INTRODUCTION TO COMPUTATIONAL MOLECULAR BIOLOGY

JOÃO SETUBAL and JOÃO MEIDANIS

University of Campinas, Brazil

PWS PUBLISHING COMPANY

I(T)P

An International Thomson Publishing Company

BOSTON • ALBANY • BONN • CINCINNATI • DETROIT • LONDON
MELBOURNE • MEXICO CITY • NEW YORK • PACIFIC GROVE • PARIS
SAN FRANCISCO • SINGAPORE • TOKYO • TORONTO

INTRODUCTION TO COMPUTATIONAL MOLECULAR BIOLOGY

PWS PUBLISHING COMPANY
20 Park Plaza, Boston, MA 02116–4324

I(T)P ™
International Thomson Publishing
The tradmark ITP is used under license.

Library of Congress Cataloging-in-Publication Data

Setubal, João Carlos.
Introduction to computational molecular biology / João Carlos Setubal, João Meidanis.
p. cm
Includes bibliographical references (p. 277) and index.
ISBN 0-534-95262-3
1. Molecular biology--Mathematics. I. Meidanis, João. II. Title.
QH506.S49 1997 96-44240
574.8'8'0151--dc20 CIP

Sponsoring Editor: *David Dietz*
Editorial Assistant: *Susan Garland*
Marketing Manager: *Nathan Wilbur*
Production Editor: *Andrea Goldman*
Manufacturing Buyer: *Andrew Christensen*
Composition: *SuperScript Typography*
Prepress: *Pure Imaging*
Cover Printer: *Coral Graphics*
Text Printer/Binder: *R. R. Donnelley & Sons Company/Crawfordsville*
Interior Designer: *Monique A. Calello*
Cover Designer: *Andrea Goldman*
Cover Art: *"Digital 1/0 Double Helix" by Steven Hunt. Used by permission of the artist.*

Printed and bound in the United States of America
97 98 99 00 — 10 9 8 7 6 5 4 3 2 1

For more information, contact:
PWS Publishing Company
20 Park Plaza
Boston, MA 02116

International Thomson Publishing Europe
Berkshire House 168-173
High Holborn
London WC1V 7AA
England

Thomas Nelson Australia
102 Dodds Street
South Melbourne, 3205
Victoria, Australia

Nelson Canada
1120 Birchmont Road
Scarborough, Ontario
Canada M1K 5G4

International Thomson Editores
Campos Eliseos 385, Piso 7
Col. Polanco
11560 Mexico D.F., Mexico

International Thomson Publishing GmbH
Königswinterer Strasse 418
53227 Bonn, Germany

International Thomson Publishing Asia
221 Henderson Road
#05-10 Henderson Building
Singapore 0315

International Thomson Publishing Japan
Hirakawacho Kyowa Building, 31
2-2-1 Hirakawacho
Chiyoda-ku, Tokyo 102
Japan

Contents

PREFACE

Biology easily has 500 years of exciting problems to work on.
— Donald E. Knuth

Ever since the structure of DNA was unraveled in 1953, molecular biology has witnessed tremendous advances. With the increase in our ability to manipulate biomolecular sequences, a huge amount of data has been and is being generated. The need to process the information that is pouring from laboratories all over the world, so that it can be of use to further scientific advance, has created entirely new problems that are interdisciplinary in nature. Scientists from the biological sciences are the creators and ultimate users of this data. However, due to sheer size and complexity, between creation and use the help of many other disciplines is required, in particular those from the mathematical and computing sciences. This need has created a new field, which goes by the general name of *computational molecular biology*.

In a very broad sense computational molecular biology consists of the development and use of mathematical and computer science techniques to help solve problems in molecular biology. A few examples will illustrate. Databases are needed to store all the information that is being generated. Several international sequence databases already exist, but scientists have recognized the need for new database models, given the specific requirements of molecular biology. For example, these databases should be able to record changes in our understanding of molecular sequences as we study them; current models are not suitable for this purpose. The understanding of molecular sequences in turn requires new sophisticated techniques of pattern recognition, which are being developed by researchers in artificial intelligence. Complex statistical issues have arisen in connection with database searches, and this has required the creation of new and specific tools.

