}$$

Completing table T in the lower right-hand corner box, $T(m, n)$, we obtain the best alignment score between S_1 and S_2 . To find the alignment(s) which obtain(s) this score⁴ we must recall which of the above alternative terms is the highest; that is, which among the three expressions ①, ②, and ③ is (are, in case of a tie) was (were) used to calculate $T(i, j)$. These possibilities represent different possible paths through the matrix T , as presented in Figure 2.8.

This approach determines which path in matrix T yields the overall maximal alignment of sequences S_1 and S_2 . Figure 2.9 gives an example of a calculation using matrix T , with the resulting maximal path, for the alignment of a fragment of the androsterone receptor with a fragment of the thyroid hormone receptor, as it was presented in Figure 2.2.

Application of the dynamic programming method to alignment is known as the algorithm of Needleman & Wunsch, who first applied it to sequences of biological molecules. Given comparison matrix M and the cost of insertion/deletion Δ , the Needleman & Wunsch algorithm finds the minimal cost alignment. For two sequences of length m and n , this algorithm has complexity $O(mn)$.

⁴ Several alignments can achieve the maximal score. Each time that the maximum number taken to calculate $T(i, j)$ is attained by at least two of the three possible alternative terms, ‘branching’ occurs in the matrix T return pathway. The number and multiplicity of branchings determines the number of equivalent maximal alignments.

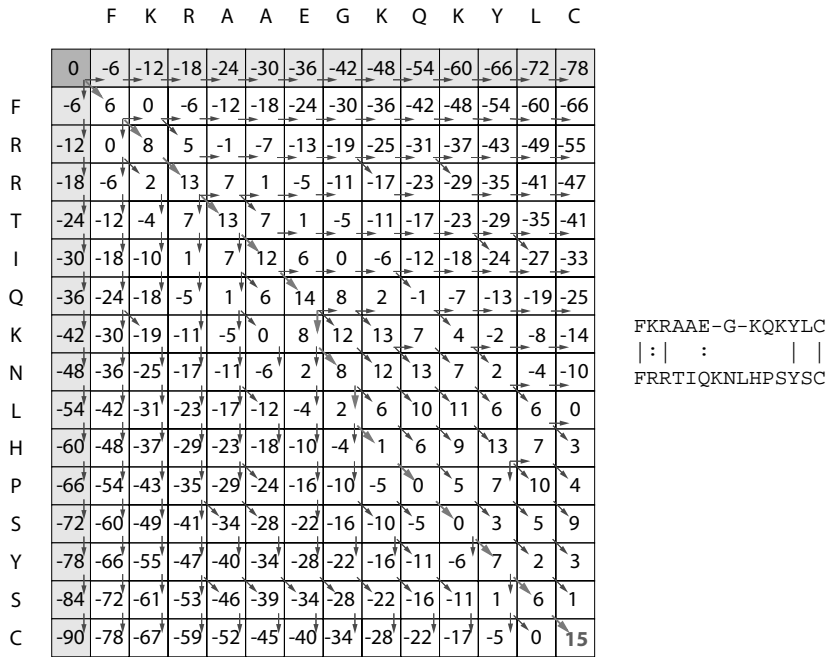


Figure 2.9 Example of matrix $T(i, j)$. The scores are indicated in cells; arrows indicate each time a given pathway is followed, in order to obtain the maximum score. The resulting maximal alignment is indicated to the right of the matrix. The value of Δ , the cost of alignment with a gap, is -6 . The first line and the first column, $T(0, j)$ and $T(i, 0)$, are respectively initialized by $i \times \Delta$ and $j \times \Delta$, effectively corresponding to alignments that begin with i or j gaps.

n) in terms of calculation time and memory space. Table $T(i, j)$ must first be completed, then followed in reverse, in order to reconstitute an optimal pathway.

The Needleman & Wunsch method permits aligning sequences in an *overall* manner, attempting to construct the best alignment of the two sequences along their full lengths. This approach is justified when two proteins are homologous throughout their entire sequences. However, very often the homology is *local*; concerning only a part of each of the two sequences being compared. In the case of the androsterone nuclear receptor, the sequence alignments presented in Figure 2.2 concern just the DNA binding zone, which consists of only around a hundred amino acids. This protein also contains domains that affect other functions, including an androsterone binding domain and a regulatory domain, as well as domains responsible for interactions with other proteins. Sometimes the homology between two proteins is limited to one or several domains that correspond to the functions they have in common. The Needleman & Wunsch algorithm is clearly not good for detecting this type of similarity in two

sequences. Indeed, the scores obtained would be severely penalized by ‘non-homologies’ outside the conserved region.

To avoid this difficulty, Smith & Waterman proposed a very simple modification to the Needleman & Wunsch algorithm: score ‘debt’ incurred in the non-homologous regions of table $T(i, j)$ must be abolished in order to locate locally similar regions. This principle consists in annulling the score as soon as it becomes negative; otherwise the calculation remains identical. There is then a quadruple alternative:

$$T(i, j) = \max \begin{cases} 0 \\ T(i-1, j-1) + M(S_1(i), S_2(j)) \\ T(i-1, j) + \Delta \\ T(i, j-1) + \Delta \end{cases}$$

Figure 2.10 is the same as Figure 2.9 with the calculation of $T(i, j)$ modified. No longer satisfied with looking at the score in the bottom-right cell of the matrix, as with the Needleman-Wunsch algorithm, we now look at all cells and

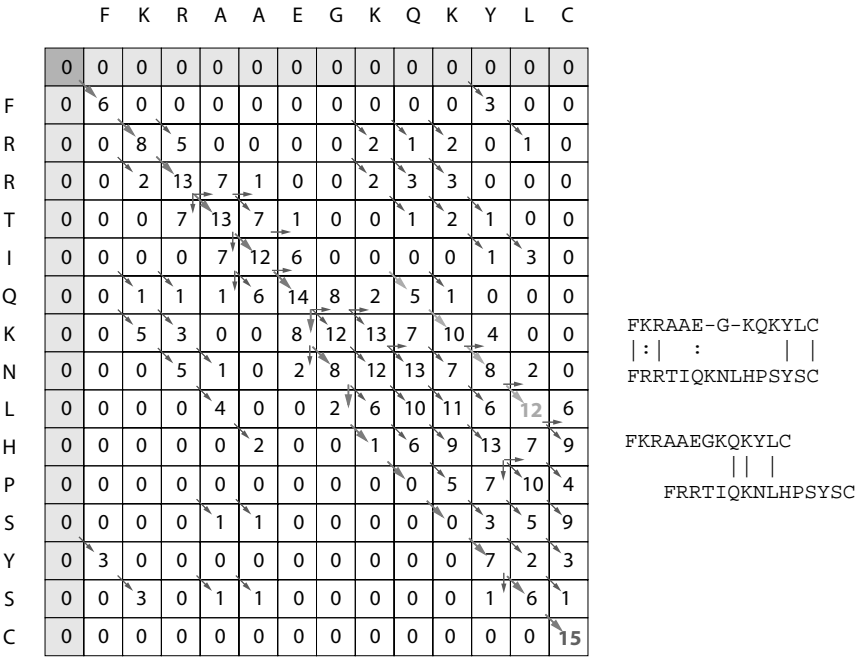


Figure 2.10 Alignments using the local method of Smith & Waterman. This algorithm finds the overall alignment, but also detects others, such as the one indicated on the right, whose score is 12.

find the one with the highest score. Using that score, we can identify the most homologous segment in the two sequences, as was done using the global alignment method. It is also possible to examine all cells whose scores exceed a threshold value chosen by the user. This finds alternative alignments, like the one indicated in Figure 2.10, which sometimes can reveal interesting biological properties, such as imperfect repetitions of a pattern in the sequence being studied.

Dynamic programming has yielded numerous variants and improvements in these methods. The principle, which is used in most programs accessible online via the web, involves calculating the cost of insertions/deletions. In the two above-described approaches for an insertion or deletion of length n , the cost is $n \times \Delta$. This linear cost in terms of n is simple to compute, but not very realistic from the biological point of view. The introduction of a discontinuity in the alignment is the penalty. However, the length of this discontinuity is not critical. In protein sequences, these breaks in alignment most often occur in peptide chain loops exposed on the surface of the 3D molecule. Indeed, the length of these loops may be relatively variable without at all disturbing the heart of the molecule, where practically no insertions or deletions are found. To take this into account, we use an affine cost for insertions/deletions of length n :

$$\text{cost} = \alpha + \beta n$$

The parameter α is the cost of opening the discontinuity, and β is the cost of extending it. Again, choosing α and of β is relatively empirical; $\alpha < \beta$, and $\alpha < \min \{M(a, b)\}$, where M is the comparison matrix (BLOSUM, PAM, etc.) For example, with the BLOSUM62 matrix, one commonly uses $\alpha = -10$ and $\beta = -2$, or neighboring values. Applying this cost function does not question the use of dynamic programming; it just makes the procedure of calculating the matrix $T(i, j)$ slightly more complicated.

2.7 Fast heuristic methods

The two variants of dynamic programming just described are efficient and optimal, but their costs in terms of calculation time and memory are quadratic; specifically, $O(n, m)$, where n and m are the respective lengths of the two sequences being studied. This cost is very reasonable when comparing two sequences of several hundred to several thousand nucleotides or amino acids, but becomes high when carrying out a systematic search of sequence databases containing around 10^9 to 10^{10} residues. However, this latter type of search is becoming one of the dominant bioinformatics applications. Indeed, massive

genomics sequencing programs lead to the identification ‘by the kilometer’ of thousands of genes whose functions are most often unknown. Each new gene must be compared with the databases in order to try to find a known homolog, which allows attribution of a possible function. This need has led to the development of much more rapid heuristic methods that provide high-quality (although usually not optimal) alignment solutions at very reasonable cost.

The diagonal strip method: FASTA and FASTP

The inspiration for this first heuristic approach is found in a graphic representation of the homology between two sequences S_1 and S_2 , known as a *dot-plot*. If their respective lengths are n and m , they describe a matrix of $n \times m$ pixels. Each pixel of coordinates i, j of the image generated is either black or white, according to whether the value of the comparison matrix, $M(S_1(i), S_2(j))$ is above or below a given threshold. In other terms, if $S_1(i)$ and $S_2(j)$ are identical or similar, the corresponding pixel will be colored. When two sequences S_1 and S_2 share homologous regions, the regions appear as diagonal lines on the image (*cf* Figure 2.11).

When two sequences share regions that present homologies, we notice the presence of very large diagonal segments localized within a narrow band. This

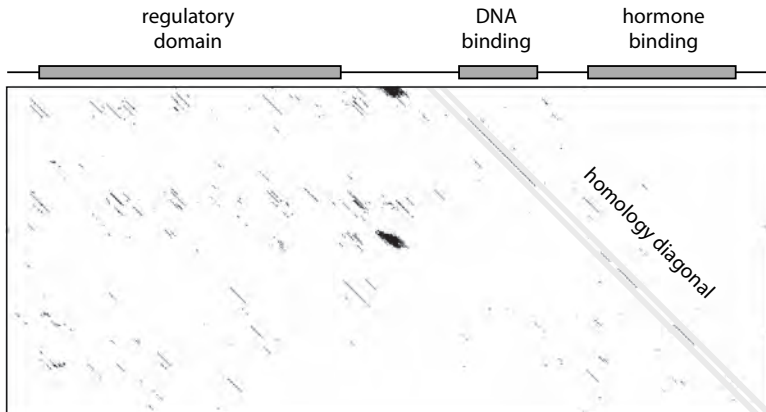


Figure 2.11 Dot-plot comparison of the sequence of the androsterone nuclear receptor (abscissa) with that of another nuclear receptor of unknown function (ordinate). The positions of the functional domains of the androsterone receptor are indicated above. The most significant alignment appears as a series of diagonal lines, which are highlighted in the figure. The unknown receptor is shorter than the androsterone receptor and does not seem to have a regulatory domain.

property has been used to develop an overall heuristic alignment method for identifying the diagonals. Two slightly different variants of this method exist, one for protein sequences, called FASTP (for FAST Protein alignment) and the other for DNA sequences, called FASTA (for FAST nucleic Acid alignment).

To improve asymptotic efficiency with respect to dynamic programming methods, it is of course necessary to avoid calculating the entire dot-plot, which would clearly be $O(m, n)$ for two sequences of respective length m and n . FASTA and FASTP simplify this step by the following:

- 1. First, tabulate the lists of k -tuples of nucleotides or amino acids appearing in sequence S_1 and of the positions at which they appear. Typically, the value of k will be 4 to 6 for nucleic acids and 2 for proteins, because of the smaller size of the alphabet used (4 nucleotides compared with 20 amino acids). For example, for the following DNA sequence,

ATGGCGCGATGGAAAAAAT

we obtain the following list of 4-tuples and their positions:

AAAA	13, 14, 15	CGAT	7	GAAA	12	TGGA	10
AAAT	16	CGCG	5	GATG	8	TGGC	2
ATGG	1, 9			GCGA	6		
				GCGC	4		
				GGAA	11		
				GGCG	3		

This list is constructed in linear time as a function of the length n of S_1 .

- 2. Create a score table for the $n + m - 1$ diagonals of the matrix. All values in this table are initialized at zero.
- 3. Sweep through the S_2 sequence looking for k -tuples tabulated for S_1 . Each time an S_1 k -tuple is found in S_2 , locate the diagonal(s) on which it appears. Each of them is identified in an unequivocal manner by the difference $i - j$, where i and j are the respective positions of the common k -tuple in S_1 and S_2 .
- 4. Once the totality of S_2 has been analyzed, look for a maximum homology band like the one highlighted in Figure 2.11, using the score table of the diagonals.
- 5. Construct the alignment limited for this diagonal band, attaching the various k -tuples end-to-end. It is possible to refine the alignment by using the Needleman & Wunsch method to fill in the remaining gaps.

The behavior of this method crucially depends on the k parameter. Increasing its value accelerates the search, since the mean number of occurrences of a k -tuple in a sequence of given length diminishes as k increases. Thus, there will be far fewer cases to treat during step 3, which is one of the costliest in terms of time. The price to be paid is a loss in the sensitivity of the method, since there is a risk of missing weak homologies for which there are only a few conserved k -tuples. If the identity rate between two DNA sequences is 60 percent, out of 100 nucleotides, an average of 13 quadruplets will be conserved ($0.6^4 = 0.129$), versus only 1 if the identity rate falls to 35 percent⁵.

Applying these simplifications renders FASTA and FASTP very efficient in aligning relatively conserved sequences. But for weakly homologous protein sequences, the limiting number of common k -tuples often requires resorting to the value $k = 1$, which causes FASTP to lose much of its efficiency. In addition, using a comparison matrix such as BLOSUM62 to account for replacements by chemically close amino acids further complicates the task. In order to search all protein databases systematically, FASTP therefore remains a useful but – since it is rather slow – limited tool. This limitation is not likely to be lifted by the constant increase in computer power, since it is offset by an at least equally rapid increase in the size of the databases to be searched.

K-tuples: a general search method

The FASTA strategy, which consists in tabulating a list of k -tuples that overlap in a sequence, is a very general strategy for seeking partial or total sequence overlaps originally proposed by J. Ninio and J.-P. Dumas in 1982. For example, it is also used in contig reconstruction algorithms during genome assembly (see the chapter on sequencing). Much longer k -tuples are used for this latter application, since we are looking for sequence identities, not homologies. When two clones overlap, one would expect that the sequence fraction they have in common be identical (within the limits of sequencing errors). Thus, we generally try to use much longer k -tuples that are present only once in the entire genome. According to the hypothesis of the equal distribution of the four nucleotides (for a discussion of DNA composition, see Chapter 5), we take $k > \log_4 L$, where L is the genome length, which is 11 nucleotides for a bacterial genome and 16 for the human genome.

⁵ Applying this simplistic calculation supposes that conserving a given position in the sequence is independent of the position of its neighbors, which is generally not the case. Fortunately for this method, identity regions often form compacted zones separated by more variable regions. This reflects the fact that the selective pressure of evolution is not uniform, but preferentially exercised at sites that are essential for the functioning of the molecule. This explains why FASTA and FASTP ‘work’ so well in spite of the drastic simplification represented by k -tuples.

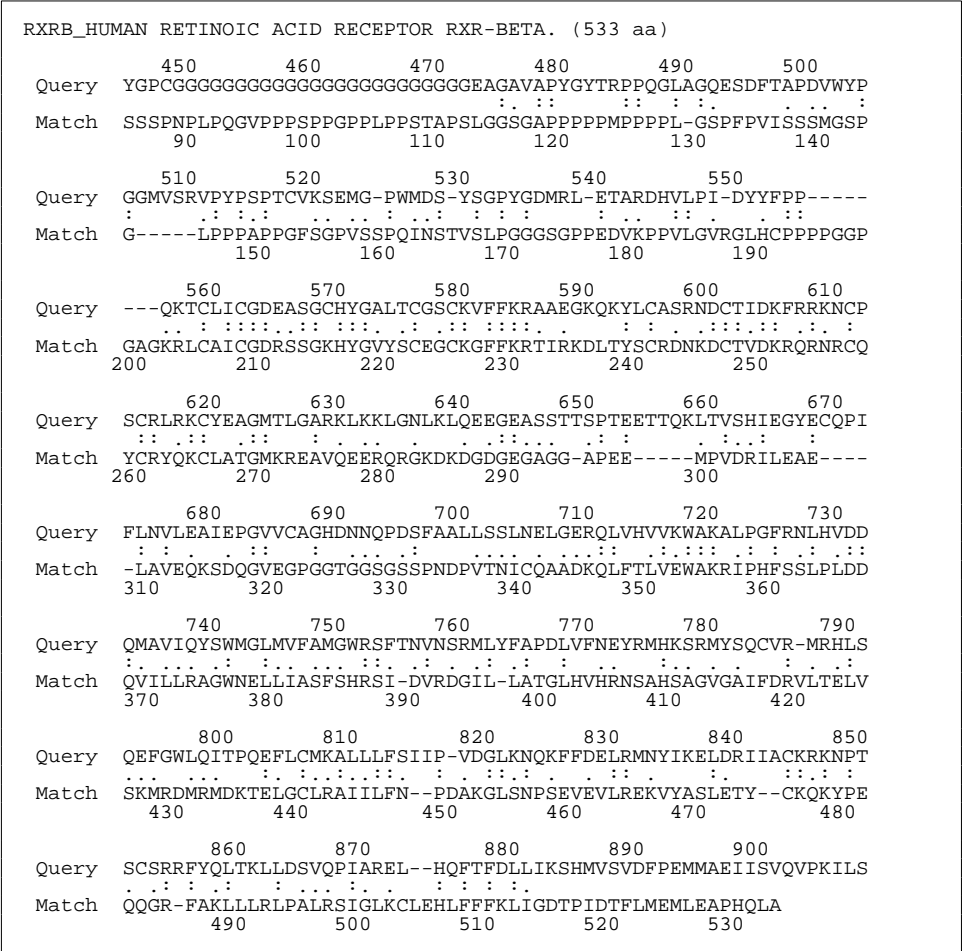


Figure 2.12 Example of an alignment produced by FASTP between the androsterone receptor sequence ('Query') and the retinoic acid receptor sequence RXRB ('Match').

In principle, *k*-tuples may be tabulated either for the sequence being studied, as FASTA does it, or for the entire sequence database. The latter approach was used very successfully when sequence databases first became available online. The *k*-tuple table, which must be recalculated each time the database is updated, is a gigantic index of the positions of the various *k*-tuples. Thus, for example, for the FFKR tetrapeptide, the table contains a list of pointers pointing toward all the sequences that contain it, with the corresponding position in the following sequence:

...

FFKQ → ...

FFKR → ... → androsterone receptor, position 582 → protein X, position y → ...

FFKS → ...

...

This strategy seems attractive, since simply reading the index table provides the list of candidates that are homologous to the sequence being studied. However, this is rarely done these days, for three reasons:

- Database size increases much more rapidly today, and the quantity of work required to retabulate the occurrence of k -tuples at each update becomes prohibitive.
- In order to vary the k parameter, as many index tables as desired k values must be compiled, which makes the calculations more complicated. In addition, there will be few k -tuples for low k -values, and each index table entry will contain a great number of references to the database, which makes the process very long.
- Because of their large size, sequence databases cannot completely fit into the main memories of computers, and are therefore generally stored in external disks. Consulting database sequences from the table of k -tuples is an operation that requires direct access (random) to the database entries, which is very inefficient. In the inverse strategy utilized by FASTA, sweeping through the base to look for k -tuples of the sequence being studied is a *sequential* operation, which is much less costly in terms of access time.

For these same reasons, the BLAST algorithm which we will now consider, also involves tabulation of the sequence studied; not that of the database.

Homology by pieces: the ultra-rapid BLAST method

In order to further improve efficiency, another approach was developed specifically for use in aligning protein sequences. Like the Needleman & Wunsch algorithm, FASTP is an overall alignment method. The homology sought often may be weak and localized, hence difficult to detect by the two methods mentioned above. BLAST (Basic Local-Alignment Search Tool) is a heuristic method specifically developed to compare an unknown protein sequence with a set of sequences found in protein databases. BLAST detects locally homologous short segments (a few dozen amino acids) at a linear cost proportional to the size of the database searched. The main application of BLAST is the use of known

proteins to rapidly detect significant homologies, in the attempt to attribute a function to the protein being studied. BLAST uses amino acid comparison matrices such as BLOSUM62 to evaluate the pertinence of these alignments and to attribute a ‘plausibility’ score to the obtained alignment in evaluating the probability that the observed resemblance is due to chance⁶.

Like FASTA and FASTP, the BLAST principle is based on the use of k -tuples of amino acids in the sequence analyzed in order to reduce complexity. BLAST still uses long k -tuples (at least four amino acids) to maintain high selectivity; that is, so as not to have to deal with too many chance occurrences of these same k -tuples when sweeping through the sequence database. In order to prevent this high selectivity from imposing a loss of sensitivity, a list of k -tuples sufficiently homologous to each k -tuple is compiled. For this purpose, a comparison matrix is used that allows calculation of a score corresponding to the alignment without inserting the two k -tuples.

Taking the example of the androsterone receptor with $k = 4$, the FFRK quadruplet mentioned above appears in position 582. Aligning it with itself, the BLOSUM62 matrix yields a score of 22:

FFKR
FFKR

score: 6655 total: 22

Looking for all tetrapeptides whose alignment scores are superior to a threshold H , a user-defined parameter, for example, with $H = 17$, the following homologous peptides for FFKR result:

Score = 22	Score = 19	Score = 18	Score = 17
FFKR	YFKR	FFQR	WFKR
	FYKR	FFER	FWKR
	FFRR	FFKQ	FFNR
	FFKK		FFSR
			FFKN
			FFKE
			FFKH

This operation is repeated for all the k -tuples of sequence S , which allows construction of a list L of k -tuples that are ‘close’ to S_1 . For any k -tuple w of

⁶ Since databases today contain several dozen billion nucleotides and amino acids, the sporadic appearance of local alignments that are credible to the eye but still due to chance is practically inevitable. More thorough critical analysis by the biologist is thus usually necessary when the homology score is low.

L there exists a k -tuple ν of S_1 such that the score $(\nu, w) \geq H$. The position at which the k -tuple appeared in sequence S also figures in list L . The position of the FFKR pattern in the androsterone receptor sequence is associated with all the quadruplets in the example in the above table.

The list L of k -tuples is then used to scour the sequence databases. Each time a k -tuple belonging to L is located, a homology begins that guarantees a minimum score for H . An attempt is then made to extend the alignment of S , using the sequence found around the homologous k -tuple. The BLAST strategy is to seek *maximal homology segments*; the alignment is prolonged progressively, first on one side, then the other, using the comparison matrix to calculate the scores. The method employed proceeds according to the following scheme:

```

max_score ← initial alignment score of the two  $k$ -tuples
max_length ←  $k$ 
Do
  Prolong the alignment of a residue on each sequence
  Calculate the resulting score, extended_score
  If extended_score > max_score then
    max_score ← extended_score; increment max_length
While extended_score ≥ max_score-tolerance

```

The tolerance factor introduced in checking the loop allows elongation of the alignment even if the score drops slightly below the current maximum, in the hope that when elongating later segments, favorable coincidences of amino acids will be encountered that raise the score. If the score descends below the `max_score-tolerance`, BLAST stops and recovers the maximal score value and the maximum length encountered. This operation is carried out for both sides of the initial k -tuple, so as to obtain local alignment of the maximum score between the two sequences, known as the Maximum Scoring Pair (MSP). Finally, BLAST displays all local alignments whose scores exceed a threshold set by the user.

The cost in terms of calculation time for a BLAST search sensitively depends on the various parameters of the heuristic algorithm: k , the length of the k -tuples H , the homology score threshold, and to a lesser degree, the tolerance utilized for extension, starting from the k -tuples found. For example, if the H threshold is raised, it will reduce the number of k -tuples in list L , since only the most homologous k -tuples will be retained. The number of coincidences between the database examined and list L will be reduced proportionally, accelerating the calculation. Regrettably, when H is increased, there is also the risk of allowing certain segments whose homologies are more distant to escape, yielding lower scores. A compromise must be found between the sensitivity of

the method and its effectiveness. After much statistical and empirical testing, the following parameter values are now conventionally used to search protein sequence databases with BLOSUM62 or PAM250:

$$K = 4 \quad H = 17 \quad \text{tolerance} = 20$$

Using these parameters, a BLAST search is around one order of magnitude faster than using FASTP, and two to three orders of magnitude faster than with the dynamic programming algorithms of Smith & Waterman or Needleman & Wunsch. As is the case for FASTA and FASTP, even though today there are numerous variants and improvements for BLAST, all start from the basic principle described above. The most recent specifically allow fusing several compatible local alignments into a single alignment, taking short insertions and deletions into account to a limited degree.

2.8 Sensitivity, specificity, and confidence level

Using FASTA, FASTP, or BLAST to search sequence databases almost always yields a series of more or less high-score alignments. In view of this result, the biologist must consider several questions:

- How much confidence can one have in the alignment result produced by the program? In more rigorous terms, is it possible to evaluate the probability that the result obtained is random?
- Has the program identified all the sequences that are homologous⁷ to the sequence sought?
- For what percentage of the sequences found is the homology really significant from the biological point of view?

These three questions concerning the level of confidence, sensitivity, and specificity of the heuristic algorithm used are partially related to each other and depend on the parameters applied. Sensitivity and specificity are somewhat antagonistic, since if one relaxes the stringency of the parameters, for example, by lowering the BLAST homology threshold H , there is a greater risk of obtaining false positives. It was thus crucial to devise a tool capable of evaluating the likelihood of an alignment. A statistical model of sequence alignment without insertions and deletions developed by the authors of BLAST provides the means

⁷ By *homologous sequences* is meant all those for which the Smith & Waterman algorithm yields a local homology score equal or superior to a threshold used by the heuristic algorithm.

for conducting this analysis. However, a detailed presentation of it would extend beyond the framework of the present discussion; therefore only its main results are presented here:

Briefly, a comparison matrix (BLOSUM62, for example) is used to compare a sequence of length n with a database sequence of length m . Take the case in which the sequence studied presents no ‘biological’ homology with the database sequences, in order to look only at ‘fortuitous’ similarities. We are interested in the distribution of the number $N(\sigma)$ of alignments whose scores exceed the value σ . Since it is roughly a Poisson distribution, the following expression describes the probability of an accidental alignment of score greater than σ :

$$p(\text{score} > \sigma) = 1 - e^{-E(\sigma)}$$

$E(\sigma)$ is the expected value of $N(\sigma)$ where λ and K are positive constants that depend on the frequencies of the amino acids in the databank and on the coefficients of the comparison matrix. One can either calculate λ and K explicitly⁸ or adjust them from a real distribution of the scores obtained with a sufficiently large sample of non-homologous sequences. The resulting probability thus has the expression:

$$p(\text{score} > \sigma) = 1 - e^{-K \cdot m \cdot n \cdot e^{-\lambda \sigma}}$$

Its shape as a function of the threshold σ may be seen in Figure 2.13, in which it drops rather sharply over a given score value.

Based on its score, it is possible to determine the probability that each alignment produced by BLAST is purely accidental. The best ‘biological’ alignments have infinitesimal probabilities of random statistical occurrence (10^{-20} to 10^{-100}), generally leaving no doubt of their significance. At around 10^{-1} to 10^{-5} , there may be a degree of ambiguity, in what is known as a *twilight zone*, in which case *a posteriori* validation by the biologist is generally necessary.

Application: BLAST search for homologs of the androsterone receptor

As stated above, the main function of BLAST is to carry out systematic searches of databases. We have undertaken the same exhaustive search for homologs of the human nuclear androsterone receptor, which is 919 amino acids long. To

⁸ For those who are curious, λ is the strictly positive solution of the equation
$$\sum_{a,b \text{ amino acids}} p(a)p(b)e^{\lambda M(a,b)}$$
 with $p(a)$ being the frequency of amino acid a and $M(a, b)$ being the coefficient of the comparison matrix (BLOSUM62). There is no closed form for K , which is the sum of the terms of a converging series that depends on the same $p(a)$ and $M(a, b)$. Since the convergence is rapid, it is possible to directly calculate a value close to K by summing the first terms.

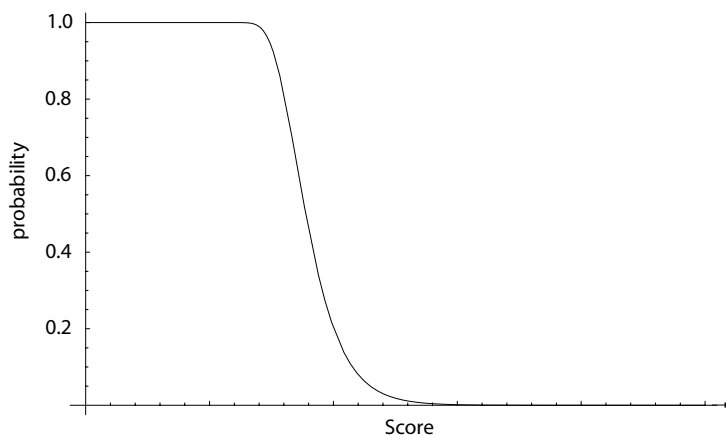


Figure 2.13 Value of the probability of an alignment score higher than a threshold σ as a function of that same threshold. The abscissa is expressed in arbitrary units.

do this, we selected a non-redundant⁹ and documented database of well-characterized protein sequences known as SWISSPROT. Its May 2005 version contained 180,652 entries, for a total of 65 million amino acids.

During analysis of the database, BLAST found more than 1.8 million ‘positive’ k -tuples for which it attempted extension, for 4,536 of which it was possible to prolong the alignment. Of these alignments, 268 satisfied the selected minimal score criteria, among which 45 implicated protein sequences of human origin, including (of course) the sequence of the androsterone receptor (*cf* Figure 2.14). Some of these sequences code for known nuclear receptors involved in responses to various stimulatory molecules, such as steroid sex hormones, vitamin D, corticoids, and thyroid hormones. This functional relationship is an *a posteriori* verification of the postulate stated at the beginning of this chapter:

More interestingly, BLAST analysis also identifies at least as many highly homologous protein sequences of previously unknown function. Their high level of similarity, evaluated by the alignment score, would seem to indicate that they are nuclear receptors. They are known as ‘orphan receptors’ (see an example in Figure 2.15). In most cases, the molecule that plays the role of cofactor, such as androsterone or corticosteroids, is unknown. The discovery of orphan receptors was made possible by bioinformatics analysis; today they pose numerous questions, as well as suggesting various therapeutic perspectives.

⁹ Today, protein sequences are most often obtained by *in silico* translation of DNA open reading frames. Numerous copies of some DNA sequences are present in databases (genomic sequences, cDNA sequences, EST, etc . . .) derived from several tissues. The SWISSPROT database is ‘curated’ so as to render it non-redundant. It thus contains only one entry per protein, plus comments concerning the protein’s activity and functional domains. This sorting and annotation work explains the reduced size of the SWISSPROT database compared with DNA sequence databases.

sequence homology \Leftrightarrow functional homology

Sequences producing significant alignments:		Score (bits)	E Value
ANDR_HUMAN	ANDROGEN RECEPTOR	1499	0.0
PRGR_HUMAN	PROGESTERONE RECEPTOR	438	e-122
MCR_HUMAN	MINERALOCORTICOID RECEPTOR	389	e-107
GCR_HUMAN	GLUCOCORTICOID RECEPTOR	386	e-107
ESR2_HUMAN	ESTROGEN RECEPTOR BETA (ER-BETA).	156	2e-37
ESR1_HUMAN	ESTROGEN RECEPTOR (ER) (ESTRADIOL RECEPTOR).	154	6e-37
ERR2_HUMAN	STEROID HORMONE RECEPTOR ERR2	141	6e-33
ERR1_HUMAN	STEROID HORMONE RECEPTOR ERR1	138	5e-32
ERR3_HUMAN	ESTROGEN-RELATED RECEPTOR GAMMA.	134	1e-30
RXRB_HUMAN	RETINOIC ACID RECEPTOR RXR-BETA.	113	2e-24
RXRG_HUMAN	RETINOIC ACID RECEPTOR RXR-GAMMA.	109	2e-23
RXRA_HUMAN	RETINOIC ACID RECEPTOR RXR-ALPHA.	108	5e-23
HN4G_HUMAN	HEPATOCYTE NUCLEAR FACTOR 4-GAMMA	105	4e-22
EAR2_HUMAN	ORPHAN NUCLEAR RECEPTOR EAR-2	105	5e-22
COT2_HUMAN	COUP TRANSCRIPTION FACTOR 2	104	8e-22
COT1_HUMAN	COUP TRANSCRIPTION FACTOR 1	102	3e-21
NR41_HUMAN	ORPHAN NUCLEAR RECEPTOR HMR	98	6e-20
RRG2_HUMAN	RETINOIC ACID RECEPTOR GAMMA-2	97	1e-19
RRG1_HUMAN	RETINOIC ACID RECEPTOR GAMMA-1	97	1e-19
HN4A_HUMAN	HEPATOCYTE NUCLEAR FACTOR 4-ALPHA	96	2e-19
NRH3_HUMAN	OXYSTEROLS RECEPTOR LXR-ALPHA	94	9e-19
RRB2_HUMAN	RETINOIC ACID RECEPTOR BETA-2	94	1e-18
NR43_HUMAN	NUCLEAR HORMONE RECEPTOR NOR-1	93	2e-18
NR42_HUMAN	ORPHAN NUCLEAR RECEPTOR NURR1	92	3e-18
RRA_HUMAN	RETINOIC ACID RECEPTOR ALPHA	92	5e-18
NR52_HUMAN	ORPHAN NUCLEAR RECEPTOR NR5A2	92	5e-18
STF1_HUMAN	STEROIDOGENIC FACTOR 1	89	3e-17
TR4_HUMAN	ORPHAN NUCLEAR RECEPTOR TR4	86	3e-16
NR13_HUMAN	ORPHAN NUCLEAR RECEPTOR NR1I3	85	8e-16
RORB_HUMAN	NUCLEAR RECEPTOR ROR-BETA	82	4e-15
ROR4_HUMAN	NUCLEAR RECEPTOR ROR-ALPHA-4	82	5e-15
ROR3_HUMAN	NUCLEAR RECEPTOR ROR-ALPHA-3.	82	5e-15
ROR2_HUMAN	NUCLEAR RECEPTOR ROR-ALPHA-2.	82	5e-15
ROR1_HUMAN	NUCLEAR RECEPTOR ROR-ALPHA-1.	82	5e-15
NRH2_HUMAN	OXYSTEROL RECEPTOR LXR-BETA	80	2e-14
RORG_HUMAN	NUCLEAR RECEPTOR ROR-GAMMA	80	2e-14
VDR_HUMAN	VITAMIN D3 RECEPTOR	77	2e-13
THA2_HUMAN	THYROID HORMONE RECEPTOR ALPHA-2	73	2e-12
THA1_HUMAN	THYROID HORMONE RECEPTOR ALPHA-1	73	2e-12
THB2_HUMAN	THYROID HORMONE RECEPTOR BETA-2.	73	3e-12
THB1_HUMAN	THYROID HORMONE RECEPTOR BETA-1.	73	3e-12
PPAT_HUMAN	PEROXISOME PROLIFERATOR ACTIVATED RECEPTOR	70	2e-11
NRD2_HUMAN	ORPHAN NUCLEAR RECEPTOR NR1D2	69	5e-11
NRD1_HUMAN	ORPHAN NUCLEAR RECEPTOR NR1D1	69	5e-11
ERBL_HUMAN	TRANSFORMING PROTEIN ERBA HOMOLOG	38	0.12

Figure 2.14 Human protein sequences identified by BLAST as presenting homologies with the androsterone nuclear receptor. The columns on the right indicate alignment scores obtained using the BLOSUM62 matrix and the mathematical expectance $E(\sigma)$ of the number of random alignments of a higher score than that value.

- What are the cofactors of these receptors?
- What are their functions? With which metabolic and/or physiological response(s) are they associated?
- Is it possible to modulate their functions using exogenous molecules? Could these molecules constitute a new class of medicines? Most known

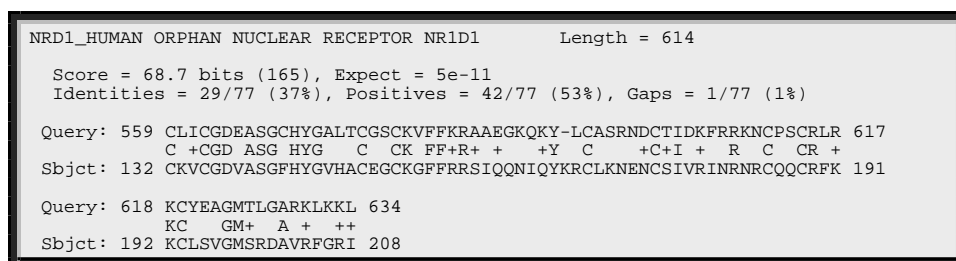


Figure 2.15 Example of local alignment produced by BLAST. It concerns the homology between the androsterone nuclear receptor and an orphan nuclear receptor of unknown function.

nuclear receptors have turned out to be interesting therapeutic targets, and the discovery of new receptors enlarges the perspectives accordingly.

In order to identify the functions of orphan receptors, and in general, the functions of proteins newly identified using bioinformatics, a new methodology has been devised: *reverse genetics*. Instead of looking for genes associated with a phenotype, then mapping and isolating them in order to derive their sequence, we start with bioinformatics sequence analysis in an attempt to discern the function or phenotype associated with an interesting candidate. For example, we can inactivate or modify the corresponding mouse gene, if it exists, then try to determine the consequences. One can also look for molecules capable of binding to a protein of unknown function, in order to study their effect in the animal, and eventually in humans.

2.9 Multiple alignments

Until now, we have not considered the problem of aligning segments pairwise. When carrying out a database search, several sequences that present similarities with the sequence being studied are often encountered. To compare all these homologous sequences among themselves simultaneously, it is natural to try to align N sequences together, in the manner shown in Figure 2.4. This problem of multiple alignment has many interesting biological applications:

- Multiple alignments allow detection of regions that are conserved over evolution. These regions very often correspond to domains associated with a key function of the molecule.
- Strictly conserved amino acids or nucleotides, like those that appear in gray in Figure 2.4, often play a direct role in the function. Using multiple alignments, we are able to identify amino acids implicated in the catalysis or selective binding of a substrate by an enzyme.

- They sometimes permit *a posteriori* validation of an alignment found by BLAST or FASTA, whose low score is in the ‘twilight zone,’ where it is difficult to ascertain its biological significance. If the regions and/or homologous amino acids are precisely the same as those conserved over evolution, there is a strong likelihood that the alignment found reflects biological reality.

Multiple optimal alignment

Dynamic programming methods (Needleman and Wunsch and Smith and Waterman) may in principle be directly generalized to N sequences. The notion of alignment score as now defined must be extended to include use with BLOSUM and PAM comparison matrices. Given a multiple alignment column, the corresponding score is generally defined as the sum of the scores of all pair-wise combinations of sequences in the alignment:

$$\text{Score}(\text{column } i) = \sum_{1 \leq k < l \leq N} M(S_k(i), S_l(i))$$

For insertions/deletions, we extend matrix M as above, defining:

$$\forall a, M(-, a) = M(a, -) = \Delta < 0, \text{ insertion penalty.}$$

The same matrix column can contain an insertion in several sequences. We can thus also define a score for the alignment of two gaps, which we generally take to be zero:

$$M(-, -) = 0$$

The fact of rendering this cost zero has the effect of favoring the grouping of insertions in the same columns in the multiple alignment. This is rather natural because, as we have already discussed, for proteins, insertions/deletions are usually grouped in polypeptide loops located on the surface and not dispersed in the interior of the molecular sequence.

These extensions in the calculation mode and of the comparison matrix permit calculation of an overall score for a multiple alignment containing insertions and deletions. This method of calculating the score in a multiple alignment is called the *sum of the pairs score*. It is thus possible to search for an optimal alignment, for which, by definition, the score of the sum of the pairs will be maximal.

To find it, we always calculate a table T of the partial scores. This table is now N -dimensional, with $T(i_1, i_2, \dots, i_N)$ = the best score for a partial alignment with the sub-sequences $\{i_1, i_2, \dots, i_N\}$. This table is calculated in the same way, taking the maximum of an alternative to $2^N - 1$ terms, and describes all pos-

sible combinations of insertions and ‘gaps’ in the N sequences. The following is an example of the seven terms of the alternative for three sequences, S_1 , S_2 , and S_3 , where the score function is what was defined above for a column and where a dash (‘-’) designates a gap in the alignment.

$$T(i, j, k) = \max \left\{ \begin{array}{l} T(i-1, j-1, k-1) + \text{score}(S_1(i), S_2(j), S_3(k)) \\ T(i, j-1, k-1) + \text{score}(-, S_2(j), S_3(k)) \\ T(i-1, j, k-1) + \text{score}(S_1(i), -, S_3(k)) \\ T(i-1, j-1, k) + \text{score}(S_1(i), S_2(j), -) \\ T(i, j, k-1) + \text{score}(-, -, S_3(k)) \\ T(i, j-1, k) + \text{score}(-, S_2(j), -) \\ T(i-1, j, k) + \text{score}(S_1(i), -, -) \end{array} \right.$$

While in principle this method is only slightly more complex than for two sequences, the computing time clearly becomes prohibitive as N increases. For N sequences of length n , there are n^N elements to be calculated in table T , for each of which it is necessary to evaluate $2^N - 1$ scores, each requiring the summation of $N(N-1)/2$ terms, which engenders an overall algorithmic cost of

$$\frac{N(N-1)}{2} n^N (2^N - 1)$$

The required amount of memory space itself increases ‘only’ as n^N . Whereas one second of computer time is needed to align two protein sequences of 100 amino acids, ten minutes are required to align three sequences, and nearly three days to align four. For nine or more sequences, the computation time required exceeds the age of the universe . . . Clearly, this method cannot be used in routine practice. Today, families of homologous proteins, such as those of nuclear receptors, consisting of several dozen or even several hundred members, are common. The development of high-performance, good quality, multiple alignment algorithms is still an open problem and an area of active research. Numerous heuristic strategies exist that generally function in a satisfactory manner when the set of proteins or DNA to be aligned consists of sequences that are rather similar to each other and contain few insertion and deletion sites. These algorithms function with more difficulty when the homology is weaker.

‘Profile’ alignment: a simple and efficient heuristic method

Since it is difficult to undertake simultaneous alignment of N sequences directly, a simple approach is to progressively align them pairwise, starting with the most similar. Other sequences are then aligned with the first alignments, and others

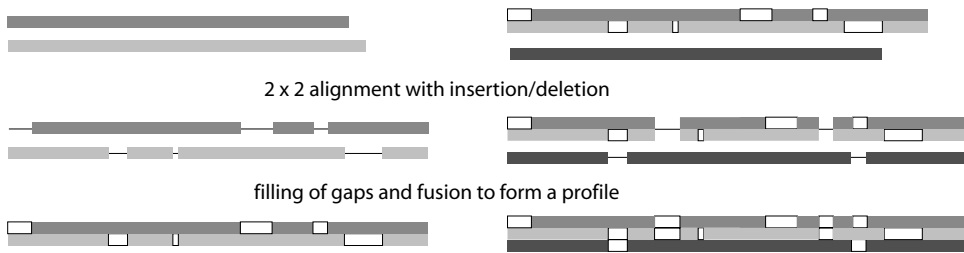


Figure 2.16 Profile alignment. On the left side, two sequences are aligned using a classical dynamic programming method. The gaps are then filled by replacing them with a neutral character. The profile obtained may then be aligned to another sequence (right side), or to another profile.

are aligned with those alignments, until all the sequences are incorporated into an overall alignment. These intermediate alignments are called *profiles*. Only two elements, either sequence/sequence, sequence/profile, or profile/profile, are aligned at each step, using a method derived from the Smith & Waterman method, which considerably limits the complexity of the algorithm.

After the two sequences have been aligned, the gaps are filled with a neutral character that does not correspond to any amino acid or nucleotide (for example, – or ×). We thus obtain two sequences of the same length that constitute a profile. We can then align two profiles, or a profile and a sequence, with each other, still using the dynamic programming approach. It is enough just to modify calculation of the alignment score by means of the sum of the pairs method, as described above.

One of the properties of this method is that gap creation is definitive (‘once a gap, always a gap’). If an insertion is created in a sequence during an early pairwise alignment step, it cannot be abolished in a later step, even if that would be favorable in terms of the overall score. This is certainly a weakness in the method, and in order to minimize its harmful effects, it is crucial to avoid aligning the least homologous sequences in the early phases of multiple alignment. Indeed, not doing so would risk introducing insertions at the wrong places. Since this process is irreversible, the overall alignment quality would be strongly affected. All methods of profile alignment use a *phylogenetic* approach, which follows the tree of evolutionary ancestry among the various sequences (*cf* the next paragraph: *Construction of a phylogenetic tree*).

The first step is usually to reconstruct this tree. The approach used consists in first carrying out pairwise $N(N - 1)/2$ sequence alignments. Starting with the scores of these alignments, we construct a table of the evolutionary distances between each pair of sequences. The lower the score, the greater the corresponding distance. The function employed to determine the distance between two sequences i and j starting from an alignment score is necessarily somewhat arbitrary. Among those most frequently used are expressions of the following type:

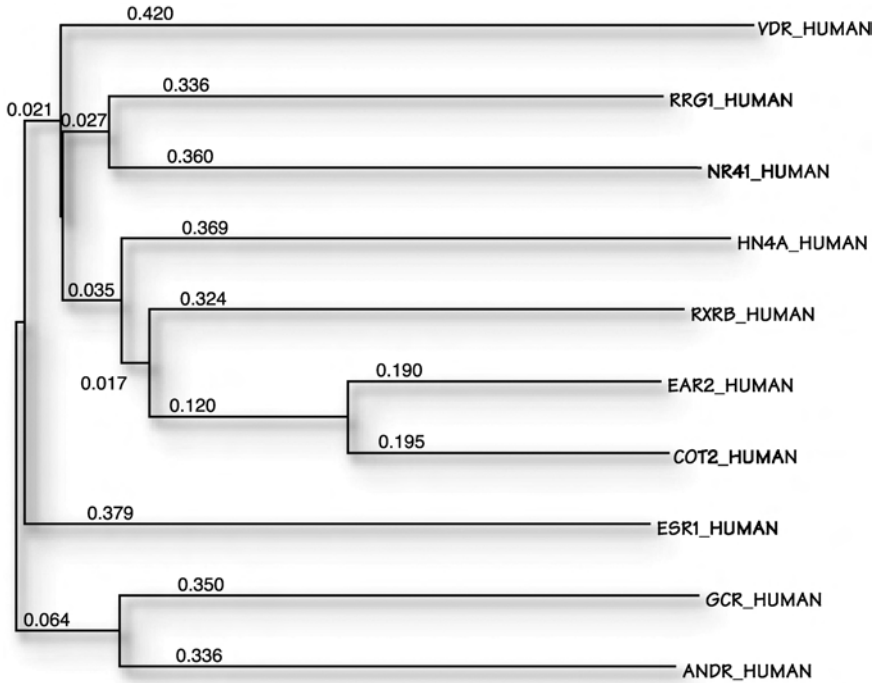


Figure 2.17 Example of a phylogenetic tree constructed from the ten sequences of nuclear receptors listed in Figure 2.14. The lengths of the tree branches are proportional to the evolutionary distances (indicated above). The multiple alignment constructed from this tree is indicated in Figure 2.18.

$$D_{i,j} = -\log\left(\frac{S - S_{\text{random}}}{S_{\text{id}} - S_{\text{random}}}\right),$$

where S is the alignment score, S_{random} is the score obtained after random permutation of the amino acids of one of the two sequences, and S_{id} is the maximum score obtained by aligning one of the two sequences with itself.

The table of distances $D_{i,j}$ is then used to construct a tree that serves as a guide used for iterative alignment. We start by aligning the sequences that correspond to the closest branches, following the branches to the roots of the tree. Figures 2.17 and 2.18 show an example of a phylogenetic tree and of multiple alignment constructed using the CLUSTAL program. CLUSTAL (for Cluster Alignment) is one of the most popular multiple alignment programs, available on most computer platforms and online on various web servers.

```

OT2_HUMAN  WARNIPFFDQITDQVALLRLTWSSEFVLNAAQCSMPHLHVP-LLAAAGLHASPMSADRVVAFMDHIRIFQEQVEKALKALHVDSAEYSCLKATVLF
AR2_HUMAN  WARRHG-FFFEVPAVDQVALLRMSWSEFVLNAAQAALPLHTAP-LLAAAGLHAAPMAAERAVAFMDQVRAFQEQVDKLGRLQVDSAEYGLKATALF
XRE_HUMAN  WAKRIEHFSSPLDDQVILLRAGNELLLASFHSRSDVRDC--ILLATGLHVHRNSAHSAGVGAIFDRVLTELVSKMRRMDKTELGLKRAILIF
N4A_HUMAN  WAKYIYAFDFEPLDDQVALLRAHAGEHLLLGATKRSMEVKDV--LLGNDYIVPRHCPKELAEMSRVSRILDELVLFPQELQIDNNEYAYLKAIFFF
RG1_HUMAN  FAKRLPCFTGCSIAQDTLLKKAACLDILMLRICTRYTPEQDT--MTFSDGLTLNRTQMHNAGFGPLTD-LVFAFAGQLLPLEMDDTETGLLSAICLI
R41_HUMAN  WAKRIYPCFVDFILHDOVHLLSECANLEILMIGLVWRSMEHPGK--LIFCSGLVLHRLQCAR-GFGWDID-SILAFSRSLHSLVDVPAFACLIALVLI
NDR_HUMAN  WAKALPGFRNHHVDQMAVIOYSWMGLMVFAMGWSFTNVNSRMLYFAPDLVFNEYRMHKSRYMSQCV-RMRHLSQEFGLWQITPQEFCLKALLLF
CR_HUMAN  WAKAIYGFNRHLLDDQMTLLQYSWMFLMAFALGWSYRQSSANLFCFAPDLIINEQRMTLPCMYPDQC-HMLYVSSELHRLQVSYEYELCKMTLLLL
DR_HUMAN  FAKMIPGFRNHTSEDDQVLLKSSAIEVIMLRNSNESFTMDMS-WTCGNQDYKYRVSDVTKAGHSELEIPLKFPQGLKKLNLHHEEHVLLMAICIV

LHVAP-LLAAAGLHASPMSADRVVAFMDHIRIFQEQVEKALKALHVDSAEYSCLKATVLF-----DACGLSDVAHVSELSQESQCALEEYVR-SQYPNQF--TRF
LHTAP-LLAAAGLHAAPMAAERAVAFMDQVRAFQEQVDKLGRLQVDSAEYGLKATALFTP-----DACGLSDPAHVSELSQEAQVAITEYVR-AQYPSQP--QRF
VRDG--ILLATGLHVHRNSAHSAGVGAIFDRVLTELVSKMRRMDKTELGLKRAILIFNP-----DAKGLSNPSEVEVLREKVYASUETYCK-QKYPEQQ--GRF
FKDV--LLGNDYIVPRHCPKELAEMSRVSRILDELVLFPQELQIDNNEYAYLKAIFFFDP-----DAKGLSDPGKIKRLRSQVQSLEDYINDRQYDSR--GRF
EQDT--MTFSDGLTLNRTQMHNAGFGPLTD-LVFAFAGQLLPLEMDDTETGLLSAICLICG-----DRMDLEEPKVDKLQEPLEALRLYAR-RRRPSQP--YMF
GEGK--LIFCSGLVLHRLQCAR-GFGWDID-SILAFSRSLHSLVDVPAFACLIALVLEIT-----DRHGLQEPKRVDELQNRILASCKEHVAAVAGEPQPA-SCL
HPGK--LLFAPNELLDNRQKQCVESGMEVIFDMLLATSSSRFRMNLQGEFFVCLKSLILNLSGVYTFSLSTLKSLEEKDHIHRLDKITDTLHLMKASGLTLQOQHRL
NVNSRMLYFAPDLVFNEYRMHKSRYMSQCV-RMRHLSQEFGLWQITPQEFCLKALLFSII-----PVDGLKNQKFFDELRMNYIKEDRIIACKRKNPTSCSRF
QSSANLFCFAPDLIINEQRMTLPCMYPDQC-HMLYVSSELHRLQVSYEYELCKMTLLLS-----VPGDLKSKQELFDEIMTYIKELKAIKVRREGNSQNWQRF
DMS-WTCGNQDYKYRVSDVTKAGHSELEIPLKFPQGLKKLNLHHEEHVLLMAICIVSP-----DRPQVQDAALIEAQDRLSNTQTYIRCRHPPPGS-HLLY

```

Figure 2.18 Portion of the multiple alignment of 10 sequences of nuclear receptors indicated in Figure 2.17. Positions at which at least 7 out of 10 sequences are identical are highlighted. Strict conservations are framed. The region represented corresponds to the cofactor liaison domain.

Construction of a phylogenetic tree

Phylogenetics consists in reconstructing filiations and ancestral links among various species in the context of the Darwinian Theory of Evolution. Such classifications were long based on sometimes rather arbitrary external characteristics. Since the introduction of DNA sequencing, the criteria of similarity among the sequences of certain genes have been used to directly deduce weak evolutionary ancestry. These criteria permit the construction of genealogical speciation trees known as *phylogenetic trees*.

Two families of algorithmic methods are used for reconstructing phylogenetic trees:

- **Progressive grouping methods**, which use a distance matrix $D_{i,j}$ between pairs of sequences. These methods utilize the metrics determined by these distances to construct a binary tree in an iterative manner. First they group the sequences into branches, then the branches among themselves, starting with the nearest ones and ending with those most distant.