There is one class of problems, however, for which what is most needed is *efficient algorithms*. An algorithm, simply stated, is a step-by-step procedure that tries to solve a certain well-defined problem in a limited time bound. To be efficient, an algorithm should not take "too long" to solve a problem, even a large one. The classic example of a problem in molecular biology solvable by an algorithm is sequence comparison: Given two sequences representing biomolecules, we want to know how similar they are. This is a problem that must be solved thousands of times every day, so it is desirable that a very efficient algorithm should be employed.

The purpose of this book is to present a representative sample of computational

problems in molecular biology and some of the efficient algorithms that have been proposed to solve them. Some of these problems are well understood, and several of their algorithms have been known for many years. Other problems seem more difficult, and no satisfactory algorithmic approach has been developed so far. In these cases we have concentrated in explaining some of the mathematical models that can be used as a foundation in the development of future algorithms.

The reader should be aware that an algorithm for a problem in molecular biology is a curious beast. It tries to serve two masters: the molecular biologist, who wants the algorithm to be *relevant,* that is, to solve a problem with all the errors and uncertainties with which it appears in practice; and the computer scientist, who is interested in proving that the algorithm efficiently solves a well-defined problem, and who is usually ready to sacrifice relevance for provability (or efficiency). We have tried to strike a balance between these often conflicting demands, but more often than not we have taken the computer scientists' side. After all, that is what the authors are. Nevertheless we hope that this book will serve as a stimulus for both molecular biologists and computer scientists.

This book is an introduction. This means that one of our guiding principles was to present algorithms that we considered *simple,* whenever possible. For certain problems that we describe, more efficient and generally more sophisticated algorithms exist; pointers to some of these algorithms are usually given in the bibliographic notes at the end of each chapter. Despite our general aim, a few of the algorithms or models we present cannot be considered simple. This usually reflects the inherent complexity of the corresponding topic. We have tried to point out the more difficult parts by using the star symbol (*) in the corresponding headings or by simply spelling out this caveat in the text. The introductory nature of the text also means that, for some of the topics, our coverage is intended to be a starting point for those new to them. It is probable, and in some cases a fact, that whole books could be devoted to such topics.

The primary audience we have in mind for this book is students from the mathematical and computing sciences. We assume no prior knowledge of molecular biology beyond the high school level, and we provide a chapter that briefly explains the basic concepts used in the book. Readers not familiar with molecular biology are urged however to go beyond what is given there and expand their knowledge by looking at some of the books referred to at the end of Chapter 1.

We hope that this book will also be useful in some measure to students from the biological sciences. We do assume that the reader has had some training in college-level discrete mathematics and algorithms. With the purpose of helping the reader unfamiliar with these subjects, we have provided a chapter that briefly covers all the basic concepts used in the text.

Computational molecular biology is expanding fast. Better algorithms are constantly being designed, and new subfields are emerging even as we write this. Within the constraints mentioned above, we did our best to cover what we considered a wide range of topics, and we believe that most of the material presented is of lasting value. To the reader wishing to pursue further studies, we have provided pointers to several sources of information, especially in the bibliographic notes of the last chapter (and including WWW sites of interest). These notes, however, are not meant to be exhaustive. In addition, please note that we cannot guarantee that the World Wide Web Universal Resource Locators given in the text will remain valid. We have tested these addresses, but due to the dynamic nature of the Web, they could change in the future.

BOOK OVERVIEW

Chapter 1 presents fundamental concepts from molecular biology. We describe the basic structure and function of proteins and nucleic acids, the mechanisms of molecular genetics, the most important laboratory techniques for studying the genome of organisms, and an overview of existing sequence databases.

Chapter 2 describes strings and graphs, two of the most important mathematical objects used in the book. A brief exposition of general concepts of algorithms and their analysis is also given, covering definitions from the theory of NP-completeness.

The following chapters are based on specific problems in molecular biology. Chapter 3 deals with *sequence comparison.* The basic two-sequence problem is studied and the classic dynamic programming algorithm is given. We then study extensions of this algorithm, which are used to deal with more general cases of the problem. A section is devoted to the multiple-sequence comparison problem. Other sections deal with programs used in database searches, and with some other miscellaneous issues.

Chapter 4 covers the *fragment assembly problem.* This problem arises when a DNA sequence is broken into small fragments, which must then be assembled to reconstitute the original molecule. This is a technique widely used in large-scale sequencing projects, such as the Human Genome Project. We show how various complications make this problem quite hard to solve. We then present some models for simplified versions of the problem. Later sections deal with algorithms and heuristics based on these models.

Chapter 5 covers the *physical mapping problem.* This can be considered as fragment assembly on a larger scale. Fragments are much longer, and for this reason assembly techniques are completely different. The aim is to obtain the location of some markers along the original DNA molecule. A brief survey of techniques and models is given. We then describe an algorithm for the consecutive ones problem; this abstract problem plays an important role in physical mapping. The chapter finishes with sections devoted to algorithmic approximations and heuristics for one version of physical mapping.

Proteins and nucleic acids also evolve through the ages, and an important tool in understanding how this evolution has taken place is the *phylogenetic tree.* These trees also help shed light in the understanding of protein function. Chapter 6 describes some of the mathematical problems related to phylogenetic tree reconstruction and the simple algorithms that have been developed for certain special cases.

An important new field of study that has recently emerged in computational biology is *genome rearrangements.* It has been discovered that some organisms are genetically different, not so much at the sequence level, but in the order in which large similar chunks of their DNA appear in their respective genomes. Interesting mathematical models have been developed to study such differences, and Chapter 7 is devoted to them.

The understanding of the biological function of molecules is actually at the heart of most problems in computational biology. Because molecules fold in three dimensions and because their function depends on the way they fold, a primary concern of scientists in the past several decades has been the discovery of their three-dimensional structure, in particular for RNA and proteins. This has given rise to methods that try to predict a molecule's structure based on its primary sequence. In Chapter 8 we describe dynamic programming algorithms for RNA structure prediction, give an overview of the difficulties of protein structure prediction, and present one important recent development in the

field called protein threading, which attempts to align a a protein sequence with a known structure.

Chapter 9 ends the book presenting a description of the exciting new field of DNA computing. We present there the basic experiment that showed how we can use DNA molecules to solve one hard algorithmic problem, and a theoretical extension that applies to another hard problem.

A word about general conventions. As already mentioned, sections whose headings are followed by a star symbol (⋆) contain material considered by the authors to be more difficult. In the case of concept definitions, we have used the convention that terms used throughout the book are in **boldface** when they are first defined. Other terms appear in *italics* in their definition. Many of our algorithms are presented first through English sentences and then in pseudo code format (pseudo code conventions are described in Section 2.3). In some cases the pseudo code provides a level of detail that should help readers interested in actual implementation.

Summaries are provided for the longer chapters.

EXERCISES

Exercises appear at the end of every chapter. Exercises marked with one star (⋆) are hard, but feasible in less than a day. They may require knowledge of computer science techniques not presented in the book. Those marked with two stars (⋆⋆) are problems that were once research problems but have since been solved, and their solutions can be found in the literature (we usually cite in the bibliographic notes the research paper that solves the exercise). Finally exercises marked with a diamond (◇) are research problems that have not been solved as far as the authors know.

At the end of the book we provide answers or hints to selected exercises.

ERRORS

Despite the authors' best efforts, this book no doubt contains errors. If you find any, or have any suggestions for improvement, we will be glad to hear from you. Please send error reports or any other comments to us at bio@dcc.unicamp.br, or at

J. Meidanis / J. C. Setubal
Instituto de Computação, C. P. 6176
UNICAMP
Campinas, SP 13083-970
Brazil

(The authors can be reached individually by e-mail at meidanis@dcc.unicamp.br and at setubal@dcc.unicamp.br.) We thank in advance all readers interested in helping us make this a better book. As errors become known they will be reported in the following WWW site:

http://www.dcc.unicamp.br/~bio/ICMB.html

ACKNOWLEDGMENTS

This book is a successor to another, much shorter one on the same subject, written by the authors in Portuguese and published in 1994 in Brazil. That first book was made possible thanks to a Brazilian computer science meeting known as "Escola de Computação," held every two years. We believe that without such a meeting we would not be writing this preface, so we are thankful to have had that opportunity.