There are several variants of progressive grouping methods. The simplest groups the two i and j vertices between which the distance is minimal at each iteration. They are then replaced by a new vertex k , and the distance between it and the other vertices is determined by calculating the arithmetic mean of the distances to the fused vertices: $D_{k,m} = \frac{1}{2}(D_{i,m} + D_{j,m})$. At each iteration the number of vertices is reduced by one. When only one remains, the tree has been constructed. This method, called UPGMA (Unweighted Pairgroup Method using the Arithmetic Mean), is very simple and rapid, but in unfavorable cases does not yield an optimal tree in terms

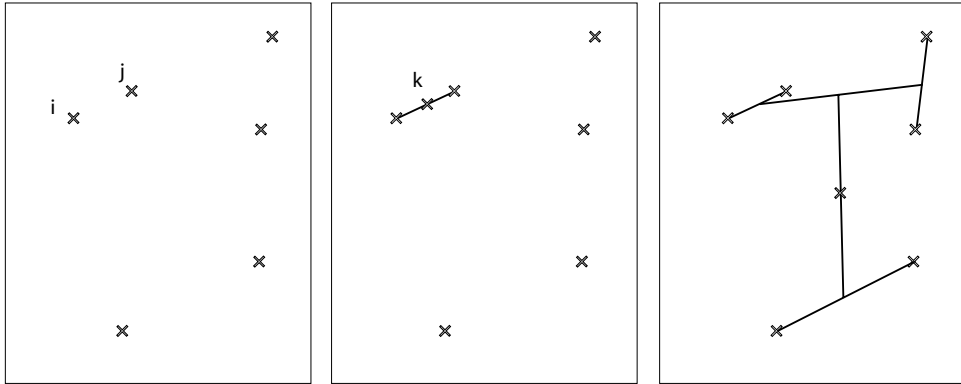


Figure 2.19 Schematic principle of the construction of a tree by the UPGMA method. The two closest vertices i and j are replaced by a new vertex corresponding to their arithmetic mean k . On the right: the finished tree. In the center: the last summit added; that is, the root of the tree. In this simplified example, the distances $D_{i,j}$ are represented by the Euclidian distances in the plane.

of total branch length. Nevertheless, with a few modifications, it is possible to overcome this shortcoming.

- **Methods based on the individual nature of each amino acid or nucleotide:** these take into account the number of mutations (changes in the sequence) necessary to go from one sequence to another. The sequence of the common ancestor of that branch is located at each vertex of the tree. The terminal extremities ('leaves') are the sequences of the current species. The lengths of the various branches that descend from a vertex correspond to the number of mutations separating the common ancestor from its descendants. The so-called *maximum parsimony method* is used to find an optimal tree the sum of whose branch lengths is minimal. The tree explains the biological diversity present starting from a minimum number of mutations. The search for this optimal tree is carried out using a systematic approach, exploring all possible configurations. In view of the explosive asymptotic character of the number of different N -leaved binary trees, parsimony methods resort to an elaborate branch-pruning strategy in their search, which consists in the computer science principle known as *branch and bound*. We will not give a detailed description of this algorithm here.

Beyond phylogenetic analysis, which is certainly interesting with respect to evolutionary theory, several groups of bioinformaticians have become interested in the systematic analysis of databases in their search for data concerning multiple alignments among conserved protein regions. They realized that Nature often reutilizes the same sequence element to perform analogous functions.

These conserved elements are generally structured into independent domains in the corresponding proteins and fulfill one of the ‘tasks’ of the molecule: DNA or RNA binding, ligand binding, enzymatic activity, association with other cell components, etc . . . Nature is thus playing with a kind of molecular *Lego*, assembling various domains in constituting a new protein. A great many of these domains have been inventoried, and today specialized databases exist that compile multiple sequence alignments for each of them. For example, the PRODOM database developed by INRA (the French *Institut National de la Recherche Agronomique*) in Toulouse, France (<http://protein.toulouse.inra.fr/prodom.html>) in May 2005 listed around 240,000 domains common to at least two proteins. PRODOM is part of the InterPro project (<http://www.ebi.ac.uk/interpro/>), which integrates the data of most protein domain and family databases. The mean length of these conserved domains is slightly more than one-hundred amino acids. For example, Figure 2.20 illustrates the breakdown of nuclear androsterone and glucocorticoid receptors into domains.

The systematic alignment of these domains provides the biologist with another very useful tool: the identification of characteristic motifs. When multiple alignments like those in Figures 2.4 and 2.19 are carried out, there are positions in conserved regions (highlighted in Figure 2.4 and framed in Figure 2.19) at which either homologous amino acids, like those in the first column of Figure 2.4, or only F or Y, two aromatic amino acids, are systematically found. A simple method for determining whether an unknown protein contains a conserved domain is to look not for a sequence alignment, but for the presence of a pattern of a few amino acids. The definition of such a pattern is of the following type:

‘F or Y, followed by any amino acid, followed by C, followed by any five amino acids, followed by C.’

This can be written in a more synthetic manner as follows:

$$[FY] \times C \times (6) \quad C$$

where the brackets represent the alternative, x indicates any amino acid, and the parentheses surround a numerical value that is the number of repetitions of

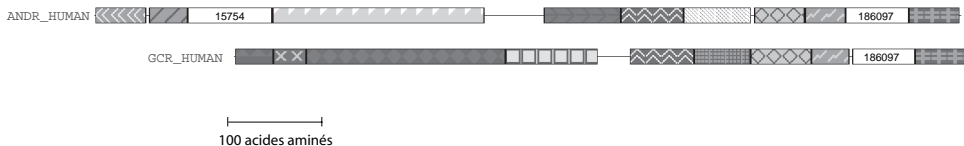


Figure 2.20 Breakdown of the nuclear receptors for androsterone and glucocorticoids by PRODOM. Each ‘box’ corresponds to a domain. The C-terminal third of the two receptors presents a certain number of common domains associated with DNA and ligand binding.

the preceding symbol. This type of pattern corresponds to what computer scientists call a ‘regular expression’ (cf, Chapter 4, *Genetic Information and Biological Sequences*). There are numerous programs to search for these, especially using UNIX (grep, lex). Several slightly different syntaxes exist for writing these regular expressions; the one presented above is different from the one provided in UNIX tools. It corresponds to one used by biologists, who have compiled a database of functional patterns and regular expressions that can be associated with them. Called PROSITE (<http://www.expasy.ch/prosite>), it is maintained by the Swiss Institute of Bioinformatics in Geneva, which also runs the SWISSPROT database (<http://www.expasy.ch/sprot>). It contains 1,400 documented patterns. The pattern that corresponds to nuclear receptors fits the following definition:

C-x(2)-C-x-[DE]-x(5)-[HN]-[FY]-x(4)-C-x(2)-F-F-x-R

In Figure 2.2, we note that the three receptor sequences conform well to the syntax of the PROSITE pattern, whereas the bacterial protein sequence whose homology is random does not. This pattern correctly ‘recognizes’ 229 nuclear receptors of the 233 identified to date, and no other proteins. We say that ‘the specificity is 100 percent’ (0 false positives out of 229) and that its sensitivity 98.3 percent (229 positives out of 233).

PQKTCLICGDEASGCHYGALTGSGCKVFFKRAAEGK Androsterone
 PQKICLICGDEASGCHYGVLTCGSGCKVFFKRAMEGQ Progesterone
 KDELCVVCGDKATGYHYRCITCEGCKGFFRRTIQKN Thyroid hormone

C	C	D	HF	C	C	FF	R	PROSITE pattern
		E	NY					

DQDKCIGCKTCVLACPYGTMEVVSRPVMRKLTLALNT Bacterial ferredoxin

Regular expressions are powerful tools, since algorithms exist that allow searching sequences in linear time ($O(n)$), using what are called *finite state automata* (cf. Chapter 4, *Genetic Information and Biological Sequences*). Owing to its great efficiency, the systematic search for known patterns is today a precious complement to research methods for similarity by alignment.

Bibliography

- Altschul S.F., *et al.* (1990). Basic local alignment search tool. *J Mol Biol* 215: 403–410.
 Altschul S.F., *et al.* (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25: 3389–3402.
 Bairoch A. (1991). PROSITE: a dictionary of sites and patterns in proteins. *Nucleic Acids Res* 19 Suppl: 2241–2245.

- Gotoh O. (1987). Pattern matching of biological sequences with limited storage. *Comput Appl Biosci* 3: 17–20.
- Henikoff S., Henikoff J.G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA* 89: 10915–10919.
- Henikoff S., Henikoff J.G. (1993). Performance evaluation of amino acid substitution matrices. *Proteins* 17: 49–61.
- Higgins D.G., Sharp P.M. (1988). CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene* 73: 237–244.
- Karlin S., Altschul S.F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc Natl Acad Sci USA* 87: 2264–2268.
- Needleman S.B., Wunsch C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 48: 443–453.
- Pearson W.R., Lipman D.J. (1988). Improved tools for biological sequence comparison. *Proc Natl Acad Sci USA* 85: 2444–2448.
- Smith T.F., *et al.* (1981). Comparative biosequence metrics. *J Mol Evol* 18: 38–46.

3

Comparative genomics

Formerly, biologists used isolated genes to study molecular evolution, but since it has become possible to completely determine genomic sequences, evolution is investigated at a higher level: the whole genome. How can genomic information be used to understand the evolution of genomes?

Genomes can be analyzed and compared according to various criteria, such as nucleotide and dinucleotide content, repetitions, coding-zone density, operons, gene family size distribution, etc. Genomes may also be studied on different *levels*, at the lowest of which they are regarded as a ‘sack of genes’, without considering interactions among their components. The next level takes these interactions into account, as well as cross-correlations among the various properties of genomes.

3.1 General properties of genomes

One can only be struck by the extreme diversity of the genomes that have already been studied, whether with respect to their general properties or to finer characteristics. The first general property of genomes is their size, which, as may be seen in Figure 3.1, ranges from 10^3 to 10^{11} basepairs if viruses are included, and from 10^5 to 10^{11} if only cellular life forms are considered. The sizes of bacterial genomes overlap those of both viruses and eukaryotes.

3.1.1 Size and structure of eukaryote genomes

Table 3.1 lists the sizes and numbers of some unicellular and multicellular eukaryote chromosomes. However, comparisons of closely related species reveal that the great variability observed in the number of chromosomes evidently does not obey a golden rule. The human genome consists of 46 chromosomes, whereas that of the chimpanzee has 48 (certain monkeys have 70), although divergence in DNA sequences between these two organisms does not exceed 1 percent. Within a single vegetal genus, millet, for example (Table 3.1, bottom),

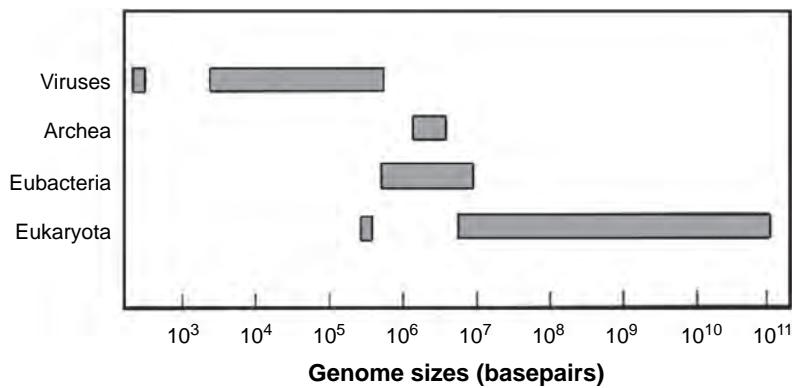


Figure 3.1 Order of magnitude of the genome sizes of viruses and of the three great realms of cellular life forms: archaea, eubacteria, and eukaryota. Deviant values correspond to viroids among the viruses and to endosymbiotic green algae among the eukaryotes.

Table 3.1 Physical analysis of eukaryote genomes of the major phylogenetic groups by size and number of chromosomes (for a haploid cell).

Common name	Size (Mbp)	Number of chromosomes
<i>Unicellular organisms</i>		
Baker's yeast	12	16
<i>Multicellular organisms</i>		
<i>Animals</i>		
Nematode	97	6
<i>Drosophila</i>	137	5
Mouse	3,000	20
Human	3,100	23
<i>Plants</i>		
<i>Arabidopsis</i>	120	5
Rice	450	12
Wheat	15,000	
Mistletoe	75,000	
Millet	370–1,200	5–9

various species have haploid genomes whose sizes vary by a factor greater than three. The number of chromosomes ranges between 5 and 9 in the haploid millet genome, and since millet ploidy can vary between two and six according to species, the number of chromosomes per cell in fact ranges between ten (2 × 5) and fifty-four (6 × 9).

The considerable diversity in genome size observed among organisms does not reflect differences in their complexity or capacity. The mistletoe genome (75,000 Mbp) is around 25 times larger than that of the human (3,100 Mbp). However, if we consider the basic criterion of the number of protein-coding genes contained in a genome, it is probably true that the genomes of unicellular organisms have fewer genes than those of multicellular organisms. Nevertheless, the spread in the number of genes between unicellular and multicellular organisms is low. Between the fly and the human, via *Arabidopsis* and nematodes, the number of genes probably varies by no more than a factor of two, although this point remains controversial, because of the difficulty in locating genes in mammals and plants.

Comparative genomics becomes particularly efficient when it is based on the study of a large number of genomes belonging to related organisms. Indeed, knowledge of such cases facilitates disentanglement of the numerous and superimposed evolutionary events that have resulted in genomes as we know them today. So far, the best case study in eukaryotes relates to yeasts. The genomes of over a dozen species of yeasts and those of a few filamentous fungi have recently been analyzed by complete or partial sequencing. These comparative genomic analyses, which have no equivalent in other phyla, reveal that yeasts evolved through interplay among gene duplication, formation of novel genes, and loss of others. The picture that emerges from these studies is that of a highly dynamic evolutionary process involving diverse mechanisms operating simultaneously. Longer-range duplications have also been demonstrated in yeasts, plants, and vertebrates. They consist of whole genome duplications, segmental duplications, and tandem gene array formation. Whole genome duplications are rare in eukaryotes, with the possible exception of polyploid plants, and are rapidly followed by extensive gene loss that leaves little trace, except for very recent events.

3.1.2 Diversity and plasticity of bacterial genome structure

Although bacteria display very limited variation in their dimensions and morphology, we should bear in mind that branches in their phylogenetic trees are separated by immense temporal distances, at least as large as those that separate the longest branches of the eukaryote tree; for example, between fungi and vertebrates.

The following section explores the structural diversity and fluidity of the bacterial genome, showing how individual bacteria within the same species differ, as well as how closely related species differ from each other. Eubacteria will be emphasized, since they are the most studied phylum in this respect. Large DNA replicons are called *chromosomes* and small ones *plasmids*, although the definitions of these terms have become blurred, since intermediate cases have been recognized.

Replicon size

Bacterial genome sizes can vary by a factor greater than ten. The smallest known bacterial genome is 0.58 Mbp (*Mycoplasma genitalium*) and the largest is 9.2 Mbp (*Myxococcus xanthus*). These values may be compared with the sizes of the largest viral genome, 0.67 Mbp (bacteriophage G), and of the smallest known eukaryote genome, 6.2 Mbp (the microsporidium *Spraguea lophii*; Figure 3.1). The average gene size in bacterial genomes known today is rather uniform, around 1 kbp. Bacterial genes all seem to be compacted in a similar manner, with around 90 percent of the DNA coding for macromolecules, proteins, and stable RNA. Large bacterial genomes therefore contain more genes than small ones. The number of genes in a bacterial genome seems to reflect the lifestyle of the organism. Small-genome bacteria (~500 genes) are *specialists*; for example, parasites that only thrive within living hosts or under other very special conditions, whereas bacteria with large genomes (~10,000 genes) are *generalists* from the metabolic point of view, or undergo certain forms of development, such as sporulation or mycelium formation.

As may be seen in Figure 3.1 and in Table 3.2, bacterial genome sizes vary widely, even within the same species. This variation can be explained by the rapid loss of genes occurring after a species has invaded a highly specific niche, which would tend to reduce genome size. However, the discovery of high rates of horizontal gene transfer among bacterial phyla (see details below) also makes increases in genome size plausible.

Table 3.2 Physical analysis of bacterial chromosomes among the major bacterial divisions with respect to size, shape (Circular or Linear), and number of chromosomes.

Bacterial division	Genome size (Mbp)	Genome shape	Number of chromosomes
Aquificales	1.1	C	1
Chlamydiae	1.0	C	1
Cyanobacteria	2.7–6.4	C	1
Fibrobacteriae	3.6	C	1
Firmicutes	0.58–0.67	C	1
Actinomycetes	1.6–8.2	C or L	1
Fusobacteria	2.4	–	1
Cytophagales	2.1–5.3	C	1
Planctomycetes	5.2	C	1
Proteobacteria	0.9–9.2	C, C and L	1–3
Micrococcus	1.7–3.6	C	1
Spirochetes	0.9–4.5	L or C	1–2

Replicon geometry

In most bacteria, the genome is circular; however there are exceptions (Table 3.2). Bacteria belonging to the genera *Borrelia* and *Streptomyces* have linear chromosomes, and often also have linear plasmids. The ends of *Borrelia* DNA are hairpin-shaped; that is, a *Borrelia* DNA strand forms a loop that becomes the second DNA strand. In one species of *Borrelia*, a plasmid that is ordinarily linear has been found to be circular, indicating that linearity is not necessary for replication. In contrast, *Streptomyces* DNA extremities are open, and specific proteins are attached to the free 5'-ends. If such a chromosome is artificially circularized, we again see that linearity is not a prerequisite for replication.

Number of replicons

Bacteria generally have a single large chromosome (Figure 3.1 and Table 3.2), and extra-chromosomal elements (plasmids) are found in many – if not all – species. However, members of several bacterial genera contain two or three large replicons (>100 kbp). The presence of multiple chromosomes is a stable characteristic of the *Brucella* and *Burkholderia* genera (Table 3.2), and ‘housekeeping’ genes are distributed among these multiple chromosomes. The possible adaptive advantage conferred by chromosome multiplicity remains a mystery. If the circular chromosome of *Bacillus subtilis* is divided into two circular parts, the bacterium displays no particular phenotype.

Number of chromosome copies

The number of copies of plasmids found in natural bacterial isolates is nearly the same as the number of principal chromosomes. It is an oversimplification to consider bacteria to be *haploid* (one copy of each chromosome per cell), since a few bacterial species have more than one chromosome in each cell. This is the case for *Deinococcus radiodurans*, a bacterium that is highly resistant to radiation. It contains four copies of its chromosome, which, after severe radiation damage, can recombine by homology to regenerate an intact chromosome.

Also, even a haploid bacterium may contain more copies of certain genes than others. During rapid proliferation, bacteria have an average of four times more copies of sequences that are close to the replication origin than of those close to the replication terminus. This is because the time between two cell divisions during rapid proliferation is shorter than the total time required for complete replication (Figure 3.2), resulting in the coexistence of two or three levels of

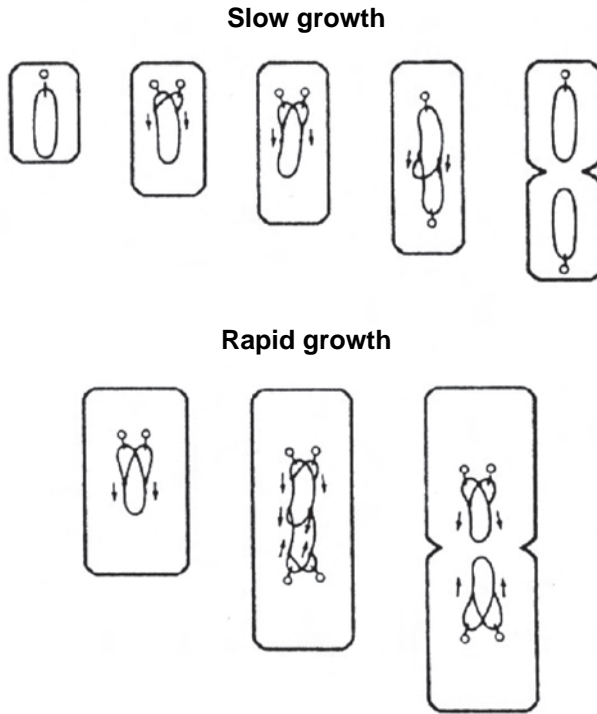


Figure 3.2 Replication of circular bacterial DNA. The circular chromosome of *Escherichia coli* replicates bi-directionally from the origin **O** toward a terminus situated opposite the origin. The period of chromosome replication is relatively independent of growth conditions and of the time between successive cell divisions (generation time). Replication takes about 40 minutes for *E. coli*. Under conditions of slow growth (above), replication time is shorter than generation time. Re-initiation of replication does not occur until the preceding replication is complete. Replication is followed by a period in which there is no DNA synthesis. Under conditions of rapid growth (below), replication time is longer than generation time. Re-initiation of replication nevertheless occurs at the same rhythm as cell division. Thus, two or three levels of replication forks coexist following successive fork departures from the origin.

replication forks in the bacterium at a given time. Thus, double, and sometimes quadruple, copies of genes close to the origin are present in bacteria during rapid growth. In the absence of downstream regulation, this leads to two-to-four times greater expression than with a single copy. It is not absurd to imagine that over the course of evolution, chromosome rearrangements have optimized the selection of these genes as a function of their particularly intensive expression during rapid growth.

3.1.3 Bias, isochores, and CpG islands

Bacterial genomes display characteristic codon usage (some codons are preferred to others that are synonymous with them). These genomes have a characteristic G+C content (the molar proportions of guanine and cytosine in DNA), and a G and C bias for one DNA strand compared with the other, as well as particular oligonucleotide frequencies (mono-nucleotides, di-nucleotides, etc). Although the mechanistic origin of these properties is not entirely clear, they evolve slowly enough to be of use in inferring the evolutionary history of horizontal transfers.

The same biases, as well as other characteristics, are observed in eukaryote genomes. *Isochores* are long DNA segments (>300 kbp) of homogeneous base composition in mammals. They may be divided into a small number of families covering a wide range of G+C content. Isochores that are poor in G+C tend to correlate with dark G bands (after staining with certain reagents, such as Giemsa stain, all human chromosomes display dark and light bands known as G bands). Isochores with a high G+C level have been observed to be richest in genes.

CpG islands (the ‘p’ is for phosphodiester bond), as their name indicates, contain numerous repetitions of the CG dinucleotide. This dinucleotide is generally underrepresented in the human genome, but found in some islands of a few hundred basepairs, usually associated with the 5′-ends of genes.

3.2 Genome comparisons

3.2.1 Orthologous and paralogous genes

To compare genomes A and B in a meaningful manner, it is useful to determine which gene *b* in genome B corresponds to gene *a* in genome A. This correspondence relationship is primarily based on homology. Functional homology without sequence resemblance indicates evolutionary *convergence* (genes with no ancestral link but identical function.) Sequence homology leads to a diagnosis of evolutionary *divergence*, starting from a common ancestral sequence.

In the frequent case of evolutionary divergence, the genes evolved from a common ancestor, but diverged after speciation or following a duplication event (Figure 3.3). When the homology is the result of *speciation*, such as when the history of the gene reflects that of the species (for example, human and mouse *alpha*-hemoglobins), the genes are called *orthologs* (ortho = ‘correct’). When the homology is the result of gene *duplication*, in which the two copies are transmitted side-by-side over the history of a species (for example, mouse *alpha*- and *beta*-hemoglobin), the genes are known as *paralogs*, (para = ‘in parallel’). Thus,

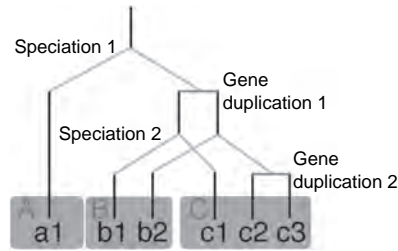


Figure 3.3 Diagram of divergent evolution illustrating orthology and paralogy. Speciation events yield species A, B, and C. Genes *a1*, *b1*, *b2*, *c1*, *c2*, and *c3* are descended from the same ancestral gene (above) via evolutionary speciation events and genetic duplication. Speciation 2, which starts from the same ancestral gene as *b1* – *c1*, gives rise to two orthologous genes, *b1* and *c1*, in species B and C, respectively. Gene duplication 2 in species C gives rise to two paralogous genes, *c2* and *c3*.

orthology and paralogy are defined only with respect to the *phylogeny* of the genes; not to their functions.

Identification of orthology using relative levels of sequence identity

Ideally, one would expect orthologous genes in the genomes of two species to have the highest similarity, considering their relatively recent divergence. The most direct approach to identifying orthologous genes therefore consists in comparing all the genes in the two genomes with each other. The pairs of genes (*a*, *b*) that have the most similarity are selected and considered orthologs. For example, *b* in B is the gene most homologous to *a*, which in A is the gene most homologous to *b*. This approach can be supported by auxiliary information, but may also encounter difficulties, described below.

Auxiliary information used for orthology detection

The first piece of auxiliary information useful in measuring orthology is *synteny* (see details below.) An example of synteny is the presence in genome B of contiguous genes *b1* and *b2*, which are the orthologs of *a1* and *a2*, themselves contiguous in genome A. When the evolutionary distance between two genomes is such that the divergence between their orthologous genes is on average greater than 50 percent with respect to amino acids, gene order is no longer conserved between the genomes. Therefore, the use of synteny to identify orthologs is mainly limited to genomes in which divergence is relatively recent.

A second kind of auxiliary information may be obtained by comparing a pair of presumably orthologous genes with a homologous gene in a third genome. If two genes in different genomes have the highest level of identity both with each other and with respect to the gene in the third genome, there is strong presumption that the two genes are in fact orthologs.

Sequence divergence

At great evolutionary distances, for example, between eubacteria and archaeobacteria, sequence similarities can be so eroded that the distances between orthologous genes are comparable to the distances between sequences from same gene family. The similarity may even become so low as to be undetectable.

Non-orthologous gene displacement

A second problem encountered in identifying orthologs is the displacement of a non-orthologous gene. This occurs rather frequently when two non-orthologous genes with no phylogenetic link carry the same function in different organisms. This is therefore a case of *evolutionary convergence*, which can be confusing.

Gene duplication and loss

A third process that limits the identification of orthologous genes is gene loss in combination with gene duplication. If genomes A and B lose two paralogs *a1* and *b1* of an ancestral gene that was duplicated prior to speciation events, the remaining genes *a2* and *b2* in genomes A and B display the highest sequence identity. According to the above definition, they are considered orthologs, whereas in fact they are not. Such cases may be detected by verifying whether the percentage of similarity between their protein products lies within a certain range of values.

Orthology of proteins consisting of several domains

Two levels of orthology can be identified in proteins that consist of several domains, one for each domain considered individually and one for the complete protein. This can lead to situations in which non-orthologous genes code for orthologous protein domains.

The concept of orthology is therefore an important refinement of the notion of homology for use in describing phylogenetic relationships among genes, as long as the problems outlined above are borne in mind and the methods used to determine orthology are well-specified.

3.2.2 Synteny

As mentioned briefly above in the discussion of orthology, synteny concerns the conservation of the respective order of genes among genomes.

When species that are not closely related are considered, extensive rearrangement of gene order is usually observed. In contrast, any conservation of relative gene order is remarkable. However, in a few cases, relative gene order is conserved over considerable evolutionary periods. In general, genes that ‘work together stay together.’ This is often the case for operons, suggesting that the advantageous co-regulation of genes in an operon could justify synteny. However, this justification may be insufficient, since, for example, the tryptophan operons of *E. coli* and *B. subtilis* genes display total gene order conservation despite utilizing different regulatory mechanisms. In addition, genes that are co-regulated within the same operon in a given species are sometimes dispersed in another, even though they are correctly co-regulated there.

Synteny within the same genus

It is possible to compare the genomes of two species of a given genus (Figure 3.4(A)). The chromosome sizes of *Mycoplasma pneumoniae* and *Mycoplasma genitalium* are, respectively, 816 and 580 kbp. An ortholog of every *M. genitalium* gene exists in *M. pneumoniae*. However, the structures of the two genomes have greatly diverged, particularly as a result of deletions and insertions. The *Mycoplasma* chromosome may be considered to consist of six segments whose order has been shuffled without affecting their orientations. It is interesting to note that these six segments are flanked by repetitive sequences with the same orientation, known as ‘MgPa,’ which could permit such rearrangements by homologous recombination.

Synteny within the same species

More surprising than the differences found between closely related species are the substantial ones observed among the genomes of natural isolates of the same species (Figure 3.4(B)). Among the 127 isolates of *Salmonella typhi* there are 17 different arrangements of the seven chromosome segments, flanked by seven

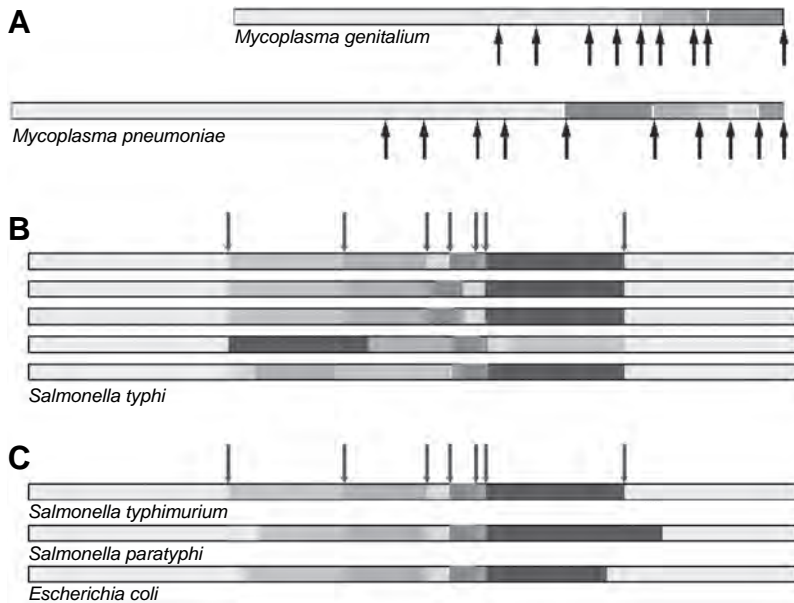


Figure 3.4 Major differences in genomic structure among species within the same genus and among strains of the same species. All the circular chromosomes in this representation have been linearized near the replication terminus. (A) Relations between *Mycoplasma pneumoniae* and *Mycoplasma genitalium*. Different shades of gray correspond to different genome segments in each strain. The same shade of gray indicates a segment that is homologous in the two strains, even though each *M. genitalium* segment includes deletions with respect to its *M. pneumoniae* homolog. Arrows indicate the positions of repetitive sequences known as 'MgPa'. (B) Rearrangements among *Salmonella typhi* strains (127 natural isolates were analyzed). Arrows indicate the positions of the seven tRNA operons. (C) Rearrangements among three enterobacteria genera. Arrows indicate the positions of the seven tRNA operons.

tRNA operons. Those most frequently found are presented in Figure 3.4(B). Many non-counter-selected rearrangements therefore seem to be produced by homologous recombination among these tRNA operons. Consider enterobacteria of closely related genomic content, but from different genera (*Escherichia* and *Salmonella*) or species (*paratyphi* and *typhimurium*) (Figure 3.4(C)). Curiously, they display less difference in this particular case (Figure 3.4(C)) than do strains of the same genus and species.

Synteny within the same strain

Even within the same strain, minor genomic differences can appear among individuals, either as random events that accumulated over time, or in response to

a particular situation. In the first case, semi-stable reversible phenotypic variations occur in which members of the population spontaneously change properties (gene expression states), with low probability at each generation. These properties may later revert, with the same low probability. Such variations are called ‘epigenetic variations’ or ‘epimutations,’ and do not affect genome sequence itself, but rather gene expression status.

In the second case, programmed changes in the genome arise irreversibly in particular cells not destined to produce descendants. Two such cases are known: the *B. subtilis* chromosome during spore formation and the heterocyst chromosome in certain cyanobacteria (heterocysts are described in Figure 8.4). In both these cases, the rearranged chromosome is no longer replicated and the rearrangement causes major changes in gene structure and expression.

‘Cassette mechanisms’ also exist, in which a DNA sequence contained in an unexpressed pseudogene is physically placed in an expression site. These mechanisms can be either *distributive* (pseudo-genes not grouped at a single site) or *organized* (pseudo-genes in single-file at one or more privileged sites.) Some members of the genus *Borrelia* have a mechanism by which cassettes code external membrane proteins, allowing the bacterium to change its surface antigen determinants, thereby escaping the response of the host immune system. In eukaryotes, a cassette mechanism allows yeasts to switch mating type. The vertebrate immune system itself utilizes a basically similar strategy to create immunoglobulin diversity.

3.2.3 Minimum gene set

Once complete genome sequences are available, it is possible to investigate whether there exists a minimum set of genes necessary to maintain life. The *M. genitalium* bacterium includes only 468 genes that have been identified as coding for proteins, which some biologists refer to as the minimum number necessary to maintain life. While the *M. genitalium* genome is the smallest one known among cellular life forms, there is not the slightest experimental evidence to indicate that it is the smallest *possible* genome. It is also interesting to compare this set of 468 genes with other small-genome bacteria, such as that of *Haemophilus influenzae*, which contains 1,703 protein-coding genes. Of the *M. genitalium* genes, 240 have orthologs in the *H. influenzae* genome. To these may be added 22 genes whose functions are not assumed by any of the 240 orthologs due to displacement of non-orthologous genes. Finally, six functionally redundant genes may be eliminated from the count. Therefore 256 remain, and have been postulated to constitute the minimum set of genes necessary for cellular life (Figure 5.5).

This approach is obviously subject to criticism, since these small-genome bacteria are parasites that have lost numerous genes necessary for autonomous life.

Nevertheless, these 256 genes do provide basic life functions and a degree of metabolic autonomy. They have inspired the choice of the 127 basic genes that have been modeled in the ‘Electronic-Cell (‘E-Cell’; see chapter 8) to simulate the functioning of a minimal cell, including notions of energy cost, such as for the synthesis of macromolecules.

The circles, drawn approximately to scale, represent the gene sets of *H. influenzae*, *M. genitalium*, and of the yeast *Saccharomyces cerevisiae*. The rectangles show the two successive stages in the construction of the theoretical minimum set of 256 genes, in comparison with the two prokaryotes, and a putative set of 96 ancestral genes derived by comparing this minimal set of genes with the genes of the yeast, a eukaryote.

If this study is extended to a eukaryote such as the yeast, once the genes that code for mitochondrial proteins have been eliminated, 96 of these 256 genes are found to have an ortholog in the eukaryote (Figure 3.5). It is amusing

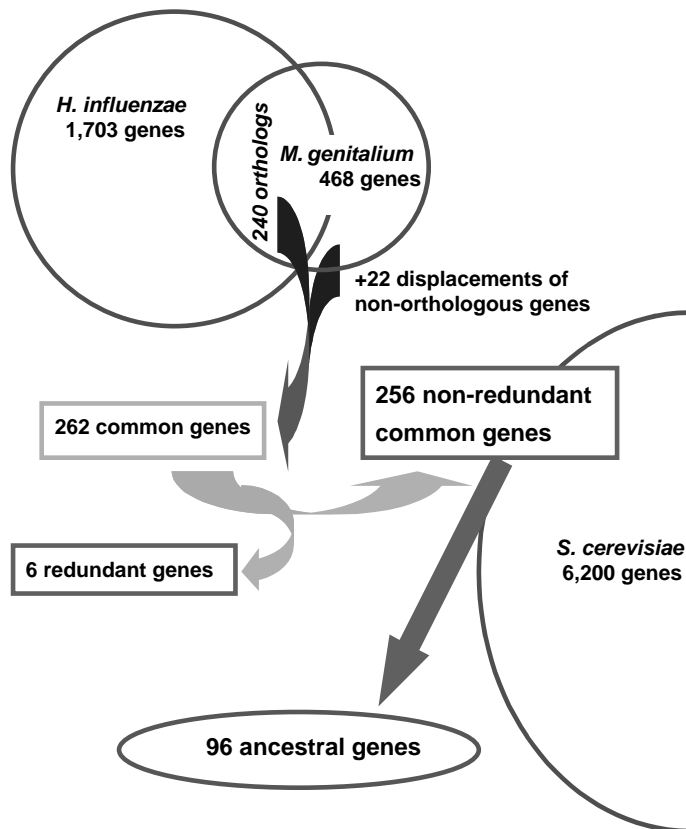


Figure 3.5 Construction of a minimum set of genes from two bacterial genomes and comparison with yeast genes.

to note that certain functional categories are totally absent from this reduced set of 96 genes, in particular, genes involved in DNA replication. This suggests that the last common ancestor of these organisms did not use DNA; perhaps it used another nucleic acid to replicate. While it is too early to tell, this study is interesting in that it is the first to prefigure a theory of comparative genomics.

3.2.4 Pathogenicity islands

Pathogenicity islands are chromosome or plasmid regions in pathogenic bacteria that assemble genes that code for virulence factors, such as toxins, pili, and other host adsorption and invasion factors. These islands also have at least one of the following characteristics: association with DNA mobility agents (integrase genes and insertion/deletion sites), non-universal distribution in natural isolates, unusual codon use, and an abnormal percentage of G+C in comparison with the other genes of the bacterium. Their size is highly variable; the largest one known is 190 kbp. By their sporadic presence and mobility, they clearly contribute to genomic content variability in numerous pathogens. Thus, four different pathogenicity islands are present, alone or in pairs, in 10, 11, 28, and 28 of the 72 pathogenic isolates of *E. coli* that have been studied.

If the genomes of two closely related strains, one pathogenic and the other benign, are compared, it is often possible by studying the difference between them to identify the genetic basis of the pathogenicity. Thus, 60 percent of the genes of the pathogen *Haemophilus influenzae* that do not have homologs in a benign strain of *E. coli* appear to be implicated in pathogenesis.

It seems likely that pathogenicity islands are just one manifestation of a more universal phenomenon that could be called 'specialization islands,' which would confer various specific capacities – metabolic, aggressive, or defensive – permitting a fraction of the members of a bacterial species to occupy very specialized niches.

3.2.5 Therapeutic targets

Comparing the genetic contents of the genomes of pathogens (bacteria and yeasts) and hosts (mainly human, but also animals and plants) can be extremely revealing. Indeed, any gene of a pathogen that has no equivalent in its host is a potential therapeutic target. The list of differences therefore amounts to a preliminary list of antibacterial or antifungal targets. The targets retained are those that are indispensable either for the survival of the pathogen, or for its invasive or pathogenic capacity (for example, the gene for a pathogenicity island.) Of course, before a drug can be considered a candidate therapeutic agent it is

necessary to ascertain that it does not have an unexpected secondary effect on a host function.

Finally, the genome of each bacterial species may be considered to consist of a universally present hard kernel of genes (the ‘endogenome’), plus a battery of accessory elements, which may be located on free replicons or well-integrated at various chromosome sites (the ‘exogenome’). The selective advantage provided by the fact that certain genes are accessory elements could be related to their genetic mobility. An accessory element may or may not be functional (for example, it may have lost its function following evolutionary degeneration). It may appear to be selfish (a pro-virus integrated into the bacterial host’s chromosome), or of great adaptive value under certain circumstances (a pathogenicity island or integrative element that confers antibiotic resistance), or both at the same time (a provirus bearing a bacterial virulence gene). It is difficult to draw similar conclusions regarding eukaryotes, since few complete genomes have so far been sequenced. However, it appears that eukaryote genomes harbor some ‘orphan’ genes with no known homolog in other genomes.

3.3 Gene evolution and phylogeny: applications to annotation

3.3.1 Gene evolution

Various criteria may be applied in evaluating the rate of genome evolution. In an initial approach, we will consider the genome as a ‘sack of genes’ and count how many orthologous genes the compared genomes have in common. Correlations between genes in these genomes will then be taken into account, and gene synteny, regulatory modifications, and co-occurrence will be evaluated.

Sharing of orthologous genes

In order to evaluate the rate of genome evolution, we can examine how the number of orthologs shared by two genomes decreases over the time since their divergence. The number of amino acid substitutions per protein per position will be used to estimate divergence time. Figure 3.6 shows that the fraction of orthologous sequences shared by two sequences rapidly decreases over evolution, faster than the decrease in the level of sequence identity between these same sequences. The most rapid decrease occurs over short time scales, when protein identity between a given pair of genomes is still greater than 50 percent.

Time since divergence is measured by the number of amino acid substitutions per protein per position in a set of 34 orthologs (abscissa). If eubacteria and archaeobacteria diverged 3.5 billion years ago, each unit of the abscissa

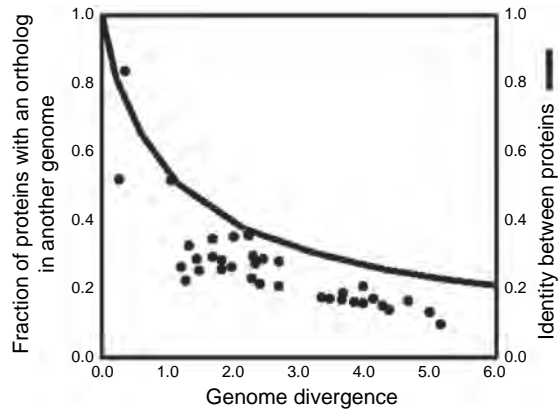


Figure 3.6 Relationship between genome similarity and evolutionary time.

represents 875 million years. Each point on the graph indicates the fraction of sequences in genome A that have an ortholog in genome B (left ordinate). The curve indicates the percentage of identity of the protein sequences (right ordinate) calculated as a function of the abscissa, according to the equation of N.V. Grishin [*J Mol Evol* (1995), 41:675–679].

Horizontal transfer

Sometimes a species shares a few orthologous genes with a phylogenetically distant one (for example, eubacteria and archaeobacteria) that it does not share with closely related species. This reveals an aspect of the evolution of the genetic content of genomes that does not derive from the ‘vertical’ notion of the ‘phylogenetic tree’: ‘horizontal’ transfer. Based on nucleotide and dinucleotide frequencies, it appears that at least 10 to 15 percent of the genome of a bacterium such as *E. coli* consists of sequences that have been transferred horizontally; that is, bacterium-to-bacterium. This phenomenon is rarer among archaeobacteria, although some particular mechanisms have recently been discovered by which they integrate mobile genetic elements. It is also believed that in the distant past, metabolic genes were horizontally transferred from eubacteria to archaeobacteria, whose genomes would therefore be chimeric.

Synteny

Gene order conservation may be analyzed in terms of genome divergence time, as above (Figure 3.7). A drastic rearrangement of genomes is observed over short

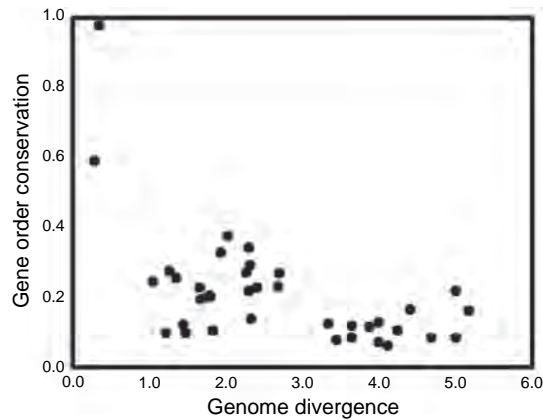


Figure 3.7 Relationship between the conservation of gene order in a genome and evolutionary time.

periods, while protein pair identity between the genomes is still over 50 percent, below which saturation is reached. Compared with the preceding figure, the *order* of orthologous genes is seen to be somewhat less well-preserved than their actual presence. Genes that are still paired by the time the saturation level is attained are generally those that code for proteins that physically interact. Gene order conserved in the absence of physical interaction indicates recent horizontal transfer.

Horizontal transfer

The above argument may be exploited to detect horizontal gene transfer by synteny. Since gene order is poorly conserved over evolution, the ordered presence of a few genes in two evolutionarily distant branches raises the suspicion that they have undergone horizontal transfer. This suspicion is reinforced if the gene order is not conserved in closer evolutionary branches.

Selfish operons and Fisher's hypothesis

Bacterial operons are DNA segments in which genes coding for several proteins are contiguous and co-regulated. Considering the universality of operons and their structuring effects, the weak conservation of bacterial gene order is surprising (Figure 3.7). Except for cases of horizontal transfer, conservation of bacterial gene order appears to be limited to genes whose products interact.

According to Fisher, this is justified because the physical proximity – genetic linkage – between genes whose products function well together tends to increase in order to prevent the separation of co-adapted allele pairs by recombination events.

The time since divergence is measured by the number of amino acid substitutions per protein per position in a set of 34 orthologs (abscissa). If eubacteria and archaebacteria diverged 3.5 billion years ago, each unit of the abscissa corresponds to 875 million years. The ordinate displays the number of genes with orthologs in the two genomes that also have at least one neighboring gene that is orthologous in the two genomes. This number is compared with the total number of orthologs shared by the pair of genomes. The rapid differentiation in gene order during evolution may then be readily observed.

These interactions should impose selective constraints, thereby slowing the evolution of genes whose products are co-adapted. As a result, sequence conservation in genes that are conserved in pairs would, on average, be greater than in other genes. In bacteria, the co-folding of two proteins (destined to belong to the same complex) during co-translation has been proposed as a possible explanation of why their genes are maintained side-by-side, producing their messenger RNAs at the same physical site.

The ‘selfish operon’ theory proposes that operons simply increase the probability that genes that function in concert stay together during horizontal transfer. Of course, the transferred DNA is more likely to be conserved over evolution if it contains genes necessary for a biochemically advantageous function. This model applies only to non-essential genes; *i.e.*, those that can be lost and then reintroduced by horizontal operon transfer (pathogenicity islands, etc). For example, it does not apply to ribosome genes, which are essential and closely grouped, and for which there is no proof of horizontal transfer.

While many operons include genes whose products do not physically interact, those operons that have been conserved over long evolutionary periods do seem to conform to Fisher’s hypothesis; *i.e.*, they contain genes whose products interact.

Regulatory modifications

Once orthologous genes and synteny have been determined, it is possible to examine the conservation of the regulatory sequences in which the links that control gene expression are located may be considered. This conservation has been found to be remarkably low and to diminish much more rapidly than the conservation of gene order. However, there are exceptions to this rule. A secondary RNA structure at the 5′-end of *E. coli* ribosome genes *rpl1* and *rpl11* has been found to be involved in the regulation of operon expression. The same secondary RNA structure has been identified in all bacterial genomes studied to date.

Gene co-occurrence

Some genomes are more organized than others

If neighboring genes in a genome express products that cooperate in a given function, as is the case for operons, these same genes should also coexist in other genomes, even in those in which they are not neighbors or located in the same operon. The absence of even one such cooperative gene should render the entire set non-functional. This phenomenon has indeed been observed. If gene *a1* in genome A has gene *a2* as a neighbor, and if ortholog *b2* of *a2* is found in genome B, the probability that ortholog *b1* of *a1* is also present in B is increased. Reversing this proposition, orthologs shared by genomes tend to be physically close in some of these genomes. Of course, *b1* and *b2* are often also neighbors; except for such cases, this tendency remains. Genomes therefore are often organized, some more so than others, in the sense that grouping is more frequent in them (Figure 3.8).

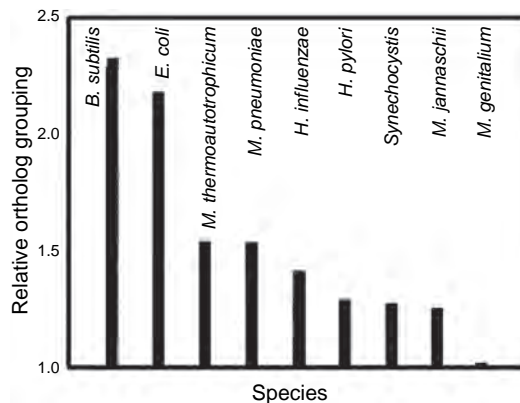


Figure 3.8 *Bacterial genomes are more or less well organized*

The ordinate gives the ratio of the number of genes in genome A that have an ortholog and at least one neighboring gene that also has an ortholog in genome B to the number obtained after randomizing gene order. This analysis includes only genes that are neighbors in A but not in B (or the inverse), so that the results will not be influenced by the phenomenon of synteny, discussed previously. The relative grouping of orthologs, given on the ordinate, is the result of the mean comparison of one genome with the other eight. The species studied are given on the abscissa. The *M. genitalium* genome (right) is the only species in the study in which there is not more gene grouping than predicted by a random model. The explanation for this could be that the very small *M. genitalium* genome makes neighbors of all of its genes that display orthology with the genes of another species. Genome size therefore does not systematically justify relative grouping, since, for example, *Synechocystis* has a large genome but displays little organization. *E. coli* and *B. subtilis* have the most organized genomes.

Gene co-occurrence and metabolic pathway conservation

Besides the spatial association of orthologs, it is also possible to determine whether orthologs reveal ‘genomic association’. Are both orthologs either coexistent or absent in a genome; *i.e.*, is one not found without the other? Such analysis makes it possible, in principle, to infer which genes are functionally linked. As we have just seen, orthologs found in two genomes have an increased probability of being grouped together (and possibly of being functionally linked) in one of the genomes, even if not in the other. This information may be used to help reconstruct metabolic and signaling pathways. This works well only if the structure of the pathway concerned is well conserved over evolution. The displacement of a non-orthologous gene, by which one gene takes over the role of another in a given pathway, suggests that orthologous gene pathways are better conserved than their presence.

These observations have consequences in selecting strategies for genomic analysis. In particular, they indicate that, according to the phenomenon being studied, the evolutionary distance between organisms included in a study must be carefully chosen. Therefore, when studying the evolution of gene regulation, it is better to compare species that are close to each other than when studying the evolution of gene order. When studying the evolution of coding sequences, it is better to compare more distant species. Finally, the most distant species should be used when studying the evolution of metabolism.

3.3.2 Using genomic context to predict functions

This section appears here because it makes use of some of the concepts that have just been discussed. However, the perspective is different now, involving the annotation of unknown genes that derive from systematic sequencing. More specifically, the aim here is to predict functional interactions among proteins, based on their gene context. In bacteria, this covers three different possibilities:

- Type 1: Gene fusion, forming a single hybrid gene that codes for a single protein in another genome;
- Type 2: Gene order conservation or gene co-occurrence in putative operons;
- Type 3: Co-occurrence of genes in genomes (phylogenetic profile).

In the first type, interaction between the two proteins of an organism is demonstrated by their covalent bonding in another species. In the second type, it has been established experimentally that proteins coded by 75 percent of conserved gene pairs interact physically, as seen above. To these 75 percent may be

added 20 percent of conserved gene pairs for which gene product interaction has been predicted but not proven. For the third type, as discussed above in ‘Gene co-occurrence and metabolic pathway conservation’, a positive result suggests that the genes concerned contribute to the same major function or pathway.

Conservation of gene order (Type 2) is the most useful of these three concepts, since it provides contextual information for 37 percent of the genes studied in the *M. genitalium* standard case. If there is gene co-occurrence in operons without order conservation, Type 2 provides information in 8 percent of the cases. Type 1 provides contextual information in 6 percent of the cases, and Type 3 in 11 percent. Combining these criteria, significant information concerning the genomic context may be obtained for 50 percent of genes (the categories overlap).

This approach based on genomic context therefore holds promise and is complementary to the more classical approach based on homology search. In principle, homology search allows prediction of a molecular function, while the genomic context approach addresses a higher functional level by predicting to which process or pathway a given protein belongs, or with which other protein it interacts.

3.3.3 The genomic tree of life

An important consequence of the availability of numerous complete genomic sequences is that it becomes possible in principle to construct a universal tree of life based on *genomic* rather than *genetic* phylogeny. The existence of the universal tree, which until recently was based on 16S rRNA genes, led during 1987–90 to a proposal of the existence of three major biological reigns: eukaryotes (Eukarya), eubacteria (Bacteria), and archebacteria (Archaea). For various reasons, this classification was criticized over the following ten years. Certain other genetic phylogenies have yielded the same result, whereas still others have proposed different topologies, and others ended up with only two realms. These difficulties in part derive from the fact that archebacteria are close to eukaryotes with respect to transcription/translation machinery and close to bacteria with respect to metabolism. These problems, as briefly mentioned above, are due to horizontal transfer, which may have been particularly intense during early evolution, and from unequal nucleotide substitution rates, according to the lineage. More generally, these problems reflect the fact that the trees represent the evolutionary distances between genes, rather than whole organisms or genomes.

The first attempts to analyze genome macrostructure for the purpose of phylogenetic reconstruction utilized DNA hybridization or restriction fragment analysis. As is the case for genetic phylogenies, these techniques ultimately depend on the degree of divergence of the sequences compared. In contrast, once the orthologs have been identified, the comparative analysis of gene order

discussed above is independent of the degree of divergence. A different but complementary approach is described here, based not on evolutionary lineage, but on the hierarchical classification of genomes according to their gene content and overall similarity.

This approach consists in comparing the full set of gene products predicted from the complete genome of an organism, either with itself or with sets deriving from each of the other organisms studied. First, using a classical method, the product of each open reading frame (ORF) is determined and compared with all the others. Then the *proportion* of ORFs in genome A with at least one similar ORF in genome B is determined. Overall comparison of n organisms generates an $n \times n$ matrix of these proportions. Multifactorial correspondance analysis is then used to reveal the axes of greater data variability (by rotating the axes while keeping them orthogonal). Each organism is represented by a point in this n -dimensional space. The distances between pairs of organisms are then calculated. Organisms are classified according to their neighborhood, yielding a hierarchy known as a 'genomic tree.' A genomic tree is the graphic representation of the relationship between sets of organisms, which indirectly depends on genome size, internal redundancy due to ancestral duplication, and overall gene loss/acquisition events. Nevertheless, this tree is independent of the functional identity of the genes. It is also possible to render it independent of duplication events by eliminating redundant genes within the same species from the set of initial data.

Applying this method to the first 20 sequenced genomes generates the graph in Figure 3.9, in which only the two axes with the greatest data dispersion are represented. Based on this graph, the distances calculated then permit con-

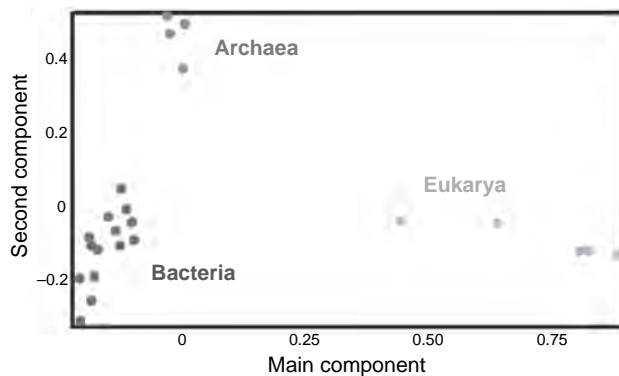


Figure 3.9 Representation of the proportion of ORFs of one organism for which there is at least one similar ORF in another organism. After multifactorial correspondence analysis, the first and second axes represented here contain 48 percent and 26 percent of the total data dispersion, respectively. Each point in this space corresponds to an organism (see their names in Figure 3.10). Groups of points correspond to major phylogenetic divisions, represented here in different shades of gray.