The present book started its life thanks to Mike Sugarman, Bonnie Berger, and Tom Leighton. We got a lot of encouragement from them, and also some helpful hints. Bonnie in particular was very kind in giving us copies of her course notes at an early stage.

We have been fortunate to have had financial grants from FAPESP and CNPq (Brazilian Research Agencies); they helped us in several ways. Grants from FAPESP were awarded within the "Laboratory for Algorithms and Combinatorics" project and provided computer equipment. Grants from CNPq were awarded in the form of individual fellowships and within the PROTEM program through the PROCOMB and TCPAC projects, which provided funding for research visits.

We are grateful to our students who helped us proofread early drafts. Special thanks are due to Nalvo Franco de Almeida Jr. and Maria Emília Machado Telles Walter. Nalvo, in addition, made many figures and provided several helpful comments.

We had many helpful discussions with our colleague Jorge Stolfi, who also provided crucial assistance in typesetting matters. Fernando Reinach and Gilson Paulo Manfio helped us with Chapter 1. We discussed book goals and general issues with Jim Orlin. Martin Farach and Sampath Kannan, as well as several anonymous reviewers, also made many suggestions, some of which were incorporated into the text. Our colleagues at the Institute of Computing at UNICAMP provided encouragement and a stimulating work environment.

The following people were very kind in sending us research papers: Farid Alizadeh, Alberto Caprara, Martin Farach, David Greenberg, Dan Gusfield, Sridar Hannenhalli, Wen-Lian Hsu, Xiaoqiu Huang, Tao Jiang, John Kececioglu, Lukas Knecht, Rick Lathrop, Gene Myers, Alejandro Schäffer, Ron Shamir, Martin Vingron (who also sent lecture notes), Todd Wareham, and Tandy Warnow. Some of our sections were heavily based on some of these papers.

Many thanks are also due to Erik Brisson, Eileen Sullivan, Bruce Dale, Carlos Eduardo Ferreira, and Thomas Roos, who helped in various ways.

J. C. S. wishes to thank his wife Silvia (a.k.a. Teca) and his children Claudia, Tomás, and Caio, for providing the support without which this book could not have been written.

This book was typeset by the authors using Leslie Lamport's LaTeX 2_ε system, which works on top of Don Knuth's TeX system. These are truly marvelous tools.

The quotation of Don Knuth at the beginning of this preface is from an interview given to Computer Literacy Bookshops, Inc., on December 7, 1993.

João Carlos Setubal
João Meidanis

CHAPTER

1

BASIC CONCEPTS OF MOLECULAR BIOLOGY

In this chapter we present basic concepts of molecular biology. Our aim is to provide readers with enough information so that they can comfortably follow the biological background of this book as well as the literature on computational molecular biology in general. Readers who have been trained in the exact sciences should know from the outset that in molecular biology nothing is 100% valid. To every rule there is an exception. We have tried to point out some of the most notable exceptions to general rules, but in other cases we have omitted such mention, so as not to transform this chapter into a molecular biology textbook.

1.1 LIFE

In nature we find both living and nonliving things. Living things can move, reproduce, grow, eat, and so on — they have an *active* participation in their environment, as opposed to nonliving things. Yet research in the past centuries reveals that both kinds of matter are composed by the same atoms and conform to the same physical and chemical rules. What is the difference then? For a long time in human history, people thought that some sort of extra matter bestowed upon living beings their active characteristics — that they were "animated" by such a thing. But nothing of the kind has ever been found. Instead, our current understanding is that living beings act the way they do due to a complex array of chemical reactions that occur inside them. These reactions never cease. It is often the case that the products of one reaction are being constantly consumed by another reaction, keeping the system going. A living organism is also constantly exchanging matter and energy with its surroundings. In contrast, anything that is in equilibrium with its surrounding can generally be considered dead. (Some notable exceptions are vegeta-

tive forms, like seeds, and viruses, which may be completely inactive for long periods of time, and are not dead.)

Modern science has shown that life started some 3.5 billions of years ago, shortly (in geological terms) after the Earth itself was formed. The first life forms were very simple, but over billions of years a continuously acting process called evolution made them evolve and diversify, so that today we find very complex organisms as well as very simple ones.

Both complex and simple organisms have a similar molecular chemistry, or *biochemistry*. The main actors in the chemistry of life are molecules called **proteins** and **nucleic acids**. Roughly speaking, proteins are responsible for what a living being is and does in a physical sense. (The distinguished scientist Russell Doolittle once wrote that "we *are* our proteins.") Nucleic acids, on the other hand, encode the information necessary to produce proteins and are responsible for passing along this "recipe" to subsequent generations.

Molecular biology research is basically devoted to the understanding of the structure and function of proteins and nucleic acids. These molecules are therefore the fundamental objects of this book, and we now proceed to give a basic and brief description of the current state of knowledge regarding them.

1.2 PROTEINS

Most substances in our bodies are proteins, of which there are many different kinds. *Structural proteins* act as tissue building blocks, whereas other proteins known as *enzymes* act as catalysts of chemical reactions. A catalyst is a substance that speeds up a chemical reaction. Many biochemical reactions, if left unattended, would take too long to complete or not happen at all and would therefore not be useful to life. An enzyme can speed up this process by orders of magnitude, thereby making life possible. Enzymes are very specific — usually a given enzyme can help only one kind of biochemical reaction. Considering the large number of reactions that must occur to sustain life, we need a lot of enzymes. Other examples of protein function are oxygen transport and antibody defense. But what exactly are proteins? How are they made? And how do they perform their functions? This section tries briefly to answer these questions.

A protein is a chain of simpler molecules called **amino acids**. Examples of amino acids can be seen in Figure 1.1. Every amino acid has one central carbon atom, which is known as the alpha carbon, or C_α. To the C_α atom are attached a hydrogen atom, an amino group (NH_2), a carboxy group (COOH), and a *side chain*. It is the side chain that distinguishes one amino acid from another. Side chains can be as simple as one hydrogen atom (the case of amino acid glycine) or as complicated as two carbon rings (the case of tryptophan). In nature we find 20 different amino acids, which are listed in Table 1.1. These 20 are the most common in proteins; exceptionally a few nonstandard amino acids might be present.

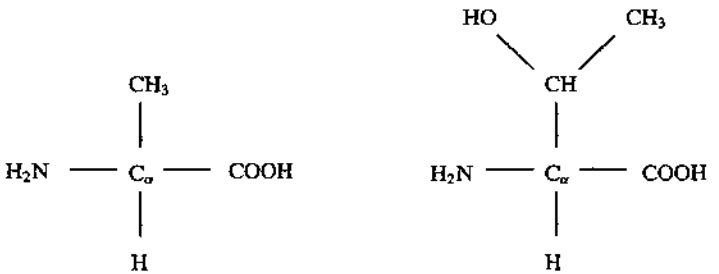

FIGURE 1.1

Examples of amino acids: alanine (left) and threonine.

In a protein, amino acids are joined by *peptide bonds*. For this reason, proteins are polypeptidic chains. In a peptide bond, the carbon atom belonging to the carboxy group of amino acid A_i bonds to the nitrogen atom of amino acid A_{i+1}'s amino group. In such a bond, a water molecule is liberated, because the oxygen and hydrogen of the carboxy group joins the one hydrogen from the amino group. Hence, what we really find inside a polypeptide chain is a **residue** of the original amino acid. Thus we generally speak of a protein having 100 residues, rather than 100 amino acids. Typical proteins contain about 300 residues, but there are proteins with as few as 100 or with as many as 5,000 residues.