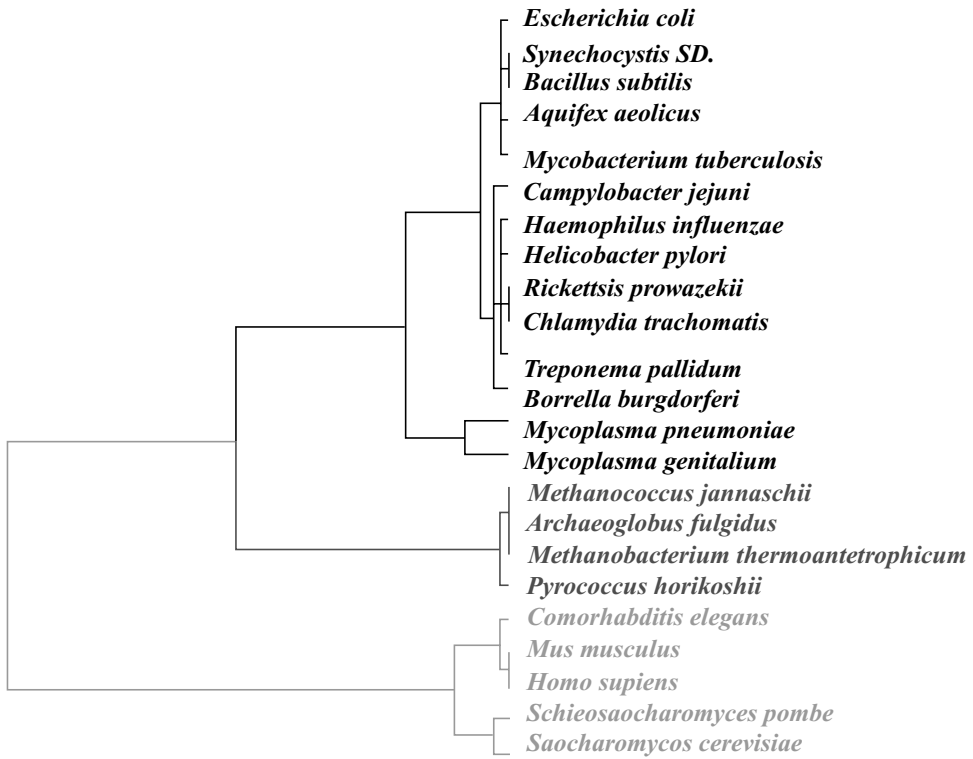


Figure 3.10 Genomic tree. This tree is obtained by the pair-wise hierarchical classification of organisms, based on their neighboring distances (see Figure 3.9). Gray-level codes are the same. Lengths of the horizontal lines between nodes are proportional to their degree of similarity.

struction of the genomic tree (Figure 3.10). Four well-defined groups of organisms appear in this tree: a) an archebacterial group; b) a eubacterial group; c) a group of mycoplasmas close to the eubacterial group, and d) a eukaryote group. This result is perfectly compatible with the ‘three kingdoms’ perspective.

Bibliography

- Casjens S. (1998). The diverse and dynamic structure of bacterial genomes. *Annu Rev Genet* 32: 339–377.
- Dujon B., *et al.* (2004). Genome evolution in yeasts. *Nature* 430: 35–44.
- Jaillon O., *et al.* (2004). Genome duplication in the teleost fish *Tetraodon nigroviridis* reveals the early vertebrate proto-karyotype. *Nature* 431: 946–957.
- Casjens S. (1998). The diverse and dynamic structure of bacterial genomes. *Annu Rev Genet* 32: 339–377.

- Dujon B., *et al.* (2004). Genome evolution in yeasts. *Nature* **430**: 35–44.
- Fisher R. A. (1930). In *The genetical theory of natural selection*. Oxford University Press, Oxford, UK.
- Huynen M.A., Bork P. (1998). Measuring genome evolution. *Proc Natl Acad Sci USA* **95**: 5849–5856.
- Jaillon O., *et al.* (2004). Genome duplication in the teleost fish *Tetraodon nigroviridis* reveals the early vertebrate proto-karyotype. *Nature* **431**: 946–957.
- Kellis M., *et al.* (2004). Proof and evolutionary analysis of ancient genome duplication in the yeast *Saccharomyces cerevisiae*. *Nature* **428**: 617–624.
- Koonin E.V., Mushegian A.R. (1996). Complete genome sequences of cellular life forms: glimpses of theoretical evolutionary genomics. *Curr Opinion Genet Develop* **6**: 757–762.
- Tekaia F., *et al.* (1999). The genomic tree as revealed from whole proteome comparisons. *Genome Res* **9**: 550–557.

4

Genetic information and biological sequences

4.1 Introduction: Coding levels

The information contained in a genome is stored at several levels, the most basic of which associates each amino acid of each protein coded by a gene to a single triplet of DNA bases (codon). Besides this elementary code, simple punctuation signals identify the beginnings and ends of genes. In addition to this 'raw' data, the genome contains expression, regulation, and alternative splicing signals (in eukaryote cells) that govern how cells implement the information it contains. The genome also contains specific signals unrelated to expression of the genetic message, which concern the metabolism of the DNA molecule itself, including replication, recombination, methylation, and restriction sites.

These data are all coded in the DNA sequence, and often mutually overlap. Genes thus contain methylation and recombination sites; certain genes partially overlap; the expression signals of one gene are sometimes located within another. . . . The unraveling of these various coding levels is of primary importance to the biologist seeking access to information contained in the genome in order to understand the functions of living matter, as well as to devise experimental strategies and to analyze results.

Information technology may be used to efficiently extract the information coded in DNA. The remainder of this chapter recalls and describes various types of signals coded in DNA, as well as specific patterns and sequences with which they are associated.

4.2 Genes and the genetic code

The principal information contained in the genome consists in the genes themselves. Confronted with the raw sequence data of a genome, the biologist first seeks to identify the various genes they contain, in order to study the proteins

TTT: Phe	TCT: Ser	TAT: Tyr	TGT: Cys
TTC: Phe	TCC: Ser	TAC: Tyr	TGC: Cys
TTA: Leu	TCA: Ser	TAA: STOP	TGA: STOP
TTG: Leu	TCG: Ser	TAG: STOP	TGG: Trp
CTT: Leu	CCT: Pro	CAT: His	CGT: Arg
CTC: Leu	CCC: Pro	CAC: His	CGC: Arg
CTA: Leu	CCA: Pro	CAA: Gln	CGA: Arg
CTG: Leu	CCG: Pro	CAG: Gln	CGG: Arg
ATT: Ile	ACT: Thr	AAT: Asn	AGT: Ser
ATC: Ile	ACC: Thr	AAC: Asn	AGC: Ser
ATA: Ile	ACA: Thr	AAA: Lys	AGA: Arg
ATG: Met	ACG: Thr	AAG: Lys	AGG: Arg
GTT: Val	GCT: Ala	GAT: Asp	GGT: Gly
GTC: Val	GCC: Ala	GAC: Asp	GGC: Gly
GTA: Val	GCA: Ala	GAA: Glu	GGA: Gly
GTG: Val	GCG: Ala	GAG: Glu	GGG: Gly

Figure 4.1 The genetic code.

for which they code. Translation of a gene into a protein associates one nucleotide triplet (*codon*) with each of the 20 natural amino acids that constitute proteins. Since there are four different nucleotides, there exist $4^3 = 64$ possible different codon triplets. The meaning of each of these 64 codons is universally conserved throughout all forms of life, and is known as the **genetic code** (Figure 4.1). Sixty-one codons specify the 20 amino acids, and three, **TAG**, **TAA**, and **TGA**, are translation stop signals, also called *termination* or *non-sense codons*. In addition, protein translation can only be initiated by certain codons known as *start* signals or codons. The main start codon is **ATG**, which specifies the amino acid methionine; however, translation can also begin with **GTG**. Whatever the start codon, the first amino acid incorporated into a protein is always methionine, even though the **GTG** codon normally specifies valine in peptide chains. A gene that codes for a given protein of length n therefore has the following structure, known as an *open reading frame (ORF)*:

$$\text{start codon} - (n - 1) \times \{\text{codon specifying an amino acid}\} - \text{termination codon}$$

When examining a sequence of cDNA or of a genome that has no introns (bacteria), analyzing the distribution of its nonsense codons furnishes valuable information concerning the potential positions of genes. In a purely random sequence in which the four nucleotides, **A**, **T**, **G**, and **C**, would be distributed equally, the statistical frequency of occurrence of these stop codons would be $3/64$, or around one stop for every 21 codons. However, a protein chain generally consists of between 100 and 1,000 amino acids, which is very significantly longer. The presence of an ORF of length equal to or greater than 100 codons in a DNA sequence is thus a very strong indication of the presence of a gene, thereby providing a predictive method. This simply requires examining stop

codon distribution in the six possible reading frames (three in each of the two DNA strands).

4.3 Expression signals

Transcription

The expression of the genes coded in DNA begins with transcription of the genetic message into messenger RNA molecules (mRNA). In eukaryotes, each mRNA molecule codes for only one gene, whereas in bacteria, several genes may be transcribed by the same RNA molecule, constituting what is known as a **transcription unit**. In all cases, the DNA sequence situated around the beginning of the transcription sequence contains a specific pattern known as a **promoter sequence**, which enables the cell to begin polymerizing mRNA.

Cells contain one or several enzyme complexes known as RNA polymerases, which carry out transcription. When transcription begins, certain protein factors or supplementary subunits whose function is to recognize the transcription promoter sequence bind to the RNA polymerase. Once transcription has begun, these factors have fulfilled their roles and detach from the RNA polymerase. For example, in *Escherichia coli*, RNA polymerase is a pentamer, $\alpha_2\beta\beta'\omega$, to which a supplementary subunit called the σ factor, which recognizes the promoter, is added at the start of transcription. The normal σ factor, known as σ^{70} , recognizes the 'classical' sequence, located a dozen nucleotides upstream from the beginning of transcription:

$5'$ TTGACA- ~17 bases- TATAAT

E. coli cells can also contain σ factors specific for other DNA sequences that determine other promoter families. Various σ factors are expressed under particular physiological conditions, such as heat shock and low nitrogen levels, in response to which they permit selective activation of the transcription of a whole group of genes.

A comparable, but much more complex situation is observed in eukaryotes, in which three distinct RNA polymerases exist, each interacting with a large number of transcription factors. RNA polymerase II, which transcribes most of the genes for proteins, binds in the vicinity of a conserved pattern known as a *TATA box*, situated around 30 nucleotides upstream from the transcription starting site:

$5'$ TATA (A or T) A (A or T)

This sequence is recognized by a specific protein known as TFIID, by which the polymerase binds. Upstream eukaryote transcription promoters usually

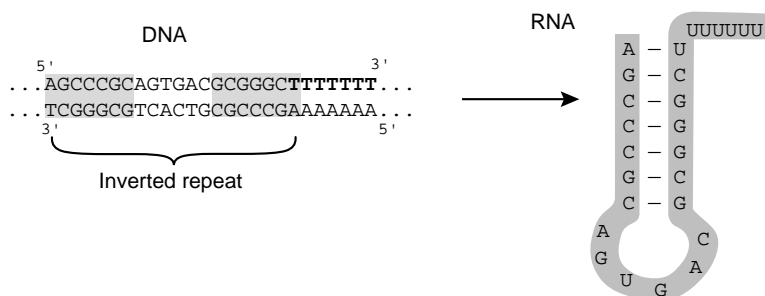


Figure 4.2 Prokaryote transcription terminators.

include other large patterns that enhance transcription efficiency, such as a 5'-GG(T or C)CAA(T or C)CT around 70 basepairs upstream from the transcription starting site, and sometimes a GC-rich motif of the 5'-GGGCGG or 5'-CCGCCC-type one-hundred basepairs upstream.

Since promoters are found upstream from transcription units, specific downstream sites exist that determine the end of transcription. In prokaryotes, these generally contain an inverted repeat (see Chapter 6 on structure prediction) able to fold into RNA, forming a hairpin structure with a stem and a single-strand loop. This sequence is followed by a series of T (U in RNA).

Transcription termination is more complex in eukaryotes. Many eukaryote messenger RNA molecules are polyadenylated at the 3'-end, and the enzyme responsible for this, poly(A) polymerase, recognizes a 5'AAUAAA3' segment downstream from the gene. This site also appears to be involved in the termination of transcription by RNA polymerase.

Alternative splicing

The product of DNA transcription in eukaryotes is precursor mRNA, which may contain non-coding segments (*introns*). The precursor RNA then undergoes a process called *alternative splicing*, which excises the introns, yielding mature messenger RNA that can then be translated into protein. This maturation is carried out by complex assemblies of proteins and nucleic acids called Small Nuclear Ribonucleo-Proteins (snRNP or SNURPs), which recognize specific sequences at intron-exon junctions:

Exon	Intron	Exon
5' ... (C ou A)AG	GURAGU ... YYUUYYYYYYNCAG	G ... 3'

Y indicates a pyrimidine (C or U), R a purine (A or G), and N any nucleotide (A, U, G, or C).

Translation

In eukaryotes, the ribosome sweeps along messenger RNA from the 5'-end, starting translation with the first ATG codon it encounters. In bacteria, where messenger RNA may be polycistronic, the translation start codon is selected by the ribosome, which recognizes the *Shine-Dalgarno sequence*, AGGAGGU, located 5 to 10 nucleotides upstream from it:

5'AGGAGGU-{5 to 10 nucleotides} (A or G)TG...

Regulation of gene expression

A wide variety of mechanisms regulate genetic expression. One or more specialized proteins exist in both eukaryotes and prokaryotes that recognize specific sites on DNA around gene promoter regions. These regulatory proteins can either activate or inhibit transcription, depending on the physiological conditions and external stimuli.

Modification signals

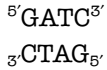
DNA is not only the medium upon which genetic information is based and that the cell uses to synthesize proteins, it is also a macromolecule that the cell must synthesize, maintain, and repair. DNA therefore also contains signals involved in its own metabolism that are recognized by specialized enzymes of which it is the substrate.

Methylation: Conservation of the genetic heritage is a concern of vital importance for cells, which have developed mechanisms for protecting the quality of their DNA. Polymerization errors may occur during the replication process, resulting in mispairing between the template strand and the complementary strand that is being synthesized. However, cells contain enzymes that can repair such defects. In the repair process, one of the two mispaired bases is excised and replaced by the base that restores correct pairing. The repaired DNA sequence differs according to the strand from which the base is excised:

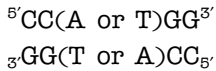
- If the excision is on the newly synthesized strand, the repaired double-stranded DNA retains the native sequence;
- If excision is on the replication template strand, the repaired double-stranded DNA bears a mutation.

The cell has developed a strategy for distinguishing the template strand from the newly synthesized strand. Specialized enzymes introduce chemical modifi-

cations (methylation of certain bases) at specific sites on the template strand, but not on the just-synthesized strand, enabling the repair enzyme to distinguish one strand from the other. For example, in *E. coli*, there are two modification systems, *dam* methylase, which methylates adenines in the symmetrical motif



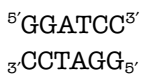
and *dcm* methylase, which methylates cytosines in the quasi-symmetrical pattern



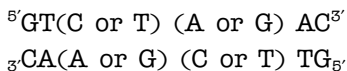
Similar specific cytokine methylations exist in animal cells for 5'-CG-3' sequences and in plants for 5'-CNG-3' sequences (N can be any nucleotide; A, T, C, or G).

Restriction: An analogous mechanism is used by prokaryotes for defense against bacteriophage viruses. The bacterial cell produces an endonuclease that recognizes a specific DNA sequence and cleaves the two strands of the double helix. This action of this enzyme permits degradation of the viral genome, preventing infection. In order to avoid cutting its own chromosome, the bacteria also contains a methylase that recognizes the same site as the endonuclease and modifies some of the site's bases, inhibiting the nucleolytic activity of the endonuclease. At least one strand of the bacterial chromosome is always methylated at these sites, protecting it from the endonuclease.

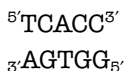
Several varieties of restriction enzymes exist, all of which recognize very local, well-defined motifs of between four and a dozen bases. The strands are usually cut at the recognized site, or in the immediate vicinity. Some endonucleases/methylases, such as *Bam*HI, have perfectly symmetrical (**palindromic**) sites:



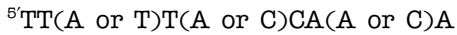
Others, such as *Hind*II, have quasi-palindromic sites, with several possible bases at certain positions:



Finally, a small number, such as *Hph*I, have non-palindromic sites:



Replication: DNA contains specific patterns involved in replication. Replication usually starts at sites known as **replication origins**, according to various mechanisms that vary as a function of the organism and type of DNA (chromosome, plasmid, or virus). Replication generally begins with the binding of a specialized protein to a nucleotide pattern located at the replication origin. This protein promotes fusion of the double helix, required to initiate replication. For example, in *E. coli*, the chromosome replication origin bears four tandem repetitions of the following nine-base sequence:



which is recognized by the initiation protein known as **DnaA**.

In general, a great number of proteins involved in DNA metabolism ‘selective’ for certain nucleotide sequences. Therefore it is possible to identify specific patterns at which certain events occur, for example, recombination, integration of a viral genome, and the action of certain gyrases.

4.4 Specific sites

Therefore, as just discussed, identification of the DNA sequence of a site that corresponds to a biological function consists in searching for a pattern. The essential characteristics of the principal types of sites encountered in biological sequences are listed below. Studying these signals reveals the existence of two distinct families, according to whether recognition of the specific motif occurs on template or transcribed RNA. The mechanisms involved are quite different, leading to the identification of two major classes.

4.5 Sites located on DNA

Proteins are able to ‘read’ double-stranded DNA bases

As seen in the above examples of signals directly recognized on double-stranded DNA, there is almost always a protein that ‘reads’ the pattern. To understand how proteins are able to decode a sequence of bases requires a review of the DNA double helical structure.

The two deoxyribose phosphate chains wind around each other on the outside of the molecule, inside which the basepair plates are stacked perpendicular to the axis of the helix. This arrangement forms two lateral grooves of unequal size in which the basepairs are accessible edgewise. Proteins can slide into either of these grooves (most often the larger one), thereby entering into specific interactions with particular basepairs. These basepairs expose different chemical functions, which play either acceptor or donor roles in

hydrogen bonding with the lateral chains of the amino acids of proteins lodged in the grooves.

Dimeric proteins bind to palindromic sites

In analyzing the various types of signals that are recognized by proteins, it becomes clear that a large number of them correspond to palindromic or quasi-palindromic sites. For the DNA molecule, this consists of two-fold symmetry around an axis centered on the middle of the motif. In the majority of cases, these sites are recognized by proteins that consist of an even number of subunits that also have two-fold symmetry, dimers, tetramers, etc. . . . Each half of the DNA site is recognized by one of the halves of the protein, which has the same axis of symmetry as the DNA in the complex.

This symmetrical subunit arrangement can accommodate a large number of regulatory proteins, methylation enzymes, and restriction enzymes, and allows the cell to increase interaction specificity by doubling the size of the pattern recognized without increasing the size of the protein subunit involved in the recognition. Figure 4.3 illustrates such an interaction between a dimeric protein and a palindromic site, the protein *cro* of bacteriophage λ . This regulatory protein binds to viral DNA on sequences of the following type:

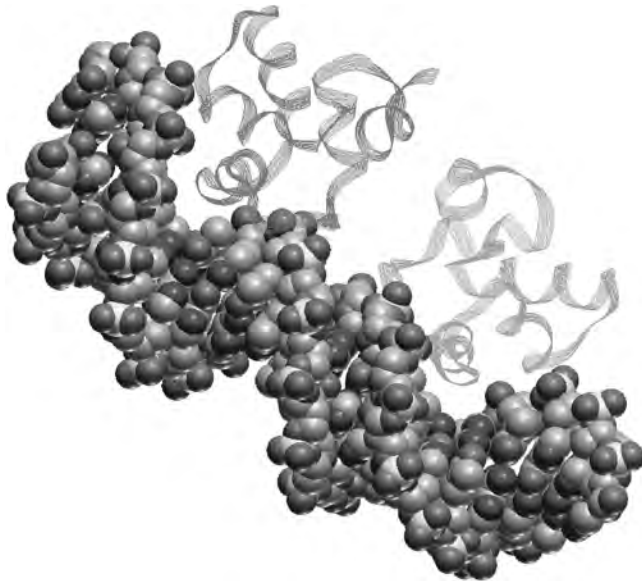
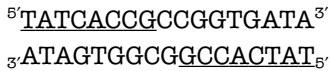


Figure 4.3 Structure of a complex consisting of the bacteriophage lambda *cro* regulatory protein and its specific DNA site.



and stimulates transcription of genes implicated in the lytic cycle of the virus.

Some complex sites consist of several subsites

Certain factors involved in the recognition of specific DNA sites are complex structures, such as nucleoprotein assemblies (for example, ribosomes) and very large proteins consisting of multiple subunits (for example, RNA polymerase). They may therefore be in contact with several distinct regions on the DNA. Each region constitutes an independently recognized sub-site, the spacing between which varies over a more or less elastic range of values. Since the contact zones are discontinuous, therefore distributed over a greater distance, this variable spacing may possibly be due to the flexibility of both the DNA double helix and the molecule with which it is in contact.

Transcription promoters are examples of the most well known complex sites, both in prokaryotes and eukaryotes. They consist of several distinct 'boxes' separated by zones of various sizes. Translation start sites in prokaryotes may also be considered complex sites, since they consist of both the Shine-Dalgarno sequence and the start codon, situated about a dozen bases downstream.

Expression signals are often fuzzy patterns

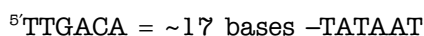
Until now, only patterns strictly defined by a given nucleotide sequence have been covered in this chapter, although in some cases, one or several ambiguities have been introduced in the form of alternatives of the type '(A or T)', which has the simple effect of multiplying the acceptable sequences; for example, GT(C or T)(A or G)AC, defines one of the following sequences: GTCAAC, GTCGAC, GTTAAC, and GTTGAC. However, the corresponding sites remain clearly defined and easily recognizable.

Some proteins, such as methylases and restriction enzymes, recognize such strict patterns and tolerate no variation in the admissible sequences. The situation is different with respect to regulation signal sites. In general, it is impossible to define a precise sequence that exactly corresponds to a regulation signal site. However, based on experimental observations, it appears that sets of sequences corresponding to regulation signal expression sites more or less always tend toward an ideal sequence, which is known as a **consensus sequence**.

Such fuzzy patterns allow cells to modulate the efficacy of various expression signals according to their needs. The closer a signal expression sequence approaches to the consensus sequence, the more effective it is. The protein(s)

involved can effectively enter into a certain number of molecular interactions (hydrogen and ionic bonding, hydrophobic interactions, etc. . . .) with the bases of the consensus sequence. At positions where the sequence of a site differs from the canonic sequence, interactions with the corresponding bases will be broken. The interaction energy will therefore be lower, which destabilizes formation of the DNA/protein complex to the extent that the site sequence differs from the consensus sequence.

Most of the expression and regulation signal sequences mentioned above are canonic consensus sequence patterns that correspond to maximum effectiveness. For example, the sequences of transcription promoters corresponding to genes that are very strongly expressed in prokaryotes are often practically identical to the following canonical sequence:



Less strongly expressed gene promoters may differ more or less significantly from this ideal sequence, either in the spacing between the two patterns or in the nature of the bases in each conserved cell. In addition, the variation observed among the bases of the consensus sequence is not constant. Constraints on certain positions seem to be greater, probably because the interactions they make with RNA polymerase are stronger and/or more numerous.

This naturally leads to the **weighting** of various bases recognized at a given site. Certain bases may be strictly conserved, and are therefore probably essential, whereas others are less strictly conserved. Figure 4.4 illustrates the results of statistical analysis of the conservation of the consensus sequence bases in a compilation of 112 experimentally characterized *E. coli* promoters.

Patterns that induce structural changes in the double helix

Until now, the signals discussed have generally constituted sites recognized by other macromolecules, essentially proteins. In addition, specific sequences exist that induce deformations or other structural changes in the DNA molecule, in the absence of an external protein factor. These regions can nevertheless also be the targets of interaction with proteins that then recognize the perturbation in the structure, without necessarily specifically 'reading' its sequence.

T	T	G	A	C	A	—	T	A	T	A	A	T
82%	84%	79%	64%	54%	61%	—	79%	95%	44%	59%	51%	96%

Figure 4.4 Average frequency of various bases in the consensus sequence of *E. coli* transcription promoters.

The most basic structural change is the appearance of a curvature in the axis of the double helix. The simplest interpretation is that the curvature is caused by an accumulation of deformations at the level of basepair stacking. The geometry of this stacking is determined by three angles, as indicated in Figure 4.5.

The twist angle determines the step of the double helix, which on the average is 34.7° for type B DNA. The roll and tilt angles vary by a few degrees, as a function of the nature of the basepairs, i and $i + 1$, which can introduce a slight bend. In a given sequence, these slight deformations cancel each other out, resulting in a nearly straight double helix axis. Nevertheless, in particular cases, the small deformations distribute in phase with the step of the helix, and their cumulative effect yields a non-zero curvature.

In particular, the dinucleotide AA (or TT on the complementary strand) causes significant variations in the roll and tilt angles. Some sequences in which series of 3 to 5 A (or T) are regularly spaced every 10 or 11 nucleotides have been described. Since the double helix contains around 10.4 basepairs per turn, all the A in such a region are located on the same side of the double helix; therefore the effects of the induced curve are cumulative.

In general, regularly repeating nucleotide sequence patterns whose periodicity corresponds to that of the double helix indicate the presence of a curved region and/or the binding site for a protein that interacts with one face of the DNA. Figure 4.6 illustrates a region located upstream from a bacterial gene that binds a regulatory protein called Lrp. This sequence is probably curved.

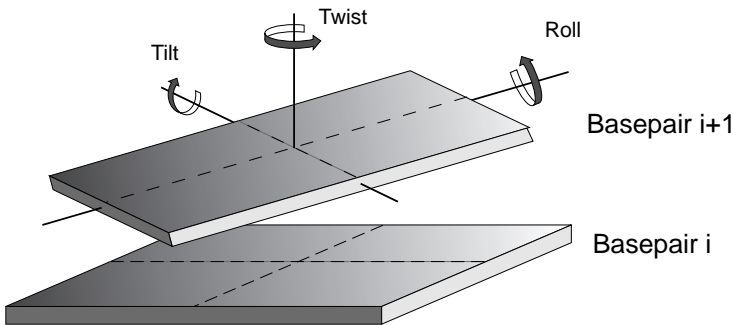


Figure 4.5 Angles that determine the local geometry of the stacking of two basepairs.

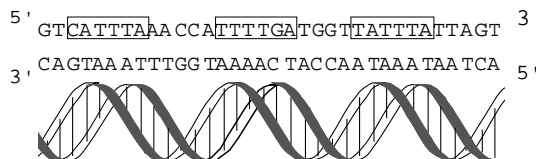


Figure 4.6 Binding site of the Lrp protein. A T-rich segment repeats approximately every 10 to 11 bases. The structure of the double helix is indicated below on the same scale.

4.6 Sites present on RNA

Certain sites are recognized by other RNA molecules with which they can pair

RNA present in the cell is usually single-stranded, which allows it to locally pair with another RNA strand. The sites recognized are usually fuzzy, and the strength of the interaction depends on the stability of the duplex RNA formed. Using the empirical rules described in chapter 6 to calculate the sites, it is generally possible to estimate their stability. To do this, the sequence of the complementary RNA strand involved in the recognition must be known, which requires studying the underlying biological mechanism.

Among the recognition mechanisms described that utilize RNA:RNA pairing are translation initiation in prokaryotes, in which the 3'-end of the 16S ribosomal RNA fragment pairs with the Shine-Dalgarno sequence of messenger RNA, and alternative splicing in eukaryotes, in which intron-extron junctions interact with small RNA fragments in the nucleus (snRNP or SNURPs).

Certain sites correspond to local stem and loop RNA folding. Single-stranded RNA can also pair with itself (internal pairing) and fold locally, forming stem and loop structures. Such secondary structures can play multiple roles; for example, they can serve as specific sites for proteins that recognize the shape rather than the sequence of an RNA segment. Inversely, RNA regions involved in internal pairing cannot interact with other RNA molecules. Some sequences may be 'masked' by the formation of a stem and loop structure. Finally, as in the case of DNA transcription terminators, the presence of such a secondary structure can hinder the progress of a polymerization enzyme, leading to a pause or block.

The existence of an RNA segment able to fold into stems and loops is revealed by the presence of extended Palindromic regions (repeated inverse regions), such as the DNA binding sites of dimeric proteins (Section 4.5). When such palindromes are detected in a sequence, the biological context must be analyzed in order to distinguish between these two alternatives: a symmetric site on the DNA molecule, and a stem-and-loop structure on the transcribed RNA.

4.7 Pattern detection methods

The sequencing programs used today generate huge quantities of raw data from which biologically pertinent information must be extracted, such as the locations of genes, expression signals, etc. This analysis essentially involves two complementary approaches:

- **Pattern detection**, which consists in locating defined sequence segments that correspond to documented biological functions, such as those described in the preceding paragraphs;
- **Content search**, which analyzes the properties of the statistical distribution of nucleotides or amino acids in a sequence. This approach will be described in Chapter 5, ‘Statistics and sequences’.

The following is an intentionally simplified description of the methods used in pattern detection. Known as *pattern matching* in basic computer science, this heavily studied domain employs high-performance algorithms that employ very formal methods. In what follows, these algorithms will be approached using only practical ‘biological’ examples. For a fully formal treatment of this subject, the reader is advised to refer to specialized works (see bibliography.)

Simple searches

Numerous patterns associated with biological functions are relatively simple in nature. For example, the eukaryotic messenger RNA polyadenylation signal is a non-degenerate hexanucleotide whose sequence is AAUAAA (AATAAA in the corresponding DNA sequence). The search for occurrences of such a non-degenerate pattern in a long DNA sequence may be conducted using the following naïve algorithm:

```

i ← 1
k ← 1
Do
  If Pattern[k] = Sequence[i+k] then
    k ← k+1
  Else
    k ← 1; i ← i+1
  If k > Length(pattern) then
    pattern found at position i
    k ← 1; i ← i+1
While i ≤ Length (Sequence)

```

This simple algorithm consists in nucleotide-by-nucleotide comparison of the pattern with the target sequence at a given position. If the comparison fails, the pattern is advanced one position and the operation repeated. This algorithm has the advantage of being simple; its cost is simply a function of the target sequence length. Nevertheless, it is not optimal, since it may require a number of com-

parisons (line shaded in gray) that are greater than the length of the sequence itself. To demonstrate this, it suffices to observe what happens when the target sequence is very rich in A and U¹: For example,

. . . AAAUAUUAAAUAUUAGACGAAUAAAAGUAUAUUUAG . . .

At the beginning of the search, the situation is the following:

```
target:  AAAUAUUAAAUAUUAGACGAAUAAAAGUAUAUUUAG
pattern:  AAUAAA
test      ++-
```

The naïve algorithm will produce two positive comparisons (+) for the two first pairs of A, before failing in the third (-). It will then shift the pattern one position and repeat the analysis. This time, it will fail in the fifth comparison, after four positive tests:

```
target:  AAAUAUUAAAUAUUAGACGAAUAAAAGUAUAUUUAG
pattern:  AAUAAA
test      +++++-
```

The algorithm compared a total of eight positions just for the two positions tested. This example was particularly unfavorable, but for strongly biased target sequences, it is possible to conduct twice as many comparisons as the number of nucleotides they include².

This method is not optimal, since when the naïve algorithm yields several coincidences between the pattern and the sequence before failing it returns to the target sequence, in which case it reads some target sequence nucleotides several times. It is possible to improve the naïve algorithm by making it ‘remember’ what has already been read, thus avoiding going back.

Searching with a finite state automaton

To keep track of information concerning nucleotides that have already been read, a virtual machine known as a **finite state automaton** is devised. The machine is characterized by a finite number of certain states. The automaton

¹ While not a textbook case, this problem is frequently encountered with the genomes of some organisms that are particularly rich in A and T (see chapter 5, ‘Statistics and sequences’).

² Even in the best case, that of a target sequence in which the four nucleotides are distributed equally, it may be shown that the cost of the naïve algorithm is 35 percent greater than that of an optimal algorithm.

begins to function in a particular state, called the *initial state*. The machine reads one nucleotide in the sequence during each cycle, then evolves toward a new state, according to the nucleotide and its present state. If the automaton reaches another specific state, known as the final state, the pattern has been found and the automaton stops.

The following three elements constitute a finite automaton:

- An alphabet Σ of characters for the target sequence (for biological applications, four nucleotides or twenty amino acids);
- A list E of n states $e_0, e_1 \dots e_n$. This list includes an initial state, e_0 , and one or more final states.
- A transition function T which is an application of $E \times \Sigma \rightarrow E$. This function determines the $i + 1$ state, starting from the current state i and the character read. This transition function is stored in the form of a matrix of dimensions (n, k) where n is the number of states and k is the size of the alphabet Σ (four nucleotides or twenty amino acids).

The automaton can then be simulated by applying a very simple algorithm:

```

i ← 1
state ← e0
Do
  State ← T[state, sequence[i]]; i ← i+1
  If state = final_state then
    Pattern found at position i; Quit
While i ≤ Length(sequence)
  
```

This algorithm must read each nucleotide or amino acid in the sequence only once and not go back over it. The problem is obviously to specify the list of states and transition table, so that the automaton will search for the pattern. Taking the example of the eukaryote polyadenylation pattern, **AAUAAA**, it is possible to achieve this with the automaton, a graphic representation of which appears in Figure 4.7.

This helpful graphic representation permits better understanding of the operating principle of the automaton. In particular, it is possible to see how reading the pattern sought, AAUAAA, leads directly to the final state (state 6). The transition table, T , that corresponds to this graph is indicated below (there is no entry for state 6, since it is the final state).

Using the above automaton to analyze the target sequence, the reader may verify whether it goes through the states indicated below and identifies the pattern (underlined):

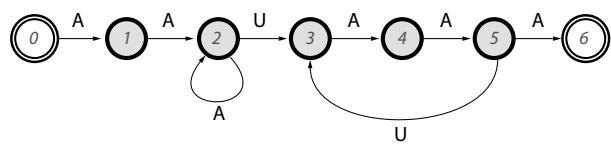


Figure 4.7 Graphic representation of a finite state automaton seeking a polyadenylation signal, AAUAAA. States are symbolized by numbered circles. The initial state is state 0 and the final state is state 6. Transition rules are represented by arrows beneath the nucleotides associated with each transition. Transitions not represented return to the initial state (for example, if the automaton reads something other than A in state 0, it remains in state 0).

		nucleotide read			
		A	G	C	U
initial state	0	1	0	0	0
	1	2	0	0	0
	2	2	3	0	0
	3	4	0	0	0
	4	5	0	0	0
	5	6	0	0	3

target: AAAUAUUAAAUAUUAGACGAAUAAAAGUAUAUUUAG
state: 01223400122340010100123456

The state of the automaton at a given instant constitutes its ‘memory’. It permits knowing which left sub-part of the pattern has already been identified. If the automaton is in state 2, this indicates that it has just read two A (adenine). If it reads a third one, which does not correspond to the sequence sought, AAUAAA, the two last A of the three read still correspond to the beginning of the pattern.

Degenerate patterns

The Knuth-Morris-Pratt (KMP) method functions well for simple, non-degenerate patterns and may readily be adapted to other, partially degenerate types of patterns, such as consensus sequence sites that give rise to alternative splicing in eukaryotes (exon-intron junction), for example, (C or A) AGGU(AA or C)AGU. Generalizing the KMP approach, it is possible to devise automata capable of seeking such patterns. For example, Figure 4.8 is a graphical representation of a finite automaton that searches for a pattern corresponding to pre-

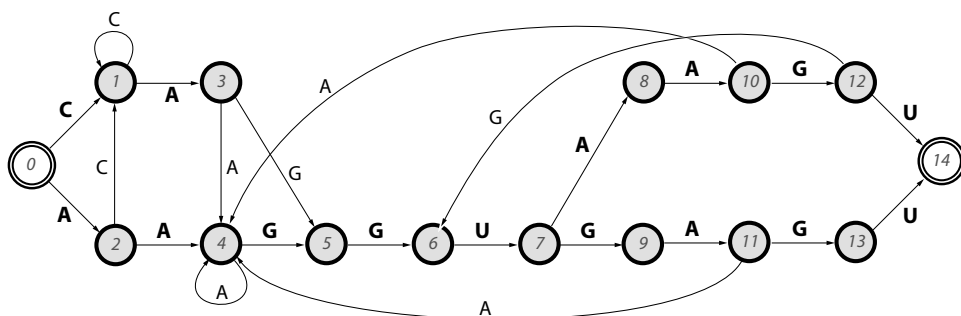


Figure 4.8 Simplified representation of a finite state automaton that searches for the pattern **(C or A)AGGU(A or G)AGU**, which corresponds to sites that give rise to alternative splicing in eukaryote RNA. Not all transitions are represented. Missing transitions, which correspond to reading a **C**, return to state ①; those corresponding to reading an **A** return to state ②, and the others return to the initial state.

sumptive alternative splicing sites. It is more complex than the one in the preceding paragraph, consisting of 13 states and presenting a branched structure that allows managing the degenerated positions of the pattern. Some arrows corresponding to the return to states 0, 1, and 2 have been omitted in order to simplify the figure.

Regular expressions

Finite state automata are extremely powerful tools that can recognize a wide variety of different patterns, not only in the context of biological sequences, but in other domains as well. They are often utilized by text and program editing software that includes sophisticated functions of the ‘search/replace’ type. In general, these search functions employ a language for identifying somewhat complex patterns in the files analyzed. In general, this rather extensive language may be reduced to what are called ‘regular expressions’.

Regular expressions are defined with respect to three basic operations:

1. *Concatenation*, the consecutive linking of two patterns: pattern__1 pattern__2
2. *Alternation* between two patterns, notated ‘or’: pattern__1 or pattern__2
3. *Repetition* a given number of times (0 or more) of a pattern, notated ‘*’: pattern*

According to the application concerned – word-processing, biological sequences, program editors, etc – the basic language can adopt slightly different syntaxes and take on operators that simplify notation, but which can always be reduced to the above three fundamental operations. The formal framework of regular expressions allows defining a very wide variety of interesting biological patterns, such as codons, open reading frames, elementary signals, and consensus signals, of which some examples follow. In order to reduce the complexity of the expressions, simple notations are introduced, some of which derive from the syntax of the Unix-based **grep** program.

For the simple alternatives using alphabetic symbols, the following notation within brackets is used:

[ATG] \Leftrightarrow (A or T or G)

[AG] \Leftrightarrow (A or G)

Sometimes ‘N’ or ‘X’ are used as wild cards to indicate any nucleotide or amino acid, respectively:

N \Leftrightarrow [ATGC]

X \Leftrightarrow ACDEFGHIKLMNPQRSTVWY

Among these simple constructions, it is possible to define ‘punctuation’ codons, which identify the beginning and end of an open reading frame:

start codon: [AG] TG

stop codon: T([GA]) or A[AG])

A coding codon, that is, a non-stop codon, is longer, but still accessible:

coding codon: [ACG]NN or T([CT]N or G[CGT] or A[CT])

Starting from these constructions, it is possible to define an open reading frame as being a start codon assembly followed by some number of coding codons (identified by a ‘*’ symbol) and ending with a stop codon.

open reading frame: [AG]TG([ACG]NN or T([CT]N or G[CGT] or A[CT]))* T([GA] or A[AG])

It is also possible to define variable spacing between patterns, like the one separating the prokaryote ribosome binding site from its start codon (between 6 and 12 nucleotides).

Ribosome binding site: AGGA or GGAG or GAGG

Translation start site: (AGGA or GGAG or GAGG)(N or NN or NNN or NNNN or NNNNN or NNNNNN or NNNNNNN) [AG]TG

This type of pattern definition by regular expression can be very readily generalized to protein sequences, using the 20-character amino acid alphabet instead of the four nucleotides.

The PROSITE pattern library (<http://www.expasy.org/prosite>, see Chapter 2) employs regular expression syntax in this way to define more than a thousand patterns and protein signatures.

Finite automata and regular expressions

The fundamental property of patterns defined by regular expressions is that they **exactly** correspond to patterns that may be sought using finite state automata. In other terms, if it is possible to represent a biological pattern of interest using a regular expression, then it is possible to construct a finite state automaton which seeks it in a sequence, and reciprocally.

The following is an example of an automaton that seeks a pattern which is a simplified version of the expression that defines an open reading frame:

[AG]TG (NNN)* T([GA] or A[AG])

Figure 4.9 displays an example of an automaton that recognizes patterns defined by the above regular expression:

This automaton is different from those described above in that its behavior is not deterministic. For example, if a T is read in state 6 of the sequence analyzed, there exist two possible actions: either going on to state 7, which amounts to trying to identify a stop codon, or returning to state 4; that is, continuing the open reading frame. Therefore the automaton has several different possible routes and *a priori* it is impossible to say which one leads to the final state, state

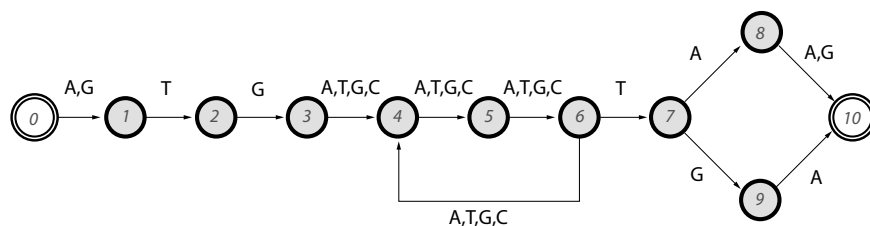


Figure 4.9 A non-deterministic finite state automaton that recognizes open reading frames. The arrows indicating the return to the initial state are not represented.

10, which is why this construction is known as ‘a non-deterministic finite state automaton’. If in reading a DNA sequence, one of the possible paths in the graph in Figure 4.9 leads to the final state (state 10), then one occurrence of the pattern sought has been found; that is, an open reading frame.

Non-deterministic automata may be used to conduct searches for complex patterns contained in biological sequences. In order to simulate the multiplicity of possible paths in graphs for a non-deterministic automaton, a *set* of states replaces the single state used in deterministic automata. The transition function T must then be modified. Deterministic automata use the application $E \times \Sigma \rightarrow E$, in which E is the list of states and Σ the alphabet utilized. For non-determinist automata, starting from a given state e_i , for a character read in Σ , it is possible to end up at several different states. In the example used in Figure 4.9, starting from state 6, when a T is read, the automaton ends up in states 7 and 4. The transition function T is therefore an application of $E \times \Sigma \rightarrow P(E)$, where $P(E)$ represents the set of all subsets of E .

		nucleotide read			
		A	G	C	T
initial state	0	{0.1}	{0.1}	{0}	{0}
	1	{0}	{0}	{0}	{0.2}
	2	{0}	{0.3}	{0}	{0}
	3	{0.4}	{0.4}	{0.4}	{0.4}
	4	{0.5}	{0.5}	{0.5}	{0.5}
	5	{0.6}	{0.6}	{0.6}	{0.6}
	6	{0.4}	{0.4}	{0.4}	{0,4,7}
	7	{0.8}	{0.9}	{0}	{0}
	8	{0.10}	{0.10}	{0}	{0}
	9	{0.10}	{0}	{0}	{0}

The above transition table corresponds to the automaton represented in Figure 4.9. The lists of states obtained after each transition are indicated in parentheses.

The functioning of such non-deterministic automata may then be simulated by applying the following algorithm, which is a generalization of the one given above for deterministic automata:

```
i ← 1
state_list ← {e0}
Do
  list ← {}
  For ei in state_list
    list ← union (T[ei, sequence[I]], list)
```

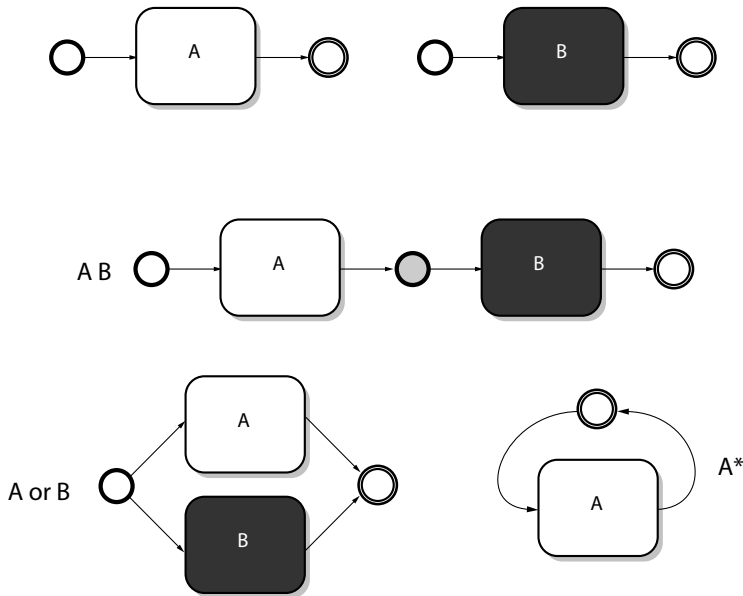


Figure 4.10 Principles of automaton assembly. The initial and final states are represented by boldface or double circles, respectively. Concatenation is achieved by merging the final state of the first automaton with the initial state of the second. Alternation is obtained by merging the initial and final states of the two automata. Repetition is obtained by merging the initial and final states of the automaton which recognizes the pattern that must be repeated.

```

state_list ← list
i ← i+1
If final_state ∈ state_list then
  pattern found; quit
While i ≤ Length (sequence)

```

The following is an example of an analysis obtained using this automaton. The sequence read is indicated above the list of states through which the automaton passes upon analyzing each nucleotide. Start and stop codons in the miniature open reading frame detected by the automaton are underlined in the sequence.

```

GGATCCTATGATAACATGACCTGG
000000000000000000000000
 1112   121121151231
      34564   6454
      796
      10

```

Non-deterministic automata may be constructed directly from regular expressions by applying the three basic rules defined above, concatenation, alternation, and repetition. Figure 4.10 is a schematic representation of how to construct automata that recognize the expressions (AB) , $(A \text{ or } B)$, and (A^*) , starting from automata that recognize the expressions A and B , respectively. The principle of these constructions consists in appropriately merging the initial and/or final states of elementary automata. By iterating these assemblies, it is possible to progressively construct a non-deterministic finite state automaton that corresponds to any regular expression.

Bibliography

- Aho A.V. (1990). Algorithms for finding patterns in strings. In *Handbook of Theoretical Computer Science, Volume A: Algorithms* (van Leeuwen, J., ed.), pp. 255–300. Elsevier, Amsterdam.
- Attwood T.K. (2000). The role of pattern databases in sequence analysis. *Brief Bioinform* 1: 45–59.
- Bairoch A. (1991). PROSITE: a dictionary of sites and patterns in proteins. *Nucleic Acids Res* 19 Suppl: 2241–2245.
- Bairoch A., Bucher P. (1994). PROSITE: recent developments. *Nucleic Acids Res* 22: 3583–3589.
- Gattiker A., *et al.* (2002). ScanProsite: a reference implementation of a PROSITE scanning tool. *Appl Bioinformatics* 1: 107–108.
- Gusfield D. (1997). Algorithms on strings, trees, and sequences: In *Computer Science and Computational Biology*. Cambridge University Press, Cambridge, UK.
- Knuth D.E., *et al.* (1977). Fast pattern matching in strings. *SIAM J Comput* 6: 323–350.
- Sigrist C.J., *et al.* (2002). PROSITE: a documented database using patterns and profiles as motif descriptors. *Brief Bioinform* 3: 265–274.

5

Statistics and sequences

5.1 Introduction

Nucleotide sequences extracted from a database usually seem quite unreadable, and at first sight may be difficult to distinguish from a random series of letters A, T, G, and C. However, this impression is erroneous, since ‘real’ nucleotide sequences are replete with redundancies and statistical biases resulting from the processes of evolution and natural selection to which they have been subjected. Studying these biases is extremely informative for the biologist, since they provide information concerning the origins of the phenomena responsible for them, leading to a better understanding of how living cells exploit their genetic information. Once the mechanisms involved in such biases have been characterized, their analysis and systematic investigation become valuable tools for use in predicting the properties of other biological sequences.

5.2 Nucleotide base and amino acid distribution

This chapter covers the distribution of nucleotide bases in genomes and of amino acids in proteins. Analysis will proceed in the direction of increasing complexity, beginning with monomer frequency, followed by more complex patterns of length n (n -tuples), including their frequencies and correlations with monomer frequency, as well as with various types of sequences (coding and non-coding strands, introns, exons, etc).

Genome composition

Since the sequence of bases constituting the genome of living species consists of the four nucleotide bases, A, T, G, and C, the first point to consider with respect to genome composition is the relative frequencies of the bases. The principle of base complementarity in the DNA double-helical structure imposes overall

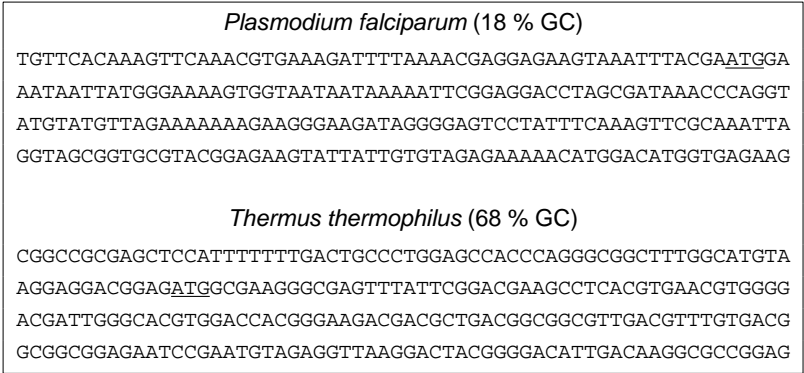


Figure 5.1 Sequence of two gene fragments derived from organisms whose genomes contain very different proportions of GC. Initiation codons are underlined.

equality in the numbers of A and T, and G, and C. What is generally studied is the $(G+C)/(A+T)$ ratio or the GC percent in a given genome or part thereof.

Well before sequencing methods had been developed, physicochemical techniques, such as the measurement of DNA density and fusion temperature, were used to demonstrate the wide disparities in the genomic GC content of various species. These differences were later confirmed by statistical analysis of sequences extracted from databases.

GC/AT distribution in the *E. coli* genome is more or less equal; around 51 to 49 percent. However, equality is not the general rule, and the GC content can range between 15 and 70 percent, according to the species. Thus, only around 18 percent GC is found in the genome of the malaria protozoan *Plasmodium falciparum*, versus 68 percent in *Thermus thermophilus*, a bacterium that lives in hot springs able to thrive in temperatures exceeding 80°C (see Figure 5.1). The overall GC level in vertebrate genomes is intermediate, ranging between 40 and 45 percent. However, the situation is somewhat more complex, since the vertebrate genome is segmented into large regions called ‘isochores’, in which the GC level is locally constant, but different from that in neighboring regions.

The origin of the variation in the GC rate among various living species is not entirely clear. In thermophilic organisms, the genome generally includes a high proportion of GC, which permits DNA to better resist thermal denaturation, since G-C pairing is more stable than that of A-T.

Comparison of proteins

In contrast to DNA, the average protein composition remains relatively constant throughout all living organisms. However, there is some variation, accord-

Alanine	A: 8.3%	Methionine	M: 2.4%
Cysteine	C: 1.7%	Asparagine	N: 4.4%
Aspartate	D: 5.3%	Proline	P: 5.1%
Glutamate	E: 6.2%	Glutamine	Q: 4.0%
Phenylalanine	F: 3.9%	Arginine	R: 5.7%
Glycine	G: 7.2%	Serine	S: 6.9%
Histidine	H: 2.2%	Threonine	T: 5.8%
Isoleucine	I: 5.2%	Valine	V: 6.6%
Lysine	K: 5.2%	Tryptophan	W: 1.3%
Leucine	L: 9.0%	Tyrosine	Y: 3.2%

Figure 5.2 Average protein composition throughout all living organisms.

ing to the type of protein considered. For example, membrane proteins are rich in hydrophobic residues, and proteins that interact with nucleic acids are often more basic, but these are general tendencies that depend little on the species. In the ‘standard’ protein composition indicated in Figure 5.2 it may be seen that the distribution of the 20 amino acids is unequal: some amino acids are relatively abundant, while others are less common. The average amino acid distribution is a kind of universal signature that may be used to distinguish a real protein sequence from a random one.

N-tuple frequency

Analysis of the incidence of simple nucleotides may be extended to dinucleotides, trinucleotides . . . and more generally, to n -tuples. Triplets will be specifically treated below, in connection with the genetic code.

In this type of analysis, the frequency of occurrence of an n -tuple of the bases $B_1B_2 \dots B_n$, $f_{B_1B_2 \dots B_n}$, is compared with the product of the frequencies of occurrence of the individual bases $f_{B_1} \cdot f_{B_2} \dots f_{B_n}$. If it is lower, the n -tuple is underrepresented, if higher, it is overrepresented.

This kind of analysis may be carried out on an entire genome, or on two families of disjointed sequences in parallel, in attempting to discriminate between them (e.g., coding or non-coding sequence; intron or exon). Among the most spectacular results of such analysis are:

- Marked underrepresentation of the CG dinucleotide throughout all vertebrate genomes. The CG sequence is a cytosine methylation signal. The 5-methylcytosines located at these sites can be transformed into T (thymine) by deamination, a common type of chemical damage. It is clear then that, little by little, owing to the effect of successive mutations, CG sequences can become TG sequences, thus becoming increasingly rare in the genome.

- Underrepresentation of the CTAG palindromic quadruplet in the *E. coli* genome:

$$f_{\text{CTAG}} = 3.6 \times 10^{-4} \ll f_{\text{C}}f_{\text{T}}f_{\text{A}}f_{\text{G}} \approx 3.9 \times 10^{-3}$$

In general, palindromes appear to be underrepresented in bacterial genomes.

Frequency of base triplets; use of the genetic code

The fact that DNA includes genes that code for proteins imposes additional constraints on the nucleotide sequence that constitutes a gene. Since translation of DNA into protein uses elementary words called *codons*, which consist of three bases, it is reasonable to examine the frequencies of the 64 possible triplets. The presence of long open reading frames without stop codons is already a first indication of the non-random character of their distribution.

Respecting the reading frames, it is possible to analyze the frequency of the 61 base triplets corresponding to the 20 amino acids. Codon distribution must necessarily follow that of the corresponding amino acids in the genetic code. For example, on average, tryptophan residues constitute 1.3 percent of proteins; therefore genes should consist of 1.3% TCG, the only codon that corresponds to that amino acid.

It is more interesting to look at what happens in the case of amino acids that are coded by several *synonymous* codons. For example, lysine, which represents 5.7 percent of the amino acids present in proteins, can be specified by two codons, AAA and AAG. Are these two triplets represented in an equal manner? For 100 lysine codons in the human genome, there are only around 38 AAA for every 62 AAG. In the *E. coli* genome, the proportion is reversed, with 60 AAA for every 40 AAG. Cells therefore express a preference for certain synonymous codons. This preference is also species-specific.

It is possible to compute the statistics of the occurrence of various codons in the genes of a species, compiling what is known as a *codon usage table*. Figure 5.3 gives the codon usage tables for *E. coli* and for the human. The various codon frequencies observed are the result of two superimposed effects: the amino acid composition of the proteins (which is not uniform) and the systematic preference for certain codons among the various possible synonymous codons. Certain triplets, such as the arginine codon AGG in *E. coli* and the alanine codon GCG in the human, which appear to be systematically avoided, are known as ‘rare codons’.

When compiling codon usage tables for different genes derived from the same organism, it is clear that they are all very similar, in general, faithfully reflecting the overall table. Thus, every gene generally conforms to these preference rules, which are a kind of ‘signature’ of the genome in which it occurs.