TABLE 1.1

The twenty amino acids commonly found in proteins.

	One-letter code	*Three-letter code*	*Name*
1	A	Ala	Alanine
2	C	Cys	Cysteine
3	D	Asp	Aspartic Acid
4	E	Glu	Glutamic Acid
5	F	Phe	Phenylalanine
6	G	Gly	Glycine
7	H	His	Histidine
8	I	Ile	Isoleucine
9	K	Lys	Lysine
10	L	Leu	Leucine
11	M	Met	Methionine
12	N	Asn	Asparagine
13	P	Pro	Proline
14	Q	Gln	Glutamine
15	R	Arg	Arginine
16	S	Ser	Serine
17	T	Thr	Threonine
18	V	Val	Valine
19	W	Trp	Tryptophan
20	Y	Tyr	Tyrosine

The peptide bond makes every protein have a *backbone*, given by repetitions of the basic block $-N-C_\alpha-(CO)-$. To every C_α there corresponds a side chain. See Figure 1.2 for a schematic view of a polypeptide chain. Because we have an amino group at one end of the backbone and a carboxy group at the other end, we can distinguish both ends of a polypeptide chain and thus give it a direction. The convention is that polypeptides begin at the amino group (*N-terminal*) and end at the carboxy group (*C-terminal*).

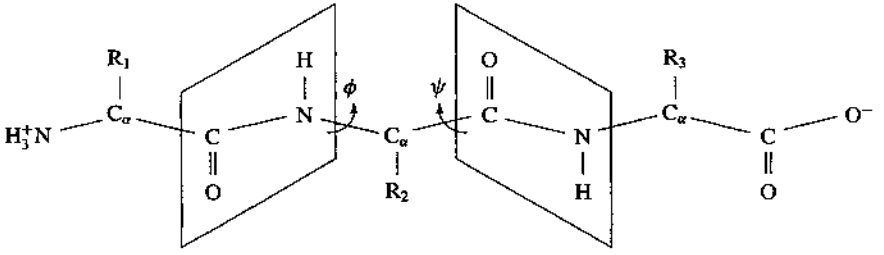

FIGURE 1.2

A polypeptide chain. The R_i side chains identify the component amino acids. Atoms inside each quadrilateral are on the same plane, which can rotate according to angles ϕ and ψ.

A protein is not just a linear sequences of residues. This sequence is known as its **primary structure**. Proteins actually fold in three dimensions, presenting **secondary**, **tertiary**, and **quaternary** structures. A protein's secondary structure is formed through interactions between backbone atoms only and results in "local" structures such as helices. Tertiary structures are the result of secondary structure packing on a more global level. Yet another level of packing, or a group of different proteins packed together, receives the name of quaternary structure. Figure 1.3 depicts these structures schematically.

Proteins can fold in three dimensions because the plane of the bond between the C_α atom and the nitrogen atom may rotate, as can the plane between the C_α atom and the other C atom. These rotation angles are known as ϕ and ψ, respectively, and are illustrated in Figure1.2. Side chains can also move, but it is a secondary movement with respect to the backbone rotation. Thus if we specify the values of all $\phi - \psi$ pairs in a protein, we know its exact folding. Determining the folding, or three-dimensional structure, of a protein is one of the main research areas in molecular biology, for three reasons. First, the three-dimensional shape of a protein is related to its function. Second, the fact that a protein can be made out of 20 different kinds of amino acids makes the resulting three-dimensional structure in many cases very complex and without symmetry. Third, no simple and accurate method for determining the three-dimensional structure is known. These reasons motivate Chapter 8, where we discuss some molecular structure prediction methods. These methods try to predict a molecule's structure from its primary sequence.

The three-dimensional shape of a protein determines its function in the following way. A folded protein has an irregular shape. This means that it has varied nooks and

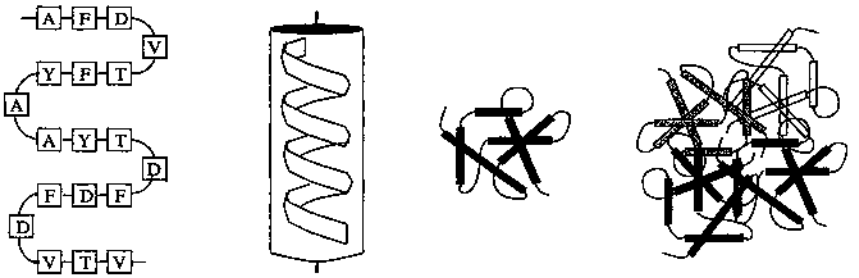

FIGURE 1.3

Primary, secondary, tertiary, and quaternary structures of proteins. (Based on a figure from [28].)

bulges, and such shapes enable the protein to come in closer contact with, or **bind to,** some other specific molecules. The kinds of molecules a protein can bind to depend on its shape. For example, the shape of a protein can be such that it is able to bind with several identical copies of itself, building, say, a thread of hair. Or the shape can be such that molecules A and B bind to the protein and thereby start exchanging atoms. In other words, a reaction takes place between A and B, and the protein is fulfilling its role as a catalyst.

But how do we get our proteins? Proteins are produced in a cell structure called *ribosome*. In a ribosome the component amino acids of a protein are assembled one by one thanks to information contained in an important molecule called **messenger ribonucleic acid**. To explain how this happens, we need to explain what nucleic acids are.

1.3 NUCLEIC ACIDS

Living organisms contain two kinds of nucleic acids: **ribonucleic acid,** abbreviated by RNA, and **deoxyribonucleic acid**, or DNA. We describe DNA first.

1.3.1 DNA

Like a protein, a molecule of DNA is a chain of simpler molecules. Actually it is a *double* chain, but let us first understand the structure of one simple chain, called **strand.** It has a backbone consisting of repetitions of the same basic unit. This unit is formed by a sugar molecule called 2′-deoxyribose attached to a phosphate residue. The sugar molecule contains five carbon atoms, and they are labeled 1′ through 5′ (see Figure 1.4). The bond that creates the backbone is between the 3′ carbon of one unit, the phosphate residue, and the 5′ carbon of the next unit. For this reason, DNA molecules also have an

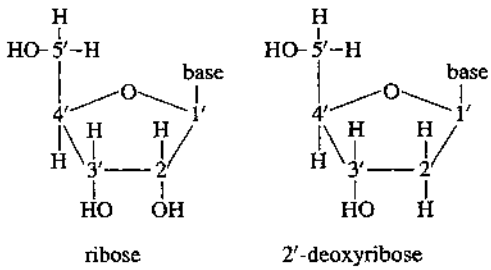

FIGURE 1.4

Sugars present in nucleic acids. Symbols 1′ through 5′ represent carbon atoms. The only difference between the two sugars is the oxygen in carbon 2′. Ribose is present in RNA and 2′-deoxyribose is found in DNA.

orientation, which by convention, starts at the 5′ end and finishes at the 3′ end. When we see a single stranded DNA sequence in a technical paper, book, or a sequence database file, it is always written in this canonical, 5′ → 3′ direction, unless otherwise stated.

Attached to each 1′ carbon in the backbone are other molecules called **bases**. There are four kinds of bases: adenine (A), guanine (G), cytosine (C), and thymine (T). In Figure 1.5 we show the schematic molecular structure of each base, and in Figure 1.6 we show a schematic view of the single DNA strand described so far. Bases A and G belong to a larger group of substances called *purines*, whereas C and T belong to the *pyrimidines*. When we see the basic unit of a DNA molecule as consisting of the sugar, the phosphate, and its base, we call it a **nucleotide**. Thus, although bases and nucleotides are not the same thing, we can speak of a DNA molecule having 200 bases or 200 nucleotides. A DNA molecule having a few (tens of) nucleotides is referred to as an *oligonucleotide*. DNA molecules in nature are very long, much longer than proteins. In a human cell, DNA molecules have hundreds of millions of nucleotides.