Phe TTT: 1.9	Ser TCT: 1.0	Tyr TAT: 1.5	Cys TGT: 0.6
Phe TTC: 1.8	Ser TCC: 1.0	Tyr TAC: 1.4	Cys TGC: 0.5
Leu TTA: 1.0	Ser TCA: 0.6	TAA: STOP	TGA: STOP
Leu TTG: 1.1	Ser TCG: 0.8	TAG: STOP	Trp TGG: 1.3
Leu CTT: 1.0	Pro CCT: 0.6	His CAT: 1.1	Arg CGT: 2.5
Leu CTC: 1.0	Pro CCC: 0.4	His CAC: 1.1	Arg CGC: 2.2
Leu CTA: 0.3	Pro CCA: 0.8	Gln CAA: 1.3	Arg CGA: 0.3
Leu CTG: 5.5	Pro CCG: 2.4	Gln CAG: 3.0	Arg CGG: 0.4
Ile ATT: 2.7	Thr ACT: 1.1	Asn AAT: 1.6	Ser AGT: 0.7
Ile ATC: 2.8	Thr ACC: 2.4	Asn AAC: 2.5	Ser AGC: 1.5
Ile ATA: 0.4	Thr ACA: 0.6	Lys AAA: 3.7	Arg AGA: 0.2
Met ATG: 2.7	Thr ACG: 1.2	Lys AAG: 1.2	Arg AGG: 0.1
Val GTT: 2.1	Ala GCT: 1.8	Asp GAT: 3.2	Gly GGT: 2.9
Val GTC: 1.4	Ala GCC: 2.3	Asp GAC: 2.3	Gly GGC: 3.1
Val GTA: 1.2	Ala GCA: 2.0	Glu GAA: 4.4	Gly GGA: 0.7
Val GTG: 2.5	Ala GCG: 3.3	Glu GAG: 2.0	Gly GGG: 0.9

Phe TTT: 1.6	Ser TCT: 1.3	Tyr TAT: 1.3	Cys TGT: 1.0
Phe TTC: 2.3	Ser TCC: 1.8	Tyr TAC: 1.9	Cys TGC: 1.5
Leu TTA: 0.5	Ser TCA: 0.9	TAA: STOP	TGA: STOP
Leu TTG: 1.1	Ser TCG: 0.4	TAG: STOP	Trp TGG: 1.4
Leu CTT: 0.1	Pro CCT: 1.6	His CAT: 0.9	Arg CGT: 0.5
Leu CTC: 2.0	Pro CCC: 2.0	His CAC: 1.4	Arg CGC: 1.1
Leu CTA: 0.6	Pro CCA: 1.4	Gln CAA: 1.1	Arg CGA: 0.5
Leu CTG: 4.3	Pro CCG: 0.6	Gln CAG: 3.4	Arg CGG: 0.4
Ile ATT: 1.5	Thr ACT: 1.3	Asn AAT: 1.7	Ser AGT: 1.0
Ile ATC: 2.4	Thr ACC: 2.3	Asn AAC: 2.3	Ser AGC: 1.9
Ile ATA: 0.6	Thr ACA: 1.4	Lys AAA: 2.2	Arg AGA: 1.0
Met ATG: 2.3	Thr ACG: 0.7	Lys AAG: 3.5	Arg AGG: 1.1
Val GTT: 1.0	Ala GCT: 2.0	Asp GAT: 2.2	Gly GGT: 1.1
Val GTC: 1.6	Ala GCC: 2.9	Asp GAC: 2.9	Gly GGC: 2.5
Val GTA: 0.6	Ala GCA: 1.4	Glu GAA: 2.7	Gly GGA: 1.7
Val GTG: 3.1	Ala GCG: 0.7	Glu GAG: 4.1	Gly GGG: 1.7

Figure 5.3 *Escherichia coli* (top) and human (bottom) codon usage tables. The frequencies of various codons are indicated in percentage. Highly preferred codons are shaded gray.

In addition, notice that the preferential use of certain synonymous codons is conserved over the course of evolution. The tables of neighboring species often reveal very similar codon usage, and to some extent, their comparison may be used to evaluate their degree of evolutionary relationship.

It has been observed that extreme (very high or very low) GC levels in the genomes of organisms have a significant effect on codon selection. In genomes that are GC-rich, codons ending in C or G are very strongly preferred, whereas the opposite is true for genomes that are AT-rich. Generally, even in species whose genomes have a nearly equal GC/AT ratio, the constraints imposed by the codon usage table result in a more accentuated bias for bases located in the third codon position.

Context-dependent codon biases

In cases in which several synonymous codons are able to code for the same amino acid, each organism expresses preferences that may be expressed as frequencies and listed in a codon usage table. Other effects may add to this scheme of preferences. When several codons of nearly equal probability are possible, the choice among them may be influenced by the neighboring codon; that is, by nucleotides located *immediately* upstream and/or downstream. In *E. coli*, it has been noticed that for the two lysine codons AAA and AAG, the former is more frequently encountered when the next codon starts with a G, and reciprocally, that AAG is preferred when there is a C immediately downstream.

These biases are collectively known as ‘context effects,’ some of the most marked of which are indicated below for the *Escherichia coli* genome:

AAA-G > AAG-G	AAG-C > AAA-C	Lysine
GAA-G > GAG-G	GAG-C > GAA-C	Glutamate
GGC-G > (GGG-G, GGA-G, GGT-G)	GGT-A > (GGT-C, GGT-T, GGT-G)	Glycine
TTT-G > TTC-G		Phénylalanine

5.3 The biological basis of codon bias

Codon utilization adapts to transfer RNA concentrations

It has been shown that the frequency of utilization of each codon in the yeast *Saccharomyces cerevisiae* and in the bacterium *E. coli* is directly proportional to the intracellular concentration of transfer RNA (tRNA) that decodes it. For example, *E. coli* has two transfer RNA isoacceptors for isoleucine, tRNA^{Ile}₁, which decodes the ATT and ATC codons (respective frequencies: 2.7 and 2.8 percent), and tRNA^{Ile}₂, which decodes the ATA codon (frequency: 0.4 percent). The intracellular tRNA^{Ile}₁/tRNA^{Ile}₂ ratio is 20/1, which is close to the ratio of the decoded codons (27 + 28)/4.

This adaptation optimizes the protein translation process by exactly adjusting the tRNA demand of the translation machinery to the amount available in the cell.

The most highly expressed genes are the fittest

Comparing the codon usage table of individual genes with the ‘canonic’ table based on the full genome reveals that the genes which most closely follow the canonic table are those most strongly expressed in the cell (ribosomal proteins,

translation factors, structural proteins, etc). The selection pressure on these genes is the greatest, resulting in stronger bias than for weakly expressed genes.

At each stage of the translation process, tRNA molecules in the cytoplasm diffuse into the active site of the ribosome, which ‘tests’ whether the codon-anticodon interaction is correct and ‘rejects’ inappropriate tRNA molecules. The ribosome ‘waits’ for the tRNA molecule that possesses the anticodon corresponding to the codon. This wait is proportional to the number of unfruitful tests that the ribosome must carry out before the appropriate tRNA molecule arrives, which depends on the relative abundance of the isoacceptor sought among the set of tRNA molecules:

$$\begin{aligned} &<\text{number of tRNA molecules tried}> \\ &= [\text{total tRNA molecules}] / [\text{tRNA molecules sought}] \end{aligned}$$

The most abundant glycine codons in *E. coli* are GGT and GGC. They are translated by tRNA^{Gly₃}, which accounts for 6.5% of all tRNA in the cell. Ribosomes must therefore carry out around 15 tRNA trials (1/0.065) when translating one of those codons. It takes the ribosome more than 300 attempts to translate a rare codon such as ATA, which codes for isoleucine, since the corresponding isoacceptor represents only 0.3% of the total tRNA. When ribosomes encounter a rare codon, there is a pause in translation. By utilizing only codons that correspond to the most abundant isoacceptors, translation of the most highly expressed genes is able to avoid delays that can slow cell growth.

The rarity of the CG dinucleotide in vertebrates affects the codon usage table

As described earlier (5.2), due to the effect of mutations, the CG dinucleotide, which is a methylation site, tends to disappear in vertebrates. As a result, codons that include the CG doublet are underrepresented in vertebrate codon usage tables. As may be seen in Figure 5.3, this phenomenon is particularly evident for proline (CCN), threonine, (CAN), and alanine (GCN), in all of which a G in the third position is strongly disfavored. This is also reflected in context effects; codons ending in C upstream from a codon that begins with G are systematically avoided.

5.4 Using statistical bias for prediction

The statistical biases described in the preceding sections may be used for purposes of prediction in answering questions of biological importance, such as the following: ‘Does this DNA region code for a protein?’ ‘Which is the coding

strand in this DNA sequence?’ ‘Which is the coding phase?’ ‘Does this sequence contain an error?’ ‘What are the boundaries between introns and exons?’ ‘Is this gene strongly expressed?’

In chapter 4, sequence characteristics were investigated according to **signals** determined by specific patterns. In this section, they will be pursued by identifying their **contents**; *i.e.*, by determining the statistical properties of their distribution as a function of the bases that constitute them.

The search for coding sequences

The most powerful method for determining whether a stretch of DNA is a coding region, and which relies on the fewest *a priori* hypotheses, consists in seeking period 3 irregularities in nucleotide distribution. Non-uniform codon usage within a gene or exon should be revealed by period 3 bias in the frequencies of occurrence of individual nucleotides. This method requires no prior knowledge of the codon usage table for the organism, or even of the genetic code. The frequency of occurrence of each base at positions $3i$, $3i + 1$, and $3i + 2$ are simply calculated and compared with the average frequency of occurrence in the sequence.

$$\Delta = \sum_{N=A,T,G,C} \sum_{3 \text{ phases}} |f_{N/\text{phase } i} - f_N|$$

The value of Δ is calculated within a window of around 100 nucleotides moved progressively along the sequence being studied. If the value of Δ is traced as a function of the position of the window, a profile is obtained whose peaks fairly well represent the coding regions.

This method is generally well adapted to seeking introns and exons in eukaryotic genomic sequences. Splice site consensus sequences are usually insufficient to accurately determine intron/exon boundaries, for which the content research method can be an effective complement.

Coding phase analysis

The above technique may be considerably refined by utilizing the codon usage table for the species from which a sequence derives, thereby allowing prediction of which of the three phases is coding. The principle is to compare the frequencies of triplets appearing in the three phases with the canonical triplet frequencies, then to determine which of the former are most similar to the latter.

Practically speaking, the frequency of occurrence f_1 of N codons in a window on phase 1 is compared with the individual frequencies of codons in the standard genetic code usage table:

$$f_1 = \prod_{i=1}^N f_{\text{codon } i}$$

By shifting first one and then two nucleotides, the calculation is repeated for phases 2 and 3, which gives two other frequencies, f_2 and f_3 . The probability of each phase being the coding phase is given by Bayes' formula:

$$p_1 = f_1 / (f_1 + f_2 + f_3)$$

$$p_2 = f_2 / (f_1 + f_2 + f_3)$$

$$p_3 = f_3 / (f_1 + f_2 + f_3)$$

By displacing the length N window along the sequence, it is possible to trace the profiles of probabilities p_1 , p_2 , and p_3 , whose peaks indicate the positions of genes that code for proteins with remarkable precision (see Figure 5.5).

This method is very sensitive and can be used with shorter windows (around 12 codons). It is extremely useful for detecting insertion and deletion sequencing errors, which can result in a reading frame shift and thus easily be detected by using the probability graph, p_1 , p_2 , and p_3 (Figure 5.4).

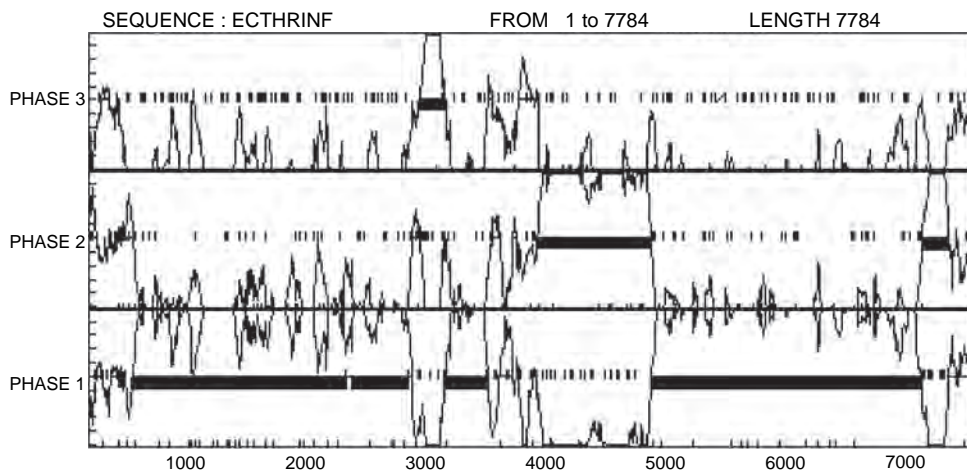


Figure 5.4 Coding probability profile obtained by analyzing genetic code usage. Thick lines correspond to actual coding regions. Short vertical lines in the phase graphs indicate the positions of stop codons.

5' nucleotide	3' nucleotide			
	A	C	G	T
A	0.102	0.054	0.071	0.074
C	0.077	0.057	0.010	0.069
G	0.059	0.046	0.054	0.048
T	0.062	0.057	0.072	0.087

Figure 5.5 Dinucleotide frequencies in the human genome.

Gene expression level

The expression level of a gene may be estimated by comparing its codon usage frequency with the standard codon frequencies given in the codon usage table for the species. The gene expression level may be quantified using a quantity known as the codon adaptation index (CAI), which is calculated as follows:

Each codon i contained in the gene is assigned a score w_i equal to the ratio of its frequency to the frequency of the most frequent codon that codes for the same amino acid. If codon i is the most frequent, then w_i equals 1. For a codon that is systematically avoided, w_i is close to 0. The codon adaptation index is the geometric mean of the w_i scores of the set L of the gene’s codons.

$$Index = \left(\prod_{i=1}^L w_i \right)^{\frac{1}{L}}$$

The CAI score obtained ranges between 0 and 1, increasing as the gene conforms to the standard utilization frequency of the species’ genetic code. For example, protein genes that are strongly expressed in the yeast, such as ribosomal proteins and histones, have scores of between 0.52 and 0.92, whereas regulatory protein genes, of which there are only a few copies per cell, have a score of 0.1.

The CAI may be used to estimate the expression level of a gene whose function is unknown. It is also useful when expressing a recombinant protein in a heterologous host, for example, a human protein in a bacterium. The CAI for a human gene, in combination with the bacterial codon usage table, allows us to predict whether a gene will be efficiently expressed, and may be used to guide the modification of certain codons in order to better adapt the gene to its new host.

5.5 Modeling DNA sequences

In his novel, ‘Jurassic Park’, the American author, Michael Crichton, tells the story of a molecular biologist, a certain Dr. Wu, who sequences the DNA of dinosaurs preserved in yellow amber in order to resuscitate the extinct animals.

To support his fiction, the author even went to the extent of devising a figure that supposedly represented a ~1.5 kbase fragment of dinosaur genome!

Is this dinosaur DNA sequencing credible? For the connoisseur of biomathematics, it does not withstand scientific scrutiny, hence it is just pseudoscience. Subjecting the fictional sequence to the statistical tests described above reveals that it does not present the biases expected for vertebrate DNA, such as the 40% G + C level and low CG dinucleotide frequency (see Section 5.2). But what *would* a novelist or biological faker have to do to concoct a pseudosequence capable of mystifying the shrewdest specialist?

Clearly, the solution would be to find a method that generated sequences displaying exactly the same statistical biases as real biological ones.

Markov chains

A very simple first approach is to attempt to reproduce both the nucleotide composition of the genome and its CG nucleotide content. Elementary statistical analysis of a body of vertebrate genomic sequences yields the following results for the relative frequencies of nucleotides:

$$f_A = 0.30, f_T = 0.29, f_C = 0.21, f_G = 0.20$$

The nucleotide frequencies are indicated in Figure 5.5, in which the frequency of the CG dinucleotide (in boldface) is observed to be lower than that of the others:

A very simple approach for generating a sequence that reproduces these frequencies involves iteration, by adding one nucleotide at a time. Suppose that a C has just been added at position i . Using the above table, it is easy to calculate the probabilities of encountering A, C, G, or T after that C. For example, the probability for an A to follow a C is:

$$p(A|C) = \frac{f_{CA}}{f_C} = \frac{f_{CA}}{f_{CA} + f_{CC} + f_{CG} + f_{CT}}$$

Equivalent probabilities may be symmetrically calculated for the other three nucleotides, C, G, and T. They are indicated in Figure 5.6 for the example of vertebrate sequences, expressed in percentages. Within rounding-off errors, line totals should equal 1 (100 percent).

Utilizing these probabilities in conjunction with a random number generator, it is possible to progressively generate a pseudosequence that mimics the biases observed in vertebrate sequences. This method of random fabrication using a probability table to generate state $i + 1$ from state i is called a **Markov chain**

5' nucleotide	3' nucleotide			
	A	C	G	T
A	34%	18%	24%	25%
C	36%	27%	5%	32%
G	28%	22%	26%	23%
T	22%	21%	26%	31%

Figure 5.6 Markov chain transition probabilities for human DNA sequences.

(or Markov **process**), well known in the field of applied mathematics. If state i is called e_i , the probability $p(e_{i+1}|e_i)$ that i will go to state e_{i+1} is given by the Markov process transition table, as indicated above. A simple Markov chain is a discrete random process that takes only the immediately preceding state into account in generating the current state.

Higher-order biases

The simple Markov chain just described is a perfect tool for reproducing DNA composition in terms of mono- and dinucleotides. Nevertheless, as seen above, higher-order irregularities exist in genomes, in particular, period 3 biases related to the translation of the genetic message by codons (*i.e.*, nucleotide triplets). To simulate these biases and produce ‘plausible’ sequences, we can use an extension of simple Markov chains. All that is required to produce state i is to utilize more complex probability tables that not only take state $i - 1$ into account, but also states $i - 2, i - 3 \dots i - k$. In this case, we speak of an ‘order k Markov process’. State e_i is then produced with probability $p(e_i|e_{i-1} \dots e_{i-k})$. With a Markov process of order $n - 1$, it is possible to produce pseudosequences that reproduce the frequencies of the k -tuples of nucleotides of a family of biological sequences for all k between 1 and n .

Using Markov processes to study DNA sequences

Up to this point, the objective has been to produce random pseudosequences that would fool a biologist, which might seem to be of quite trivial interest. Indeed, the utility of Markov chains lies elsewhere, since they are powerful probabilistic tools for testing hypotheses concerning the nature of DNA sequences. Using a Markov transition table makes possible *a posteriori* computation of the probability that a given sequence $e_1, e_2 \dots e_n$ could have been produced by the corresponding Markov process. This equals the product of the probabilities of the individual transitions:

$$\text{prob}(e_1 e_2 \dots e_n) = \prod_{1 \leq k \leq n} p(e_k | e_{k-1})$$

For example, in the above model, it is possible to calculate the probability of sequence CGCG and to compare it with probability of AATG. The first sequence, which is rich in CG, is thus shown to be much less probable than the second:

$$\begin{aligned} \text{prob (CGCG)} &= 0.05 \times 0.22 \times 0.05 = 5.5 \times 10^{-4} \\ \text{prob (AATG)} &= 0.34 \times 0.25 \times 0.26 = 2.2 \times 10^{-2} \end{aligned}$$

This kind of calculation is useful when comparing various hypotheses for the sequence of a given DNA segment, as long as it is possible to associate a Markov process with each hypothesis envisaged. The following is a concrete example illustrating this type of application.

A biologist has just determined some cDNA sequences obtained from messenger RNA derived from a mammalian cell culture. Even when all the required precautions are taken, cell cultures are sometimes contaminated by intracellular bacterial parasites known as mycoplasmas. The RNA extraction process does not permit separating the nucleic acids of the mammalian cells from those of the mycoplasmas. At the end of the experiment, some of the sequences obtained therefore could have come from the mycoplasmas rather than from the mammalian cells. How can the intrusive sequences be distinguished from the ones of interest?

Several mycoplasma genomes have already been completely sequenced, therefore it is possible to study their dinucleotide composition and to deduce a Markov process transition probability table for them, as was done above for the vertebrate sequences. The resulting values, compiled for *Mycoplasma genitalium*, are indicated in Figure 5.7.

Given a fragment of a sequence obtained by our biologist, it is possible to calculate the probability that it was produced by Markov processes, utilizing the probabilities derived from either the vertebrate or the mycoplasma genomes. Comparing these two probabilities, it is often also possible to decide which hypothesis is the most likely. Consider the following sequence as an example:

		3' nucleotide			
		A	C	G	T
5' nucleotide	A	42%	15%	17%	26%
	C	40%	18%	6%	36%
	G	31%	19%	18%	32%
	T	26%	14%	19%	42%

Figure 5.7 Markov chain transition probabilities for mycoplasma DNA sequences.

$S = 5' - \text{TTTCAAATAATCGTGAAATATCTTC} - 3'$

A *posteriori* calculation utilizing the two transition tables (vertebrate and mycoplasm) given above yields the following results:

$$p_{\text{vertebrate}}(S) = 4.3 \times 10^{-15}$$

$$p_{\text{mycoplasm}}(S) = 18.7 \times 10^{-15}$$

Despite the short length of this sequence (only 25 nucleotides), there is a rather clear difference, since the Markov process based on mycoplasm DNA produces a result that is four times more probable than the Markov process based on vertebrate DNA. This sequence may therefore be attributed to contamination with a mycoplasm bacterium. The differences are even clearer (by several orders of magnitude) in longer sequences, usually eliminating any ambiguity.

5.6 Complex models

When examining protein-coding regions in a given organism, it is preferable to utilize the codon usage table for the species, such as those shown in Figure 5.3 for the human and for *E. coli*. However, the simple Markov processes described above do not allow this to be done directly, since codon usage tables only consider triplets that are in phase with reading frames that code for proteins. Triplets that are shifted +1 or +2 nucleotides with respect to the reading frame are not taken into account, unlike in a simple Markov process.

For sequences that encode proteins, it is possible to compile triplet frequency tables for both the correct phase (phase 0), which corresponds to the classical usage table of the genetic code, and for the two other phases (phase +1 and phase +2). If this statistical approach is used for *E. coli*, the three tables shown in Figure 5.8 are obtained.

These three tables are rather complicated, but carefully examining them reveals certain interesting features: While some triplets, such as TAG and GGG, are systematically lacking in all three phases, others have very different frequencies, according to the phase considered. For example, whereas GAA is relatively abundant in the coding phase (4.05%), it is rare in phase +1 (only 0.36%). This observation calls for two important remarks:

- The simple Markov processes discussed above cannot capture the full complexity of biases imposed by using the genetic code within coding regions.
- To predict the coding phase, it must be possible to use biases that are not the same among the three phases.

phase 0	TTT	188	CTT	104	ATT	268	GTT	196
	TTC	194	CTC	108	ATC	276	GTC	141
	TTA	110	CTA	29	ATA	29	GTA	116
	TTG	107	CTG	553	ATG	257	GTG	250
	TCT	92	CCT	55	ACT	106	GCT	207
	TCC	88	CCC	50	ACC	278	GCC	257
	TCA	54	CCA	88	ACA	59	GCA	219
	TCG	78	CCG	262	ACG	133	GCG	337
	TAT	157	CAT	119	AAT	142	GAT	309
	TAC	131	CAC	111	AAC	220	GAC	226
	TAA	19	CAA	141	AAA	353	GAA	405
	TAG	0	CAG	309	AAG	87	GAG	185
	TGT	42	CGT	222	AGT	62	GGT	280
	TGC	58	CGC	223	AGC	153	GGC	323
	TGA	4	CGA	29	AGA	16	GGA	58
	TGG	145	CGG	42	AGG	11	GGG	99
phase 1	TTT	111	CTT	79	ATT	94	GTT	115
	TTC	151	CTC	87	ATC	188	GTC	145
	TTA	185	CTA	86	ATA	183	GTA	157
	TTG	310	CTG	210	ATG	261	GTG	188
	TCT	120	CCT	105	ACT	110	GCT	120
	TCC	139	CCC	111	ACC	148	GCC	146
	TCA	205	CCA	186	ACA	169	GCA	166
	TCG	253	CCG	271	ACG	262	GCG	325
	TAT	39	CAT	61	AAT	102	GAT	18
	TAC	90	CAC	111	AAC	228	GAC	28
	TAA	83	CAA	93	AAA	242	GAA	36
	TAG	71	CAG	157	AAG	346	GAG	24
	TGT	156	CGT	99	AGT	99	GGT	38
	TGC	303	CGC	278	AGC	183	GGC	111
	TGA	302	CGA	167	AGA	126	GGA	65
	TGG	406	CGG	266	AGG	174	GGG	83
phase 2	TTT	149	CTT	150	ATT	97	GTT	203
	TTC	102	CTC	90	ATC	48	GTC	71
	TTA	108	CTA	98	ATA	36	GTA	65
	TTG	40	CTG	118	ATG	38	GTG	52
	TCT	184	CCT	150	ACT	150	GCT	311
	TCC	105	CCC	100	ACC	93	GCC	158
	TCA	160	CCA	156	ACA	131	GCA	234
	TCG	121	CCG	138	ACG	83	GCG	173
	TAT	219	CAT	201	AAT	156	GAT	252
	TAC	141	CAC	173	AAC	120	GAC	143
	TAA	217	CAA	227	AAA	134	GAA	224
	TAG	32	CAG	125	AAG	44	GAG	41
	TGT	153	CGT	208	AGT	108	GGT	233
	TGC	289	CGC	281	AGC	173	GGC	277
	TGA	329	CGA	347	AGA	197	GGA	252
	TGG	406	CGG	266	AGG	174	GGG	83

Figure 5.8 Relative frequencies of various triplets in the three reading phases within *E. coli* genes (occurrences per 10,000 genes).

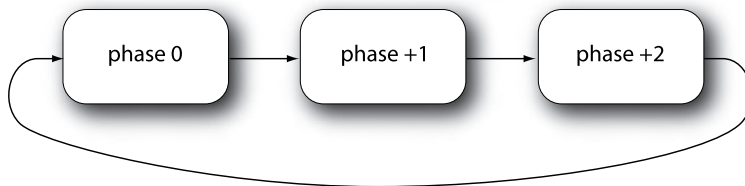


Figure 5.9 Phase shifting in a standard coding sequence.

It is nevertheless possible to devise a sequence-coding model that includes the notion of phase. This requires construction of a kind of Markov process that utilizes three probability tables deduced from the frequencies listed in Figure 5.8 in rotation. During each cycle, this process generates a nucleotide that shifts its reading phase one position, as per the diagram in Figure 5.9.

This is a modified order 2 Markov process, because knowledge of the two preceding nucleotides, e_{i-1} and e_{i-2} , is used to complete the nucleotide triplet by referring to probability tables: $p_0(e_i|e_{i-1}, e_{i-2})$, $p_{+1}(e_i|e_{i-1}, e_{i-2})$ and $p_{+2}(e_i|e_{i-1}, e_{i-2})$ for each of the three phases, 0, +1, and +2. This Markov process could be considered an automaton, permitting fabrication of very convincing *E. coli* coding pseudosequences. These pseudosequences also reproduce the preferential codon usage of this bacterium by utilizing the phase 0 probability table, as well as the context effects described above, which are captured by +1 phase probabilities for the 3' context and +2 phase probabilities for the 5' context. An example of a generated sequence is:

```
ATGCTATTCAGCTTCATCCTGACAAAACCGGGGACGGTAACGAGGCGCTG
ATCTATATC . . .
MetLeuPheSerPheIleLeuThrLysProGlyAspGlyAsnGluAlaLeuIleTyrIle
. . .
```

As for classical Markov processes, the main utility of this type of model obviously is not to produce pseudosequences, but rather to test *a posteriori* whether a biological sequence does indeed correspond to a coding phase. To do this, it is possible, as shown above, to calculate the probability that a given sequence had been produced by the modified Markov process:

$$\text{prob}(e_1 e_2 \dots e_n) = \prod_{2 \leq k \leq n} p_{k \bmod 3}(e_k | e_{k-1}, e_{k-2})$$

This probability evaluation alone does not provide much information, but it is interesting to compare it with other probabilities. As discussed above, a direct application is prediction of the species from which a sequence derives; for

example, determining whether a sequence has been contaminated by exogenous genetic material. To do this requires calculating and comparing the probabilities, using Markov processes adapted to various frequencies in the organisms studied.

Another interesting application concerns searching for the coding phase. Given a cDNA sequence presumed to code for a protein, is it possible to determine which is the coding phase? The following three probabilities may be calculated, each of which is shifted by one nucleotide with respect to the preceding one, and which correspond to the three phases, 0, +1, and +2:

$$prob(e_1e_2 \dots e_n); prob(e_2e_3 \dots e_{n+1}); prob(e_3e_4 \dots e_{n+2})$$

Their comparison usually allows unambiguous prediction of which of the three phases is coding. In the following example, the 25 first codons of a natural *E. coli* gene are:

ATGAAAGGCGGAAAACGAGTTCAAACGGCGCGCCCTAACCGTATCAATGG
CGAAATTCGCGCCCCAGGAAGTTTCG

In spite of the short length of this sequence, the results obtained using the modified Markov model described above are very clear: the correct phase (phase 0) has a probability between ten and one-hundred million times greater than the other two phases:

$$prob(\text{phase } 0) \approx 10^{-97}; prob(\text{phase } +1) \approx 10^{-105}; prob(\text{phase } +2) \approx 10^{-104}$$

This result is all the more remarkable in that it uses only statistical composition information and does not directly depend on the presence of start or stop codons, which, as seen in the preceding chapter, is necessary for pattern recognition. This approach is therefore very robust and may be applied within a fragment of any coding sequence. It can also be applied to very short open reading frames (a few dozen codons) to predict whether they are in fact real genes coding for small proteins or peptides.

5.7 Sequencing errors and hidden Markov models

Recognition of coding phases is a fundamental problem faced by biologists attempting to annotate genomes. Despite the technical progress that has been made in sequencing methods (see Chapter 1), errors inevitably slip into raw data. When this concerns the insertion or deletion of a nucleotide, the result is a shift in the DNA reading frame. The pattern recognition methods mentioned in the preceding chapter cannot be used to solve this type of problem, which results in possible failure to detect a gene in a genome.

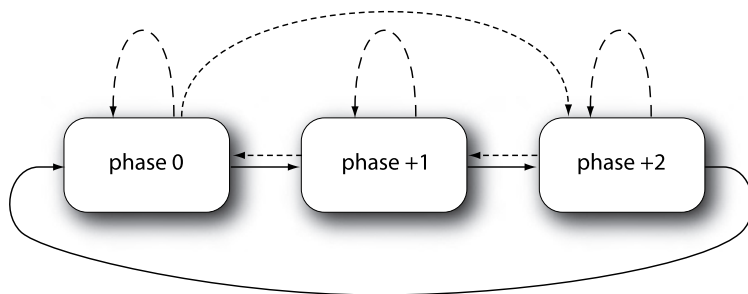


Figure 5.10 Schematic representation of the modified model, allowing phase jumps. Solid arrow shafts correspond to normal transitions and dashed arrow shafts to transitions associated with insertions or deletions. The diagram is simplified, each arrow in reality representing four transitions, each of which is associated with one of the four nucleotides, A, T, G, and C.

The sensitivity of the statistical approach provided by Markov processes in locating coding phases makes it possible to detect this type of error. In explaining how to proceed, the following section will again hypothesize using a Markov-type process to fabricate false genomic sequences containing errors.

Sequencing errors are relatively rare, with an incidence of around 1 insertion or deletion per 1,000 nucleotides in the raw data. The model employed in Figure 5.9 may be used, in which the phases are incrementally cycled, utilizing the associated probability table each time. The only modification that will be introduced is that the system will occasionally be allowed to ‘slip’; *i.e.*, jump to a phase other than the one expected. The new diagram of the process (Figure 5.10) presents frequent transitions (solid lines), which correspond to those of the correct model, as well as rare transitions, (dotted lines), which correspond to insertions/deletions.

This system is a probabilistic automaton consisting of 48 possible states. In fact, each transition depends on the two preceding nucleotides e_{i-1} and e_{i-2} , which represent 16 different dinucleotides. These 16 possibilities combined with the three phases amount to 48 states. The automaton goes from one state to another, as a function of its probability tables. At each transition, it also produces a nucleotide that is added to the end of the sequence being fabricated.

The objective now is to utilize this model for *a posteriori* calculation of the probability of a given sequence, as was done earlier for classical Markovian processes. However, this cannot be done, since if only the final sequence is available, it is impossible to reconstitute the route taken in Figure 5.10. In fact, unlike all the automata constructed earlier in the chapter, in this one, several different transitions can yield the same nucleotide at each step, according to whether a normal transition or an insertion/deletion (dotted arrow shafts) occurs. For this automaton, it all depends on whether the current state is in phase 0, +1, or +2. Such a model, in which the sequence produced does not allow reconstruction

of the progression of events of the automaton, is called a *hidden Markov process*, precisely because its states are unknown.

For hidden Markov processes, there is a very large number of possible routes, therefore a series of possible states yielding the same sequence. In the above example, given the three phases, 3^{n-2} routes exist for a sequence of length n . However, the probabilities associated with them may be very different. In particular, since transitions associated with phase jumps (insertions/deletions) have very low probabilities, the routes that take them are usually highly unlikely.

The problem at hand is to identify the coding phase and any shifts in it caused by sequencing errors. This information corresponds to the sequence of the states of the automaton. The aim therefore is to reconstitute the pathway shown in Figure 5.10.

According to which criteria should the route be selected among the 3^{n-2} possible? The overall probability of each route may be calculated by computing the individual probabilities associated with each transition as it proceeds. The route with the highest probability can then be selected in order to reconstruct the one with the most plausible sequence of states.

The Viterbi algorithm

There are obviously too many probabilities associated with each route to calculate them exhaustively. The Viterbi algorithm avoids this difficulty by using a dynamic programming approach. For a sequence $e_1e_2 \dots e_n$ of length n , it begins by calculating the probability of the best route from e_1 to e_k for k progressively varying between 1 and n and terminating in each of the possible states of the automaton. In the example of the search for coding phase, this amounts to seeking the probability of the best route over the k first nucleotides and terminating in either phase 0, phase +1, or phase +2. A table P of dimension $3 \times (n - 1)$ containing all these possibilities is drafted. The table is initialized at 1 in the three phases for the second nucleotide e_2 . By recurrence, it is then possible to calculate the k^{th} column of table P :

$$P(i, k) = \max_{j=0,1,2} (P(j, k-1)p_{j,i}(e_k|e_{k-1}, e_{k-2}))$$

Variable i and j represent the index of the phase and $p_{j,i}(e_i|e_{k-1}, e_{k-2})$ represents the probability of transition of the dinucleotide $e_{k-2}e_{k-1}$ toward the nucleotide e_k passing through phase j toward phase i . If $i = j + 1$ modulo 3, then it is a standard transition, corresponding to the solid-stem arrow in Figure 5.10, and the probability can be calculated from the tables in Figure 5.8. In the contrary case, the transition corresponds to an insertion or deletion (dotted-shaft arrow in Figure 5.10) and the probability is lower, on the order of the error level, estimated to be around 10^{-3} .

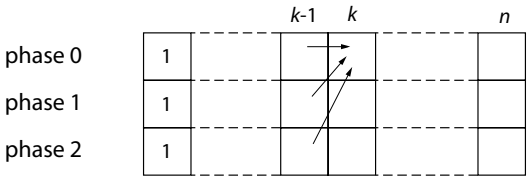


Figure 5.11 Completion of table P. For each cell, the probabilities of the three possible transitions are evaluated, then the maximal one is selected. The table is completed from left to right.

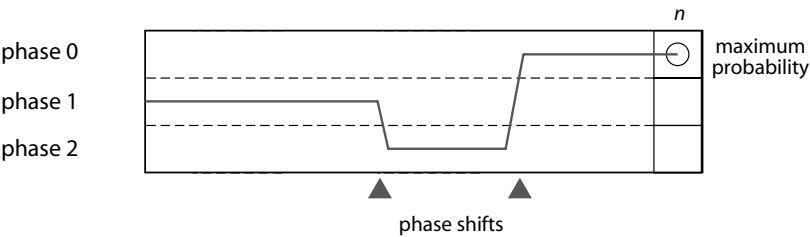


Figure 5.12 Reverse reconstitution of the path across table P, starting from the final maximum probability score. In this diagram only the route of the first nucleotide of each codon is drawn; this eliminates the normal cycling across the three phases that would make the figure illegible.

When the table is complete, the last (n^{th}) column indicates which of the three phases has the highest probability. The inverse route is then taken in the table in order to determine the path it followed. For each column k , this amounts to finding the value of j in the above equation corresponding to the maximum of the three terms. This value of j indicates the most probable state of the hidden Markov process at the level of the $k - 1^{\text{th}}$ nucleotide. The list of the most probable states as a function of k yields a prediction of the correct coding phase (Figure 5.12).

In practice, it is not these probabilities that are calculated directly, but their logarithms, which has two advantages: (i) calculation of a *product* is replaced by calculation of a *sum*, which is faster, since all that is necessary is to add the logs of the transition probabilities $\log(p_{j,i}(e_i|e_{k-1},e_{k-2}))$; (ii) using logarithms avoids the risk of exceeding the capacity of the machine (underflow), since the overall probability of long sequences rapidly becomes ridiculously small.

Predictions carried out using the Viterbi algorithm are often very accurate and of much higher quality than those produced using profile score methods, such as those presented in Figure 5.4. In the following example, which uses the nucleotide sequence of an *E. coli* gene, two errors have been introduced intentionally: one deletion and one insertion. This kind of double error can be espe-

```

ATTAAAGGCGGAAAACGAGTTCAAACGGCGCGCCCTAACCGTATCAATGG
CGAAATTCGCGCCCCAGGAAGTTCGCTTAACAGGTCTGGAAGGCGAGCAG
CTTGGTATTGTGAGTCTGAGAGAAGCTCTGGAGAAAGCAGAAGAAGCCGG
AGTAGACTTAGTCGAGATCAGCCTAACGCCGAGCCGCCGTTTGTCTGTAT
AATGGATTACGGCAAATTCCTCTATGAAAAGAGCAAGTCTTCTAAGGAAC

```

Figure 5.13 Sequence of a fragment of the *Escherichia coli* infC gene with two phase shifts; first a deletion, then an insertion. The gray-highlighted area indicates the zone between the two errors. The Viterbi algorithm phase shift prediction is underlined. Only the first nucleotide of the shifted zone is incorrectly predicted.

cially difficult to detect, since the effects on the phase mutually compensate for each other such that the length of the reading frame is not affected. Nevertheless, the coding sequence between the two errors is profoundly affected. However, the Viterbi algorithm nearly perfectly detects the two ill-timed changes in the sequence (Figure 5.13).

5.8 Hidden Markov processes: a general sequence analysis tool

The phase shift search discussed above is just one relatively simple example developed to illustrate the material treated in this chapter. However, the range of Markov model applications in the domain of biological sequence analysis is much greater. Refining this type of model and having it take into account frequencies in non-coding regions and in start and stop codons can convert it into an automatic genome gene-seeking tool. In eukaryote genomes, in which coding regions are interrupted by introns, statistical analyses have been conducted in both intron and exon regions and the frequencies obtained used to construct hidden Markov processes. The most probable route reconstituted using the Viterbi algorithm may then be utilized to predict spliced regions in messenger RNA. Finally, hidden Markov processes have also been used with protein sequences to predict secondary structures (α -coils and β -sheets; see Chapter 6).

5.9 The search for genes – a difficult art

The exhaustive search for all the genes contained in a large genome is a complex task. In higher eukaryotes, it is complicated by the ‘dilution’ of relevant information in non-coding sequences of repetitive DNA and in intergene regions. The

presence of lengthy introns separating short exons renders precise assembly of coding zones rather delicate. In such case, none of the methods discussed until now is by itself sufficient to identify genes. In general, several different approaches are necessary to achieve a reliable result. Currently used approaches are based on the following strategy:

1. Predict exon and intron regions, using a statistical method, most often a hidden Markov process.
2. Search for consensus patterns of donor and acceptor splicing sites, using a finite automaton method.
3. Combine the two types of information in order to precisely predict intron and exon borders.
4. Assemble the predicted exons and compare the sequence obtained with cDNA and EST sequence databases of the same organism (see Chapter 1). All or part of the predicted sequence may be found in one of these sequences, which derive from messenger RNA.
5. If this search fails, it is still possible to compare the protein sequences predicted from the genomic DNA (translated into all possible phases) with protein sequence databases. If the genomic DNA region being considered effectively codes for a protein that possesses a homolog identified in another species, a BLAST-type search based on translated sequence fragments may be carried out to identify these homologies. This would permit identifying or confirming some coding parts, thus some intronic parts of the gene.

Several very elaborate tools that combine several of these approaches are accessible on the web, for example, GenScan (<http://genes.mid.edu/GENSCAN.html>), which has been utilized to analyze and document the entire human genome (accessible on <http://www.ensembl.org>).

Bibliography

- Burge C., *et al.* (1992). Over- and under-representation of short oligonucleotides in DNA sequences. *Proc Natl Acad Sci USA* 89: 1358–1362.
- Burge C., Karlin S. (1997). Prediction of complete gene structures in human genomic DNA. *J Mol Biol* 268: 78–94.
- Burge C.B., Karlin S. (1998). Finding the genes in genomic DNA. *Curr Opin Struct Biol* 8: 346–354.
- Durbin R., *et al.* (1998). Biological sequence analysis. In *Probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, UK.

- Eddy S.R. (2004). What is a hidden Markov model? *Nat Biotechnol* **22**: 1315–1316.
- Gautier C. (2000). Compositional bias in DNA. *Curr Opin Genet Dev* **10**: 656–661.
- Lukashin A.V., Borodovsky M. (1998). GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res* **26**: 1107–1115.
- Zhang M.Q. (2002). Computational prediction of eukaryotic protein-coding genes. *Nat Rev Genet* **3**: 698–709.

6

Structure prediction

6.1 The structure of RNA

As the material basis of genes and the vector of heredity, DNA has for many years occupied a preeminent place in the family of nucleic acids. In 1953, Watson and Crick predicted its double-helical structure, now recognized as practically universal, one of the essential advances in modern molecular biology.

Compared with DNA, RNA seemed to be a poor relative, and was relegated to secondary roles. Three main families of RNA molecules could be identified:

- Messenger RNA (mRNA) molecules, considered to be ephemeral copies of DNA used in protein translation;
- Ribosomal RNA molecules, long reduced to the role of internal ribosome scaffolding, with no real function, since ribosomal functions were attributed to their protein components;
- Transfer RNA (tRNA) molecules, considered to be little more than molecular adapters for ferrying amino acids to the ribosome.

The rehabilitation of RNA began in the early 1980s, when the introns of some mRNA molecules were found to be capable of excision in the absence of proteins. Other RNA molecules were later found to be endowed with catalytic properties; for example, ribonuclease P, an enzyme responsible for the maturation of tRNA. This amounted to a conceptual revolution leading to the recognition of these RNA enzymes, which came to be known as *ribozymes*.

There is now good reason to believe that RNA is responsible for polymerase activity in the ribosome, and that the proteins that adorn RNA molecules are not there just to stabilize and/or enhance that activity. This has recently been confirmed by resolution of the three-dimensional structure of the ribosome, revealing the active site of the ribonucleoproteic assembly to consist exclusively of RNA.

Thus, one of the essential reactions in all living cells, the translation of genetic message into protein, is carried out mainly by RNA molecules, not proteins. RNA not only carries genetic information and is replicated like DNA, but also catalyzes the reactions necessary for life. In 1986, the versatility of RNA led Francis Crick, co-discoverer of the double-helical structure of DNA, to propose that life itself developed via an ancestral organism that used only RNA to carry out its functions. This hypothesis, known as the ‘RNA world,’ obviates the chicken-or-egg dilemma of whether proteins or DNA came first during evolution.

Our knowledge of the family of RNA molecules has also considerably expanded. We now know of RNA molecules that play important, if not essential roles in the control of plasmid replication, in the regulation of genetic expression, in eukaryote intron splicing machinery (the ‘spliceosome’), in the manufacture of telomeres (special structures found at chromosome ends), and in numerous processes in RNA viruses, retroviruses, for example.

This chapter covers how RNA acquires the complex and varied structures that allow it to exercise these diverse functions. It may even be stated that ‘unstructured linear RNA does not exist *in vivo*,’ since under physiological conditions, practically all RNA, including mRNA, is more or less elaborately folded.

Predictive methods exist today that permit *a priori* calculation of the secondary structure of RNA, starting from its primary sequence. It is possible to validate and clarify these theoretical structures by phylogenetic comparison and biochemical experimentation, which in certain cases has led to the development of three-dimensional models of the folding of some very complex RNA molecules.

6.2 Properties of the RNA molecule

In the cell, RNA may be distinguished from DNA by the presence of a hydroxyl group at the 2′-position of the ribose molecule (Figure 6.1) and by the fact that RNA is mostly present as a single strand. The thymine residues in DNA are replaced by uracil in RNA (Figure 6.2).

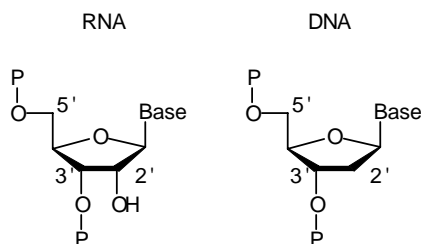


Figure 6.1 RNA has an additional hydroxyl group on the ribose 2′-carbon atom.

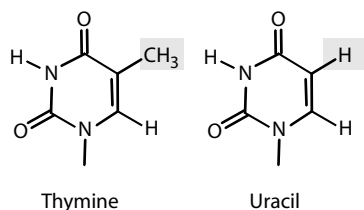
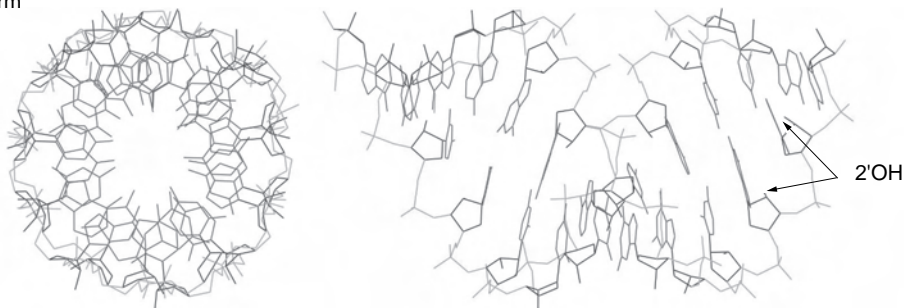


Figure 6.2 Uracil (U) is the thymine analog in RNA. It has the same pairing capacity as thymine, but lacks the methyl group at position 5.

A Form



B Form

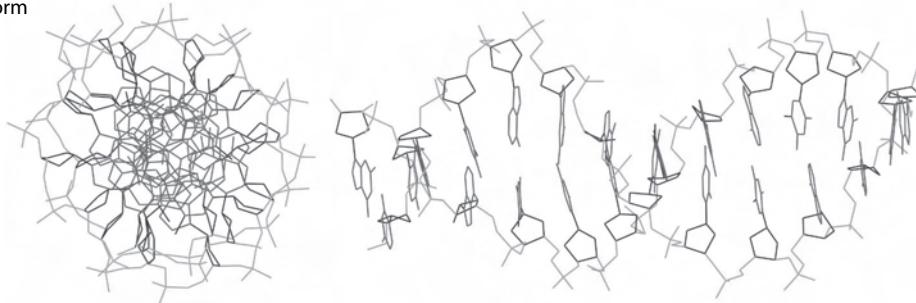


Figure 6.3 Structures of the A and B types of double helices formed by polynucleotides.

Like DNA, RNA can form Watson–Crick-type nucleotide pairs (A:U and G:C) and locally fold into a double-helix, either with another RNA strand or by forming a heteroduplex with a DNA strand. However, the presence of the ribose 2'-OH group imposes additional steric constraints and usually prevents formation of a B-type DNA, the major DNA conformation. RNA thus forms an A-type helix, which is more open and includes 11 nucleotides per turn instead of 10. Figure 6.3 displays the differences between the A and B helical forms. In A-form RNA, the large groove is appreciably deeper and very narrow, whereas

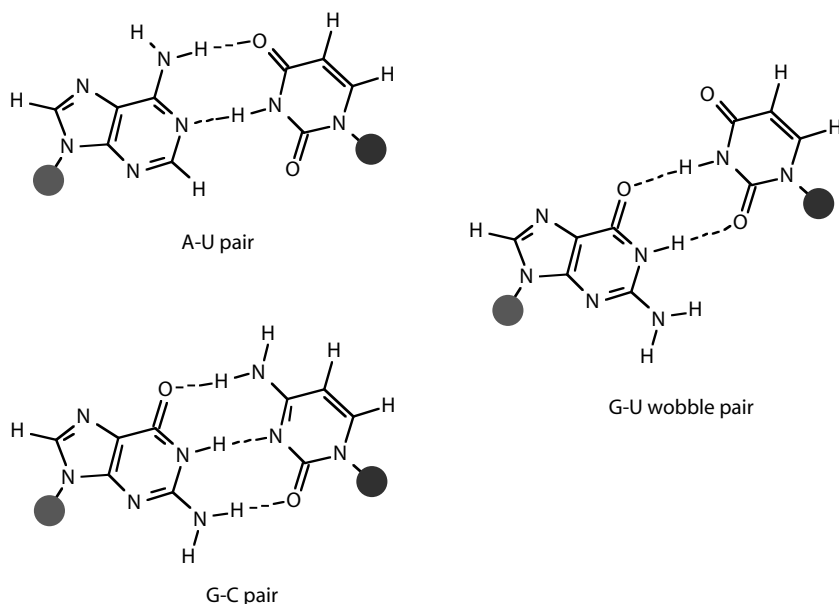


Figure 6.4 Watson-Crick A-U, G-C, and 'wobble' pairings in RNA structures.

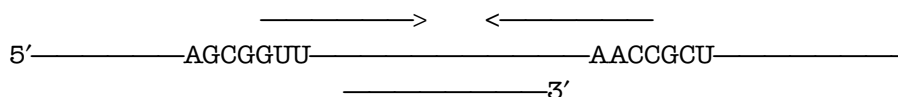
the small groove is much shallower. Finally, the basepair stacks are inclined and displaced with respect to the axis of the helix.

Finally, RNA forms non-canonical pairs that are different from standard Watson-Crick pairs (A:T and G:C), among which the most frequently observed are G:U, known as 'wobble pairs.' Compared with Watson-Crick pairs, wobble pairs require displacement of the pyrimidine nucleotide (uracil) toward the major groove, which deforms the ribose-phosphate backbone (Figure 6.4).

6.3 Secondary RNA structures

Classical structures

Given a segment of an RNA sequence, for example, 5'-AGCGGUU-3', it is simple to deduce the complementary sequence 5'-AACGCU-3' by reversing the order of the nucleotides and replacing each with its complement. When an RNA sequence has both a sequence and its complement on the same strand, it is said to contain an *inverted repeat*.



When a single-strand RNA molecule contains an inverted repeat, it can fold onto itself locally, forming small double-helical segments called ‘stems’, separated by single-stranded regions known as ‘loops.’ Four types of loops may be identified, according to the local folding topology: bulges, terminal loops or ‘hairpins’, internal loops, and branched or multiple loops (Figure 6.5).

This planar representation of pairing topology and helix formation in an RNA molecule is called its *secondary structure*. Determining this topology is the first step in the three-dimensional modeling of helix layout, known as *tertiary structure*.

The curved lines connecting the tandem reverse sequences in the diagram at the top of Figure 6.5 do not cross each other. Such pairings are said to be *classical*, since there is no entanglement of the strands of two helical regions. This non-entanglement constraint is analogous to that used for writing parenthetical

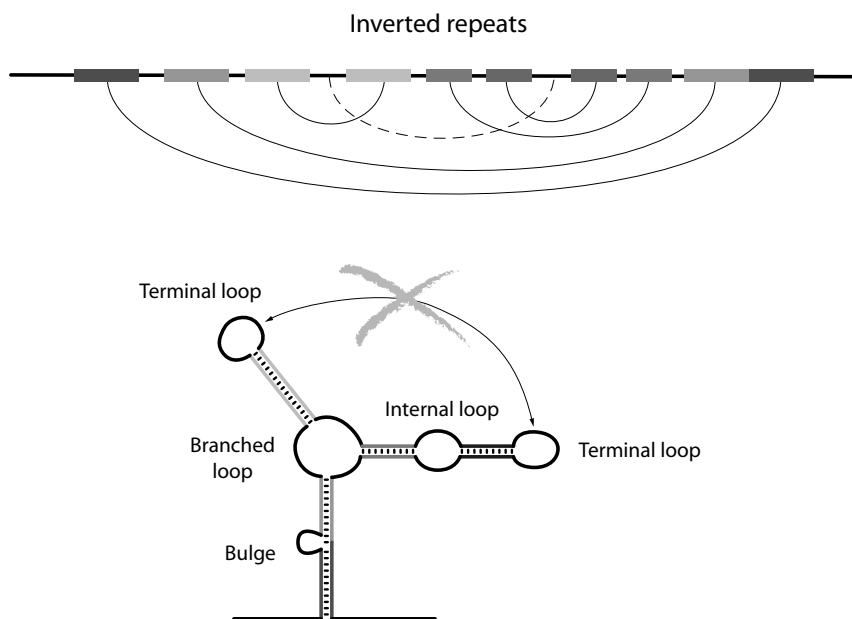


Figure 6.5 Example of secondary structures formed by the pairing of complementary sequences in an RNA molecule, showing the various types of loops encountered. ‘Classical’ structures do not have the long-distance interactions among loops that correspond to the complementarity indicated in dashed lines in the upper part of the figure.

expressions in mathematics. If an opening parenthesis is associated with the first strand of a paired region in a classical secondary structure and a closing parenthesis is associated with the second, the syntax of the expression obtained is correct. For example, in the structure represented in Figure 6.5, the parenthetical expression associated with the diagram is

$$([() [()]])$$

We arbitrarily chose to alternate parentheses and brackets in order to make the expression more understandable, but whatever the symbol (parentheses, brackets, braces, etc), the expression remains correct as long as two of the same symbols are attributed to each strand of a paired region. If non-classical pairing occurs between two terminal loops (indicated by the dashed line in Figure 6.5) the associated parenthetical expression becomes

$$([([) [()]])$$

which clearly is incorrect.

The reason for making this rapprochement between parenthetical expressions and secondary RNA structure (which at first sight may appear somewhat artificial) will become apparent later. The methods used to analyze and predict the classical secondary structure of RNA derive from tools developed by computer scientists for analyzing the syntax of mathematical expressions and the structures of computer programs.

True knots and pseudoknots

The preceding paragraph described classical secondary RNA structures and the limitation they impose on long-distance interactions. However, there is no *biological* reason for RNA folding to be limited to classical folds. In fact, there even exist known and well-described secondary RNA structures called **pseudoknots** that do not obey this rule. Pseudoknots can occur in a simple structure consisting of a stem and a terminal loop when part of the loop sequence is complementary to a region situated outside the stem. A second helical region then forms, often as a prolongation of the first region, in which the ribose-phosphate backbone takes on an S-shape (Figures 6.6 and 6.7).

They are called pseudoknots because the RNA strand does not really form a knot in the topological sense, which would be physically unrealistic and biologically catastrophic. This imposes a major topological constraint on these pseudoknot structures:

The non-canonical paired region of a pseudoknot cannot exceed 9 or 10 basepairs

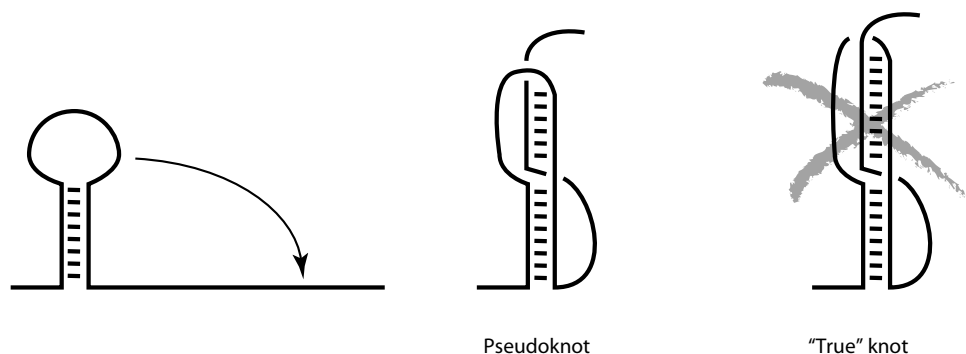


Figure 6.6 Formation of a pseudoknot by a loop pairing with a region outside its stem.



Figure 6.7 Three-dimensional structure of a pseudoknot present at the 3'-end of the genomic RNA of the tobacco mosaic virus.

This constraint is the result of the helical character of the stem that is formed. The helices formed by the RNA contain 10 or 11 basepairs per turn. Starting with this length, the full-turn formed by the rolling-up of the two strands transforms the pseudoknot into a real knot (*cf.* Figure 6.6).

Strategy for analyzing and predicting RNA secondary structures

This constraint on the length of pseudoknots is an advantage for the bioinformatician, since it considerably limits the number and extent of pseudoknots in real structures, greatly simplifying analysis. In fact, the only efficient algorithmic tools that exist are those used for seeking classical secondary structures without pseudoknots.

After identifying the most probable classical secondary structure heuristically, the search continues for possible pseudoknots that the RNA molecule might

contain. This approach yields satisfactory results; since pseudoknots are short, they usually do not significantly contribute in a definitive manner to overall secondary structure stability.

6.4 Thermodynamic stability of RNA structures

A priori, there are numerous ways an RNA molecule of known sequence and length can fold into stem and loop structures. One would expect these *in vivo* structures to correspond to the lowest energy conformations. A method is needed that can be used to develop approaches for predicting the relative stability of these various structures for quantitative evaluation.

The ‘nearest neighbors’ hypothesis

The factors that play a role in stabilizing nucleic acids in duplex form are known. There are three principal contributions: First, hydrogen bond formation between the two bases of a pair; then Van der Waals interactions between consecutive layers of helically stacked basepairs; and finally, the ‘hydrophobic effect,’ which also promotes the stacking of basepairs by protecting their hydrophobic faces from the aqueous environment.

In a classical helix structure, these three interactions take only the nature of the basepair and its immediate neighbors (the basepairs before and after it) into consideration. The absence of long-distance effects suggests that it could be possible to calculate the energy associated with the formation of a helical region (pairing ΔG^0), by ‘cutting elementary slices’ of a basepair and summing the energy contributions of each slice:

$$\Delta G_{\text{helix}}^0 \approx \sum \Delta G_{\text{basepairs}}^0$$

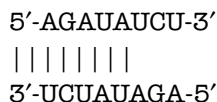
Each ΔG^0 depends on the nature of the basepair (A-U or G-C), as well as on that of its nearest neighbors, the basepairs situated immediately before and after it in the helix. Empirically, it turns out that this simplification, while rather drastic, allows accurate prediction ($\leq 10\%$) of the stability of RNA helices (without loops).

Empirical measurement of parameters

The thermodynamic parameters associated with basepair formation in a helical structure must be available in order to implement this simplified method. These parameters are based on sets of experimental measurements carried out on short

synthetic oligoribonucleotides. For example, by placing a solution of the octameric palindrome sequence 5'-AGAUAUCU-3' in the cuvette of a spectrophotometer, it is possible to follow the UV absorption of the RNA bases as a function of temperature.

At low temperature, the oligonucleotide self-pairs, forming a duplex:



Heating melts the Watson–Crick pairs, separating the two stands. In the duplex form, the bases are stacked on top of each other, which places them in a hydrophobic environment in which their UV absorption is lower than for the single-strand form, in which the bases are exposed to an aqueous environment¹.

Analysis of the melting curves obtained with various concentrations of RNA strands provides access to the thermodynamic equilibrium parameters:



In particular, it is possible to calculate variation in the values of the enthalpy parameter ΔG^0 , ΔH^0 , and ΔS^0 associated with duplex formation². For example, for the octamer AGAUAUCU mentioned above, ΔG^0 was found to be -6.58 kcal/mol at 37°C by experimentation.

Systematic analysis of a large number of oligonucleotides permits determination of all the incremental ΔG^0 corresponding to various stacking combina-

¹ This property of paired nucleic acids is called *hypochromicity*.

² The following relations are utilized:

$$\Delta G^0 = -RT \log K = \Delta H^0 - T \cdot \Delta S^0 \text{ where } K = [\text{duplex}]/[\text{single-strand}]^2$$

Utilizing the fact that at the melting temperature T_m , $2[\text{duplex}] = [\text{single strand}]$, we obtain

$$RT_m \log(2[\text{single-strand}]) = \Delta H^0 - T_m \Delta S^0.$$

Finally, the conservation of the total quantity of RNA strands is written

$$2[\text{duplex}] + [\text{single strand}] = [\text{RNA}]_{\text{total}}$$

from which the following relation is extracted:

$$T_m = \frac{\Delta H^0}{\Delta S^0 + R \log[\text{RNA}]_{\text{total}}}$$

Studying the variation in the melting temperature T_m (in $^\circ\text{Kelvin}$) as a function of the RNA concentration in the spectrophotometer cuvette, it is easy to extract the values of ΔH^0 and ΔS^0 . These values allow calculation of the free enthalpy associated with pair formation at any desired temperature. For RNA folding, the value at the physiological temperature of 37°C is used: $\Delta G_{37^\circ\text{C}}^0$.

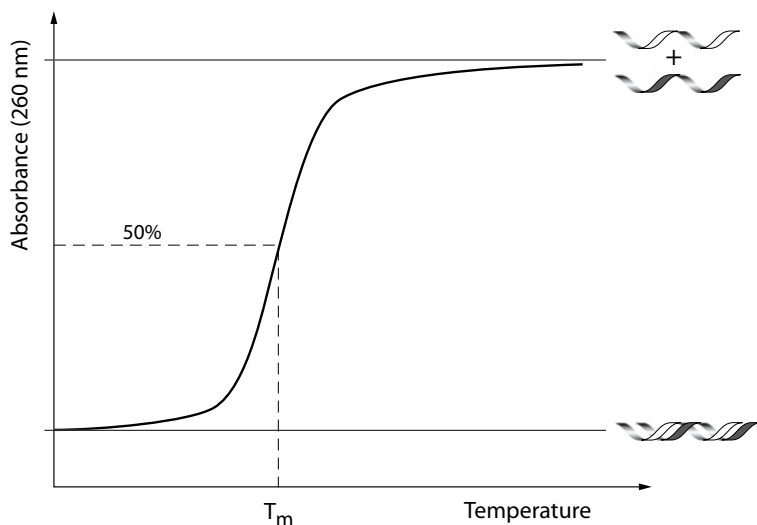


Figure 6.8 Fusion of a duplex RNA molecule as a function of its UV absorbance. The fusion temperature T_{fus} corresponds to the temperature at which half of the RNA strands are in duplex form.

tions of a new basepair onto an already formed basepair. Taking into account G-U pairs, which are rather frequent in RNA, there are three types of pairs: G-C, A-U, and G-U, each with two orientations. The set of incremental ΔG^0 at 37°C is presented in Figure 6.10.

Systematically repeating this operation with a great number of oligonucleotides makes it possible to calculate the individual contributions of each type of basepair stacking. For example, by comparing the ΔG^0 associated with the formation of a duplex of n nucleotides and the ΔG^0 associated with the formation of a duplex of $n + 1$ nucleotides containing the n first basepairs of the former (see Figure 6.9), it is possible to deduce the incremental ΔG^0 obtained by adding the supplementary basepair.

This is known as the Freier–Turner Rules table, after the names of its authors, and merits comment:

Several other authors have identified sets of thermodynamic parameters for use in evaluating the stability of RNA structures, finding the same qualitative tendencies but obtaining slightly different values. The Freier–Turner rules are more recent, and were calculated for the folding of RNA at 37°C, whereas those of other authors were calculated at different temperatures (e.g., 25°C). The Freier–Turner rules are also more accurate, since the determinations are based on a more complete set of experimental data.

Note that the table is asymmetric. For example, the contributions of G-C followed by C-G, and of C-G followed by G-C (from 5' toward 3') are different

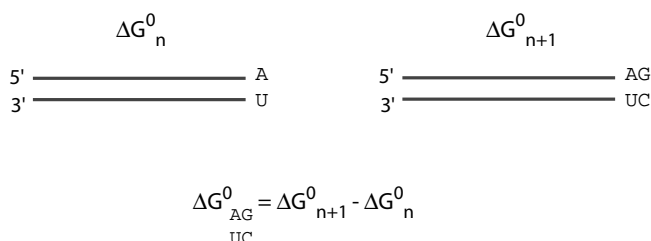


Figure 6.9 Incremental determination of ΔG^0 obtained by stacking a G-C basepair at the 3'-end of an A-U basepair.

		5' basepair					
3' basepair	GU	GU	UG	AU	UA	CG	GC
	GU	-0.5	-0.6	-0.5	-0.7	-1.5	-1.3
	UG	-0.5	-0.5	-0.7	-0.5	-1.5	-1.9
	AU	-0.5	-0.7	-0.9	-1.1	-1.8	-2.3
	UA	-0.7	-0.5	-0.9	-0.9	-1.7	-2.1
	CG	-1.9	-1.3	-2.1	-2.3	-2.9	-3.4
	GC	-1.5	-1.5	-1.7	-1.8	-2.0	-2.9

Figure 6.10 Helices: table of incremental $\Delta G_{37^\circ\text{C}}^0$ in kcal/mol (Freier-Turner rules).

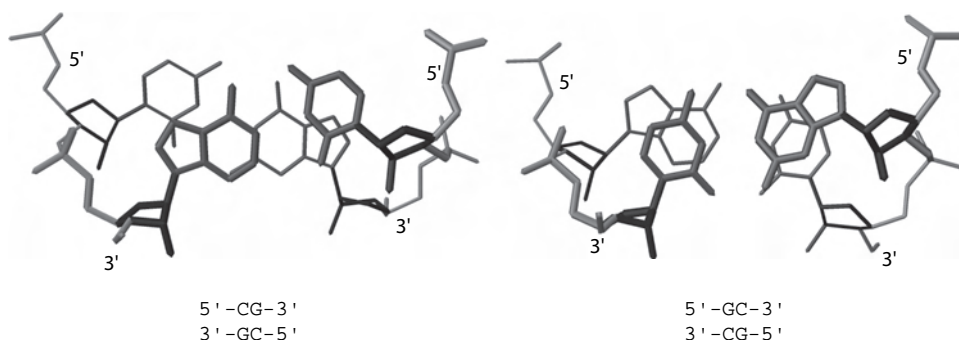


Figure 6.11 Pyrimidine-purine (left) and purine-pyrimidine (right) nucleotide stacking interactions.

(-3.4 and -2.0 kcal/mol, respectively). This may be explained by the difference in stacking interactions between the two types of duplex, which are much stronger for the former than for the latter. Indeed, coverage is better when a 5'-pyrimidine is stacked onto a 3'-purine than for the reverse, as may be verified in Figure 6.11.

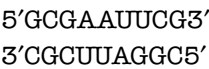
To obtain the total ΔG^0 of the pairing of a duplex RNA molecule, the contributions of all the nucleotides it contains plus the nucleation ΔG^0 (corresponding to the formation of the first pair) are summed. This nucleation term has also been determined by thermodynamic study of short duplexes:

$$\Delta G^0 = +3.4 \text{ kcal/mol}$$

which is positive, since it corresponds mainly to entropic loss during association of the two RNA strands.

Example of calculation

For the formation of the following duplex:



contributions are summed:

$$\begin{aligned} &\Delta G^0_{\text{nuc}} \Delta G^0_{\text{GC/CG}} + \Delta G^0_{\text{CG/GC}} + \Delta G^0_{\text{GC/AU}} + \Delta G^0_{\text{AU/AU}} \\ &+ \Delta G^0_{\text{AU/UA}} + \Delta G^0_{\text{UA/UG}} + \Delta G^0_{\text{UG/CG}} + \Delta G^0_{\text{CG/GC}} \end{aligned}$$

giving:

$$+3.4 - 3.4 - 2.0 - 2.3 - 0.9 - 0.9 - 0.5 - 1.3 - 2.0 = -9.9 \text{ kcal/mol}$$

Freier–Turner rules are relatively dependable for helical regions, allowing prediction of the ΔG^0 with around 5 to 7 percent accuracy.

Loops

The same kind of thermodynamic analysis of thermal denaturation curves may be carried out on oligonucleotides that include loops in order to try to quantify the destabilizing contributions they make to the overall structure. The values obtained by Turner and Freier at 37°C for ΔG^0 associated with the loops are indicated in Figure 6.12.

Length	Loop size																
	1	2	3	4	5	6	7	8	9	10	12	14	16	18	20	25	30
Bulges	3.3	5.2	6.0	6.7	7.4	8.2	9.1	10.0	10.5	11.0	11.8	12.5	13.0	13.6	14.0	15.0	15.8
Terminal loops	∞	∞	7.4	5.9	4.4	4.3	4.1	4.1	4.2	4.3	4.9	5.6	6.1	6.7	7.1	8.1	8.9
Internal loops	—	0.8	1.3	1.7	2.1	2.5	2.6	2.8	3.1	3.6	4.4	5.1	5.6	6.2	6.6	7.6	8.4

Figure 6.12 Loops: table of incremental $\Delta G^0_{37^\circ\text{C}}$ in kcal/mol.

Bulges are considered not to interrupt the continuity of the helix. The basepair situated immediately before a bulge is stacked onto the one immediately after it. Both its stabilizing contribution and that of the bulge are taken into account. Since there are not enough experimental data for multiple loops, they are usually treated as though they were internal loops.

Freier–Turner rules also take into account certain terminal mispairings in loops, as well as the fact that unpaired 5′ and 3′ nucleotides in helices can be stacked onto the last basepair, stabilizing the structure. (These are special cases that will not be developed here.)

Hyperstable tetraloops

Contributions to the empirical free energy associated with loops are in general much less accurate than those associated with the formation of paired regions. This low accuracy results mainly from a lack of data on loop conformation. Certain terminal loop sequences appear to be able to assume considerably more stable conformations than predicted by Freier–Turner or other empirical rules. Four-base terminal loops (*tetraloops*) that include the sequences GNRA, UNCG, and CUYG (Y = [C or U], R = [A or G], and N = A, U, G, or C]) have been found to be exceptionally stable. Structural studies using NMR and crystallography have revealed that the stability of GNRA and UNCG tetraloops is due to a particular nucleotide conformation that allows numerous base-base and base-phosphate interactions (Figure 6.13). Such loops are found in the ribosomal RNA of numerous species, in the genomic and mRNA of some viruses, and in certain prokaryote transcription terminators.

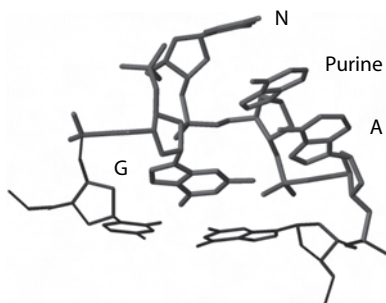


Figure 6.13 Structure of a hyperstable GNRA tetraloop. The G and A form a non-canonical pair called a ‘sheared G-A’. The two other bases are stacked onto the A.

The systematic search for motifs corresponding to these hyperstable loops between two inverse repeat sequences is a very efficient method of detecting of such stem and loop structures.

6.5 Finding the most stable structure

These empirical thermodynamic rules may be used to calculate the ΔG^0 associated with the formation of any RNA secondary structure. Finding the most stable structure; that is, the one with the lowest ΔG^0 , simply requires calculating the ΔG^0 of every possible structure. The obvious problem is that the number of different ways to fold an RNA molecule of length N becomes very great as N increases.

As discussed above, there is a parallel between classical secondary nucleotide structures and parenthetical expressions in mathematics, and this parallel is strict for the topology of secondary structures that alternate between helices and loops. It is thus possible to ascertain that there are 14 ways to write an expression consisting of four pairs of parentheses, and that 14 different topologies exist for the secondary RNA structures of an RNA molecule that includes four helical paired regions (cf. Figure 6.14).

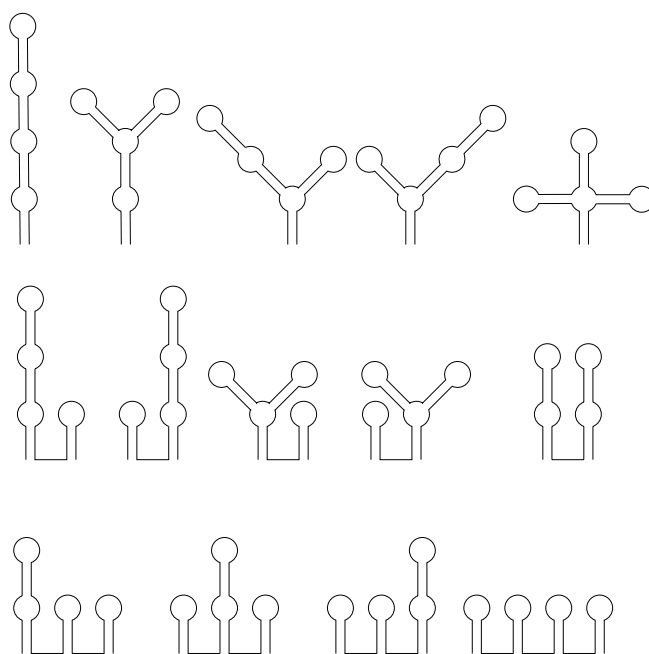


Figure 6.14 The 14 different possible topologies for an RNA molecule containing four helices.

If this approach is generalized to the level of individual pairings, associating a pair of parentheses to each basepair formed within the structure, it becomes possible to estimate the number of ways to form classical secondary structures in an RNA molecule of length N . This number will be of the same order as the number of ways to form expressions containing $p = N/2$ pairs of parentheses³. That number is known and related to the central coefficient of the binomial that we have already mentioned in chapter 2 (Sequence Comparisons). The number $S(p)$ of ways to form a syntactically correct expression containing p pairs of parentheses is expressed as:

$$S(p) = \frac{1}{p+1} \binom{2p}{p} \text{ which increases as } 2^{2p}, \text{ for large } p$$

For example, whereas for $p = 4$ there are only 14 possibilities (*cf.* Figure 6.14), for $p = 20$ the number of possibilities exceeds 6.5 billion.

It is clearly impossible to test all the possible combinations and to calculate the associated ΔG^0 , even for very low values of N (or p). As in the case for the optimal alignment of two sequences, a more astute strategy is required. Some elements or substructures are shared by a great number of different possible structures. For example, the five structures illustrated in the top line of Figure 6.14 share the bottom helix, which forms the ‘trunk’ of these various tree-like topologies. The problem may be considerably simplified by calculating the energy of a given subsequence of length N associated only once with the formation of an RNA sequence structure. The recurrent calculation of the $\Delta G^0_{i,j}$ associated with the optimal local foldings for longer and longer subsequences from i to j is used to calculate the optimal ΔG^0 for the whole sequence, $\Delta G^0_{1,N}$. This constitutes a dynamic programming method similar to that used for the optimal alignment of two sequences.

The Nussinov algorithm

Nussinov proposed the first application of dynamic programming for the prediction of secondary RNA structure in 1978. Based on a drastic simplification of the rules governing energetics, it takes into account only the contributions of paired regions (helices), attributing a score to each basepair and disregarding the destabilizing contributions of bulges, loops, and unpaired extremities. However,

³ In fact, the number of ‘realistic’ secondary structures is a bit smaller than the number of parenthetical expressions, since certain structures thus formed contain very short loops or non-canonical pairs. However, this approximation provides an idea of the asymptotic behavior of the number of structures. A probabilist asymptotic estimation of the number of ‘biologically reasonable’ structures $S(N)$ that an RNA sequence of length N can form in which there is equal distribution of the four nucleotides A, G, C, and U was given by Zuker and Sankoff: $S(N) \approx 1.8^N$.

its functioning principle is developed in the next sections, since it remains the basis of most of the more sophisticated algorithms, which merely improve on it. As for optimal sequence alignment, the Nussinov method consists of two steps: first completing a table, then identifying the optimal path in it.

For a given RNA sequence of length N , we will call $E(i, j)$ the energy of the most stable structure in a sequence extending from nucleotide i to nucleotide j ($1 \leq i \leq j \leq N$). With the Nussinov algorithm, the set of these ‘partial’ energies is progressively tabulated in an $N \times N$ table, using the following relation:

$$E(i, j) = \min \begin{cases} E(i+1, j-1) + e(i, j) \\ \min_{i < k \leq j} (E(i, k-1) + E(k, j)) \end{cases}$$

where $e(i, j)$ is the individual pairing score of base i with base j .

While this expression may appear complex at first glance, it corresponds to the example schematized in Figure 6.15.

In the best structure for the i, j subsequence, either base j is paired to base i (the left-hand example in Figure 6.15) and the stem that had nucleotides $i+1$ and $j-1$ as its extremities is prolonged, or the structure is divided into two parts, from i to $k-1$ and from k to j , and the best combination of two such pieces sought.

This recurrence relation over $E(i, j)$ makes it very easy to complete the energy table progressively, proceeding by successive diagonals that correspond to a constant sub-segment length $L = j + 1 - i$. Since it is impossible for a structure to consist of a single nucleotide, diagonal values in table $E(i, i)$ are initialized at zero. In order to prohibit loops that are too small, $E(i, i+1)$ and $E(i, i+2)$ may also be initialized at zero. The diagonal terms corresponding to segments of

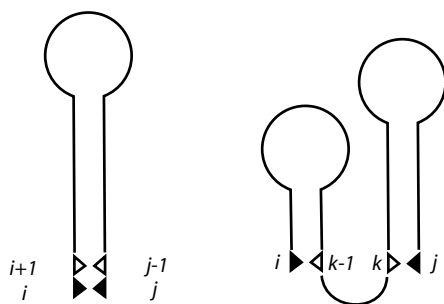


Figure 6.15 Seeking the minimal energy conformation for the $[i, j]$ subsequence. The i, j pair is either formed (left) or not formed (right). Substructures are represented in simple stem-and-loop form, but may be more complex, or even unpaired (single strand).

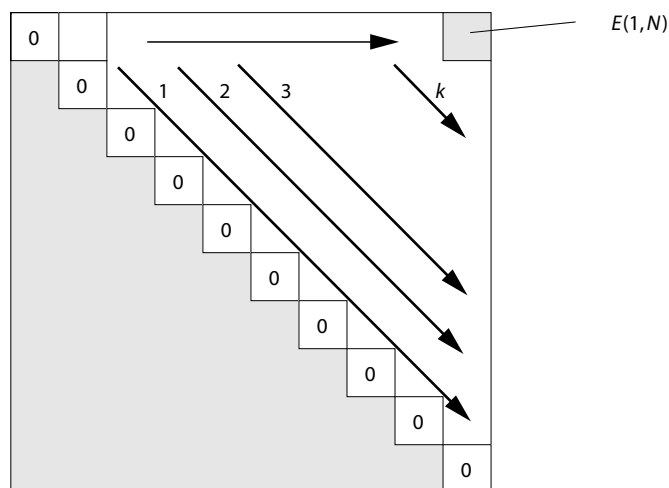


Figure 6.16 Completing the energy table.

increasing length are then progressively calculated. The energy corresponding to the complete sequence $E(1,N)$ is obtained upon arriving at a corner of the table.

In order to reconstitute the structure associated with this global minimum energy, it is necessary to go backwards through the table, seeking the path that led to the minimum energy value. The term in the above alternative that was used to calculate the calculation of $E(i,j)$ must be identified each time. The limiting step in the Nussinov algorithm is the completion of the table, which is $O(N^3)$ for computation time and $O(N^2)$ for memory space.

The Zuker algorithm

In the early 1980s, Michael Zuker perfected Nussinov's method, adopting energy rules that were more realistic, such as those of Freier and Turner, including stacking interactions between nearest neighbors in helices and the contributions of unpaired regions. An exhaustive description of the Zuker algorithm is complex, since it takes into account a great number of special cases. Only its main principles are outlined in the following sections.

The principal modification with respect to the Nussinov method is the introduction of a second table. In addition to the $E(i,j)$ table that still gives the energy of the best structure of the $[i,j]$ segment, Zuker introduced the $V(i,j)$ table, which

contains the energy of the best structure in which the $i - j$ pairing is formed. Since the $i - j$ pair is forced, it might be a sub-optimal structure. Thus $V(i, j) \geq E(i, j)$.

The $V(i, j)$ and $E(i, j)$ tables are then progressively calculated, using the following recurrence relations:

$$E(i, j) = \min \left\{ \begin{array}{ll} E(i+1, j) & \text{unpaired } i \\ E(i, j-1) & \text{unpaired } j \\ V(i, j) & i-j \text{ pairing} \\ \min_{i < k < j} (E(i, k) + E(k+1, j)) & \text{broken down into 2 substructures} \end{array} \right.$$

$$V(i, j) = \min \left\{ \begin{array}{l} \text{Terminal loop } (i, j) \\ V(i+1, j-1) + ee(i, j) \\ \text{Internal loop } (i, j) \\ \text{Multiple loop } (i, j) \end{array} \right.$$

Calculation of the $E(i, j)$ table is the same as for the Nussinov algorithm, except that two terms are explicitly added in order to take into account cases in which either base i or base j is unpaired. The table and the path search are completed in the same way. Calculating $V(i, j)$ is more complex: The $ee(i, j)$ function represents the gain in energy obtained by stacking the i, j basepair onto the $i+1, j-1$ basepair. Among other reasons, it is for the purpose of calculating this term that E and V must be tabulated separately, since it must be possible to use $V(i+1, j-1)$ to calculate $V(i, j)$.

The various Loop (i, j) functions represent the energy of the best structure in which the i, j basepair close an internal loop, a terminal loop, or a multiple loop. Calculation of the *terminal loop* function is rather simple using empirical thermodynamic rules, but the two other loop types are much more complex, often requiring the use of *ad hoc* hypotheses to limit calculation time (loop size upper limit, simplification of free energy rules). The more complex of the two, *multiple loop*, is at best $O(N^3)$ in calculation time for each $V(i, j)$ term, which makes the Zuker method an algorithm of complexity $O(N^5)$. With currently used machines, the Zuker algorithm allows handling of sequences of between several hundred and several thousand nucleotides, which is generally adequate for most common biological problems.

Limits of in silico predictive methods

The computer methods for predicting secondary RNA structure like the ones presented above have two types of limitation:

- (i) The inaccuracy of empirical rules combined with certain approximations necessary for the efficient calculation of energy associated with loops produces a relatively high degree of error in the energy values obtained;
- (ii) The algorithms presented yield only a single solution, which corresponds to the minimal energy.

The combination of these two limitations may result in failure to detect the correct structure, especially if there are several alternative solutions whose free energies are very similar. One necessary improvement in predicting secondary RNA structure is therefore to generate suboptimal solutions whose energy scores fall a few percent below the highest score. The biologist must then sort the structures within the family of candidates, based on additional criteria.

6.6 Validation of predicted secondary structures

One of the main problems encountered in trying to predict RNA folding, for example, ribosomal RNA folding, is that there are usually many alternative conformations with rather similar calculated energies. However, biologists have two complementary approaches for solving this problem: i) phylogenetic analysis; ii) enzyme or chemical probes for use in characterizing effectively paired regions.

Phylogenetic analysis

When the RNA whose folding is being determined belongs to a family for which several sequences from related organisms are available, comparisons may be used to confirm the existence of a paired region. The various secondary structure components are usually conserved, and the appearance of a mutation on one strand in a helical region must be accompanied by the appearance of a **compensatory mutation** on the other strand, in order to restore a basepair. Observation of this type of covariation between two positions in several RNA sequences of the same family is an extremely strong argument for the existence of an interaction between the two corresponding bases in the two- or three-dimensional structure.

Analyses of this type are systematically carried out on ribosomal RNA and on several families of autocatalytic introns, for which dozens, even hundreds of homologous sequences have been compiled in databases. This work has allowed reconstitution of the secondary structures of RNAs with sequences exceeding several kilobases with a high degree of certainty.

If a reliable multiple alignment in a family of RNA homologs is available, it is possible to describe a quantitative way of measuring the correlation between

variations in the bases at positions i and j . This relation is called *mutual information* $M(i,j)$, and is written:

$$M(i, j) = \sum_{b_i, b_j} f_{b_i b_j} \log \frac{f_{b_i b_j}}{f_{b_i} f_{b_j}}$$

where b_i and b_j are the various bases possible at positions i and j , f_{b_i} , f_{b_j} , their individual frequencies of appearance at positions i and j , and $f_{b_i b_j}$ the frequency of simultaneous appearance of b_i , and b_j at positions i and j .

This quantity, assigned a value between 0 and 1, increases as the correlation between positions i and j increases. If there is no correlation, or if the positions do not vary, it is 0. Since mutual information analysis makes no *a priori* hypothesis regarding the nature of the pairs (Watson–Crick, wobble, or other), it allows detection of standard interactions among helical segments, as well as more exotic, non-canonical interactions. Systematic study of $M(i,j)$ can also detect more complex interaction zones of the pseudoknot type, as well as among secondary structural components, permitting construction of a primitive three-dimensional folding model.

6.7 Using chemical and enzymatic probing to analyze folding

If purified RNA is available, probes may be used that react differently with single- and double-stranded regions to provide information concerning its structure in solution.

RNA can be subjected to limited enzyme attack. Ribonucleases (RNases) exist that preferentially cleave either unstructured regions (e.g., RNaseT1, nuclease S1) or helical regions (e.g., RNase V1). Comparison of the experimental distribution of phosphodiester bonds cleaved by these two classes of

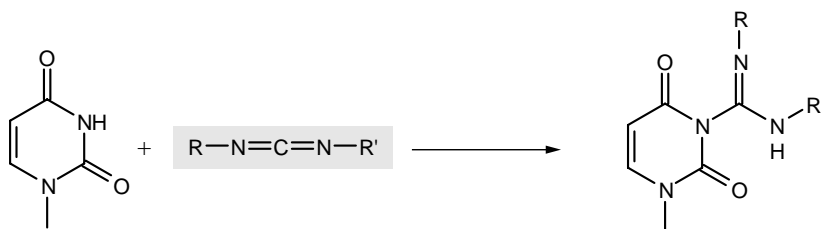


Figure 6.17 Carbodiimides (highlighted) react with the G and U imino groups. This reaction is possible only if the base is not involved in pairing. Cartography of the reactive positions identifies single-strand regions of the RNA sequence.

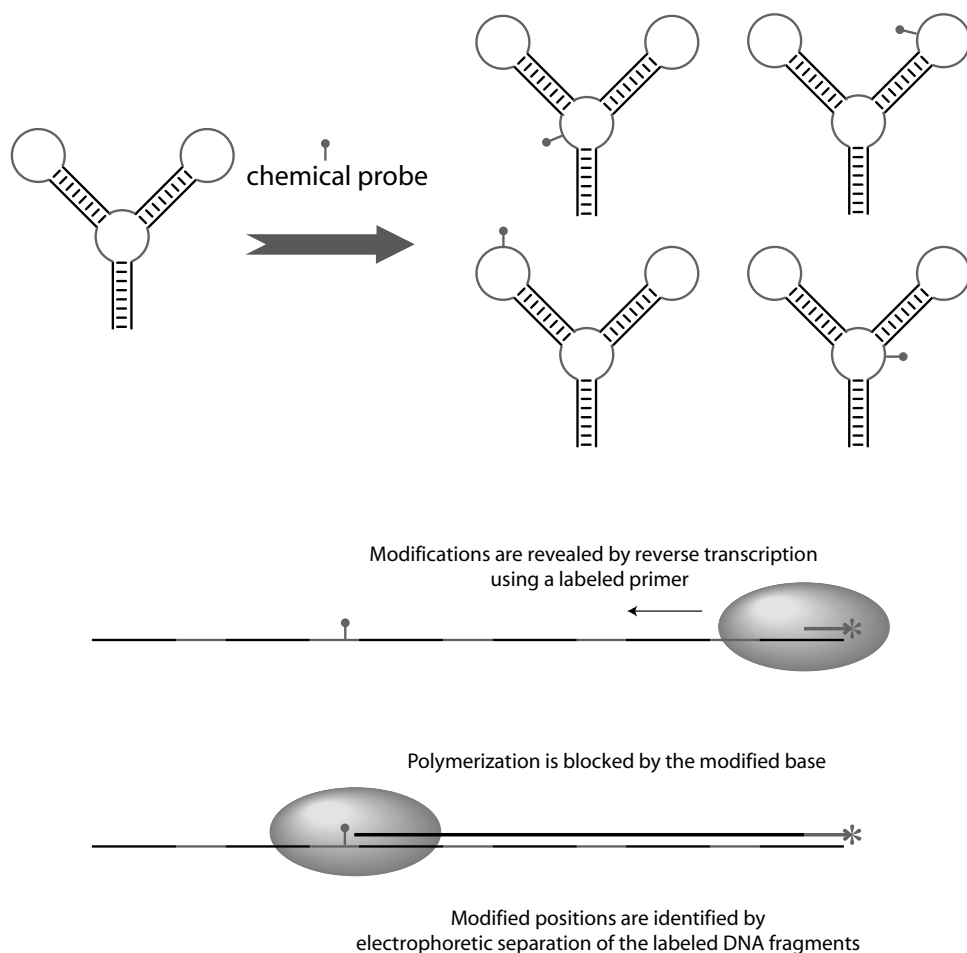


Figure 6.18 Principle of secondary RNA structure analysis using chemical probes, such as carbodiimides.

enzymes with paired regions in various computer-predicted models allows elimination of those that are inconsistent with experimental data.

The use of enzyme probes is not always sufficient, since ribonucleases are macromolecules and therefore occupy a non-negligible volume. For steric reasons, they sometimes do not reach all the theoretically cleavable sites of the RNA sequence being studied, which results in ambiguities and difficulties in interpretation for certain regions. In order to complete the analysis, smaller chemical probes, such as alkylating reagents and metal ions, are often used. Combining these probing strategies usually allows precise identification of paired RNA regions in the structure.

The ‘signature’ thus obtained directly from the molecule may then be used to screen the various secondary structures predicted by computer.

6.8 Long-distance interactions and three-dimensional structure prediction

The three-dimensional structure of highly structured RNA, such as ribosomal RNA, is well defined. As in the folding of peptide chains of proteins, it is stabilized by long-distance interactions between secondary structural elements. Three types of long-distance RNA structural interactions are known today: pseudoknots, tetraloop–receptor interactions, and triple helices. Systematic phylogenetic analysis permits location of some of these interactions. If enough of these long-distance contacts are known, in some cases it is possible to construct a more or less detailed model of the three-dimensional structure of the RNA sequence. The formation of pseudoknots was mentioned previously, and other types of interactions will be described in this section.

Triple helices

Watson–Crick-type pairing is not the only form of basepairing possible between a purine (A and G) and a pyrimidine (C and U). Purines have a second face, known as the *Hoogsteen face*, which can form hydrogen bonds with a pyrimidine. Both *Hoogsteen* and *reverse Hoogsteen* pairing are possible, according to the orientation of the pyrimidine (see Figure 6.19). Hoogsteen pairings may be

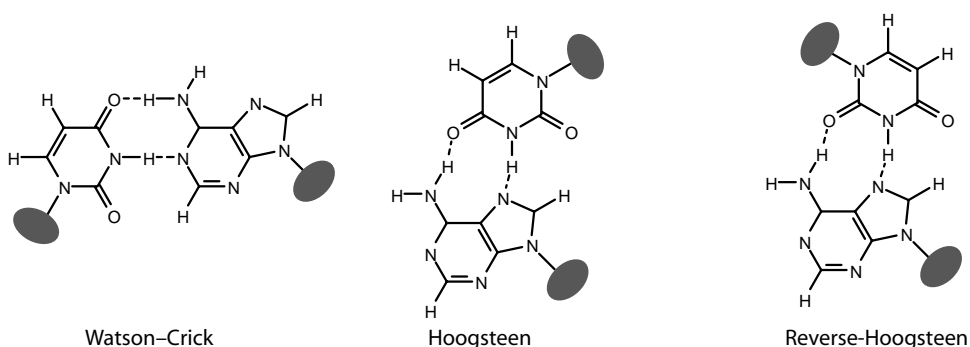


Figure 6.19 Hoogsteen-type pairings. In direct Hoogsteen pairing, the two RNA strands are parallel, whereas in reverse Hoogsteen pairing, the two strands are antiparallel, as in Watson–Crick pairing.

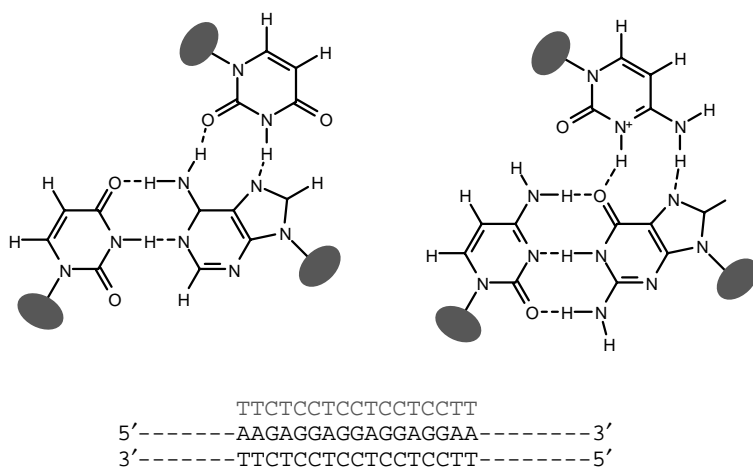


Figure 6.20 When one of the two classical Watson-Crick strands contains a homopurine tract, Hoogsteen or reverse Hoogsteen pairing can form a triple helix with a third homopyrimidine strand, as shown here.

formed between A and U (Figure 6.19) and between G and C, provided that the N3 carbon atom is protonated.

Examining the geometry of these various basepairings reveals nothing that can simultaneously form Watson-Crick and Hoogsteen pairings. A triplet of bases therefore forms. The stacking of several such triplets leads to the formation of a triple helix that consists of a standard double helix plus a third strand inserted into the major groove of the duplex.

However, a major criterion must be fulfilled for the stacking of several base triplets to occur: *the purines must succeed each other on the same strand* (Figure 6.20). Therefore, a triple helix can form only in regions that contain consecutive purine sequences, known as ‘homopurine sequences.’ These triple helical structures can interact at long distances in an RNA sequence. An unpaired region, such as a loop of appropriate sequence can in effect insert into the major groove of a stem containing a homopurine strand and a homopyridine strand (Figure 6.21).

Tetraloop–receptor interactions

Tetraloops of the GNRA family (see *Hyperstable tetraloops*, above) are able to interact over long distances with more or less regular helical structures. These structures, known as *tetraloop receptors*, are specific for the associated tetraloop. As an example, Figure 6.22 shows the structure of tetraloop GAAA, which was found in the structure of the core of a self-splicing intron.

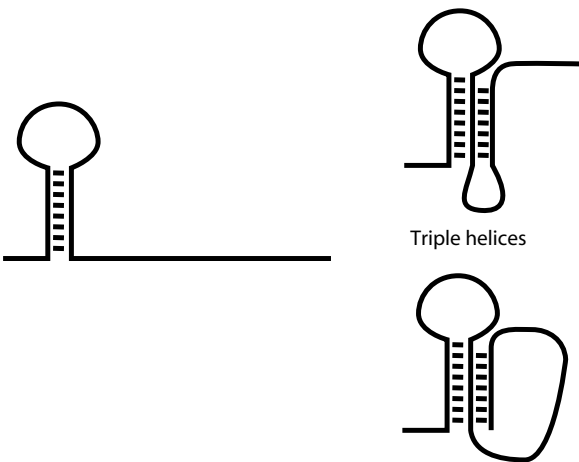


Figure 6.21 Triple helix formation.



Figure 6.22 Tetraloop–receptor interaction. The sequence of the receptor indicated is specific for the GAAA tetraloop. Other GNRA loops have different receptors associated with them.

Prediction of three-dimensional structure

When a sufficiently full set of related sequences is available, systematic analysis of the covariation between tetraloops and receptors or between sequences and loops able to pair and form pseudoknots, and the search for homopurine and homopyridine sequences able to form triplexes, can generate enough con-



Figure 6.23 Model of the structure of a self-splicing intron. The essential characteristics of this model were subsequently validated by determination of the crystallographic structure of the catalytic heart of the intron.

straints to unequivocally model the three-dimensional structure of some RNA molecules. These models are often constructed manually or semiautomatically, with fixed helical regions in the form of rigid cylinders. These models can be tested and validated by incorporating chemical and enzymatic reactivity data. In some cases, covalent bridging between nucleotides may be obtained, either by ultraviolet radiation or the action of a chemical reagent. The incorporation of supplementary data can refine and confirm the model. Several RNA complexes have been modeled in this way, permitting remarkable functional predictions.

6.9 Protein structure

The stakes and the difficulties

The rapid high-throughput genomic sequencing programs operating today are filling protein sequence databases, whose sizes double every 15 to 18 months. At the same time, although progress has been made, resolution of three-dimensional protein structure remains a relatively difficult task. At present, such structures are available for only a small fraction of proteins. The hiatus between protein sequences and three-dimensional structures will persist in the

foreseeable future, resulting in an obvious interest in predictive methods that allow avoiding the experimental step, at least for some applications.

Regrettably, the problem of predicting protein folding remains very difficult and much more complex to formalize than that of RNA folding. Four types of difficulty exist:

- There are no simple interaction rules, such as Watson–Crick pairing for RNA.
- Energy-prediction methods are inadequate. The free energy associated with folding ($\Delta G^0_{\text{folding}}$) is relatively small, on the order of a few dozen kcal/mol, or less. The imprecision of the methods used for energy calculation makes any attempt at prediction impractical.
- The more complex alphabet used for proteins (20 amino acids versus 4 nucleotides) makes using phylogenetic methods based on comparative sequence analysis more difficult. For the same reason, the wide chemical diversity of amino acids complicates the table by increasing the number and nature of interactions encountered in three-dimensional protein structures (e.g. hydrophilic, ionic, polar).
- Proteins adopt a very wide variety of three-dimensional structures. A ‘topological’ classification based on the nature of the secondary structural elements (α -helices and β -sheets) they contain has been established (see Figure 6.24). Some contain only α -helices or β -sheets, while most contain a combination of the two.
- Finally, proteins longer than 300 amino acids are almost always organized into two more or less separate domains. Starting from the sequence, this morphological richness renders *ab initio* prediction of three-dimensional protein structure more difficult.

However, these obstacles are not totally insurmountable, and it has been possible to make some progress, most often of limited scope, but sometimes leading to spectacular success.

Given a protein for which only the primary amino acid sequence is known, it is at present reasonable to envisage three approaches to predicting three-dimensional protein structure:

- In the absence of all other information, it may be possible to predict the positions of the secondary structural elements in the protein (α -helices, β -sheets);

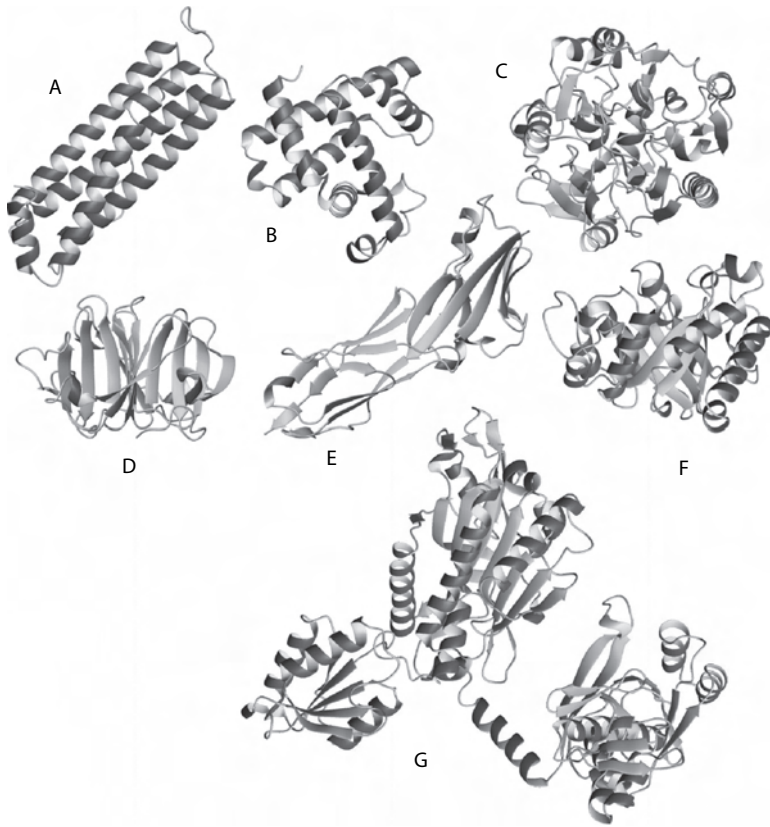


Figure 6.24 Various types of three-dimensional protein structures: A) parallel α -helices (ferritin); B) α -helices of mixed orientations (myoglobin); C) α -helices and antiparallel β -strands (amidino-transferase); D) β -sheets (collagenase); E) β -sheet sandwich (CD4 antigen); F) barrel formed by β -strands alternating with α -helices (triose phosphate isomerase); G) multi-domain protein (aminoacyl-tRNA synthetase).

- If the protein contains sequence homologies with a protein of known structure, it is possible to construct a more or less accurate three-dimensional folding model based on alignment of the two sequences;
- It is possible to determine whether the three-dimensional folding of the protein corresponds to folding that has already been determined and referenced in databases. To do so, *ad hoc* methods based on the distribution of secondary structural elements predicted for the sequence may be used, as well as more systematic methods that consist in attempting to ‘thread’

the protein sequence onto a three-dimensional fold and evaluating the degree of correspondence, using more or less elaborate cost functions.

These three strategies are briefly developed below.

6.10 Secondary structure prediction

There exist a great number of methods for predicting elements of secondary protein structure, which it would be impossible to present here. However, they all more or less directly depend on a combination of three physicochemical and geometric properties of proteins:

- In soluble proteins, hydrophilic amino acids are usually exposed on the surface, where they are in contact with the aqueous environment, while hydrophobic amino acids face the interior of the protein.
- β -sheets are structures of periodicity 2 (along a strand, the lateral chains are alternatively above and below the plane of the sheet) and the α -helices are structures of periodicity 3.6 (the lateral chains point toward the outside of the cylinder of the helix, whose pitch is 3.6 residues per turn.)
- A protein domain contains an average of 100 to 300 amino acids. The **average** length of secondary structural elements therefore corresponds to the diameter of such a domain, approximately 30 to 40 Å. This length corresponds to between 15 and 20 amino acids for an α -helix and to between five and 12 amino acids for a β -sheet whose peptide chain is completely extended. Between these elements of regular structure, the protein backbone forms loops, most often exposed on the surface.

There are two main classes of predictive methods, both based on amino acid distribution in the sequence: pattern recognition methods, which try to use the above observations directly and statistical methods, which use them in a more indirect way, calculating a probabilistic score.

Most methods used at present utilize a three-state secondary structure model: (1) α -helix; (2) β -sheet; (3) loop or irregular structure. The model aims to predict in which of these three states each amino acid in the sequence is located.

Pattern recognition

The principle of pattern recognition methods is to try to recognize the specific distribution of residues in each type of secondary structure. For example, if a

strand of a β -sheet is located on the surface of a protein, one would expect half the residues to be hydrophilic and half to be hydrophobic. The same would be the case for an α -helix located on the surface, which would be expected to have a hydrophobic side facing the inside of the protein and a hydrophilic side facing the solvent.

By examining hydrophilic and hydrophobic residue distribution in the sequence of a protein of unknown structure, one can discern the specific alternation of α -helices and β -sheets. Surface loops are usually continuous regions composed of highly hydrophilic residues.

One of the first methods used to detect helices was a graphic representation known as the 'helical wheel', in which the sequence of the backbone is wound onto an α -helix viewed perpendicularly along its axis (Figure 6.25). It is possible to directly visualize the presence or absence of hydrophilic and hydrophobic faces on the cylinder. Other graphic methods that use the surface of the cylinder have also been proposed.

Two approaches may be used to identify these distributions, one of which is purely formal, while the other is more quantitative. The first approach, proposed by the Russian biologist Valery Lim, consists in searching for certain patterns of hydrophobic residues that are characteristic of various types of secondary structures. It defines a set of 22 complex rules that allow prediction of the whole sequence. Some examples of the Lim rules are indicated below (Φ designates a hydrophobic amino acid, and $-$ a hydrophilic one).

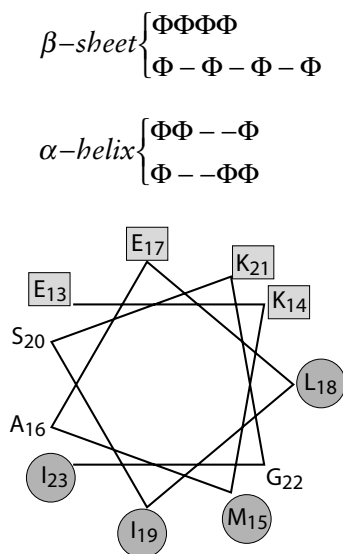


Figure 6.25 Representation of residues 12 through 23 of flavodoxin in the form of a helical wheel. Highly hydrophilic residues are in squares and hydrophobic ones in circles. The asymmetric distribution characteristic of helical regions is clearly demonstrated.

The quantitative approach consists in attributing a score to each amino acid. Several empirical hydrophobicity scales exist, based on either physicochemical parameters (e.g. solubility, partition coefficient) or protein-surface distribution statistics. For a protein sequence of length N , it is possible to use this score to calculate a vector quantity known as the *hydrophobicity moment*:

$$\vec{\mu} = \frac{1}{N} \sum_{i=1}^N H_i \begin{pmatrix} \cos \delta_i \\ \sin \delta_i \end{pmatrix}$$

where H_i is the hydrophobicity score of the i^{th} amino acid of the segment and δ is a periodicity parameter taken to be equal to 180° in order to detect β -sheets and to 100° to detect α -helices. The vector $(\cos \delta_i, \sin \delta_i)$ therefore corresponds to the orientation of the lateral chain of the i^{th} residue in the corresponding periodic structure. If the sequence of N residues studied does fold according to a structure of periodicity δ , the hydrophobicities H_i will sum in a constructive manner and the moment module $\langle \vec{\mu} \rangle$ will be large; if not, the hydrophobicities will on the average cancel out and the $\langle \vec{\mu} \rangle$ module will be very small.

It is also possible to sweep a window of N residues through a sequence being studied (N typically being on the scale of the average length of secondary structural elements). The positions in which $\langle \vec{\mu}_{helix} \rangle$ or $\langle \vec{\mu}_{sheet} \rangle$ are higher than a threshold value respectively indicate the probable presence of a helix and a sheet at that place.

Statistical methods

The statistical methods are based on systematic study of amino acid distribution in three-dimensional protein structures available in databases. The principle underlying these methods is determination of whether there is a bias in the composition of helices, sheets, and loops that could be used for prediction. It was noticed very early that the amino acids glutamate, methionine, and alanine were over-represented in α -helices. This simple frequency analysis led to an early simple predictive method, known as the method of Chou and Fasman.

The principle of this method was later modified and extended by Garnier and Robson to take into account effects at a distance. The principle is to observe the influence of the residue $i + k$ ($-8 \leq k \leq +8$) on the conformation (helix/sheet/loop) of residue i by carrying out detailed frequency analysis. This influence is quantified in terms of information, I , in the Shannon sense. Given two events X and Y , this is:

$$I(X; Y) = \log[P(X|Y)/P(X)]$$

Starting with frequency analysis in the structure database, these authors tabulated the following information regarding the conformation of residue i , S_i :

$$I(S_i = \alpha\text{-helix}; a_{i+k}), I(S_i = \beta\text{-sheet}; a_{i+k}), \text{ and } I(S_i = \text{loop}; a_{i+k})$$

Starting with this information, Garnier and Robson were able to reconstitute the information associated with subsequence $a_{i-8} \dots a_i \dots a_{i+8}$ for the three types of S_i structures.

$$I(S_i; a_{i-8} \dots a_{i+8}) = \sum_{k=-8}^8 I(S_i; a_{i+k})$$

These information profiles can attribute a probable conformation to each residue i .

Efficiency and limits

At present, and despite numerous improvements, when applied to proteins of unknown structure, the accuracy of these predictive methods does not exceed 70% correctly predicted residues. Two kinds of difficulty are encountered; on one hand, there are boundary effects, since it is always difficult to precisely determine the borders of secondary structure elements. On the other hand, these methods often miss small β -sheets, since they are too short to give a detectable signal.

Among the efficient improvements made to this method, one approach consists in using not just a single protein sequence, but a set of homologous sequences previously subjected to multiple alignment analysis. At position i in the sequence, there is not just a single amino acid, but a set of *possible* amino acids corresponding to all the residues found in the i^{th} column of the multiple alignment. This information is precious, since it provides data concerning the various amino acids that are compatible with the structure of the protein.

Several methods using multiple alignment of homologous sequence profiles have been implemented. To date, one of the most efficient remains *PredictProtein*, a neural network system that is ‘trained’, using the profiles of known structural proteins (www.embl-heidelberg.de/predictprotein/predictprotein.html). This system is related to statistical methods, since it learns by effectively taking into account the frequencies of the various types of amino acids. However, it is a ‘black box’; therefore it is difficult to figure out how it functions.

6.11 Three-dimensional modeling based on homologous protein structure

The *Protein Data Bank* protein structure database library (www.rcsb.org/pdb/) contains a great number of homologous protein structures. Its use makes it possible to see how the observed similarities in protein amino acid sequences are

reflected in their three-dimensional structure. This systematic study was carried out by Chothia and Lesk around fifteen years ago. After eliminating protein surface loops, the conformation of which is often variable, from the study and retaining only the protein core, they found a very strong correlation between sequence conservation and structural similarity.

The similarity of two structures may be quantitatively evaluated by calculating the standard deviation of the positions of atoms in the peptide backbone. When the sequence identity level (percentage of identical residues) is 50%, the standard deviation of the atom positions is of the order of 1Å or less. Even with 25% identity, the standard deviation generally remains under 2Å. This is illustrated in Figure 6.27, which shows the structural similarity between human collagenase, an enzyme that digests collagen, and serralysin, a bacterial protease. However, alignment of the two sequences indicated below reveals the similarity to be tenuous, with only 27% identity and major insertions/deletions.

Reciprocally, this correlation between sequence homology and structural similarity can be used to construct a three-dimensional model of a protein of

```
AFQVWSDVTPLRFSRIHDGEADIMINFGRWEHGDGYFPDGDGLLAHAFAPGT----GVGGDSHFDDDEL---WSLGKGVG
+ Q W+DV + F+ + G+ I FG + +D G A+AF P T +GG + ++ ++ + G
SLQSWADVANITFTEVAAGQK-ANITFGNYSQDRPGHYDY--GTQAYAFLPNTIWQGGDLGGQTWYNVNQSNVKHPATEDYG

YSLFLVAAHEFGHAMGLEHSQDPGALMA-PIY---TY---TKNFRLSQ-----DDIKGIQELYGAS
F HE GHA+GL H D A P Y TY T+ F L DDI IQ LYGA+
RQTF---THEIGHALGLSHPGDYNAGEGDPTYADVTYAEDTRQFSLMSYWSETNTGGDNGGHYAAAPLLDDIAAIQHLYGAN
```

Figure 6.26 Alignment of the sequence of human collagenase with the catalytic domain of serralysin, a protease of the bacterium *Serratia marcescens*.

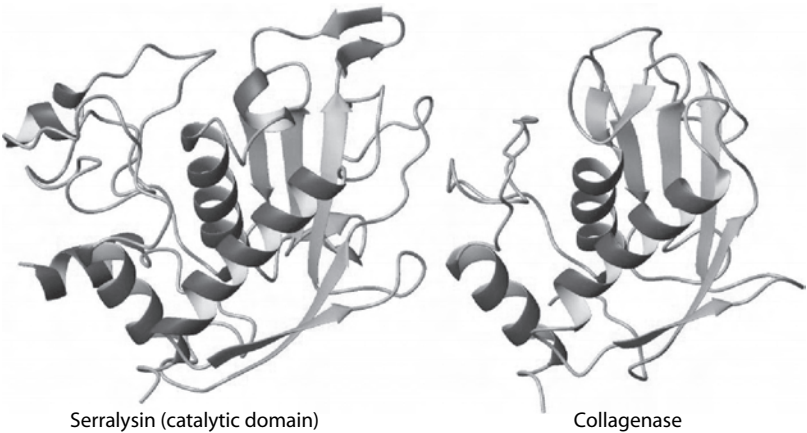


Figure 6.27 Three-dimensional structures of serralysin and collagenase.

unknown structure, starting from the known structure of a homologous protein. The protocol consists of four successive steps:

Core construction: Conserved regions corresponding to secondary structural elements in the core of the known structure are identified. Corresponding protein backbone regions are then placed onto the unknown structure.

Loop construction: Protein surface loops are extracted from the Protein Database Library. The loop of the unknown protein sequence that best corresponds in terms of length, sequence, and end-orientation is sought. The backbone of the selected loop is then grafted onto the backbone of the core constructed in the preceding step.

Sidechain budding: Following the first two steps, the polypeptide backbone is constructed, after which the two lateral chains are added. The conformation of the lateral chain of the reference protein is generally used directly at positions where there is sequence identity between the reference protein and the protein under construction. For positions at which the sequences diverge, as many as possible of the atoms the two lateral chains have in common are used.

Model refinement: Following the sidechain construction step, the model usually contains a number of defects, such as steric violations due to atoms being too close to each other, or unfavorable electrostatic interactions. It is possible to quantify these defects by using a classical formal energy calculation to sum the various potential energy terms associated with the molecular conformation. This includes the terms associated with deformation of the covalent geometry and the terms of non-covalent interaction: the electrostatic potential and Van der Waals interaction.

Covalent geometry

$$E_{\text{cov}} = \sum_{\text{bonds}} a_i (l_i - l_i^0)^2 + \sum_{\text{angles}} b_j (\vartheta_j - \vartheta_j^0)^2 + \sum_{\text{dihedral angle}} c_k (1 + \cos(n\varphi_k - \varphi_k^0))$$

The first term represents the energy associated with modifications in the lengths of the covalent bonds; the second corresponds to the modifications in the valence angles; and the third to deformations in the dihedral angles (torsion of planar bonds and aromatic cycles). Parameter n describes the periodicity of the dihedral angle (for example, $n = 2$ for a planar liaison, corresponding to *cis*- and *trans*-isomers.) The a_i , b_j , and c_k parameters are related to spring stiffness and have been empirically determined from the vibration frequencies of small molecules.

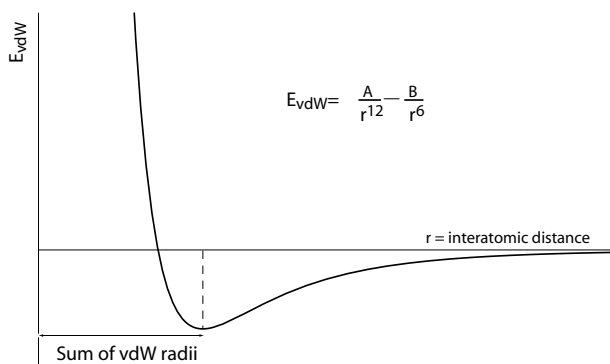


Figure 6.28 Potential of the Van der Waals interaction.

Non-covalent interactions

$$E_{non-cov} = \sum_{i \neq j} \frac{q_i q_j}{D r_{ij}} + \sum_{i \neq j} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right)$$

The first term corresponds to classical electrostatic interaction, in which parameter D describes the dielectric constant of the medium. The second term corresponds to the Van der Waals interaction, with a strongly repulsive $1/r^{12}$ part, which prevents interpenetration of the atoms and an attractive $1/r^6$ part. The A_{ij} and B_{ij} parameters are adjusted to yield a minimum that corresponds to the sum of the Van der Waals radii of atoms i and j (Figure 6.28). The two sums are carried out on all the pairs of atoms of the molecule.

Minimizing the energy of the reconstructed molecule

Starting from a conformation of the molecule defined by the Cartesian coordinates x_i, y_i, z_i of its N atoms ($1 \leq i \leq N$), it is possible to calculate the associated potential energy $E(x_i, y_i, z_i)$, using the above terms. This function of $3N$ variables can then be minimized, using classical numerical methods (e.g., steepest descent, conjugated gradient, allowing descent into a potential energy well). However, this approach is usually far from satisfactory, since the potential function is very rough and contains a great number of local minima. Direct minimization usually leads to one of these local minima, which often are far from an acceptable solution (Figure 6.29). To avoid these potential traps, a dynamic method called ‘simulated annealing’ is used.

Movements of atoms in the molecule can be simulated by the action of the above potential. To do this, initial random velocities are assigned to each atom

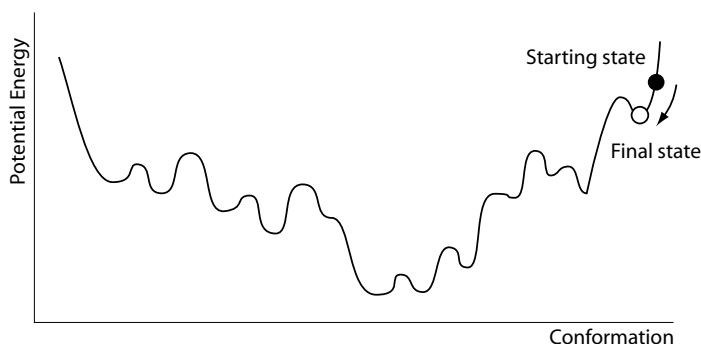


Figure 6.29 Because of the presence of numerous local minima in the potential function, the descent minimum does not give satisfactory results.

of the molecule. The distribution of these velocities is selected such that it follows a Boltzmann distribution at a given temperature (e.g., 300°K). The equation for the dynamics of the movement is then numerically introduced for each atom i :

$$\vec{F}_i = m_i \frac{d^2 \vec{r}_i}{dt^2} = - \frac{\partial E}{\partial \vec{r}_i}$$

This gives the movement trajectories of the atoms in the molecule. Following the few dozen picoseconds it takes for the system to equilibrate, the velocities of the atoms⁴ fall progressively during digital simulation of cooling. The molecule then drops into a much lower potential well, approximating a reasonable conformation (Figure 6.30). This method remains heuristic, and it is usually necessary to conduct several simulations in order to test convergence. It also still consumes much calculation time. An ‘average’ protein contains several thousand atoms, which makes calculation of the potential and trajectories rather cumbersome. In addition, it is necessary to resort to slow cooling, requiring numerous (10^4 to 10^6) integration steps, which takes several, or even dozens, of hours of CPU time. Happily, this kind of calculation lends itself to parallel processing, and most existing molecular dynamics programs are able to work on multi-processor platforms.

The final conformation obtained using this approach is usually a good approximation of the real structure of the molecule, especially in the conserved core.

⁴ For a given particle, $mv^2 = 3kT$, where k is the Boltzmann constant.

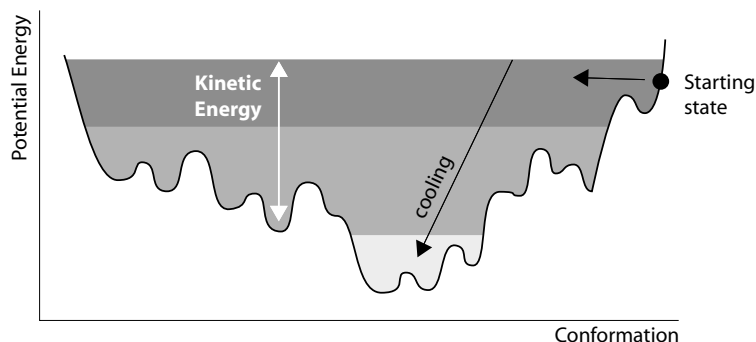


Figure 6.30 Annealing principle: Adding kinetic energy to the system first exceeds the potential barriers, then descends ‘below’ them in the potential function during cooling.

6.12 Predicting folding

As seen in the preceding paragraph, two proteins that present amino acid sequence homologies usually have closely related three-dimensional structures. However, the reciprocal of this observation is not true, and it has been shown that two proteins can have very closely related topologies without necessarily having detectable sequence homology.

It is therefore possible for a protein of unknown structure to take on a known three-dimensional topology in the absence of detectable sequence homology with other proteins that adopt the same folding topology. There is even a widely admitted postulate according to which there are a finite number of folding topologies for individual protein domains, which would explain these structural coincidences. From geometric or structural arguments, some authors even conjecture this number to be of the order of 1,000 to 2,000.

The protein structure database already includes several hundred different topologies. Therefore, when a new protein or family of proteins of unknown structure is analyzed there is a reasonable probability that one or several domains within it correspond to an already known topology. On the basis of this property David Jones and Janet Thornton several years ago proposed a new method for predicting global folding called *threading*. Their idea was to select a set of representative proteins from among all the known topologies included in protein structure databases and place the sequence of the unknown protein over the backbone of one of the selected proteins. A cost function is then applied to evaluate the quality of the interactions in the protein core and of the solvation of the surface polar groups. To optimize this structural threading, it is usually necessary to introduce insertions and deletions, especially in loops. Structural alignment by threading is therefore similar to sequence alignment, mentioned earlier, which dynamic programming methods may also be

used to achieve. During recent ‘competitions’, in which structural biologists challenge bioinformaticians to predict the folding of newly determined structures prior to making them public, this method has been able to correctly predict the global topology of 60 to 70 percent of the proteins submitted. This amounts to remarkable progress in a domain that continues to undergo major expansion and advancement.

Bibliography

- Chou, P.Y., Fasman, G.D. (1974). Prediction of protein conformation. *Biochemistry* **13**: 222–245.
- Freier, S.M., *et al.* (1996). Improved free-energy parameters of RNA duplex stability. *Proc Natl Acad Sci USA* **83**: 9373–9377.
- Garnier, J., *et al.* (1996). GOR method for predicting protein secondary structure from amino acid sequence. *Methods Enzymol* **266**: 540–553.
- Jones, D., Thornton, J. (1993). Protein fold recognition. *J Comput Aided Mol Des* **7**: 439–456.
- Lim, V.I. (1974). Structural principles of the globular organization of protein chains. A stereochemical theory of globular protein secondary structure. *J Mol Biol* **88**: 857–872.
- Michel, F., Westhof, E. (1990). Modelling of the three-dimensional architecture of group I catalytic introns based on comparative sequence analysis. *J Mol Biol* **216**: 585–610.
- Moult, J. (2005). A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Curr Opin Struct Biol* **15**: 285–289.
- Nussinov, R., Jacobsen, A.B. (1980). Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc Natl Acad Sci USA* **77**: 6903–6913.
- Pley, H.W., *et al.* (1994). Model for an RNA tertiary interaction from the structure of an intermolecular complex between a GAAA tetraloop and an RNA helix. *Nature* **372**: 111–113.
- Rost, B., Sander, C. (1993). Prediction of protein secondary structure at better than 70% accuracy. *J Mol Biol* **232**: 584–599.
- Sternberg, M.E. (1997). *Protein Structure Prediction: A practical approach*. IRL Press.
- Zuker, M. (1989). On finding all suboptimal foldings of an RNA molecule. *Science* **244**: 48–52.
- Zuker, M. (1989). Computer prediction of RNA structure. *Methods Enzymol* **180**: 262–288.

7

Transcriptome and proteome: macromolecular networks

7.1 Introduction

The traditional approach in molecular biology research is local; it consists in examining and collecting data on one gene, one protein, or one reaction at a time. We may recognize the classical reductionist approach, *understand the parts in order to understand the whole*, which has permitted remarkable advances over the years and resulted in the development of extremely precise biochemical models.

However, the advent of genomics has led to the emergence of an entirely new class of abundant data, which until now have principally been exploited in studying previously unknown genes, genes that are over- or under-expressed in certain circumstances, etc. While these data do constitute an important resource for researchers working on individual genes, is it reasonable to try to characterize all relevant molecular interactions one at a time when devising a predictive model for a human disease, given that the objective is to identify pharmacotherapeutic targets?

The changes introduced by (post-)genomics may also be described in terms of information theory: Knowledge of a biological system may be defined as the ratio of extracted information to the relevant information potentially present in the system. Until 1990, this ratio (which was very low) determined the overall strategy of biological research, which may be summarized as follows: Since access is available to only a limited quantity of data, experimental strategies must be found that focus on those parameters (genes, for example) whose effects predominate. The extraction of information from biological systems is less skewed today. Indeed, by furnishing data on thousands of genes, genomics has increased the ratio of extracted to potentially relevant information present in a given biological system by several orders of magnitude.

New methods are clearly required in order to comprehend these data on a global basis and to analyze these large-scale systems at an intermediate level

without descending to the level of precise biochemical reactions. At the very least, such analysis would be useful in guiding traditional pharmacological and biochemical approaches toward the genes that deserve the most attention, among the thousands recently discovered. Ideally, a sufficiently predictive and explicative model at the intermediate level would obviate the need for an exact understanding of the system at the biochemical level.

In this chapter, after examining the methods and data made available by post-genomics, we will undertake global analysis of the data.

7.2 Post-genomic methods

The use of (post-)genomic tools allows the acquisition of massively parallel molecular data, such as those that concern genomic DNA sequences (genomes), as well as the concentration, activity, localization, and interaction of messenger RNA (transcriptomes¹) and proteins (proteomes). This section describes the principal techniques used to acquire transcriptome and proteome data.

7.2.1 Proteomics

Separation, identification, and quantification of protein

The method of choice for fractionating a large number of proteins contained in a natural extract is two-dimensional gel electrophoresis (Figure 7.1). This technique separates proteins in a plane, first in one direction, as a function of their isoelectric point², then in the orthogonal direction, according to their molecular weight. After staining, the result is a two-dimensional image consisting of a large number of spots corresponding to the constituent proteins. The intensity of spot coloring with certain stains is approximately proportional to the quantity of protein present. However, spot resolution may not be sufficient to separate all the proteins; therefore the results obtained using two-dimensional electrophoresis gels are subject to problems of reproducibility and artifacts. Recently, these problems have been partially resolved by using highly standardized protocols and high-precision techniques. Today, it is possible to separate 2000 spots on a single gel. Proteins that have extreme isoelectric points are under-represented (supplementary gels with an extreme pH can partly remedy

¹ Although the notions of activity, localization, and interaction also apply to mRNA, they are at present inaccessible on a large scale.

² pH at which the overall charge on the molecule is neutral; otherwise stated, the pH at which the molecule does not migrate in the applied electric field. The first dimension of the gel therefore corresponds to a stable pH gradient.

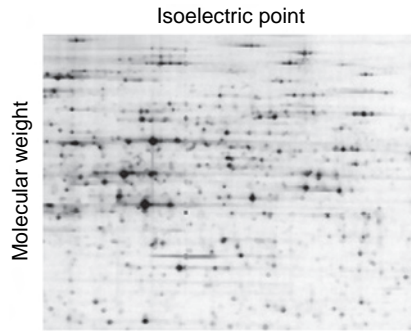


Figure 7.1 Two-dimensional protein electrophoresis gel. Proteins contained in a natural extract are fractionated in a polyacrylamide gel subjected to an electric field. They are separated first according to their isoelectric point in a stabilized pH gradient and a weak detergent (horizontal axis), then by their molecular weight, in the presence of a strong ionic detergent (vertical axis). Finally, they are stained in the gel to render them visible.

this), as are particularly hydrophobic proteins, which are not solubilized by the weak detergent used in the first separation.

It is impossible to know at the outset which proteins are present in a spot. Identification may be achieved in part by referring to a database containing two-dimensional gel electrophoresis results. Such databases exist, notably for bacteria, yeast, fruitfly, mouse, rat, and human proteins. When such information is not available in a database, and if the sequences of the proteins are known, their positions may sometimes be estimated by calculating their isoelectric points and molecular weights, providing they have not undergone post-translational modification. Otherwise, the spots must be removed and the proteins they contain eluted from them. If the sequence of a protein is known, it may be identified after mild hydrolysis, either by microsequencing or by mass spectrometry of the polypeptides obtained. In rare instances, an eluted protein can be renatured and its biological activity tested.

Intracellular localization

It is more difficult to evaluate the cellular localization of proteins on a large scale; however, some attempts have been made to do so for yeast proteins. All coding sequences have been fused to a fluorescent reporter protein, such as green fluorescent protein (GFP), each strain containing only one such fusion sequence. The strain collection includes at least one representative of each coding sequence fused to GFP. The fluorescence is then located in the cells of each strain, using a light microscope equipped with a fluorescence device. This method is limited

by the low resolution of the light microscope ($0.3\mu\text{m}$) compared with the size of the cell, as well as by localization artifacts sometimes caused by the fused reporter protein, or by over-expression.

Protein–protein interactions

The vast majority of proteins interact with other proteins, either in a stable manner, in which case they are known as a *complex of polypeptide subunits*, or in a more or less transitory manner. All degrees of stability are possible, and may be expressed in terms of the half-dissociation time, or of the *dissociation constant*, to which the half-dissociation time is inversely proportional. The variation in free energy is proportional to the logarithm of the association/dissociation equilibrium constant. Like the tertiary structure of individual polypeptides, association between polypeptides involves weak chemical bonds or covalent disulfide bridges between two cysteine residues. These interactions may be demonstrated in various ways.

Traditional molecular biological methods

Classically, various molecular biological methods are used to demonstrate that proteins co-purify, thus interact in a non-transitory manner. Biochemists generally rely on a succession of chromatographic and biophysical methods when trying to purify a molecular entity responsible for a biological activity that they know how to assay. They evaluate the nature of such a molecular agent by fractionating it on a dissociation or so-called ‘denaturing’ gel; *i.e.*, under conditions that dissociate molecules from their neighbors but do not cleave them. If the most active purified sample corresponds to several protein bands in the dissociation gel, there is reason to suspect that the biological activity is due to a polypeptide complex. The biochemist can then try to identify the genes that code for the various polypeptides, using well-tested techniques inappropriately known as ‘reverse genetics’.

If available, antibodies against individual polypeptides presumed to be contained in the complex can be used to test co-purification directly. A polypeptide is immunoprecipitated with its antibody under controlled conditions, and different antibodies are used to verify whether other polypeptides are also present in the immunoprecipitate (for example, by immunoblotting, or by successive immunoprecipitations with various antibodies). Determining the most dissociating conditions that maintain co-immunoprecipitation (for example, by varying the salt concentration) permits qualitative evaluation of polypeptide interaction stability. But even this relatively direct approach is difficult to implement on the

scale of the complete set of proteins in an organism. One limitation of the immunoprecipitation technique is its poor ability to detect transitory interactions, unlike the approach described in the next section.

Two-hybrid approaches

An alternative to the classical approaches described above is to test the interaction between polypeptides A and B by their ability to bring artificial elements added to them by genetic engineering close enough to each other. In a version of this approach (Figure 7.2), one of the manipulated genes encodes a chimeric protein containing polypeptide A fused to the DNA-binding domain of a regulatory protein. The other manipulated gene encodes a chimeric protein consisting of polypeptide B fused to the transcription-activating domain of a regulatory protein (the same or another). Both genes are then expressed in the same cell (usually yeast), which also includes a reporter gene whose transcription is activated by the regulatory protein. The transcription-activating domain alone has no effect, since it lacks the capacity to bind DNA; the DNA-binding domain alone has no effect because it cannot activate transcription; hence, the reporter gene is not expressed. However, if polypeptides A and B form a complex, the two domains are brought close enough to each other to restore the full activity of the regulatory protein. The complex then binds to the DNA, thereby activating transcription of the reporter gene, which codes for a protein that is readily assayed, usually by a colorimetric method, providing semi-quantitative data on the binding of the two polypeptides. As a precaution, it is wise to test the inter-

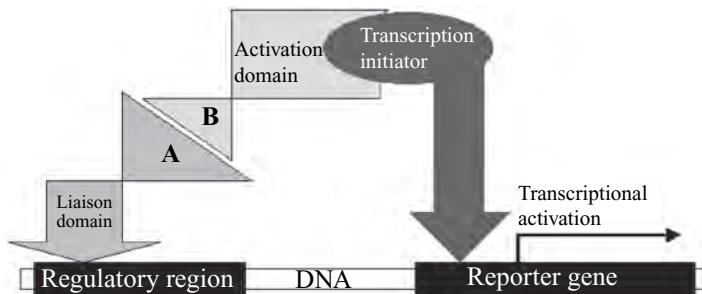


Figure 7.2 Two-hybrid assay. A functional activation complex is reconstituted by interaction between two tested polypeptides, A and B. A is fused to the DNA-binding domain of a regulatory protein. B is fused to the transcription-activating domain of a regulatory protein. Once the DNA binding domain is associated with DNA upstream from the coding sequence, interaction between A and B causes the cell transcription machinery to activate expression of the gene. The gene codes for a protein whose biological activity serves as a reporter for the transcription level.

action by reversing the roles of A and B, this time fusing A to the transcription-activating domain of the regulatory protein and B to its DNA-binding domain. Strong interaction in both configurations indicates true association of the two polypeptides. In certain real cases, it may be possible to demonstrate interaction in one direction but not in the other, justifying the use of a one-way arrow to symbolize the interaction. However, this reaction between proteins is fundamentally symmetrical: A binds to B \Leftrightarrow B binds to A. In other real cases, the reporter gene is highly expressed whatever B may be, since A is capable of binding to DNA by itself, or of activating transcription, or both; for example, if A is a regulatory protein.

In another version, A and B are both fused to fluorescent polypeptides, one of which emits blue light under ultraviolet excitation. If the other is very near, it absorbs the blue light and emits green. A fluorimeter set to detect the green light records a signal only when A is very close to B, since the energy transfer is proportional to the distance between the two fluorophores to the sixth power.

Various other possibilities exist; for example, testing a polypeptide fused to one regulatory protein fragment against a 'two-hybrid bank' of random fusions to the other regulatory protein fragment. The result of such an assay is synthesis of a dye that visibly stains the yeast colony, or of a protein essential for cell survival, thereby permitting direct selection of cells that bear the winning combination. In the 'reverse two-hybrid' technique, the reporter gene is a 'killer,' allowing only cells bearing combinations that do not associate to survive. The two-hybrid technique is binary, which also constitutes a major limitation: if a third protein is required, two polypeptides being tested will never be found to associate. A so-called three-hybrid variant technique therefore also exists, in which the interaction between modified polypeptides A and B is measured in the presence of a required third protein sandwiched between them. Note that if the two-hybrid approach is carried out in the organism from which the tested polypeptides come, a sort of 'accidental' three-hybrid may occur, since the third component is already present in the organism.

It is equally possible to test portions or domains of proteins rather than complete polypeptides. This has several advantages: The domain that interacts with the other polypeptide is clearly specified. The validity of the result may be estimated by the degree of redundancy and connectivity of the observed interactions of domains from the same natural polypeptide. Additionally, it becomes possible to test a polypeptide (for example, a membrane protein) that in its natural state would be distant from the nucleus, where the reporter gene is located.

The two-hybrid technique is sometimes used on the full set of proteins of an organism. If the number of proteins to be tested in both configurations (bait and test) is N, there are obviously 2N genetic constructions, and around N² tests would have to be carried out. This is an operation that calls for major automa-

tion. One limitation of this technique is that it is possible to test only proteins, which is not the case for the technique described below.

Protein chips

The affinities of a given protein for other molecules may be tested by purifying it and attaching it to a microarray. This may be systematically carried out by establishing as many strains as there are genes in the organism being studied. Each strain over-expresses a different gene, which encodes its natural product followed by a constant stretch of a few amino acids with a strong affinity for a highly specific reagent (for example, the stretch may be an epitope against which specific antibodies are available). A microarray is coated with the specific reagent and a crude protein extract of each culture is deposited at each spot. The microarray is washed under conditions that keep the modified protein fixed and eliminate the others. The microarray is then incubated in the presence of the substance to be tested (generally a protein, nucleic acid, lipid, etc.) that binds at spots where its binding partner is located, which can then be assayed.

Unlike the two-hybrid technique, this approach has the disadvantage of being able to test interactions only under completely artificial conditions, and is limited to proteins that maintain their functional conformation when purified and deposited onto the solid surface of a microarray. An older and less frequently used alternative consists in attaching a different specific antibody at each spot on the microarray and testing for the presence of proteins that are recognized by the antibodies in a cell-free extract. Antibody preparation is obviously a time-consuming task with this technique. The protein chip described here would require constructing N strains in order to prepare an array consisting of N protein probes.

A future type of protein chip might be based on attaching a different DNA fragment at each spot and programming *in situ* synthesis of each protein of interest. Such a protein would simply have to remain bound to its mRNA, which is itself bound to the attached DNA. This could easily be achieved by removing the translation and transcription termination signals from the DNA.

Systematic identification of protein complexes

New techniques feature relatively mild purification of complexes containing several polypeptides. The complexes obtained are resolved by ultrasensitive mass spectrometry, which, as discussed in section 2.1.1, often permits identification of polypeptides on the basis of their molecular weight, predicted with the aid of a sequence databank. This method overcomes the binary nature of the two-hybrid technique.

Bioinformatics

As discussed in the chapter on comparative genomics (Chapter 3), conservation of gene proximity over long evolutionary periods, at least in bacteria, is a strong indication that such genes code for proteins that physically interact.

7.2.2 Transcriptomics

Large-scale monitoring of genetic expression is inspired by the premise that the functional state of an organism is largely determined by the expression status of its genes. The latter may be described as the quantity of each mRNA molecule present in the cell at a given instant. These quantities evolve over time as a function of the interactions between regulatory proteins and regulated genes. Methods for measuring the quantity of mRNA, as well as for investigating the relationship between regulatory proteins and their gene targets, are briefly examined below. These methods are somewhat similar.

Complementary DNA microarrays

Used throughout the world today, microarrays are plates (usually glass) onto which complementary DNA (cDNA) probes are deposited by high-speed robotized printing methods. They are very well adapted to analyzing the expression of up to 10,000 genes derived from sequencing projects (Figure 7.3).

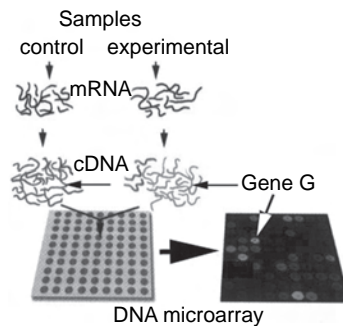


Figure 7.3 A microarray. Complementary DNAs (cDNAs) prepared from two messenger RNA (mRNA) sources (for example, the undisturbed sample on the left and the stimulated sample on the right), are marked by two different fluorophores, one red and one green. The cDNA is then probed in a microarray bearing gene probes representative of the entire genome, for example, gene G. The probes themselves consist of cDNA that has been attached to the microarray.

Measurement is carried out by differential hybridization, in order to minimize errors due to variations in cDNA printing. mRNA from two different sources (for example, a drug-treated assay and an untreated control) is usually reverse-transcribed into cDNA and marked with either of two fluorophores, one for the assay and one for the control. Differential hybridization of the microarray probes is carried out with a mixture of both liquid phase cDNA preparations, thereby minimizing systematic errors (Figure 7.3). After hybridization and washing, each fluorescent signal is independently evaluated and used to calculate the ratio of the concentrations of the experimental and control nucleic acids. This ratio is generally used as the starting point for interpreting the results.

Such microarrays have already been used to measure the genetic expression levels of the complete *S. cerevisiae* genome (around 6,400 distinct cDNA sequences) during various kinds of treatment, such as transition from sugar- to ethanol-based metabolism, or sporulation, or throughout the entire cell cycle. Such data sets are available in the public domain. Guides explaining how to construct devices that deposit cDNA onto the surfaces of microarrays, as well as how to analyze fluorescence levels, can also be found on the Internet. In addition, pre-prepared microarrays are commercially available for an increasing number of organisms, including human, rat, mouse, plant (*Arabidopsis*), and some bacteria. However, as already mentioned, they are readily adaptable to a wide range of available cDNA probes corresponding to individual laboratory requirements.

Oligonucleotide chips

These chips are produced mainly by Affymetrix®, and consist of small silicon plates to which thousands of short oligonucleotides (20 nucleotides or more) are attached. The oligonucleotides are synthesized directly on the surface of the chip by photolithography and light-controlled chemical synthesis. Due to the combinatorial nature of the process, it is possible to probe a very large number of mRNA molecules simultaneously. Today, chips consist of as many as 200,000 different probes, usually including several for each mRNA molecule (Figure 7.4). Chips may contain different exons of an intronic gene or some perfect-pairing probes as well as a few mispaired probes with a single nucleotide mismatch.

However, the preparation and reading of oligonucleotide chips requires expensive equipment, and at present only commercially produced standard chips are available at an accessible price. This does not provide laboratories the opportunity to pose specific questions.

The future of the microarray/oligonucleotide chip industry would appear to lie in the development of synthetic probes that are longer than the earlier versions (but shorter than cDNA sequences) and that are attached after synthesis and verification. A disadvantage of the methods described above is their lack of sensitivity, which means that tens of thousands of cells have to be mixed in order



Figure 7.4 Portion of an oligonucleotide chip. Each fluorescent patch corresponds to a specific gene probe.

to obtain enough material for a microarray experiment. This can be a problem if only a small amount of tissue is available, e.g., from an isolated embryonic tissue. In addition, the response dynamics are diminished by auto-absorption of fluorescence, steric hindrance reducing access to the probe, etc. These problems are exacerbated by the cost of microarrays, which limits kinetic experiments, since each time point corresponds to the consumption of one chip. A factor of two is usually estimated to be the practical limit of resolution for massively parallel quantification; therefore two values must be separated by a factor greater than two in order to be sure they are different. Of course, this factor is only an order of magnitude, since it is a function, among others, of the absolute values involved. In any case, it imposes limits on the quantity of useful information that can be obtained from massively parallel measurements.

Reverse-transcription–polymerase chain reaction

In order to evaluate gene expression using the reverse-transcription–polymerase chain reaction (RT–PCR), mRNA is first reverse-transcribed into cDNA, then amplified by PCR until detectable levels are reached. Using internal calibration techniques, it is possible to obtain exceptionally high levels of sensitivity (on the order of 1 molecule per microliter of sample volume) and dynamic range (6 to 8 orders of magnitude). This method requires using primers for all genes of

interest and, unlike the techniques described above, cannot be run in parallel. It is therefore crucial that the procedure be automated in order for it to function on a large scale. In practice, it is used to obtain more precise data on a small number of genes that have previously been identified as especially interesting.

Serial analysis of gene expression (SAGE)

Serial analysis of gene expression (SAGE) utilizes a very different technique to measure mRNA levels. First, cDNA is synthesized from mRNA; then a DNA tag long enough (~10–20 basepairs) to unambiguously identify a gene is cut from each cDNA fragment at a precise site. The tags are then concatenated into a long double-stranded DNA sequence, and the concatenated DNA is amplified and sequenced. If tag T1 is 10 times more frequent than tag T2 in the concatenated DNA sequence, it indicates that the mRNA containing T1 is 10 times more abundant than the mRNA containing T2.

The SAGE method has two advantages: (i) it is not necessary to know the mRNA sequence in advance, therefore allowing detection of unknown genes, and (ii) it utilizes a sequencing technology already commonly used in numerous laboratories. However, SAGE involves a somewhat complicated procedure and necessitates massive sequencing. SAGE has already been used to analyze the entire set of *S. cerevisiae* genes expressed during various phases of the cell cycle, as well as the expression of tens of thousands of human genes, comparing healthy and cancer cells.

Chromatin immunoprecipitation

A recently developed approach has permitted direct investigation on a genomic scale of the gene targets of proteins that regulate transcription by binding to DNA³ (Figure 7.5). During the first phase, a chemical fixative, such as formaldehyde, is added to a culture of living cells. The bivalent reagent rapidly enters the cells, where it forms covalent bonds between two chemical groups in close spatial proximity. In particular, it can form a solid bridge between a regulatory protein and its DNA binding site, if one happens to be bound at that instant. The cells are then ruptured and their DNA. The extracted and broken mechanically into random fragments of about 1 kb. DNA fragments associated with the protein of interest are then immunoprecipitated, using an antibody specifically directed against the protein. The second phase consists in deproteinization, after which PCR may be used to amplify the DNA; alternatively, the DNA can be

³ This method may also be used to study other DNA-binding proteins.

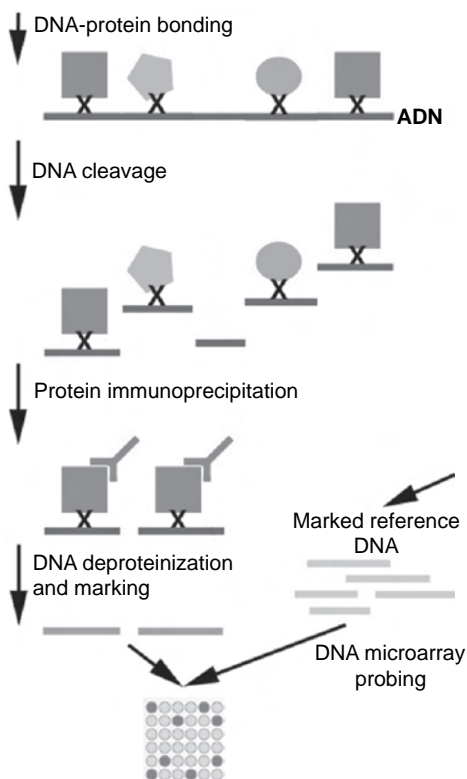


Figure 7.5 The ChIP–Chip technique. Proteins associated with DNA are covalently bonded by a bivalent reagent such as formaldehyde, which penetrates and kills living cells. The DNA is mechanically broken into random fragments of average length on the order of the size of a gene. A DNA-binding protein is then immunoprecipitated with specific antibodies. Fragments of the co-precipitated DNA are deproteinized and marked with a red fluorophore (for example). In parallel, DNA fragments representative of the entire genome are marked with a green fluorophore (for example). The red and green DNA fragments are then mixed, and the mixture used to probe a microarray or biochip that contains both gene and intergene regions of the organism being studied. After washing, the intensity of the two fluorescences is measured. For each microarray probe, the intensity ratio indicates the level of enrichment in protein-associated DNA.

identified by hybridization to a microarray or biochip. Ideally, the microarray should include probes that are representative of both gene and intergene regions. Since regulatory regions are usually intergenic, in principle, one would expect the precipitated DNA to hybridize with intergene rather than gene probes. The first phase is known as Chromatin ImmunoPrecipitation (ChIP). Since the second phase requires the use of a microchip, the whole method is called the ChIP–Chip technique.

However, this approach has its problems, especially those involving background noise due to low antibody specificity. These problems require the use of

relatively arbitrary thresholds to analyze the results. Nevertheless, as expected, the precipitated DNA preferentially hybridizes with the intergene probes, rather than with the gene probes. It also hybridizes more often with DNA regulatory regions that contain the motif recognized by the regulatory protein than with zones that do not. The results indicate that the number of regulated targets per sequence-specific regulatory protein is higher than previously believed based on classical genetics and biochemistry experiments. Thus, the yeast (*S. cerevisiae*) regulatory protein Rap1p has around 300 targets, which is around 5 percent of that organism's genes.

Bioinformatics

A protein that regulates transcription preferentially binds to DNA sites identified by a certain sequence or by a small subset of neighboring sequences (Figure 7.6). According to principles discussed in the preceding chapters, these potential sites may be detected using textual analysis of chromosome sequences. If the predicted position of such a site relative to the coding part of a gene matches the actual site, it is reasonable to assume that transcription of the gene is regulated by the DNA-binding protein. However, in practice, this approach encounters various difficulties, the most serious of which is that such sequences are usually very short and degenerate. The number of potential protein binding sites is therefore disproportionately large and identification of real sites is not based on solid criteria, except for eliminating sites that are located in coding regions, which are often as numerous as those located in regulatory zones. It is sometimes possible to obtain better results by noting when these potential sites appear three or more times in rapid succession in the regulatory region of the same



Figure 7.6 Visual representation in the form of a logo of a consensus DNA site at which a regulatory protein is likely to bind. The motif presented in this example is 12 nucleotides long. Only one DNA strand is shown. The complete data could be represented in the form of a 4-row (ACGT), 12-column (positions) array giving the percentage of each nucleotide at each position calculated from a compilation of sequences shown to bind to the regulatory protein. The sum of the heights of the letters at all the alignment positions indicates the information content in bits. The relative sizes of the letters correspond to the frequency of the nucleotide at each position. For example, only A can be found at position 11; A is nearly always found at position 10; both G and A are found at position 6.

gene. Empirically, such repetition often corresponds to effective regulation, which may be demonstrated using a direct approach. In higher eukaryotes, the respective positioning of binding sites for different regulatory proteins may be used in a similar manner as a criterion for improving the quality of predictions.

The emerging practice is to use three approaches simultaneously to discriminate among significant interactions between genes. Genes are considered to be the probable targets of a regulatory protein if they simultaneously satisfy the following three conditions: i) the regulatory region contains at least one site that is recognized by the regulatory protein (bioinformatics); ii) the DNA co-precipitates with the regulatory protein (ChIP–Chip); iii) the transcription level is modified following a stimulus known to trigger a response involving the regulatory protein (kinetic experiment using a microarray or biochip).

7.3 Macromolecular networks

Most cellular processes are rooted in dynamic interactions among a great number of biological molecules that both implement and undergo regulation. This section covers such interactions involving macromolecules. Distinctions will be made among interactions between proteins, between an enzyme protein and its substrate, and between a regulatory protein and DNA. Other types of interactions, which are potentially as relevant to the physiology of the organism but are not currently subject to massive genome-wide investigation, will not be covered here.

Before analyzing these networks separately, it is important to recall that they do not act independently in the cell. Biological information is usually described as flowing from DNA to RNA to protein to function (Figure 7.7, left). However, this view overlooks the fact that function emerges from a network of interactions among active macromolecules and small molecules, and only exceptionally from a single macromolecule. The network constitutes a filter between the isolated macromolecule and the function. This view also fails to take into account that the state of the protein interaction network feeds back onto the RNA state (affecting alternative splicing, for example). Likewise, the states of the protein and RNA networks feed back onto the DNA state (for example, the pattern of active and inactive genes) (Figure 7.7, right side).

7.3.1 Protein-protein interactions

When a large number of binary interactions between proteins are detected, an interaction map may be drafted. This map summarizes knowledge of the *protein interactome*, which is the set of all interactions among proteins in a cell. The

Information and its flow in

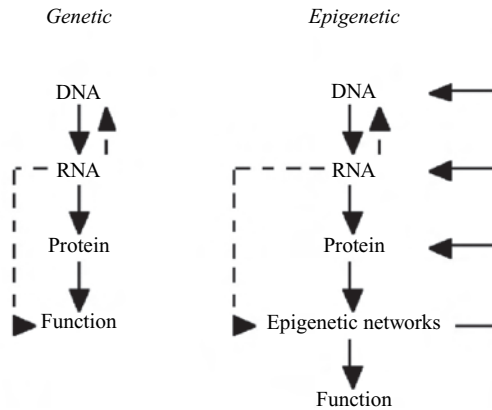


Figure 7.7 Genetic and epigenetic concepts of information flow in biology. In the all-genetic view, represented on the left, information flows from DNA to RNA to protein (except for reverse-transcription of RNA into DNA), and the bioactive macromolecule directly determines a function. In the epigenetic view, represented on the right, the network of interactions among bioactive molecules determines the biological functions. These networks feed back onto the state of the molecules represented higher up; for example, protein bioactivity, RNA alternative splicing, and transcriptional activity.

interactome is one component of what is known as the *proteome*. This map may be abstracted into a graph in which each node represents a protein and each edge indicates a non-directional interaction between two proteins⁴. As an example, Figure 7.8 presents a small portion of such a graph. When the technique employed permits, a number between 0 and 1 may be assigned to each edge in order to quantify the stability of the link. These quantitative and semi-quantitative aspects can strengthen purely topological considerations intended to distinguish subsets of proteins in the graph. The idea is that a subset (such as the one shaded in dark gray in Figure 7.8) has dense internal connectivity but few connections to other subsets (such as the one shaded in light gray).

7.3.2 Enzyme–substrate interaction

Interactions between enzymes and their substrates roughly correspond to what is known as *metabolism*, although some cases are borderline. For example, what if the substrate is a protein that the enzyme modifies irreversibly? The subject

⁴ It is tempting to orient links involving an asymmetric transitory interaction between two proteins, one of which is the substrate of the other (for example, when one protein is phosphorylated by another). This point is discussed below, precisely in the context of enzymatic reactions.

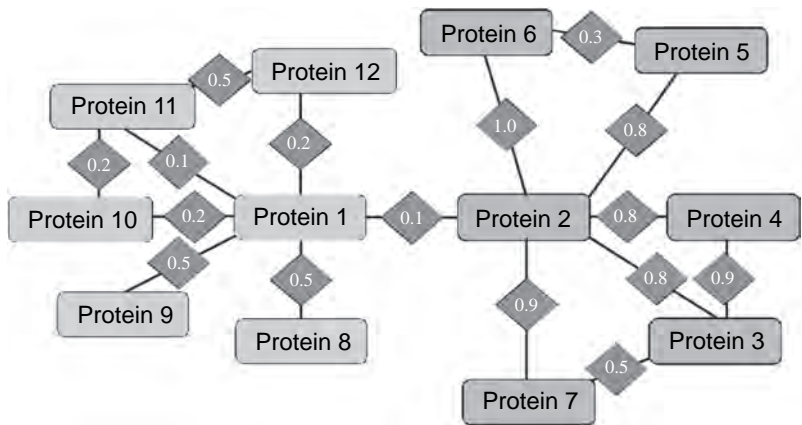


Figure 7.8 Protein–protein interaction map. In principle, links between proteins are non-directional; if protein A is associated with protein B, the reciprocal is also true. In certain cases, it is possible to assign weights to edges, represented here as a number between 0 (no link) and 1 (strong link). Such interaction charts may be obtained by means of either molecular biological or bioinformatics methods. The molecular methods are traditionally based on the co-purification of two polypeptides, indicating that they probably belong to the same complex. More recently, the two-hybrid approach, co-precipitation of protein complexes, and the use of protein chips have permitted more systematic experimentation. On the bioinformatics side, conservation of the proximity of two genes over major phylogenetic distances often indicates physical interaction between their products.

here is the *metabolome*, the set of metabolic pathways in a given organism. Metabolisms were essentially described several decades ago, using what are now considered classical biochemical and genetic approaches. These approaches are not the subject of this section, which will focus on first the local, then the global characteristics of metabolism.

Metabolic pathways

A metabolic pathway consists of a series of successive chemical reactions that transform a substrate into a product. Each reaction is catalyzed by an enzyme that is characterized by its catalytic properties (Figure 7.9). In classical cases, the effective catalytic rate is a hyperbolic function of the substrate concentration, involving an affinity parameter, K_m , and a maximum rate, V_{max} (see Insert 7.1, below).

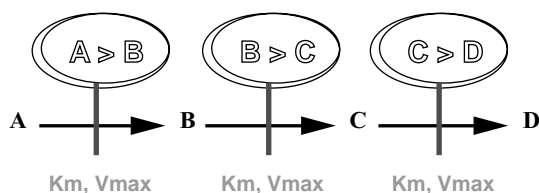
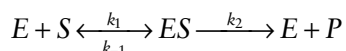


Figure 7.9 In this metabolic pathway, three successive chemical reactions transform one small molecule A into another D. Each reaction is catalyzed by a different enzyme. The first one, notated $A > B$, accelerates the transformation of substrate A into product B. Product B in turn serves as the substrate for enzyme $B > C$, which accelerates its transformation into product C, etc. Biochemists measure two parameters for each enzyme, purified separately: the Michaelis–Menten constant, related to the reciprocal of the affinity for its substrate (K_m), and the maximum rate of catalysis (V_{\max}).

Insert 7.1

The Michaelis–Menten Law (1913)

Given a chemical reaction that transforms substrate S into product P when catalyzed by enzyme E , it is possible to account for experimental observations by postulating the formation of a transitional complex ES between E and S . This complex then dissociates, recycling E and releasing P . The forward rate constants are k_1 and k_2 , and the reverse constants are k_{-1} and k_{-2} . The latter constant (k_{-2}) is neglected (see below).



Michaelis and Menten found a simple relationship between the catalysis rate V and the substrate concentration $[S]$, given three hypotheses. This relationship involves only two parameters, the Michaelis–Menten constant K_m and the maximum catalysis rate V_{\max} . These hypotheses in turn impose certain experimental constraints.

1) Initial rate conditions:

In practice, enzyme kinetics is measured only during the initial reaction period, when substrate consumption is a linear function of the reaction time. During this initial period, so little P is generated that the reverse reaction may be neglected. Concentrations appear in brackets.

$$v \text{ (synthesis of } P) = k_2[ES]$$

$$v_{-2} \text{ (disappearance of } P): \text{ negligible}$$

$$\nu_1 \text{ (synthesis of } ES) = k_1[E][S]$$

$$\nu_{-1} \text{ (consumption of } ES) = (k_{-1} + k_2)[ES]$$

2) Stationary conditions:

The first fractions of a second of the initial reaction period, during which the ES complex has not reached a constant concentration, are ignored. When the reaction becomes stationary; that is, when ES synthesis equals ES consumption:

$$\nu_1 = \nu_{-1}$$

$$k_1[E][S] = (k_{-1} + k_2)[ES]$$

$$[ES] = \frac{[E][S]}{K_m}$$

$$\text{Letting } K_m = \frac{k_{-1} + k_2}{k_1} \quad \text{Michaelis-Menten constant}$$

3) Enzyme dose and mass conservation:

The hypothesis is that the enzyme-catalyst is added in much lower concentration than the substrate. In addition, the masses of the enzyme and the substrate are conserved.

Let $[E]_T$ be the total concentration of the enzyme and $[S]_T$ that of the substrate (experimentally controlled quantities):

$$[E]_T = [E] + [ES]$$

$$[S]_T = [S] + [ES] \text{ and } [E]_T \ll [S]$$

$$[S] \approx [S]_T$$

$$\Rightarrow [ES] = \frac{([E]_T - [ES])[S]_T}{K_m}$$

$$\Rightarrow [ES] \frac{(1 + [S]_T)}{K_m} = \frac{[E]_T [S]_T}{K_m}$$

$$\Rightarrow [ES] = \frac{[E]_T [S]_T}{[S]_T + K_m}$$

$$\Rightarrow V = k_2 [ES] = k_2 [E]_T \frac{[S]_T}{[S]_T + K_m}$$

The special case of enzyme saturation:

If the substrate concentration is 'infinite' (in practice, very high), all the enzyme is complexed to the substrate and the reaction rate is maximum.

$$\begin{aligned} [ES] &\approx [E]_T && \text{maximum reaction rate} \\ V_{\max} &= k_2 [E]_T \end{aligned}$$

The relation sought may then be written as:

$$V = V_{\max} \frac{[S]_T}{[S]_T + K_m} \quad \text{Michaelis-Menten equation}$$

This hyperbolic relation can be linearized by double inversion:

$$1/V = 1/V_{\max} + K_m/V_{\max} \cdot 1/[S]_T \quad \text{Lineweaver-Burk equation}$$

It suffices to represent the reciprocal of the initial slope of the formation of P as a function of the reciprocal of the concentration of substrate added, varying the latter, then drawing a straight line through all the experimental points to calculate the values of the two parameters, K_m and V_{\max} (Figure 7.10).

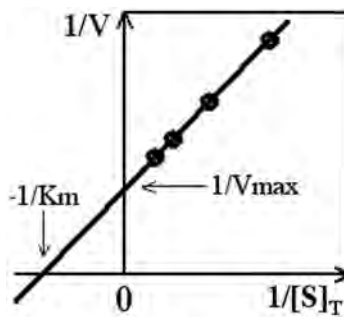


Figure 7.10 Linear relationship between the reciprocals of the measured reaction rate and the total substrate concentration, which may be experimentally varied. The two parameters of the Michaelis-Menten equation, K_m and V_{\max} , are, respectively, read directly on the abscissa at the ordinate at the origin. The slope is K_m/V_{\max} .

Remember that V_{\max} is the maximum reaction rate obtained for a given dose of enzyme acting on a substrate of infinite concentration. The K_m is the concentration of substrate that gives a catalysis rate one-half the V_{\max} . It represents the reciprocal of enzyme–substrate affinity.

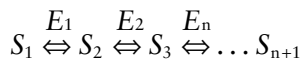
Regulation of cell metabolite concentrations

The stable metabolite at the end of a metabolic pathway (an unbranched linear chain) often has an inhibitory effect on the enzyme that catalyzes the first step of the pathway; this is called *negative feedback*. The concentration of the final metabolite is regulated by the sensitivity of the initial enzyme to its inhibition. Other, more specific, regulatory mechanisms exist, operating either by positive feedback or by connecting to two different pathways in competition with each other.

Ignoring these specific regulations, the metabolic network may be regarded as a set of interconnected pathways in which the first step of each pathway is regulated by feedback inhibition. Flux across this pathway would therefore be expected to depend strongly on the concentration of the rate-limiting enzyme. However, this is not the case. Indeed, comparison of two diploid organisms that possess respectively one and two copies of a gene that codes for an enzyme whose concentration is the limiting factor in a given metabolic pathway, (surprisingly) reveals their metabolic fluxes to be rather comparable. Artificially increasing the quantity of enzyme also has little effect on metabolic flux. A metabolic pathway is significantly affected only by total loss of enzymatic activity. Thus, negative feedback is not the principal way that cells regulate their metabolic pathways. Control is exercised in a more distributive fashion.

Distributive control of metabolic pathways

In 1973, Kacser and Burns studied the integrated kinetics of a linear sequence of reactions catalyzed by enzymes that convert one stable metabolite into another. They described how genes influence flow through a metabolic pathway as:



According to Haldane's modification of the Michaelis–Menten equation, the conversion rate of S_i into S_j may be expressed as:

$$v_i = \{V_i / K_{mi} [S_i - (S_j / K_i)]\} / \{1 + (S_i / K_{mi}) + (S_j / K_{mj})\} \quad (7.1)$$

in which v_i is the effective conversion rate, V_i is the maximum conversion rate (which depends on the enzyme concentration E_i and the rate constants); K_{mi} is the Michaelis constant for S_i ; K_{mj} is the Michaelis constant for S_j ; and K_i is the equilibrium constant for the $S_i \rightleftharpoons S_j$ reaction.

Most enzymes function under conditions in which their substrates are not saturating, such that $S_i \ll K_{mi}$ and $S_j \ll K_{mj}$. Thus, the value of the denominator in Equation 7.1 is approximately 1.

At equilibrium, all v_i rates become equal to the overall rate of the sequence of reactions. Let us call that overall rate 'Flux F ', and write the following series of corresponding equations:

$$[K_{m1} * F] / V_1 = S_1 - [S_2 / K_1]$$

$$[K_{m2} * F] / [V_2 * K_1] = [S_2 / K_1] - [S_3 / (K_1 * K_2)]$$

...

$$[K_{mn} * F] / [V_n * K_1 * K_2 \dots * K_{n-1}] = [S_n / K_1 * K_2 \dots * K_{n-1}] - [S_{n+1} / (K_1 * K_2 \dots * K_n)]$$

In summing all these elements, note that the terms on the right cancel out two-by-two, except (of course) for the first and last. This gives:

$$\begin{aligned} & F * [K_{m1} / V_1 + K_{m2} / (V_2 * K_1) \dots + K_{mn} / (V_n * K_1 * K_2 \dots * K_{n-1})] \\ &= S_1 - [S_{n+1} / (K_1 * K_2 \dots * K_n)] \end{aligned}$$

Metabolites S_1 and S_{n+1} are the molecules at the beginning and end of the reaction, whose concentrations are relatively constant. Thus, the term on the right side of the above equation is a constant, which we will call C_s .

Each V_i is the product of a rate constant K_i and the enzyme concentration E_i . These two parameters are genetically determined. Each term on the left side of the equation may be replaced by $1/e_i$, a composite term for the genetically determined parameters that contribute to the overall flux. The following simplified equation is obtained:

$$F = C_s / (1/e_1 + 1/e_2 \dots 1/e_n) \quad (7.2)$$

The essential point expressed by Equation 7.2 is that control of flux across a metabolic pathway does not occur in a single key step – the initial one, for example – but is distributed among all the enzymes in the pathway. It thus becomes possible to examine the dependence of F on the concentration of a single enzyme E_i , which could change due to inactivation of one of two cell copies of the gene that codes for E_i , for example. The sum of the terms in the denominator of equation (7.2) is a constant C_e for all the enzymes except E_i , thus:

$$F = Cs/(Ce + 1/e_i)$$

or even

$$F = Cs \times e_i / (1 + Ce \times e_i) \quad (7.3)$$

This hyperbolic relationship between F and e_i yields an initial slope Cs , then asymptotically attains the value Cs/Ce . Consider the case in which F is near the optimum, when there are two active genes coding E_i in a diploid organism. We see that with only one gene, and a halved E_1 concentration, F is little altered. The overall flux through the metabolic pathway weakly depends on each individual enzyme. This conclusion is supported by the experimental observation that enzymes function physiologically in the region of the hyperbolic curve near the asymptote, where the slope is shallow.

The metabolome

The entire set of interactions among enzymes and small molecules contained in a cell is known as the metabolome. It may be represented as a graph in which each node represents a small molecule and each edge corresponds to a chemical reaction catalyzed by an enzyme (Figure 7.11). Linear and unbranched metabolic pathways may be detected with the naked eye. Although going from a map to a graph is straightforward, it raises the issue of directionality. Since each chemical reaction is reversible, the graph must be non-directional. Whereas this is certainly true under conditions in which the reactions are likely to attain equilibrium, it is rarely the case in living cells, which function far from equilibrium. In practice, most cell reactions are drawn in one direction by coupling to a strongly exergonic reaction and by continuous consumption of the final product. For example, during protein synthesis, an amino acid is consumed as soon as it is produced. This is what the directed representation in Figure 7.9 is meant to convey. It is therefore justified to consider the network of metabolic interactions to be a directed graph. However, there are some exceptions, in which a pathway functions in one direction or the other, according to the circumstances. But even in this case, one direction is privileged at any given time.

7.3.3 Interactions between regulatory proteins and DNA regulatory regions

Regulatory proteins and DNA regulatory regions merit an introduction prior to discussing their interactions.

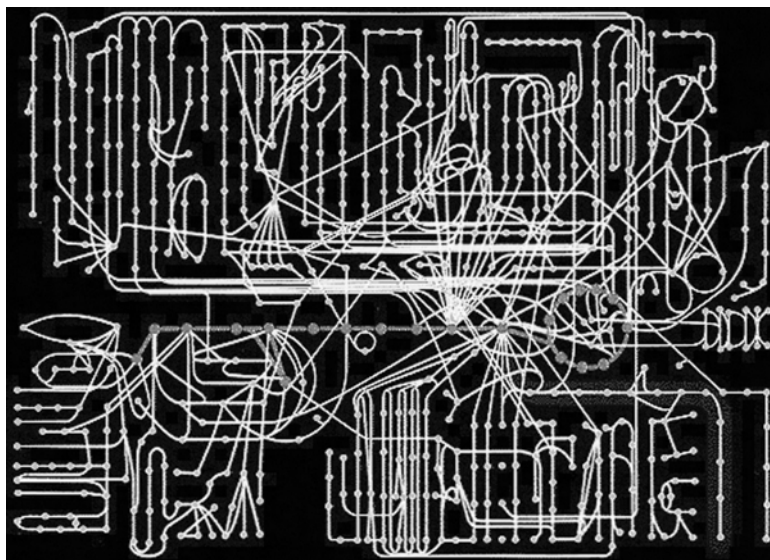


Figure 7.11 Metabolic network of a bacterial cell. Strictly speaking, this simplified map represents interactions between small substrate molecules (round nodes), each linked by catalyzed chemical reactions (light edges). Reciprocally, it would also be possible to represent the enzymes as vertices and the substrates/products of the reactions they catalyze as edges. The darker pathway corresponds to a major metabolic pathway, glycolysis (glucose breakdown), ending in the Krebs cycle, on the right.

Genes and DNA regulatory regions

According to one definition, a gene is an abstract entity bearing two kinds of information: the first is the code for the sequential assembly of a macromolecule; the second specifies the quantity of that macromolecule that is to be synthesized. The physical support of genes is always a nucleic acid polymer, usually DNA. The coding region of DNA bears the first piece of information and the regulatory region bears the second. If the coded macromolecule is a protein, the DNA coding zone is the open reading frame (ORF), and the DNA regulatory region is usually (often only, in microorganisms) located upstream from the coding region (see Gene A, on the left of Figure 7.12).

Regulatory proteins

When a regulatory protein binds to a DNA regulatory region, it modulates the expression of one or more genes. These genes are either grouped together and share the same DNA regulatory region, to which the regulatory protein binds

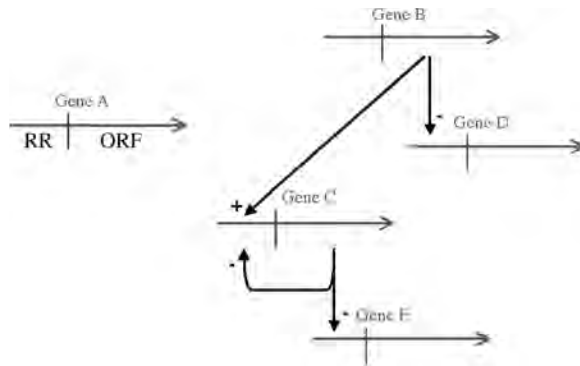


Figure 7.12 Interactions among some fictional genes. DNA, the material support of all genes, is symbolized in two parts: the regulatory region (RR) on the left and the coding region (ORF, or open reading frame, on the right). Gene A is neither regulatory nor regulated. Gene B is not regulated, but its product is a regulatory protein, which, when it binds to the regulatory regions of genes C and D, activates transcription of the former and inhibits transcription of the latter. The product of gene C inhibits transcription of gene E as well as of itself. In so doing, gene C constitutes a feedback circuit, or unary self-regulating loop. Gene C also participates in a genetic regulation pathway that links genes B and E, since it is both regulatory and regulated. In this figure, only B and C are regulatory genes, and only C, D, and E are regulated genes.

(bacterial operons, for example), or are dispersed along various points of the DNA molecule, where copies of the regulatory protein can bind.

What are the limits of the definition of a regulatory protein? We are interested here in proteins that regulate a subset of the genes of an organism. Other proteins also called regulatory play a more general role in initiating transcription (for example, most eukaryote transcription factors of type II). In principle, these other proteins, like RNA polymerase itself, are essential for the transcription of all genes that code for proteins. However, since their action is non-specific, they are not covered in this section. Nevertheless, the boundaries between ‘generalist’ and ‘dedicated’ regulatory proteins have recently become blurred. Indeed, the new post-genomic ChIP–Chip technique (see section 7.2.2.5) may be used to locate regulatory proteins at their DNA binding sites. This technique is sometimes used in combination with specific inactivation of each regulatory gene. Studies with yeasts have shown that certain dedicated regulatory proteins affect up to 5% of the genes of an organism, whereas certain generalists affect only 3%. Therefore, there is a quasi-continuum of regulatory proteins that covering the whole spectrum of genes in the genome. According to current knowledge, the threshold between generalist and dedicated regulatory proteins may be empirically established at around 3 to 5% of all genes. It is the dedicated regulatory proteins that are of interest here.

Genetic interactions

Regulatory proteins are coded by genes called ‘regulatory.’ ‘Regulated’ genes bear regulatory regions. Regulatory genes may themselves be regulated. It is customary to speak of interactions between regulatory and regulated genes, or ‘genetic interactions.’ However, this terminology poses a problem, since there is a fundamental asymmetry between regulated and regulatory genes. Regulatory genes exert their effects via their products, which are regulatory proteins. This implies that a graph of genetic interactions is basically of the directed type. Bearing that in mind, we will continue to use the customary terminology.

Genetic interaction map

At the qualitative level, a regulatory gene can activate (positive effect) or inhibit (negative effect) a regulated gene target. A dual effect is also sometimes observed, which may be either positive or negative, according to the circumstances. The target gene may itself be regulated, in which case it participates in a genetic regulatory pathway. If such a regulatory pathway is closed onto itself, it forms a feedback circuit. Since the incoming and outgoing connections of a gene can be multiple, the circuits and pathways are sometimes linked in a fully connected component of variable size. Some of these situations are represented in Figure 7.12, including the case of a gene that self-regulates, which is otherwise said to form a unary feedback circuit.

7.4 Topology of macromolecular networks

The set of molecular networks in a cell no doubt constitutes a heterogeneous web with respect to both its nodes and edges. If we emphasize the genetic network, then information rapidly flows from the patterns of genetic activity and through a cascade of intercellular and intracellular signaling functions, before slowly returning toward the regulation of gene expression. DNA regulatory and coding sequences thus unfold into spatiotemporal structures that define the organism. Other perspectives are possible, starting from and returning to protein (and possibly membrane) activity patterns (Figure 7.7). The challenge is therefore to identify significant connections in these regulatory networks and to determine the abstract principles underlying the architecture and dynamics of the network that allow it to function in a reliable and flexible manner.

The accumulation of data has made network architecture accessible. Expressed in the symbolism of graphs, network architecture consists in describing links (edges) that connect nodes (vertices), and eventually in the rules,

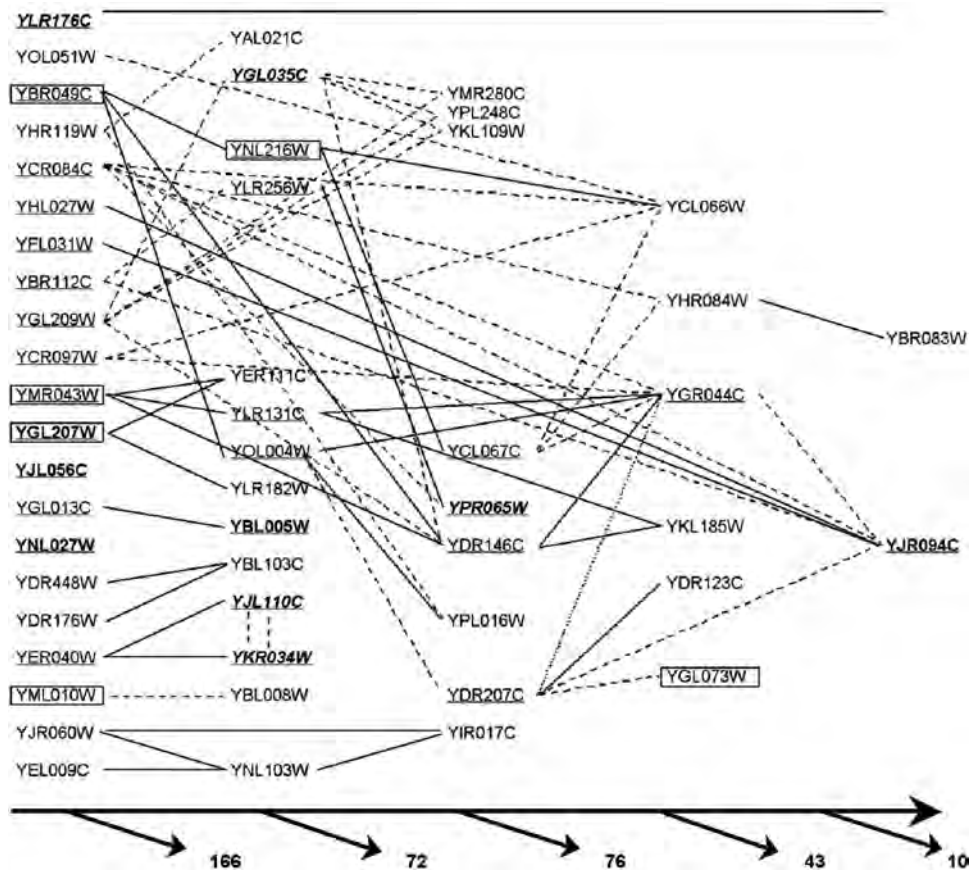


Figure 7.13 Partial map of genetic interactions in *Saccharomyces cerevisiae*. Genes are identified by the systematic names that were assigned to them during sequencing projects. For clarity, arrows are not used, but transcriptional influences go from left to right, as indicated by the thick arrow at the bottom. One-quarter of the 500 genes studied are regulatory, 52 of which are linked to at least one other regulatory gene. These 52 genes are represented here in a graph indicating causal relationships. In order to reflect the causal flow, genes that lack a known regulator (except for self-regulation) are listed in the left-hand column. The other genes are then placed in the left-most column, such that all their regulators are to their left. Respecting this rule, the numbers of non-regulatory genes regulated by the regulatory genes in each column are indicated on the bottom of the figure. For example (right column), YBR083W and YJR094C together regulate a total of 10 genes. Self-activation is indicated in bold, self-inhibition in bold italics, activation in solid lines, inhibition in dashed lines, dual regulation (one case) in dotted lines, and essential genes (whose knock-out is lethal) in frames. Genes belonging to the recurrent kernel are underlined. (A recurrent kernel is a set of genes with at least one direct or indirect target in the same group. According to this definition, a recurrent kernel includes feedback circuit genes and all genes that regulate them.)

functions, and weights that may be assigned to links. Often the only information available is whether or not a link exists, which is insufficient for modeling network dynamics. In addition, such dynamics would require introducing the notion of temporal delay; thus it often continues to escape us.

The following section presents some elements of global and local network topology analysis, which will then be separately applied to each homogenous macromolecular network. Only one, the genetic network, will be examined in detail. In doing so, we will bear in mind the reductionism to which this perspective subjects cell functioning, both by artificially separating networks and by disregarding the spatial organization of biological objects.

7.4.1 Topological analysis

Global topology

Empirical and theoretical results indicate that networks may be divided into two major categories, according to the connectivity distribution pk , which indicates the probability that a node is connected to k other nodes. The first category of networks is characterized by a pk that reaches a maximum at an average value k_{average} and that exponentially diminishes for higher values of k : $pk \sim C e^{-\beta k}$, where β and C are constants. In such exponential networks, each node has approximately the same number k_{average} of links. In the second category of networks, pk decreases according to a power law: $pk \sim C k^{-\gamma}$, where γ and C are constants. Thus, the distribution tail for high k is thicker, making the node population much less homogeneous than for the case of exponential distribution. There would be many nodes with few links and a small number of nodes with many links.

One property of a network is its average diameter, which is the minimum number of edges connecting any two nodes in the network, averaged over the set of all possible pairs⁵. Intuitively, the diameter of a network must have some impact on its dynamics. For example, information takes longer to flow through a large-diameter network. One of the expected properties of an inhomogeneous network that follows a power-law distribution that has been demonstrated by numerical simulation is that destruction of a randomly chosen vertex has little chance of modifying the diameter of the network, since most vertices are not highly interconnected. Such networks are said to be ‘robust’ with respect to accident. On the other hand, destruction of a well-chosen, strongly connected vertex – a hub – will significantly increase the diameter of the network. Such networks

⁵ It is also possible to use the maximum diameter, which is the minimum number of edges connecting the most distant pair.

are said to be ‘vulnerable’ to sabotage attack. A homogeneously distributed exponential network does not display these two characteristics.

Another general network property is the presence of a giant component, *i.e.*, a large sub-network in which a path connects any pair of vertices. The threshold size that the word ‘giant’ applies to is of little importance, since the giant component phenomenon corresponds to a sudden change of the network phase from fluid to solid. This frank jump in the number of vertices in the largest connected component marks the phase transition into a network that includes a giant component. In fact, the existence of a giant component, a global feature, if there is one, also depends on local topology.

Local topology

For a given average connectivity (the total number of edges divided by the total number of vertices), edges may be apportioned in extremely diverse ways. Independent of the type of overall distribution, edges may be distributed uniformly among vertices or display various degrees of local clustering. Figure 7.14 presents the case of increasingly stronger clustering, from left to right. While the edge distribution is rather uniform in the left panel, it is locally clustered in the center panel, as would be typical of ‘small worlds’. In the right panel, the clustering is so extreme that the network has broken down into small fragments not connected with each other, resulting in the loss of the connected giant com-

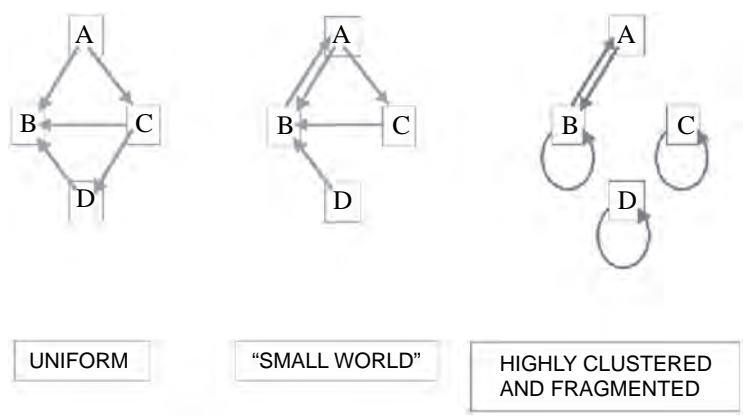


Figure 7.14 For the same number of vertices and edges, edge-per-vertex apportionment can vary from uniform to highly clustered. The example shown here, consisting of 4 vertices and 5 directed edges, is easy to visualize, but too small to be realistic. Distribution is rather uniform on the left and unequal in the center, with strong clustering around B and A. However, the graph is not fragmented. In a much larger network, this would correspond to a ‘small world’ (see text for a more detailed definition). On the right, excessive local clustering has fragmented the graph into three disconnected parts. The largest connected component is A–B.

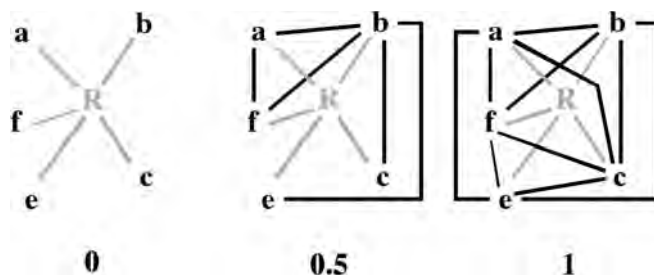


Figure 7.15 The *cliquishness* or clustering coefficient is a measure of local clustering. Taking node R as the reference, the clustering coefficient is the number of edges connecting its immediate neighbors a–f, yielding the maximum number of such edges. For N nodes, the maximum number is $[N(N-1)/2]$ for an undirected graph such as in this example, and $[N(N-1)]$ for a directed graph such as shown in Figure 7.14. Here $N = 5$; therefore the maximum number of undirected edges among neighboring nodes is 10. In the left panel, there are no edges connecting the neighbors of R to each other, so the clustering coefficient is 0. All 10 possible edges are present in the right panel, so the coefficient is 1; therefore it is a clique. The center panel has 5 edges, so the coefficient is $5/10$, and it is not a clique.

ponent. The formal criterion determining the presence of a ‘small world’ is reduced diameter, as in random networks, combined with strong local clustering, as in regular networks. In spite of such strong local clustering, a ‘small world’ still includes a connected giant component.

A clique is a set of vertices that are all interconnected. Local clustering is evaluated by the clustering coefficient, that is, by a number between 0 and 1, where 1 corresponds to a true clique. The clustering coefficient of a vertex taken as the reference is the ratio of the number of its actual connections to the number of possible connections (Figure 7.15).

7.4.2 The interactome

Maps of interactions among yeast proteins, which for the moment are quite incomplete and contain errors, generate undirected graphs whose connectivity distribution obeys a power law. The most heavily connected vertices usually correspond to products of essential genes whose inactivation is lethal. Much more complete data must be obtained before any definite conclusions can be drawn.

7.4.3 The metabolome

The data concerning metabolic pathways and their some 800 metabolites are relatively mature; it is therefore possible to draw some conclusions. If the metabolites are the vertices in an undirected graph and if the reactions connecting these molecules – each catalyzed by an enzyme – are the edges, the con-

nectivity distribution of the metabolic network obeys a power law for which (according to the author) the exponent varies between -1.5 and -3 . The results vary, particularly depending on whether or not very common small molecules, such as water and ATP, are included.

7.4.4 The genetic network

Data concerning the genetic network are obtained from classical genetic experiments, as well as from using the more exhaustive ChIP–Chip technique (Figure 7.5). For the moment, the genetic networks of *S. cerevisiae* and *E. coli* have been sufficiently investigated to allow some observations.

The graph of genetic interactions is *signed*; that is, each edge bears an interaction sign, positive for activation and negative for inhibition. It is also *directed*, for the molecular reasons discussed above. As a consequence, incoming and outgoing connectivities will be considered separately.

The distribution of incoming interactions obeys an exponential law whose exponent is -0.45 for *S. cerevisiae* and -1.2 for *E. coli* (Figure 7.16A). In practice, the shallower slope observed for the yeast indicates that the maximum number of different regulatory proteins that may regulate the same gene is higher than in bacteria. The average and maximum connectivities are, respectively, 2.3 and 13 for *S. cerevisiae*, and 2 and 6 for *E. coli*. This no doubt indicates the greater sophistication of the machinery regulating eukaryotic transcription.

Outgoing connectivity has no such limits; the total number of targets and the protein concentration are its only molecular limits. Outgoing connectivity does not obey an exponential law, but approaches a power law (Figure 7.16B). The exponent is around -1 for both organisms, indicating that the number of outgoing connections kp_k is distributed equally over k . This -1 value also corresponds to the phase transition of a generalized random graph. Average connectivity is 8.3 for *S. cerevisiae* and 3 for *E. coli*. It is among the essential genes (framed in Figure 7.13), therefore the most sensitive to disruption, that those richest in direct and indirect targets are found.

In any case, in these networks we observe a small maximum diameter (6 steps; see Figure 7.13), general fragmentation, and very strong local clustering (case shown in Figure 7.14, right). The number of feedback circuits seems small, even if it is greatly superior to what would be predicted for a random graph constrained by the same empirical connectivity distributions. In *E. coli*, self-inhibition (unary negative circuits) widely predominates. In *S. cerevisiae* there is a slight excess of positive circuits, nearly all of which are self-activations (Figure 7.13). Positive circuits are implicated in various differentiation programs, of which there are more examples in the eukaryotic yeast than in *E. coli*, which is a non-sporulating bacterium. The overwhelming predominance of the

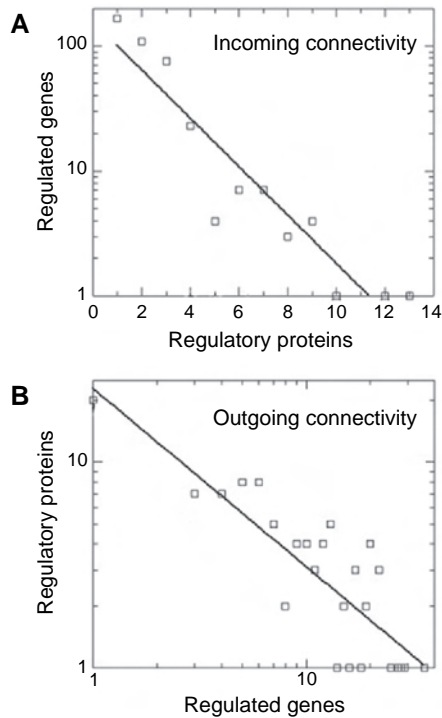


Figure 7.16 Connectivity of the genetic regulatory network of yeast. This network includes 500 genes and 900 interactions, a small number of which are represented in Figure 7.13. (A) Incoming connectivity (semilog plot): the number of regulatory proteins per regulated gene follows an exponential distribution. (B) Outgoing connectivity (log-log plot): the number of regulated genes per regulatory protein approximates a power law distribution. Nil values have been discarded.

shortest possible circuits could reflect the savings in both stimulus response time and biosynthetic energy required at each step.

Finally, overall fragmentation of the genetic network could limit information crosstalk at the transcriptional level. It is also important to recall here that the number of feedback circuits would greatly increase and fragmentation would diminish, if the genetic network were thrust into the heart of the cell; that is, if it were in contact with other molecular networks.

7.5 Modularity and dynamics of macromolecular networks

In the preceding section, we saw that the remarkable progress of post-genomics leads to a map – now much closer to completeness – of the elements that constitute *Life*. This map should be used to understand the regulatory logic of living

organisms. However, that will be possible only if we know how to *read* the map, a task beyond the grasp of the unequipped human mind. Two complementary approaches are of great importance in trying to interpret this map. First, it seems that the implications of regulatory logic will be revealed only when new experiments are undertaken that combine laboratory bench and computer approaches. Second, it is important to shed light on the inherent modularity of the map in attempting to reduce the otherwise insoluble overall problem. The relevance of this modularity is discussed below.

7.5.1 Modularity – why and how

The modularity gamble

Partitioning a molecular network into sub-networks is of interest only under three conditions: First, such modules must be biologically relevant; for example, by underlining functionality. For instance, a module could include all the actors involved in the response to a hormone. If this condition is not satisfied, partitioning is just a mathematical game of no biological interest.

The second condition is that it must be possible to attribute a characteristic dynamics to a module. For example, a negative circuit (see below) could generate homeostatic behavior. If this condition is not satisfied, partitioning will not help achieve the objective of understanding the functioning of the whole.

The third condition is that it be possible for the modules to be self contained (or that when present as part of a larger network they retain their principal properties, especially their dynamics). For example, the negative circuit, which in isolation engenders homeostasis, must retain that property when placed in a wider context. When this condition is satisfied, modularity permits the merging of certain representations and facilitates comparison among organisms. Modularity loses much of its interest if this condition is not met.

Implementing modularity

It is possible to imagine two ways of partitioning. In both cases, the three conditions described above must be satisfied. *Constructively*, partitioning consists in selecting and assembling a small number of vertices into a sub-network. *Reductively*, a mathematical criterion, based for example on local connectivity, is applied to the whole network, so as to fragment it into sub-networks.

Statistical analysis can demonstrate the interest of the modules detected, using an analytic approach or numerical simulation to demonstrate that they are significantly over- or under-represented in a natural macromolecular network.

Note, however, that there is not necessarily a correlation between the natural representation of a module and its biological relevance.

Until now, the constructive approach, in conjunction with generally rudimentary statistical analysis, has been used more frequently.

Properties of a module

Let us again use the example of the negative circuit in Figure 7.17. This small module is characterized by its topology, e.g., a self-regulating vertex; by its dynamic property, e.g., homeostasis; and by its biological properties, e.g., stable regulation or oscillation. Its behavior may be investigated by numerical simulation or by *in vivo* simulation (see Chapter 8). In the next section, these properties will be examined for some examples of modules.



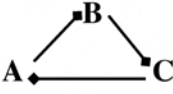

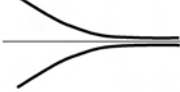

MODULES	FEEDBACK CIRCUITS		
SIGN	POSITIVE	NEGATIVE	NEGATIVE
# negative interactions	Even	Odd	Odd
Dynamic property	Multistationarity	Homeostasis	Oscillator
Biological property	Differentiation	Stable regulation	Stable regulation or oscillation
Topology			
Qualitative dynamics (<i>in vivo</i> and numerical simulations carried out)	Out (A) 		

Figure 7.17 Main properties of two families of feedback regulatory circuits. The two families are distinguished by the number of inhibitory interactions (negative interactions are represented by a square arrowhead) connecting a vertex to itself. The number of activating (positive) interactions present along the regulatory path is no more relevant than the total number of steps in the path; only the number of inhibitory steps counts. The positive circuit on the left consists of two inhibitory interactions: A self-activates via B. If A is high, it will remain so, and B will remain low, and vice-versa. The negative circuit in the center includes an inhibitory interaction: A self-inhibits. If A is high, it further self-inhibits, thus will diminish. If A is low, it self-inhibits less, thus will increase. The negative circuit on the right consists of three inhibitory interactions. If time delays are introduced for each step, and the products of A, B, and C are short-lived, the system may behave as an oscillator.

7.5.2 A sketch of module taxonomy

A few modules obtained constructively from directed graphs, which have been subjected to various degrees of investigation, are gathered here, along with an example of topology and some other properties, in particular, a probable dynamics.

Feedback circuits

A directed interaction pathway that is closed on itself constitutes a loop or feedback circuit. Interactions can be activating or inhibiting. Two types of feedback circuits may be identified (Figure 7.17): ‘positive’ circuits, which include an even number of inhibitory interactions, and ‘negative’ circuits, which include an odd number of inhibitory interactions. This terminology is justified by the fact that a vertex has an activating (positive) effect on itself in a positive circuit and an inhibitory (negative) effect on itself in a negative circuit. These two types of circuits have very different dynamic and biological properties.

A positive circuit can contribute to differentiation (‘multi-stationarity’; i.e., several possible stationary states), since if the concentration of the molecule that corresponds to node N increases, its formation will be further activated, whereas if its concentration diminishes, its formation will be less activated. Thus, as a first approximation, the dynamics tends toward either the maximum or minimum value, and any intermediate equilibrium is metastable. A number of positive circuits are found in natural genetic networks and their frequency seems to increase in proportion to the complexity of the developmental program of the organism involved.

A negative circuit contributes to homeostasis (parameter stability), since if the concentration of a molecule that corresponds to node N increases, its formation will be further inhibited, whereas if the concentration of the molecule decreases, its formation will be less inhibited; it therefore tends towards a stable equilibrium value. A negative circuit can also lead to more or less dampened oscillation, according to how the parameters for the interactions are set and to the half-lives of the molecules that correspond to the nodes. Intuitively, such half-lives must be short in order to produce oscillation. Unary negative circuits (auto-inhibition) are considerably over-represented in *E. coli*.

Regulatory triangles (‘feedforward loops’)

A ‘triangle’ in directed graphs is also known as a *feedforward loop*, and consists of an input vertex (‘In’) that influences a second vertex. These two vertices jointly influence an output vertex (‘Out’). The feedforward loop is said to be ‘coherent’ if the direct effect of the input vertex on the output vertex has the

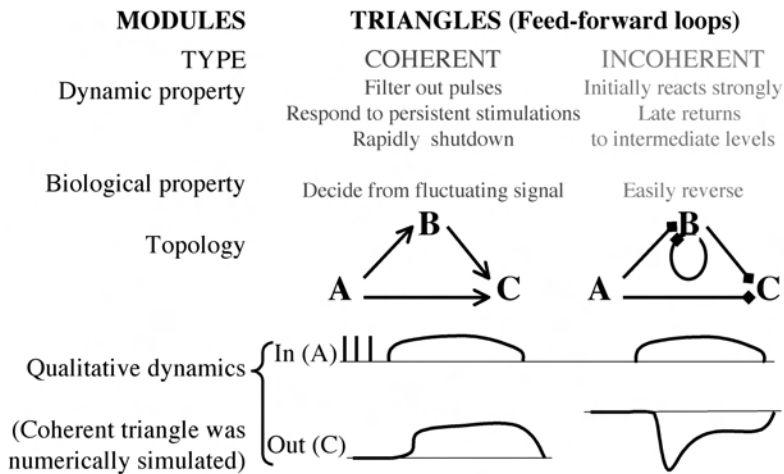


Figure 7.18 Main properties of two families of regulatory triangles ('feedforward loops'). Several topologies exist for each family. These two families are distinguished by the coherence of the action of A (Input) on C (Output), whether this action is via B or direct. For example, on the left, A activates C directly (bottom arrow) and indirectly via B (upper arrows); the action of A on C is coherent, and the resulting dynamics is rather easy to predict qualitatively. On the right, A inhibits C directly (bottom arrow), and activates C indirectly (upper arrows); the action of A on C is incoherent. To predict the resulting dynamics, some biological knowledge must be introduced (based here on some real cases, in which B also self-inhibits).

same sign (activating or inhibiting) as its net effect through the indirect path. If not, the loop is said to be 'incoherent' (Figure 7.18). Each of these two circuit families includes four possible topologies, with different dynamic and biological properties.

Coherent triangles of the type represented at the left of Figure 7.18 are found to be over-represented in genetic networks. If activation of C requires simultaneous activation of A and B (*A and B*), B must progressively accumulate under the effect of A in order to cross its threshold, finally allowing activation of C. Thus, this triangle filters the transients (which do not leave time for B to accumulate), responds only to persistent stimulation (which does allow B to accumulate), and quickly shuts down when A ceases to activate B and C. More generally, numerical simulations indicate that coherent triangles introduce a delay into the response when the signal goes either up or down.

The same approach suggests that incoherent triangles introduce acceleration into the response when the signal goes either up or down. While the incoherent triangle represented on the right in Figure 7.18 has been observed, it is infrequent. It also includes an additional interaction (B self-inhibits). These characteristics make any prediction of dynamic behavior difficult, unless the model is constrained by some prior biological knowledge.

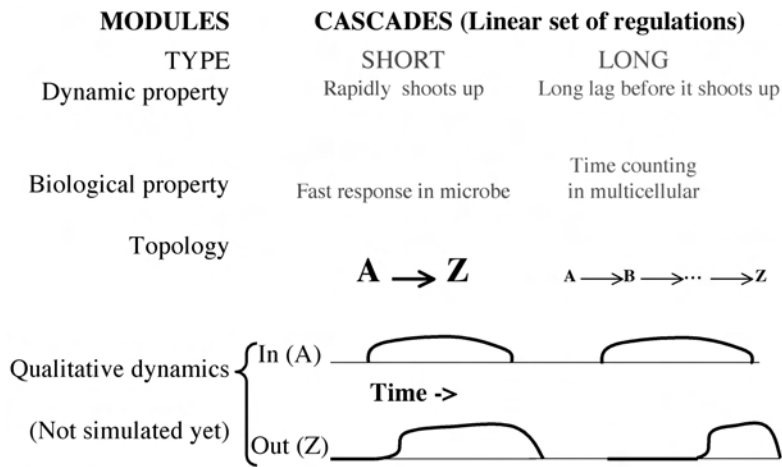


Figure 7.19 Main properties of regulatory cascades. The number of steps in a cascade, as well as the duration of each step, obviously determines the delay of the response to the initial stimulus. Note the predominance of short cascades in microbes and of long ones in multicellular organisms. This probably corresponds to the physiological requirements for rapid reaction in microbes and to long delays between successive developmental events in multicellular organisms.

Cascades

A cascade is a chain of vertices that influence one another sequentially. For the case in which each interaction introduces a non-negligible temporal delay (for example, the time necessary for biosynthesis in a genetic network), the delay introduced by the cascade is a function of its length, as well as of individual delays (Figure 7.19). It is also interesting to note that cascades are often short in microorganisms, for which a quick reaction to an external stimulus is essential. In contrast, cascades in multicellular organisms are often long, thereby introducing delays that the organism can use for its developmental program. This phenomenon is accentuated by the larger quantity of introns in multicellular organisms, which increase the time required for mRNA synthesis.

If several steps each introduce an amplification factor (for example, kinase cascades), the cascade permits strong amplification globally. If several interactions are cooperative, the bottom of the cascade responds in a quasi all-or-none manner.

Combinations of cascades and positive circuits

A developmental program consists of a series of irreversible steps, each of which takes a relatively precise length of time. One way to satisfy these constraints

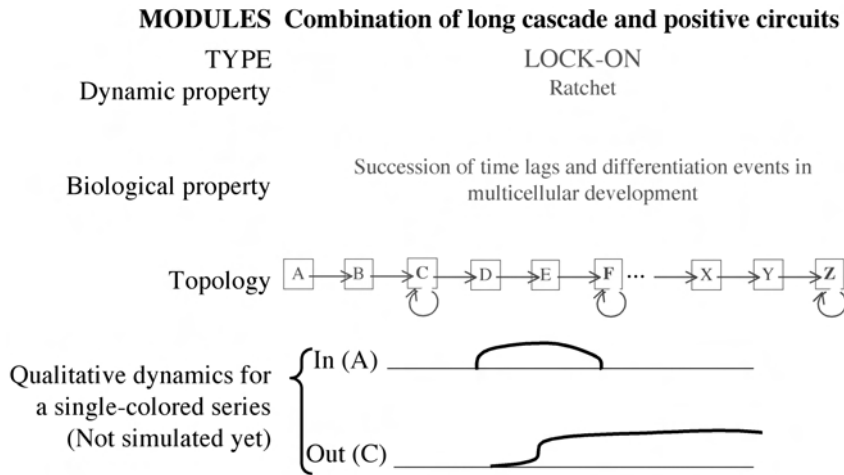


Figure 7.20 Principal properties of some long regulatory cascades combined with positive-feedback circuits. Each cascade, for example, A–C, introduces a long time delay, corresponding to preparation of the next developmental step. This developmental event is then locked on by a positive circuit, with no possibility of return (‘ratchet mechanism’), for example, self-activation of C. Successive developmental events in multicellular organisms therefore correspond to the successive triggering of C, then F, then Z.

would be to introduce a cascade that inserts a delay, followed by a positive circuit that irreversibly locks the mechanism. A linear suite of such mechanisms would implement the series of steps that constitute the entire program (Figure 7.20). Control points could be added to this simplified diagram, each of which would represent a prerequisite for the next key step. For example, attaining a minimum mass would be a condition for the next cell division.

Fans

A fan consists of a few upstream vertices (‘In’) that influence some downstream vertices (‘Out’) under closure conditions. All the incoming influences of the downstream vertices are in the fan as are, reciprocally, all the outgoing influences of the upstream vertices. At present, we can say nothing about the general case of fan dynamics; we are discussing only single-output and single-input modules.

A single-output module is a set of vertices that jointly and exclusively regulate a single output vertex (‘multigene regulation’), allowing, for example, fine regulation of genes by combining numerous inputs, each of which represents one aspect of the state of the cell. In known cases, the single-output module

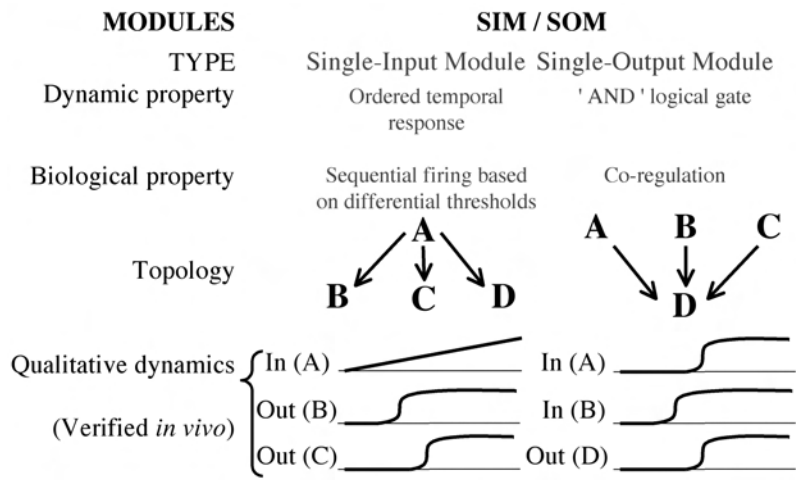


Figure 7.21 Main properties of two sub-families of fans. Only the single-input (SIM) and single-output modules (SOM) are represented. The SIM may provide a temporal expression program, according to an activation threshold hierarchy for regulated genes. When the single regulator A rises or falls, gene B, which is sensitive to the lowest threshold, will be activated first and deactivated last. Gene C, which is sensitive to the highest threshold, is the last activated and first deactivated. The SOM can provide an ‘AND’ logical gate. The single target, D, is triggered only after all A–C regulators are active.

implements an ‘AND’ logical gate; that is, all the regulators must be present and activating in order for the output to be activated (Figure 7.21).

A single-input module includes a vertex that regulates a set of exiting vertices with no other input (‘pleiotropic regulation’). Single-input modules have been found to be under-represented in genetic networks. In some cases it has been possible to demonstrate that a single-input module permits time-staged triggering of exiting genes. It is only necessary that the dose of the single regulator increase over time and that the regulated genes respond to different doses of the regulator. The most sensitive gene then triggers the early events in the global response to the common regulator, whereas the least sensitive gene triggers the late events (Figure 7.21).

7.6 Inference of regulatory networks

Inference consists in learning from a model of the relations among variables, starting from observations of those variables. The variables here are genes and proteins, and the objective is to infer maps of their molecular interactions, starting from post-genomic measurements.

7.6.1 The data

Post-genomic data obtained from kinetic experiments may be considered to contain the information required for inferring the underlying network. However, two kinds of difficulties lie in the path of this inference. On one hand, experimentally measured parameters reveal intrinsic variability and dispersion, since the observations almost always concern a population, not an individual. On the other hand, experimental options, which are often constrained by practical considerations, engender extrinsic variability (due to measurement), as well as data whose structure may not be well adapted to inference.

More generally, a typical set of post-genomic data concerns thousands of variables, either genes or proteins. However, even in the best of circumstances, this set includes only a few hundred experimental situations, thus a few hundred numerical values for each variable. Under these conditions, in which the model is under-determined by the experimental facts, one would expect the inference method to propose a model which, in ‘attempting’ to account for the facts, contains some false correlations.

These data can be analyzed using a variety of methods capable of inference at various levels, such as clustering analysis, correlation analysis, and mutual information content. Abstract computational models serve as the basis for developing these inference techniques. Such models are required in order to link the dynamic behavior of variables (trajectory, attractor) to a specific network topology.

7.6.2 Models

In this reverse-engineering work, the choice of a modeling formalism is, of course, crucial. Since modeling will be described in Chapter 8, only a few models used in inference will be mentioned here, without details.

The first models used were Boolean; that is, based on variables that could take on only one of two values. To infer a Boolean network of N potentially completely connected genes theoretically requires measuring 2^N pairs of inputs/outputs, which is obviously inconceivable. Assuming that the genes do not have more than k inputs from other genes, the quantity of independent experimental data required becomes proportional to $2^k \log N$. In practice, algorithms proposed that are based on a Boolean model function correctly only with artificial data.

Bayesian models have sometimes been successful in real situations. In the more general framework of machine learning, they allow deciphering the probabilistic structure of dependence among observed variables. Variants have been proposed whose training is more or less robust with respect to the paucity of data. For instance, it is possible to group variables with the same statistical

behavior (e.g., genes having the same regulatory inputs) in order to reduce the space of the models and parameters. This recent approach led to some predictions that have subsequently been validated at the laboratory bench. However, until now, nearly all attempts have taken a temporal series of N experimental points to be N independent experimental situations. Temporal information is therefore lost in current approaches, except in the dynamic Bayesian framework.

7.6.3 Prior knowledge

The above-mentioned under-determination suggests that it would be useful to have supplementary data available to constrain the model. Such data could derive from prior knowledge of molecular interactions, or of the consensus binding sequences. It is also sometimes possible to convert prior knowledge obtained from higher organizational levels into constraints that are expressed in the same language as the model.

Constraining the model by using prior information concerning what is known or plausible from the biological point of view probably remains the best tool for tackling the curse of dimensionality! How to include this information in the inference process is the real art of the modeler.

Bibliography

- Atlan H. (1999). *La fin du 'tout génétique'*. INRA Éditions.
- Bonnet G., *et al.* (1999). Thermodynamic basis of the enhanced specificity of structured DNA probes. *Proc Natl Acad Sci USA* **96**: 6171–6176.
- Fields S., Song O. (1989). A novel genetic system to detect protein–protein interactions. *Nature* **340**: 240–246.
- Gavin A.-C., *et al.* (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415**: 141–147.
- Guelzim N., *et al.* (2002). Topological and casual structure of the yeast genetic network. *Nature Genetics* **31**: 60–63.
- Ho Y., *et al.* (2002). Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature* **415**: 180–183.
- Kacser H., Burns J.A. (1973). The control of flux. *Symp Soc Exp Biol* **27**: 65–104.
- Lee T.I., *et al.* (2002). Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* **298**: 799–804.
- Legrain P., *et al.* (2001). Protein–protein interaction maps: a lead towards cellular functions. *Trends Genet.* **17**: 346–352.
- Milo R., *et al.* (2002). Network motifs: simple building blocks of complex networks. *Science* **298**: 824–827.
- Perrin B.E., *et al.* (2003). Gene networks inference using dynamic Bayesian networks. *Bioinformatics* **19**: Suppl 2: II138–II148.

- Segal E., *et al.* (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics* **34**: 166–176.
- Thomas R., d'Ari R. (1990). *Biological Feedback* CRC Press, Boca Raton, FL.
- Zhu H., *et al.* (2001). Global analysis of protein activities using proteome chips. *Science* **293**: 2101–2106.

8

Simulation of biological processes in the genome context

The advent of the genomics era will strongly influence the field of modeling and simulation in biology. By massively producing data at the molecular level, genomics effectively pulls the center of gravity of experimental biology towards the ground level of the molecule. It is tempting to try to build cells – and even organisms – from their genomes and related molecular data, in order to rapidly and cheaply simulate their functions and dysfunctions. As we shall see, such attempts are doomed to failure if they omit knowledge that originates from the analysis of upper levels in living systems. One reason for the failure of the purely bottom-up approach is that it is based on the common misconception of the genome as by far the major cellular information store. Another reason is that, despite all the advertized abundance of postgenomic data, models and theories in biology are still largely under-determined by the available facts, except in carefully defined and small domains. In the end, this cornucopia of genomic data may be a boost to the field of modeling/simulation, not because it has filled the gaps in our knowledge, but rather because, compared with the previous situation, the gaps are so much smaller that it has become reasonable to hope that models will legitimately fill in some of these gaps, making simulations more fruitful than ever before.

One central problem in post-genomic biomedical research is to forge new tools and improve our capacity to anticipate a cellular or organismal phenotype, starting from the data generated by high-throughput biology: genomic DNA sequences (genome), RNA (transcriptome) and protein (proteome) concentrations, activities, localizations, and interactions. The typical approach towards this goal has been to try to establish statistical correlations between a given molecular polymorphism and an individual feature (Figure 8.1). However, these correlations have no validity outside the feature under scrutiny, and they do not entail any causal link. In contrast, simulation demands that causal links of general validity be established. The famous example of sickle cell anemia can serve to illustrate this point (Figure 8.2). The phenotype at the organism level

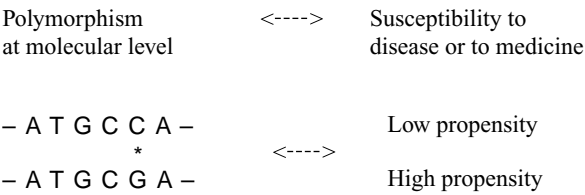


Figure 8.1 Statistical correlation between individual variation at the molecular level and an individual phenotype. As an example, a single nucleotide polymorphism (SNP) on the genome has been correlated with the propensity of the individual for diabetes. This correlation does not entail any causal link. A causality tree may later be built, either by costly and lengthy clinical or by laboratory bench work, or sometimes by simulation.

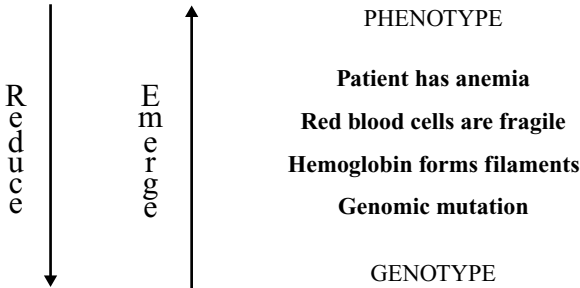


Figure 8.2 Causal analysis. In this instance, the phenotype includes anemia (organism level) caused by the brittleness of the red blood cells (cell level), due to abnormal formation of fibers by the hemoglobin (protein level), a consequence of a nucleotide substitution in the gene that encodes hemoglobin (genotype level). The genomicist receives abundant information, mainly at the molecular level, and must therefore learn how to connect the causal links in the opposite direction, on largely unknown ground.

(symptoms) has been reduced to a genotypic cause in several steps. In the opposite direction to this triumph of reductionism, and on largely unknown ground, the goal could be to re-establish a causal tree strongly rooted in the molecular data. In networks, straightforward causality is replaced by a ‘diluted’ causality that obviously constitutes a major difficulty in achieving this goal. It is, however, a highly desirable goal, since, in the long run, present users of statistical correlations (biotechnologists, clinicians, etc) would benefit more than anyone else from this causal and more generic approach that cuts the tremendous costs of benchwork.

8.1 Types of simulations

In biology, simulations have been employed in three distinct ways: *in vitro*, *in vivo*, and *in silicio*. It is important to bear in mind the necessity of validating the results of any of these simulations by observations conducted *in vivo*.

The oldest way consists in cell-free assays, which reproduce a certain function in the test tube (*in vitro*) with a mixture of chemicals and components prepared from live material. Although such assays are quite useful in determining the minimal set of components that supports the desired function, they will not be discussed further, since they fall outside the scope of this book.

The most recent way consists in implementing small genetic circuits comprising one, two, or three foreign genes in live cells (*in vivo*). The reporter of the network state is in all cases a fluorescent protein. Among the early attempts, the following functionalities have been tested: Homeostatic gene regulation by a self-inhibitory gene; a toggle switch made from two mutually inhibitory genes; an oscillator comprising three genes that inhibit each other in a circular permutation. Typically, this approach is preceded or accompanied by computational simulation in order to predict the behavior and tune the parameters of the biological circuit, yielding the desired features.

Computational (*in silicio*) simulations are the main subject of this chapter and will be developed further below.

8.2 Prediction and explanation

Before discussing some simulation approaches, let us illustrate them with a fictitious example that will take us from live cells all the way to simulation, while emphasizing predictivity and control (Figure 8.3).

Differential gene expression has been measured on a microarray in tumor versus normal cells. Let us assume that it is possible to infer the underlying genetic network from such experimental data (see Chapter 7). A portion of the resulting network can be used as a model for running a simulation. This simulation may, for instance, predict that the network can exist in either of two states, depending on the level of mitogenic stimulation of the main regulator 'REG'. The network state in a normal cell corresponds to its survival, while that in a tumor cell leads to a 'CDK'-mediated cell division. As this example implies, simulation can be useful to orient a costly laboratory bench or clinical experiment, or to attribute a probable function to a gene (annotation), and more generally, to generate a prediction that can be tested on live material. For instance, how would the genetic network re-equilibrate once its REG' node was inactivated by a drug (Figure 3, lower right)? Running a simulation might indicate that in the presence of a drug that inactivates REG' (but not REG''), division

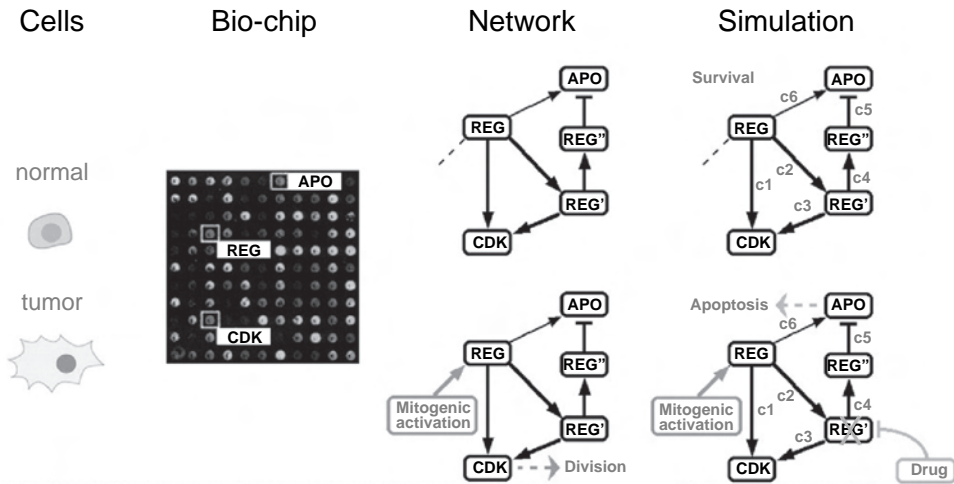


Figure 8.3 From biological sampling to simulation to control. Messenger RNAs (mRNAs) are extracted from normal or tumor cells. The ratio of their concentrations is measured on a biochip. The mRNAs transcribed from the 'REG' or 'CDK' genes are more abundant in the tumor cells than in normal cells, giving a red color (not observable on this B&W picture) at the corresponding spot. The opposite holds true for the 'APO' gene, giving a green spot. The other spots are yellow, indicating that the concentrations of the corresponding genes are similar in both samples. These partial results are compatible with the idea that 'REGulator' encodes a protein that activates the expression of the 'Cell Division Kinase' gene and inhibits that of the 'APOptosis' gene. By inference, the results obtained with the whole biochip may allow us to postulate a portion of a genetic network, such as the one shown on top, with REG' and REG'' encoding intermediate regulators (\rightarrow , activation; \dashv , inhibition). In this model, activation of REG by a mitogenic agent leads to hyperactivation of CDK, directly and via REG'. This activation results in cell division and tumor proliferation (bottom). In the absence of a mitogenic agent, division and apoptosis remain balanced and the cell survives without dividing (top). This equilibrium can be studied through simulation, e.g., by providing measured or calculated coefficients 'c' to the arrows that link genes (top). Similarly, in the presence of a drug that inactivates REG', the simulation may predict in which direction the network will re-equilibrate. In this simple example, it can readily be seen that REG' inactivation opposes division and relieves apoptosis inhibition (bottom). The prediction is that the tumor cell will therefore be killed by this drug. Please note that the simulation could have been based on a network model provided by a classical approach or by theoretical considerations, just as well as by inference from transcriptomic data (like here).

would be less favored than in an untreated tumor cell. Furthermore, apoptotic cell death (via APO), being no longer inhibited via REG' (or via REG''), would be directly triggered by REG, which is abundant in tumor cells. This prediction can then be tested on the bench.

Besides its predictive capacities, simulation may have the explanatory power to falsify or validate the coherence of a model. For instance, the growth of cell chains of the *Anabaena* microbe (Figure 8.4) has been simulated, using a rewrit-

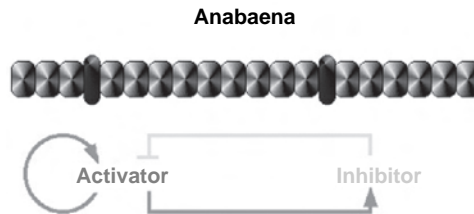


Figure 8.4 Model that explains the relative constancy of the distance between two successive heterocysts in a chain of *Anabaena* cells. This microorganism forms chains of cells (bright, wider) and differentiates into heterocysts (dark, narrower), present about every 10 cells. The simulation indicates that through the action of a couple of genes, it is possible to account for the relative constancy of the distance separating two successive heterocysts, with no critical dependence on the parameter values. It is both necessary and sufficient to assume that the activator induces the production of the inhibitor and of itself, while the inhibitor represses the production of the activator. Simulation demonstrates the coherence of the model.

ing approach called Lindenmayer's systems. The first modelling attempt included only an inhibitor of the differentiation into cells called heterocysts. It accounted well for the average distance of 10 undifferentiated cells between two heterocysts, but was hypersensitive to the actual parameter values. Addition of an activator coupled to the inhibitor by a specific regulatory circuit (Figure 8.4) made the model more robust (*i.e.*, no longer critically dependent on the parameter values). Independently, molecular biological studies later proved this model essentially correct. In essence, simulation was used as an investigative tool that demonstrated the incoherence of the first model. The expert consequently proposed the minimal organization that would re-establish the explanatory coherence of the model, as verified by running a simulation. The molecular structure that implemented the proposed organization was later discovered at the laboratory bench.

8.3 Simulation of molecular networks

The dynamic implications of the underlying logic of regulatory and metabolic networks cannot be deduced solely from laboratory experiments, in particular, because the molecular components are entangled in a complex web of interactions. Increasingly, formal models and computational simulations are required in conjunction with laboratory bench studies. This section reviews the formalisms that are commonly used to describe and study regulatory and metabolic networks. Although it is very interesting to simulate both types of networks in an integrated manner, which often requires the use of hybrid formalisms, here we will address one formalism at a time.

8.3.1 Graphs and their derivatives

A graph is a set of vertices and edges connecting some pairs of vertices. It is customary at all scales of biology to use graphs to represent interaction maps. At the supramolecular level, they have been used, for instance, to represent protein interaction maps, metabolic charts and genetic networks. At the supra-cellular level, they have traditionally been used to represent neural and immunological networks. One potential problem with pure graphs is that they describe a static relational topology, while data are sometimes more informative than just the presence or absence of a relation between vertices. Starting with graphs, it is possible to add various types of conditional, directional, and spatiotemporal information to the relations between vertices. For example:

- Biological networks may be directed, when the relation connects two objects that play different roles (e.g., transcriptional regulator/target) or undirected, in other cases (e.g., protein interactions).
- Vertices may be of the stoichiometric or catalytic type. Stoichiometric vertices describe resources that are consumed (e.g., a metabolic graph in which vertices denote metabolites). Catalytic vertices are recycled to the identical form (e.g., a metabolic chart in which vertices denote enzymes, or a transcriptional regulator/target map, in which the gene is still there after its product has activated – or has been regulated by – another gene).
- A function may be assigned to each edge that formally describes the interrelation between the vertices it connects, depending on the availability of specific knowledge about the interaction they represent. The minimal information is whether or not there is an edge. When available, more accurate information may be supplied in the form of a sign (activation or inhibition), or an amplification factor (how much more or less of this mRNA is produced per hour from the target gene when the regulator is present). Notions of space and time may also be introduced in the edge function, allowing more dynamic representation of the phenomena captured by the graph (e.g., when enough of the transcription factor has accumulated in the head of the embryo, that gene will be turned on, after a certain delay). Predicates may be embedded within the edge function, allowing some refinements (e.g., if the A gene is on and galactose is present, then the B gene is turned off). Such refined edge-assigned functions effectively provide information relevant to the dynamics of the graph, thereby allowing it to escape its inherent limitation of depicting a static topology.
- Graphs are not suitable for expressing interactions that involve more than two partners in an obligate fashion (e.g., three proteins assemble into a

stable complex, provided that they are all present simultaneously; two alone do not form a stable complex). The formalism of hypergraphs allows expression of non-binary interactions, and could be used in the above context. If computational efficiency requires using graphs, at the cost of a limited loss of information, a n -ary complex can be broken down into a set of all possible binary interactions between subcomponents (a ‘clique’, see Chapter 7).

8.3.2 Boolean modeling

An idealized model based on elementary mechanisms may sometimes capture the essence of a complex behavior. In a Boolean model, each gene may receive one or several inputs from other genes or from itself. Assuming a sigmoidal (highly cooperative) relation between input and output, a gene may be considered as a first approximation to be either *on* (1, transcribed) or *off* (0, untranscribed) (Figure 8.5). Time takes discrete values and all gene states are simultaneously computed at each time point. The output at time $t + 1$ is calculated from the input at time t according to Boolean functions.

Consider a simple Boolean network with three genes, each receiving inputs from the other two (Figure 8.6(a)). Gene A is an *AND* gate, meaning that genes B and C must both be active before A is activated at the next time point. Genes B and C are *OR* gates, meaning that they will be activated if one or the other of their inputs is on. With three genes that may take two values, the network can assume eight possible states, from (000) to (111). Reading from left to right, the table in Figure 8.6(b) shows, for each current state at time t , the next state of the genes, at time $t + 1$. For instance, starting from state (001) at $t = 0$ (second line), the system will move to state (010) at the next tick of the clock, at $t = 1$.

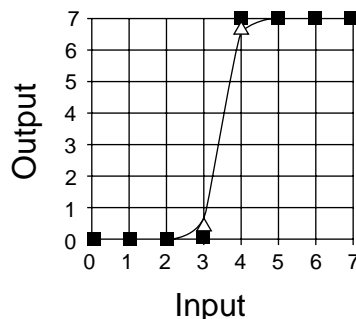


Figure 8.5 From continuous to discrete. Assume that the output is some sigmoidal function of the input (triangles), due to cooperative intermolecular interactions. The Boolean simplification replaces this curve with a step function (squares). Only two output states remain, 0 and 7.

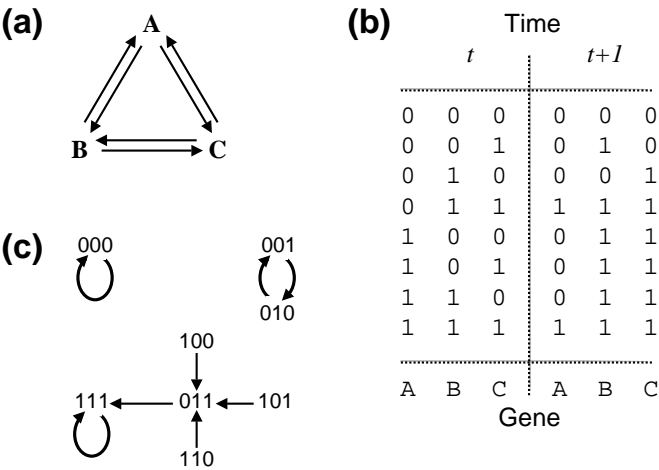


Figure 8.6 A small Boolean network. (a) The wiring diagram in a Boolean network with three genes (A, B and C), each an input to the other two. (b) The Boolean rules for the diagram shown in (a), assuming that gene A represents an *AND* gate, while genes B and C each represent an *OR* gate. Given three binary elements, there are eight (2^3) possible states at any given time t . This table shows the successor state at time $t + 1$ for each state at time t . (c) The state transition graph of the Boolean network depicted in (a) and (b). Each triplet of digits corresponds to a state for genes A–C, from left to right. State transition to a successor state is shown by an arrow. Three state cycles are observed, a point attractor (000), a state cycle (001 and 010), and a basin of attraction centered on (111).

This table therefore provides state transitions. Starting in one state, over time, the system will flow through some sequence of states. Such a sequence constitutes a trajectory. Given a finite number of states (eight here), the system will necessarily reach a state it has previously encountered. From that moment on, it will loop forever through the same cycle of states, called for this reason an *attractor*. This cycle of states may be limited to one single state, in which case it is called a *point attractor*. The collection of trajectories flowing into an attractor constitutes its *basin of attraction*. For example, we have seen that (001) moves to (010), and the third line indicates that (010) moves to (001). Therefore, the system oscillates between these two states which, together constitute a *state cycle* (Figure 8.6(c)). Its basin of attraction is limited to the state cycle, since no other state flows into either of those two. This is in contrast to state (111), a point attractor with a basin of attraction consisting of five states, including itself. For instance, state (101) moves to (011), which in turn moves to the steady state (111).

Boolean modeling has been used for network inference (see Chapter 7). It has also been used to study the global dynamic properties of large-scale regulatory systems, in particular genetic networks, given local rules that bear, for instance,

on the average degree of connection between genes. It is efficient, even for large genetic networks, at the expense of greatly simplifying assumptions regarding the absence of intermediate gene expression levels.

8.3.3 Generalized logical modeling

Transcriptomic data generally do not show extreme gene expression values, but rather intermediate ones. Although this observation may often reflect a mixture in varying proportions of cells that are each in different extreme states, careful studies conducted on small scales have suggested that at least some genes are expressed at more than two levels. More importantly, transcription factors likely have different thresholds for their different target genes. The generalized logical method in part corrects this problem by allowing logical variables to assume several discrete values. Here, a variable is an abstraction for the cellular concentration of one transcription factor. If a transcription factor encoded by gene A influences k genes, each with a different threshold, then the logical variable for A can take at most $k + 1$ values, one for each threshold and 0 if no threshold is crossed. In practice, A will have only a limited number of significant thresholds, and consequently the number of values A can take will often be smaller than k . State transitions are not necessarily synchronous for this formalism. Indeed, a synchronous step would sometimes entail jumping several thresholds at once, which cannot occur *in vivo* because the processes are continuous (e.g., protein accumulation). In logical networks, transitions are made more realistic by being desynchronized, *i.e.*, by jumping one threshold at a time. Furthermore, time delays, such as those arising from biosynthetic steps, can be taken into account.

Consider a regulatory graph consisting of two genes A and B that encode transcription factors, where A activates B and itself and B inhibits A (Figure 8.7(a)). Thus, A has two output links, corresponding to two thresholds in the most general case, so that A can take a value among 0, 1, and 2, while B is either 0 or 1. The asynchronous state graph in Figure 8.7(b) indicates possible trajectories in the space defined by all possible states for A and B. This allows computation of the next possible states. Note that this asynchronous state graph is only one among all possible state graphs and parameter sets that are compatible with the regulatory graph and associated knowledge, viewed in a static manner. Finally, by applying temporal logic, models that are consistent with the temporal properties of the system can be automatically extracted from the remaining set of all graphs that were compatible with the static view.

Generalized logical networks have been used to model various small regulatory systems, including developmental networks and bacterial genetic switches. In the case of developmental networks, it has been possible to manually introduce notions of compartmentalization. This approach seems to be an efficient

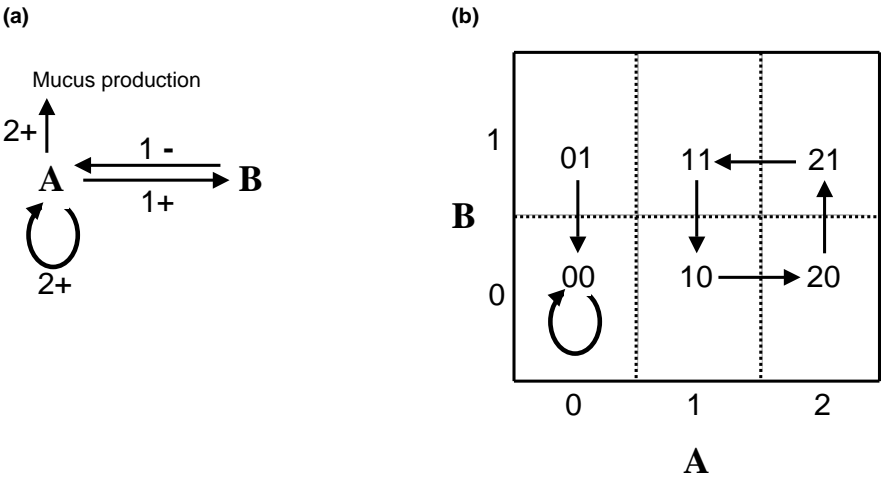


Figure 8.7 A small logical network. (a) The regulatory interactions for mucus production in the opportunistic pathogen *Pseudomonas aeruginosa*. Two genes, encoding an activator (A) and an inhibitor (B) of mucus production, are considered. Each edge in the graph is labeled with the rank number of the threshold, followed by the sign of its regulatory influence (–, inhibition; +, activation). Given parameters (not shown here), the dynamics may be deduced. (b) The asynchronous state graph. This graph is one among several graphs that would fulfill the constraints based on biological knowledge or hypotheses. A can take any value among {0, 1, 2}, and B among {0, 1}. Thresholds are represented by dashed lines, and transitions by arrows. The graph shows two steady states, one for A = 0, and one cycle: 11 → 10 → 20 → 21 → 11.

compromise between the wild simplifications of the Boolean model and the excessive parameter-dependence of the differential approaches. Moreover, it offers the possibility to exhaustively verify the temporal properties of a system, by taking advantage of the whole body of formal methods from computer science.

8.3.4 Petri nets

A Petri net typically allows the description and modeling of concurrent systems. Although so far they have mostly been used to model technological systems (seat reservations, communication protocols, etc.), they have also been proposed to describe and model biological networks, such as metabolic pathways or genetic networks. A Petri net consists of places, transitions, and arcs. A place contains tokens that may flow through arcs according to some general rules. An *arc* connects a place to a transition, and *vice versa*. A transition comprises incoming and outgoing arcs that connect to places (Figure 8.8). When a transition is trig-

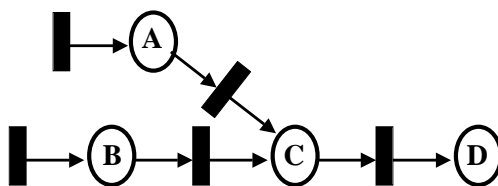


Figure 8.8 A Petri net representation of a set of regulatory interactions. Circles denote places identified by a letter, black rectangles are transitions and arrows are arcs. Only discrete elements are shown.

gered, a token is taken from each input place and one is added to each output place through the corresponding arcs. This is used to represent reactions that consume their substrates and produce new elements. A ‘test’ arc checks for the presence of a token in its source place but does not consume it. Therefore, it may be used to represent enzyme activity or gene regulation, since the enzyme or the gene is not consumed by the reaction.

Some extensions to the classical Petri net have become popular for biological applications, including the hybrid Petri net, which has two kinds of places and two kinds of transitions, discrete and continuous. The discrete places and transitions are defined above. A continuous place holds a real positive number. A continuous transition continuously fires at a rate determined by parameters assigned to the transitions in hybrid Petri nets, or to the places in hybrid dynamic nets, or to both, in hybrid functional Petri nets.

Petri nets have been employed to model a variety of small biological systems, including signaling and metabolic pathways and regulatory gene networks. It is a very natural formalism for stoichiometric networks, such as metabolic pathways, and has been adapted and extended to handle catalytic networks, such as regulatory systems. It presents the visual inconvenience of requiring a large number of symbols and links to represent a small network (Figure 8.8).

8.3.5 Bayesian networks

In Bayesian formalism, a chart of regulatory interactions is represented by a directed acyclic graph, *i.e.*, a graph with oriented edges deprived of circuits. In this graph, a vertex corresponds to a molecular entity, such as a gene or a protein, bearing a random variable representing the gene expression level or protein concentration (Figure 8.9(a)). A conditional probability distribution is defined for the variable of each vertex, given the variables of its direct inputs in the directed graph (Figure 8.9(b)). A joint probability distribution is finally defined from all conditional distributions (Figure 8.9(c)).

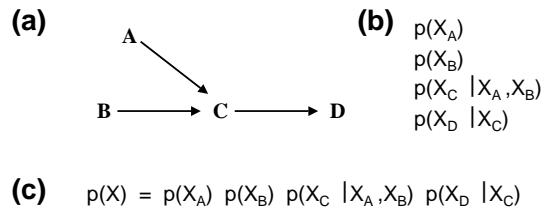


Figure 8.9 A Bayesian network. (a) In this directed acyclic graph, a vertex corresponds to a molecular entity such as a gene or a protein, and holds a random variable representing the gene expression level or the protein concentration. (b) A conditional probability distribution is defined for the variable of each vertex, given the variables of its direct inputs in the directed graph. (c) A joint probability distribution is finally defined from all conditional distributions.

As such, this formalism allows propagation of information within the model. Moreover, when data are available, it is possible to apply algorithms of statistical inference to estimate parameters of the conditional probability distributions, and to identify plausible structures. In this vein, Bayesian modeling has been successfully used for small network inference of microbial transcriptional interactions (see Chapter 7). This formalism is interesting because it is strongly anchored in statistics, and appears to be well-suited for handling noisy data. Furthermore, it can be used even under conditions of incomplete knowledge, and prior knowledge can be introduced. Bayesian formalism can be extended to time-dependent variables, thus allowing the inclusion of regulatory dynamics.

8.3.6 Ordinary differential equations

Using the widespread formalism of ordinary differential equations, molecular concentrations are represented by time-dependent variables. Regulation is modeled by expressing the rate of synthesis of a molecule as a function of the concentrations of all molecules, using rate equations of the form:

$$dx_i/dt = f_i(x),$$

or of the more complete form:

$$dx_i/dt = f_i(x; P) - \gamma_i x_i,$$

where i is an integer that denotes the molecule under consideration (between 1 and the number of molecules in the system); f_i is a function (non-linear in the general case); x is the vector storing all the molecular concentrations as real

positive numbers; P is a parameter. A degradation term of the form $-\gamma_i x_i$, where γ_i is a degradation constant, can be added to account for the exponential decay of the i^{th} molecule, which may result from true degradation, but also from dilution due to growth or diffusion. In this case, the equation represents a balance between synthesis and decay. Time delays can also be easily introduced into the function.

Analytical solutions are often impossible to reach for non-linear functions, and practitioners usually resort to numerical simulations that calculate approximate values for the variables at each successive time point. However, it is sometimes possible to analyze specific features of the dynamic system known as steady states and limit cycles. The robustness of these steady states or limit cycles to alteration of parameter (P) values may additionally be assessed using bifurcation analysis. Numerical simulations of non-linear ordinary differential equations have been used to study systems such as the regulatory switch of bacteriophage λ between host cell lysis and lysogenic growth, where the kinetic parameters are few and have been measured very carefully. A general difficulty with ordinary differential equations is that they rely on accurate knowledge of the numerical parameters, and this knowledge is seldom available at present, although this situation can only improve. In the absence of proper experimental measurements, e.g., in cell cycle models, parameter values can be chosen, using a manual or semi-automatic procedure, to fit the experimentally observed behavior. However, mere fitting does not guarantee that the parameters are right or that the numerical model is relevant to the biological situation. Furthermore, predictions following fitting are unsafe.

One popular non-linear function that accounts for real cases of sigmoidal response curves is the Hill function:

$$H^+(x_i; \theta_i; S) = x_i / (x_i + \theta_i)$$

for activation, and

$$H^- = 1 - H^+$$

for inhibition. The function H^+ ranges from 0 when x_i is null, to 1 when x_i tends to infinity. It equals 0.5 when $x_i = \theta_i$, hence θ_i is called the threshold of the regulatory influence of the i^{th} molecule on its targets (Figure 8.10). The Hill function admits a parameter S that reflects the steepness of the curve at $H = 0.5$. As S increases, so does the sigmoidicity of the curve, corresponding to increased cooperativity between interacting molecules. Extreme sigmoidicity brings us back to a step function, as discussed in the paragraph on Boolean networks (Figure 8.5).

One way to circumvent the analytical difficulties encountered with non-linear differential equations is to approximate them with a series of linear differential

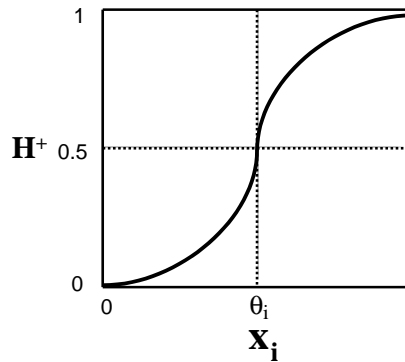


Figure 8.10 Hill activation function H^+ . θ_i is the threshold of the regulatory influence of the i^{th} molecule on its targets, for which $H = 0.5$. $S > 0$ is the steepness parameter. For $S > 1$, Hill curves show a sigmoidal shape, as shown on this graph. As S increases, so does the sigmoidicity of the curve.

equations, yielding ‘piecewise-linear differential equation’ models. For instance, the sigmoidal relation depicted in Figure 10 could be approximated by a step function. Often, this approximation does not change the qualitative properties of the solutions. Thus, piecewise-linear differential equation models stand between non-linear ones and logical models and have the advantage of strongly constraining local behavior in the phase space. They can be drawn even closer to logical models through qualitative analysis, at the expense of scalability. Qualitative analysis spots transitions that join different qualitative states through trajectories: a state transition graph can thus be elaborated, akin to a state graph in the generalized logical formalism. This approach is adapted to the usual lack of quantitative knowledge about regulatory mechanisms, but it cannot be readily scaled up, because qualitative constraints are often not available in sufficient number or strength.

8.3.7 Partial differential equations

So far, spatial aspects have not been considered, except in a rather superficial way, in a few applications. In all other cases, spatial homogeneity was assumed, or the formalism could not handle spatial aspects. However, simple considerations of how a cell functions, even a prokaryotic cell lacking any internal membranes, tell us that this assumption is wrong within a cell, not to mention the case of multicellular organisms.

Consider a set of n cells arranged in a row. Each cell is identical to all the others, and within each cell c ($1 < c < n$), gene expression is ruled by an identical rate equation such as that shown in the previous section, using some func-

tion f . Now consider a vector $x^c(t)$ holding the time-dependent concentration of molecules in cell c . Between two adjacent cells c and $c + 1$, diffusion of molecule i occurs proportionally to a diffusion constant d_i and to the concentration difference $x_i^{c+1} - x_i^c$. When n is large enough, the integer c is replaced by a real number c between 0 and C . Hence the partial differential equation:

$$\partial x_i / \partial t = f_i(x_i) + (d_i \partial x_i / \partial c^2).$$

This type of equation, and its close relative, the reaction–diffusion equation, have been extremely popular in studies of morphogenesis and pattern formation, especially in their two-molecule version, one activator and one inhibitor. Such approaches may be extended to cope with higher spatial dimensionality. However, the predictions made on the basis of partial differential and reaction–diffusion equations are generally sensitive to parameter values, boundary conditions, and domain shape. This is in stark contrast to the relative robustness of the developmental processes observed *in vivo*.

8.3.8 Stochastic equations

Bacteria may contain as few as 10 molecules of a given transcription factor, or one molecule of a given mRNA. It is thus questionable to assume, as has been done so far with differential approaches, that molecular concentrations vary continuously. It is equally questionable to neglect fluctuations (internal noise) in the timing of molecular processes and assume a perfect determinism, *i.e.*, that two identical systems starting from the same initial states will follow an identical trajectory. Accordingly, regulatory systems have also been modeled in a stochastic fashion, to account for the imperfect determinism, and in a discrete fashion, to account for the small number of molecules. One possibility with respect to the lack of determinism is to add a term to a rate equation that accounts for the noise in the system (Langevin's equation). Another possibility is to simulate step-by-step the temporal evolution of the system. In this latter case, stochasticity is introduced at the level of two variables, which represent the time interval between two successive steps, and the next reaction to occur (Gillespie's algorithm). Care is taken that the value distributions of these variables allow a large set of stochastic simulations to approximate, on average, the behavior of a so-called master equation, not discussed here. In this way, the master equation provides the average number of molecules and associated variances, while each stochastic simulation represents one individual trajectory.

Stochastic simulations have been notably applied to the developmental choice made by bacteriophage λ between host cell lysis and lysogenic growth. An interesting outcome of the observed fluctuations is that stochasticity may be one good way to account for phenotypic diversity, *i.e.*, the fact that different indi-

viduals in an apparently homogeneous population have different behaviors. In practice, the stochastic approach may yield realistic simulations, provided that the reaction mechanisms are known in great detail, at the cost of heavy computations, given the number of complex simulations to be run. When it is possible to expand either the time or the space scale, the phenomenon under study may be approximated by less costly deterministic models.

In conclusion, the choice among the formalisms discussed above must be based on careful consideration of their shortcomings and their strong points, given the problem and data, and the available computer power. Finally, it may be anticipated that the most successful approaches will involve the union of computational and bench experiments.

8.4 Generic post-genomic simulators

8.4.1 State of the art

A small number of highly funded, highly publicized, projects in the U.S.A. and in Japan propose simulation frameworks or platforms. To save the cost of developing new specific software for each model, these frameworks or platforms may in principle be generically applied to a variety of problems. Although they revolve around metabolism, they each have specific properties. Some, like Bio-Drive, provide for extracellular signaling and signal transduction through the cell membrane. Electronic-Cell (E-Cell) starts from a small set of genes encoding the minimal house-keeping metabolic functions, and implements the notions of energetic cost and protein degradation. Virtual-Cell (V-Cell) provides a computational framework that accommodates several formalisms, takes into account cell geometry and includes the notion of transmembrane flux, using it, for example, to simulate a calcium wave in the neuron. However, all these computational simulators are deeply rooted in a similar philosophy. Their starting point is a list of a few molecular components and their initial concentrations, and a set of reactions among these components. Generally, at each time point, a few differential equations are integrated and the list of concentrations is updated.

To illustrate one of the criticisms that can be made regarding all the existing generic simulators, let us consider the simple case of a cellular metabolon, *i.e.*, a complex of several enzymes that each catalyze one of the successive reactions of the same metabolic chain (Figure 8.11). Each enzyme of this complex has been purified separately by biochemists over the past decades, and their parameters measured in the test tube. These parameters are now fed into the simulators (Figure 8.11(a)). However, even the aqueous cellular compartments are highly organized. As has been proven in an increasing number of cases, cells exploit local concentration effects to allow for a sufficiently rapid metabolic

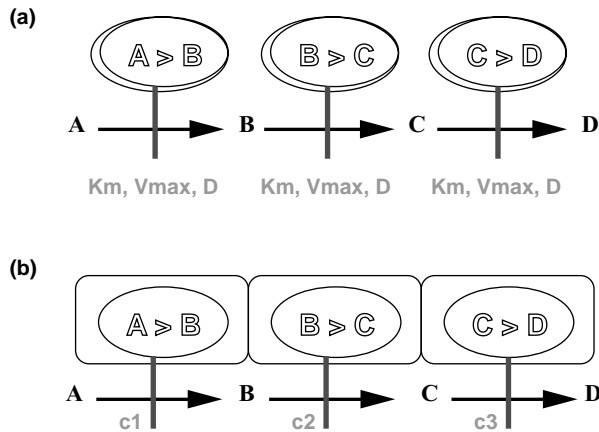


Figure 8.11 Metabolic chain and metabolon. In this short metabolic chain (a), three successive chemical reactions transform molecule A into molecule D. These three reactions are catalyzed by three enzymes. The first enzyme, named 'A > B', accelerates the transformation of substrate A into product B. Product B serves as a substrate for enzyme 'B > C' which accelerates its transformation into product C, etc., hence the notion of a 'chain'. This chain is part of the whole metabolism of the organism; it is a metabolic chain. The parameters of each enzyme have been measured separately following purification (affinity K_m , kinetic V_{max} , and diffusion D). Proper fulfilment of this set of chemical reactions could rely on either a) the chemical specificity of the interaction between an enzyme and its substrate (order based on thermodynamics), or b) spatial isolation that would prevent unwanted interactions (order based on localization). It appears that these two mechanisms are simultaneously active to varying degrees. In the absence of a membrane border, how can a metabolic chain be spatially isolated in an aqueous compartment? It suffices that the enzymes of this chain have a tendency to associate, either among themselves – dependent or not on the presence of their substrate – or to form fibers (a certain class of the cell inner skeleton). Numerous cases of multi-enzymatic complexes that process their substrates/products efficiently have already been described, including the glycolytic enzyme complex, a major and central metabolic path. For a metabolon, the most relevant parameter may be the coefficients (c) that relate to the fluxes traversing it.

pace. In particular, a metabolon maintains an elevated local concentration of the successive products of the reactions, or even *funnels* them in a channel in a few proven cases ('solid-state metabolism'; Figure 8.11(b)). At such high local concentrations, the enzymes turn over at full speed and diffusion times become negligible. Taking these facts into account would allow us to reduce the number of parameters, while obtaining fewer erroneous simulations. More generally, for the purpose of a realistic simulation, it is often an unacceptable first approximation to consider the cell as a stirred mini-reactor and to ignore the ubiquitous local concentration effects. Since the existing generic simulators use parameters obtained outside the organizing context of the cell in a purely bottom-up (from molecules up) approach, they have to introduce *ad hoc* para-

meters to fit the observations made on living cells. Hence, they lose any predictive value, retaining little explanatory value.

One way to overcome these problems would be to rely preferentially on data generated from live cells rather than *in vitro*. Along these lines, the potential of some *in vivo* approaches has not yet been fully evaluated. This is typically the case for metabolic flux measurements using either liquid chromatography or isotopic labeling and nuclear magnetic resonance, complemented by calculations based on, for example, flux balance analysis.

8.4.2 Problems and outlook

In a back-and-forth loop between laboratory bench and computer experimentation, metabolic simulations require three steps: 1) set-up of stoichiometric and kinetic equations; 2) experimental determination of the distribution of fluxes in a stationary state; 3) simulation of flux dynamics. It is clear that the present generation of post-genomic and generic simulators fall short of allowing this back-and-forth loop. We shall now briefly examine the ontogenic and epistemologic obstacles to fruitful simulations.

One of the major challenges ahead of us is to make the best use of the massive amount of molecular data generated by the tools of (post-)genomics. It is clear that the abundance of such data has increased tremendously in the recent past. Yet a closer look reveals that, with the exception of genome sequencing in most cases, the exhaustivity of such data is a 'sales pitch', and their quality is generally poor, although slow improvements can be foreseen. To take just one simple example, the lack of resolution of transcriptomic data is such that roughly speaking, one cannot distinguish values such as 1.0 and 1.9, or 1.9 and 3.8. In contrast, one can tell 1.0 from 3.8. Thus, the curve in Figure 8.12 should be

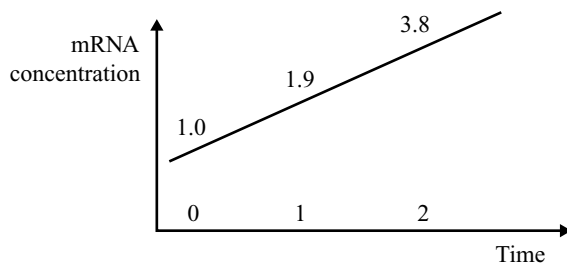


Figure 8.12 Low resolution of the numerical data generated using biochips. As a general rule, the resolution of these data is close to a factor of 2, *i.e.*, it is not possible to contend here that the first value is different from the second, or the second from the third. However, the first value is definitely different from the third one. This is just one example of the general problem that high-throughput biology produces non-exhaustive, non-quantitative data.

honestly described as: ‘This concentration increases in two hours’, which is a qualitative statement. For a long time to come, the lack of exhaustivity and the poor quality of the data will be a serious ontogenic obstacle to quantitative predictions. Besides, it would be wasteful to convert the few available quantitative data into qualitative results for the sake of format homogeneity. It will therefore be crucial to succeed in combining, in the same simulation, the use of quantitative and qualitative results.

In terms of prediction, the difficulty is to attain sufficient accuracy, so that the prediction is useful or testable. The number of variables in a biological system of interest is such that even small errors in their values may prevent the generation of a useful prediction. If, however, the purely quantitative approach turns out to be deceptive, other approaches described briefly in the previous sections may sometimes permit making useful predictions from qualitative or semi-quantitative results. Importantly, it often suffices to be able to give the evolutive trend of the final parameter, in order to determine the outcome.

Coming back to the example of the drug effect on the network (Figure 8.3, lower right), if apoptosis outweighs division in the tumor cell, cancer will regress, which is the single most important fact. This qualitative approach is common among biologists, as evidenced by the widespread use of ‘models’, little symbolic drawings, at the end of numerous primary publications. In other terms, qualitative reasoning is fundamental for the elaboration of knowledge in biology. More generally, and to cut down on costly laboratory benchwork, it would probably be a good idea to organize and improve the synergy between conventional and computational experimentations. The *Anabaena* case (Figure 8.4) pleads for such a synergy.

The temptation expressed in the introduction to this chapter; to build a cell and an organism from their genomes and related molecular data, implicitly relies on the concept that all the cellular information resides in the genome. This viewpoint is not supported by facts, yet is widespread, thus creating an epistemologic obstacle. An obvious counterexample is that two human cells endowed with an identical genome may exhibit extremely different phenotypes (compare, for instance, a stomach cell and a neuron). It is thus clear that part of the cellular information is held in its genetic material, while another part is distributed elsewhere in the cell. The genetic information is easier to identify, since it is contained mostly in one DNA molecule and is relatively accessible (in the form of the genome sequence). This may explain why the genetic part is so strongly emphasized. Another explanation may be traced back to the wrongly named ‘central dogma’ of molecular biology (Figure 7.7, left), according to which a gene encodes a one-dimensional protein sequence and the encoded protein folds into a unique three-dimensional structure to fulfill one function. Yet it has been known for a long time that several genes may contribute to the expression of one phenotypic character, and that a gene may participate in the expression of several characters. In addition, most proteins have the potential

to fold into more than one structure. A famous case is that of the prion protein, whose ‘abnormal’ conformation is transmitted to the descendants of its cellular host. This is a case of epigenetic heredity. The very fact that a stomach cell, by division, produces two stomach cells, is really another case, albeit non-pathological.

What is lacking in the scheme of the unidirectional flow of information to get closer to the epigenetic point of view (Figure 7.7, right)? Firstly, the notion of a network between the macromolecule and its function, which now results from the dynamics of several interacting molecules in the network, not from just one molecule. Secondly, the notion of feedback loops linking molecular types positioned at different levels in the flow of information, *i.e.*, regulatory proteins and DNA. The existence of such feedback loops tells us that a cell can transmit not only the sequence of a gene to its offspring, but also its activity level. These observations suggest that we should discard the naïvety of a purely genetic determinism, but certainly not replace it with an indeterminism that is not warranted by macroscopic observations (think of the robust developmental process of an animal). Instead, they suggest that we face up to an epigenetic determinism, more difficult to comprehend, which connects the constituents of the biological object under study in a complex web. Importantly, this scientific approach must integrate knowledge that originates from the analysis of upper levels in living material. Perhaps we should use the term ‘Epigenomics’, which, by analogy to ‘Epigenetics’, alludes to the bottom-up construction of biological objects at increasing levels of integration, while referring to the genome instead of the gene.

Bibliography

- Arkin A., *et al.* (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage Lambda-infected *Escherichia coli* cells. *Genetics* **149**: 1633–1648.
- Becskei A., Serrano L. (2000). Engineering stability in gene networks by autoregulation. *Nature* **405**: 590–593.
- Bernot G., *et al.* (2004). Application of formal methods to biological regulatory networks: extending Thomas’ asynchronous logical approach with temporal logic. *J Theor Biol* **229**: 339–347.
- De Jong H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol* **9**: 67–103.
- Doi A., *et al.* (2004). Constructing biological pathway models with hybrid functional Petri nets. *In Silico Biol* **4**: 271–291.
- Elowitz M.B., Leibler S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature* **403**: 335–338.
- Friedman N., *et al.* (2000). Using Bayesian networks to analyze expression data. *J Comput Biol* **7**: 601–620.

- Gardner T.S., *et al.* (2000). Construction of a genetic toggle switch in *Escherichia coli*. *Nature* **403**: 339–342.
- Gillespie D.T. (1977). Exact stochastic simulation of coupled chemical reactions. *J Phys Chem* **81**: 2340–2361.
- Glass L. (1975). Classification of biological networks by their qualitative dynamics. *J Theor Biol* **54**: 85–107.
- Hammel M., Prusinkiewicz P. (1996). Visualization of developmental processes by extrusion in space-time. *Proceedings of Graphics Interface '96*: 246–258.
- Kauffman S.A. (1993). *The origins of order: self-organization and selection in evolution*. Oxford University Press, New York.
- Kyoda K.M., *et al.* (2000). Construction of a generalized simulator for multi-cellular organisms and its application to SMAD signal transduction. *Pacific Symposium on Biocomputing* **4**: 317–328.
- Meinhardt H. (1982). *Models of biological pattern formation*. Academic Press, London.
- Murray J.D. (2003). *Mathematical Biology I & II, 3rd edition*. Springer, New York.
- Schaff J., *et al.* (1997). A general computational framework for modeling cellular structure and function. *Biophysical J* **73**: 1135–1146.
- Segal E., *et al.* (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics* **34**: 166–176.
- Thomas R., D'Ari R. (1990). *Biological Feedback*. CRC Press, Boca Raton, FL.
- Tomita M., *et al.* (1999). E-Cell: software environment for whole-cell simulation. *Bioinformatics* **15**: 72–84.
- Turing A.M. (1951) The chemical basis of morphogenesis. *Philos Transact Royal Soc B* **237**: 37–72.
- Tyson J.J. (1999). Models of cell cycle control in eukaryotes. *J Biotechnol* **71**: 239–244.

Index

- α -helices 156–7, 158–61
- alternating sequences 101, 105
- alternative splicing 88
- amino acid distribution 107–12
- androstosterone receptors
 - BLAST method 44, 47–50
 - dynamic programming 35–6
 - homologies 27–8
 - structure 26
- annealing principle 165–6
- APO genes 214
- automatic sequencing 1–4

- β -sheets 156–7, 158–61
- BACs *see* bacterial artificial chromosomes
- bacteria
 - chromosome copies 65–6
 - diversity and plasticity 63–6
 - gene evolution 75–81
 - genomic phylogeny 81–3
 - Haemophilus influenzae* genome 19–20
 - minimum gene set 72–4
 - number of replicons 65
 - pathogenicity islands 74
 - replicon geometry 65
 - replicon size 64
 - synteny 70–2, 76–8
 - therapeutic targets 74–5
- bacterial artificial chromosomes (BACs) 10–12, 18
- Bayesian models 207–8, 221–2

- bias 67
- BioDrive 226
- bioinformatics 176, 181–2
- BLAST method 43–50, 128
- block substitution matrices (BLOSUM) 30–3
 - BLAST method 44, 46–7
 - dynamic programming 38, 51
 - insertions and deletions 33
 - multiple alignments 51
- Boolean models 207, 217–19
- bulges (RNA structures) 142–3

- CAI *see* codon adaptation index
- carbodiimides 150
- cascades (macromolecular networks) 204–5
- cassette mechanisms 72
- causal analysis 211–12
- CDK genes 213–14
- cDNA *see* complementary DNA
- Centre d'étude du polymorphisme humain (CEPH) 10
- chemical probing 150–2
- chromatin immunoprecipitation (ChIP–Chip) 179–81, 182, 192
- chromosomes 61–6
- classical structures 134–6
- clone cell lines 6–8
- CLUSTAL (cluster alignment) program 54
- co-occurrence of genes 79–81

- coding levels 85
- coding phase analysis 114–16
- codon adaptation index (CAI) 116
- codons
 - biases 112–13
 - usage tables 110–11, 113
- coherent triangles 203
- collagenase 162
- comparative genomics 61–84
 - bacteria 63–6, 70–8, 81–3
 - bias 67
 - chromosomes 61–6
 - co-occurrence of genes 79–81
 - conservation of gene order 80–1
 - CpG islands 67
 - eukaryote genomes 61–3, 67, 73–4, 82–3
 - Fisher's hypothesis 77–8
 - function prediction 80–1
 - gene evolution 75–80
 - genome properties 61–7
 - genomic phylogeny 81–3
 - homologies 67–72, 81
 - horizontal transfer 76, 77
 - isochores 67
 - metabolic pathway conservation 79–80
 - minimum gene sets 72–4
 - non-orthologous gene displacement 69
 - orthologous genes 67–72, 75–8
 - paralogous genes 67–8
 - pathogenicity islands 74
 - plasmids 63–5
 - regulatory modifications 78
 - replication 65–6
 - selfish operons 77–8
 - synteny 68, 70–2, 76–8
 - therapeutic targets 74–5
- comparisons 25–59
 - BLAST method 43–50
 - comparison matrices 28–38, 44, 46–7, 51
 - confidence levels 46–50
 - deletions and insertions 33, 38, 51–2
 - diagonal strip method 39–41
 - dynamic programming 34–8, 53
 - fast heuristic methods 38–46
 - homologies 27–8, 36–7, 43–50
 - human androsterone receptors 26–8, 35–6, 44, 47–50
 - k*-tuples 40–5
 - maximum parsimony method 56
 - multiple alignments 50–8
 - phylogenetic trees 53, 54–8
 - profile alignment 52–5
 - progressive grouping 55–6
 - reverse genetics 50
 - sensitivity/specificity 46–50
- compensatory mutations 149
- complementary DNA (cDNA)
 - expressed sequence tags 20–2, 25, 128
 - microarrays 176–7
- concatenation 101, 105
- consensus sequences 13, 93
- conservation of gene order 80–1
- content searches 97
- context-dependent codon biases 112
- contigs 12–13, 18
- convergence (evolutionary) 67–8, 69
- core construction 163
- cosmids 10–12, 18
- CpG islands 67
- ddNTP *see* dideoxyribonucleotide
- degenerate patterns 100–1
- deletions 33
 - dynamic programming 38, 51–2
 - multiple alignments 51–2
- diagonal strip method 39–41
- dideoxyribonucleotide (ddNTP) method 1–4
- differential equations 222–5
- dinucleotides
 - phylogenetic analysis 150
 - RNA structure 133–4, 139–42
 - thermodynamic stability 139–42
 - triple helices 152–3
- divergence (evolutionary) 67–8, 69
- DNA-DNA hybridization 12

- duplication of genes 67–8, 69
- dynamic programming 34–8
 - multiple alignments 51–2, 53
 - predicting folding 166–7
 - protein structure 166–7
- Electronic Cell (E-Cell) 226
- electrophoresis 170–1
- endogenomes 75
- enzymatic probing 150–2
- enzyme saturation 187
- enzyme–substrate interactions 183–90
- epigenetic variations 72
- epigenomics 230
- ESTs *see* expressed sequence tags
- eukaryotes
 - genome size and structure 61–3
 - genomic phylogeny 82–3
 - isochores 67
 - minimum gene set 73–4
 - transcription 87–8
- exogenomes 75
- expressed sequence tags (ESTs) 20–2, 25, 128
- expression signals 87–91
 - codon biases 112–13, 116
 - fuzzy patterns 93–4
- fans (macromolecular networks) 205–6
- fast heuristic methods 38–46
- FASTA/FASTP 39–41
 - BLAST method 43–4
 - sensitivity/specificity 46
- feedback circuits 202
- feedforward loops 202–4
- finite state automata
 - pattern detection 98–100, 103–6
 - phylogenetic trees 58
 - regular expressions 103–6
- Fisher's hypothesis 77–8
- flavodoxin 159
- four-fluorophore technique 2–4
- fractionation 170–1
- fragmentation strategies 8–12
- Freier–Turner rules 140–3
- function prediction 80–1
- functional homologies 27–8, 67–8, 69
- gel electrophoresis 170–1
- gene evolution 75–80
- generalist bacteria 64
- generalized logical models 219–20
- generic post-genomic simulators 226–30
- Généthon 10–11
- genetic
 - code 85–7, 110–11
 - interactions 193–4
 - maps 15
 - marker identification 12
 - networks 198–9
- genome sequencing *see* sequencing
- genomic phylogeny 81–3
- GenScan 128
- Gillespie's algorithm 225
- Haemophilus influenzae* genome 19–20
- haploid bacteria 65–6
- hidden Markov processes 123–7
- homologies
 - comparative genomics 67–72, 81
 - comparisons 27–8, 36–7, 43–50
 - functional 27–8, 67–8, 69
 - protein structures 161–6
 - sequence 67–8, 69
- homopurine sequences 153
- homopyrimidine sequences 153
- Hoogsteen dinucleotides 152–3
- horizontal transfer 76, 77
- human androsterone receptors
 - BLAST method 44, 47–50
 - dynamic programming 35–6
 - homologies 27–8
 - structure 26
- human genome libraries 10–12
- hybridization 12
- hydrophobicity moment 160
- hyperstable tetraloops 143–4, 153–4
- hypochromicity 139

- immunoprecipitation 172–3, 179–81
- in vitro/in vivo* simulations 213
- incoherent triangles 203
- inference of regulatory networks 206–8
- insertions 33
 - dynamic programming 38, 51–2
 - multiple alignments 51–2
- Institute of Genome Research, The (TIGR) 19–20
- integration of genetic maps 15
- interactomes 182–3, 197
- intermediate mapping 11
- internal loops 148
- International Human Genome sequencing consortium 10–11
- intracellular localization 171–2
- inverted repeats 134–5
- isochores 67, 108
- iterative contig assembly 13
- k*-tuples 40–5
- Knuth-Morris-Pratt (KMP) method 100–1
- Langevin's equation 225
- large clones 18–19
- large-genome bacteria 64
- Lineweaver-Burk equation 187
- local homologies 36–7
- logical models 219–20
- loops
 - protein structures 163
 - RNA structures 142–3, 146, 148
- macromolecular networks 182–206
 - cascades 204–5
 - distributive control 188–90
 - dynamics 200–6
 - enzyme saturation 187
 - enzyme-substrate interactions 183–90
 - fans 205–6
 - feedback circuits 202
 - genetic interactions 193–4
 - genetic networks 198–9
 - global topologies 195–6
 - inference of regulatory networks 206–8
 - interactomes 197
 - Lineweaver-Burk equation 187
 - local topologies 196–7
 - metabolic pathways 183–90
 - metabolomes 184, 190, 197–8
 - Michaelis-Menten equation 185–7, 188–9
 - modularity 199–206
 - protein-protein interactions 182–3
 - regulation of cell metabolites 187–8
 - regulatory protein-DNA interactions 190–3
 - regulatory triangles 202–4
 - topologies 193–9
- Markov chains 117–20, 123–7
- maximal homology segments 45
- maximum parsimony method 56
- maximum scoring pair (MSP) 45
- messenger RNA (mRNA)
 - hyperstable tetraloops 143
 - proteomics 175
 - simulations 213–14
 - structure prediction 131–2, 143
 - transcriptomics 176–9
- metabolic
 - chains 226–8
 - pathways 79–80, 183–90
- metabolomes
 - enzyme-substrate interactions 184, 190
 - topology of macromolecular networks 197–8
- metabolons 226–8
- methylation 89–90, 93
- Michaelis-Menten equation 185–7, 188–9
- minimal cost pathway 14
- minimum gene sets 72–4
- modeling DNA sequences 116–27
 - complex models 120–3
 - hidden Markov processes 123–7
 - higher order biases 118
 - Markov chains 117–20, 123–7

- sequencing errors 123–7
- Viterbi algorithm 125–7
- see also* simulations
- modification signals 89–91
- molecular networks 215–26
 - Bayesian models 221–2
 - Boolean models 217–19
 - generalized logical models 219–20
 - graphical methods 216–17
 - ordinary differential equations 222–4
 - partial differential equations 224–5
 - Petri nets 220–1
 - stochastic equations 225–6
- mRNA *see* messenger RNA
- MSP *see* maximum scoring pair
- multicapillary sequencers 5
- multiple alignments 50–8
- multiple loops 148
- mutual information 150
-
- n -tuples 109–10
- nearest neighbor hypothesis 138
- Needleman & Wunsch algorithm 35–8
- non-deterministic automata 103–6
- non-homologies 37
- non-orthologous gene displacement 69
- nonsense codons 86
- nucleotide base distribution 107–12
- Nussinov algorithm 145–7
-
- oligonucleotide chips 177–8
- oligonucleotide primers
 - automatic sequencing 1, 3
 - sequencing strategies 4, 6–8
- open reading-frames (ORFs)
 - comparative genomics 82–3
 - genetic code 86–7
 - macromolecular networks 191
 - pattern detection 102
 - sequencing 22, 25
- optimal alignment 34–8, 51–2
- ordinary differential equations 222–4
- ORFs *see* open reading-frames
- orthologous genes 67–72, 75–8
- overlap identification 14
-
- palindromic sites
 - DNA sites 92–3
 - restriction enzymes 90
 - RNA sites 96
- PAM *see* probability of acceptable mutation
- paralogous genes 67–8
- partial differential equations 224–5
- pathogenicity islands 74
- pattern detection 96–106
 - degenerate patterns 100–1
 - finite state automata 98–100, 103–6
 - protein structures 158–60
 - regular expressions 101–6
 - simple searches 97–8
- PCR *see* polymerase chain reaction
- Petri nets 220–1
- phase analysis 114–16
- phase shifting 122–7
- phylogeny
 - genomic 81–3
 - phylogenetic trees 53, 54–8
 - profile alignment 53–5
 - secondary structure validation 149–50
- plasmids 63–5
- polymerase chain reaction (PCR) 15–16, 18–19
- post-genomics 169–82
 - bioinformatics 176, 181–2
 - chromatin immunoprecipitation 179–81, 182, 192
 - complementary DNA microarrays 176–7
 - inference of regulatory networks 206–8
 - information theory 169–70
 - intracellular localization 171–2
 - oligonucleotide chips 177–8
 - protein–protein interactions 172–6
 - proteomics 170–6
 - reverse-transcription–polymerase chain reaction 178–9
 - serial analysis of gene expression 179
 - simulations 226–30
 - transcriptomics 176–82

- PredictProtein neural network 161
- probability of acceptable mutation
 - (PAM) matrices 30–3
 - BLAST method 46
 - dynamic programming 38, 51
 - multiple alignments 51
- PRODOM database 57
- profile alignment 52–5
- progressive grouping 55–6
- prokaryotes 87–8
- promoter sequences 87–8
- PROSITE 58, 103
- Protein Data Bank 161
- proteins
 - α -helices 156–7, 158–61
 - amino acid distribution 107–12
 - β -sheets 156–7, 158–61
 - bioinformatics 176
 - core construction 163
 - covalent geometry 163
 - difficulties with predicting structures 155–8
 - dynamic programming 166–7
 - efficiency and limits 161
 - genetic interactions 193–4
 - homologies 161–6
 - interactomes 182–3
 - intracellular localization 171–2
 - loop construction 163
 - model refinement 163
 - non-covalent interactions 164
 - pattern recognition 158–60
 - preparative methods 170–1
 - protein chips 175
 - protein–protein interactions 172–6, 182–3
 - proteomics 170–6, 182–3
 - regulatory protein–DNA interactions 190–3
 - secondary structures 158–61
 - sidechain budding 163
 - statistical methods 160–1
 - structure prediction 155–67
 - systematic identification 175
 - thermodynamic stability 164–6
 - two-hybrid assays 173–5
- proteomics 170–6, 182–3
- pseudoknots 136–8, 150
- random fragmentation 8–10
- REG genes 213–14
- regular expressions 101–6
- regulation of gene expression 89, 92–3
- regulatory
 - modifications 78
 - protein–DNA interactions 190–3
 - triangles 202–4
- repetitive sequences
 - finite state automata 105
 - reconstruction 16–17
 - regular expressions 101, 105
- replication 65–6, 91
- replicons 61–6
- restriction enzymes
 - endonuclease sites 6–8
 - fuzzy patterns 93
 - palindromic sites 90
 - segmentation after mapping 12
- reverse genetics 50, 172
- reverse transcription 20–1
- reverse-Hoogsteen dinucleotides 152–3
- reverse-transcription–polymerase chain reaction (RT-PCR) 178–9
- ribonucleases 150–2
- ribonucleic acid (RNA)
 - bioinformatics 181–2
 - chemical/enzymatic probing 150–2
 - chromatin immunoprecipitation 179–81, 182, 192
 - classical structures 134–6
 - complementary DNA microarrays 176–7
 - dinucleotides 133–4, 139–42, 150, 152–3
 - empirical measurements 138–42
 - Freier–Turner rules 140–3
 - hyperstable tetraloops 143–4, 153–4

- limitations of predictive methods 148–9
- loops 142–3, 146, 148
- messenger RNA 131–2, 143, 176–82, 213–14
- molecular properties 132–4
- nearest neighbor hypothesis 138
- Nussinov algorithm 145–7
- oligonucleotide chips 177–8
- phylogenetic analysis 149–50
- pseudoknots 136–8, 150
- reverse-transcription–polymerase chain reaction 178–9
- ribosomal RNA 131, 143
- secondary structures 134–52
- serial analysis of gene expression 179
- structure prediction 131–55
- tertiary structures 135, 152–5
- tetraloop–receptor interactions 153–4
- thermodynamic stability 138–49
- topologies 144–5
- transcriptomics 176–82
- transfer RNA 112–13, 131
- triple helices 152–3
- true knots 136–7
- validation of predicted structures 149–50
- Zuker algorithm 147–8
- ribosomal RNA 131, 143
- RT–PCR *see* reverse-transcription–polymerase chain reaction
- SAGE *see* serial analysis of gene expression
- Sanger method 1–4
- secondary protein structures 158–61
- secondary RNA structures 134–52
- segmentation after mapping 10–12, 18
- self-splicing introns 155
- selfish operons 77–8
- sequence homologies 67–8, 69
- sequenced fragment assembly method 12–13
- sequencing 1–23
 - amino acid distribution 107–12
 - automatic 1–4
 - base triplets 120–1
 - coding levels 85
 - codon biases 112–13
 - comparison matrices 28–38, 44, 46–7, 51
 - comparisons 25–59
 - complementary DNA 20–2
 - complex models 120–3
 - complex sites 93
 - degenerate patterns 100–1
 - deletions and insertions 33, 38, 51–2
 - DNA sites 91–5
 - dynamic programming 34–8, 53
 - errors 123–7
 - expressed sequence tags 22
 - expression signals 87–91, 93–4
 - fast heuristic methods 38–46
 - filling gaps 9–10, 14–16, 18–19
 - finite state automata 98–100, 103–6
 - fragmentation strategies 8–12
 - genetic code 85–7, 110–11
 - genetic information 85–106
 - Haemophilus influenzae* genome 19–20
 - homologies 27–8, 36–7, 43–50
 - integration of genetic maps 15
 - large clones 18–19
 - modeling DNA sequences 116–20
 - multiple alignments 50–8
 - nucleotide base distribution 107–12
 - overlap identification 14
 - pattern detection methods 96–106
 - phase shifting 122–7
 - polymerase chain reaction 15–16, 18–19
 - prediction using biases 113–16
 - regular expressions 101–6
 - repetitive sequences 16–17
 - RNA sites 96
 - search procedures 127–8
 - segmentation after mapping 10–12, 18
 - sequence assembly 12–18

- statistical methods 107–29
 - strategies 4–8
 - unclonables 17–18
- serial analysis of gene expression (SAGE) 179
- serralsine 162
- shotgun method *see* random fragmentation
- sickle cell anemia 211–12
- sidechain budding 163
- simulations 211–31
 - Bayesian models 221–2
 - Boolean models 217–19
 - causal analysis 211–12
 - explanation 213–15
 - generalized logical models 219–20
 - graphical methods 216–17
 - in vitro/in vivo* 213
 - molecular networks 215–26
 - ordinary differential equations 222–4
 - partial differential equations 224–5
 - Petri nets 220–1
 - post-genomics 226–30
 - prediction 213–15
 - problems and outlook 228–30
 - stochastic equations 225–6
 - transcriptomics 219
 - types 213
- single nucleotide polymorphisms (SNPs) 211–12
- small nuclear ribonucleo-proteins (SNURPs) 88
- small-genome bacteria 64
- Smith & Waterman method 37–8
- SNPs *see* single nucleotide polymorphisms
- SNURPs *see* small nuclear ribonucleo-proteins
- specialist bacteria 64
- specialization islands 74
- speciation 67–8
- start codons 86
- statistical methods
 - amino acid distribution 107–12
 - base triplets 110–11, 120–1
 - coding phase analysis 114–16
 - codon biases 112–13
 - complex models 120–3
 - gene expression level 116
 - hidden Markov processes 123–7
 - higher order biases 118
 - Markov chains 117–20, 123–7
 - modeling DNA sequences 116–27
 - nucleotide base distribution 107–12
 - phase shifting 122–7
 - prediction using biases 113–16
 - search procedures 127–8
 - sequencing 107–29
 - sequencing errors 123–7
 - Viterbi algorithm 125–7
- stochastic equations 225–6
- structure prediction 131–67
 - chemical/enzymatic probing 150–2
 - dinucleotides 133–4, 139–42
 - proteins 155–67
 - ribonucleic acid 131–55
 - secondary protein structures 158–61
 - secondary RNA structures 134–8
 - tertiary RNA structures 135, 152–5
 - thermodynamic stability 138–49
 - validation 149–50
- sum of the pairs scores 51–2, 53
- SWISSPROT database 48, 58
- synteny 68, 70–2, 76–8
- terminal loops 148
- termination codons 86
- tertiary RNA structures 135, 152–5
- tetraloop-receptor interactions 153–4
- tetraloops 143–4
- therapeutic targets 74–5
- threading 166
- thymine 132–3
- TIGR *see* Institute of Genome Research, The tiling pathways 11
- transcription 87–8, 93
- transcriptomics 176–82
 - bioinformatics 181–2
 - cDNA microarrays 176–7

- chromatin immunoprecipitation 179–81
 - expressed sequence tags 21
 - generalized logical modeling 219
 - oligonucleotide chips 177–8
 - reverse-transcription-PCR 178–9
 - serial analysis of gene expression 179
 - transfer RNA (tRNA) 112–13, 131
 - translation 89, 113
 - triple helices 152–3
 - tRNA *see* transfer RNA
 - true knots 136–7
 - two-hybrid assays 173–5
 - ultrasound 8
 - unclonables 17–18
 - universal primers 6, 8
 - unweighted pairgroup method using the arithmetic mean (UPGMA) 55–6
 - uracil 132–3
 - V-Cell *see* Virtual Cell
 - vector clone sites 6–8
 - Virtual Cell (V-Cell) 226
 - viruses 61–2
 - Viterbi algorithm 125–7
 - Watson–Crick dinucleotides
 - phylogenetic analysis 150
 - RNA structure 133–4, 139–42
 - thermodynamic stability 139–42
 - triple helices 152–3
 - wobble pairings 134, 150
 - yeast artificial chromosomes (YACs) 10–12
 - yeasts 62–3, 73
 - Zuker algorithm 147–8
- Index compiled by Neil Manley*