As already mentioned, DNA molecules are double strands. The two strands are tied together in a helical structure, the famous double helix discovered by James Watson and Francis Crick in 1953. How can the two strands hold together? Because each base in one strand is paired with (or *bonds to*) a base in the other strand. Base A is always paired with base T, and C is always paired with G, as shown in Figures 1.5 and 1.7. Bases A and T are said to be the **complement** of each other, or a pair of **complementary bases**. Similarly, C and G are complementary bases. These pairs are known as *Watson–Crick base pairs*. Base pairs provide the unit of length most used when referring to DNA molecules, abbreviated to **bp**. So we say that a certain piece of DNA is 100,000 bp long, or 100 kbp.

In this book we will generally consider DNA as string of letters, each letter representing a base. Figure 1.8 presents this "string-view" of DNA, showing that we represent the double strand by placing one of the strings on top of the other. Notice the base-pairing. Even though the strands are linked, each one preserves its own orientation, and the two orientations are opposite. Figure 1.8 illustrates this fact. Notice that the 3′ end of one strand corresponds to the 5′ end of the other strand. This property is sometimes expressed by saying that the two strands are *antiparallel*. The fundamental consequence of

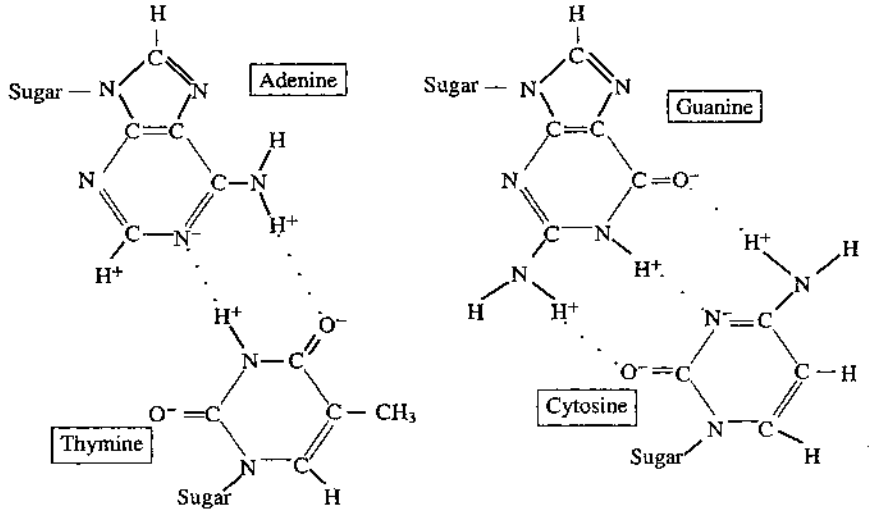

FIGURE 1.5

Nitrogenated bases present in DNA. Notice the bonds that can form between adenine and thymine and between guanine and cytosine, indicated by the dotted lines.

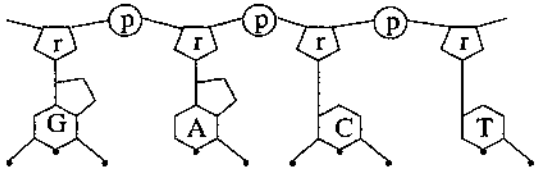

FIGURE 1.6

A schematic molecular structure view of one DNA strand.

this structure is that it is possible to infer the sequence of one strand given the other. The operation that enables us to do that is called **reverse complementation**. For example, given strand s = AGACGT in the canonical direction, we do the following to obtain its reverse complement: First we reverse s, obtaining s' = TGCAGA, and then we replace each base by its complement, obtaining $\overline{s}$ = ACGTCT. (Note that we use the bar over the s to denote the reverse complement of strand s.) It is precisely this mechanism that allows DNA in a cell to *replicate*, therefore allowing an organism that starts its life as one cell to grow into billions of other cells, each one carrying copies of the DNA molecules from the original cell.

In organisms whose cells do not have a nucleus, DNA is found free-floating inside each cell. In higher organisms, DNA is found inside the nucleus and in cell organelles called *mitochondria* (animals and plants) and *chloroplasts* (plants only).

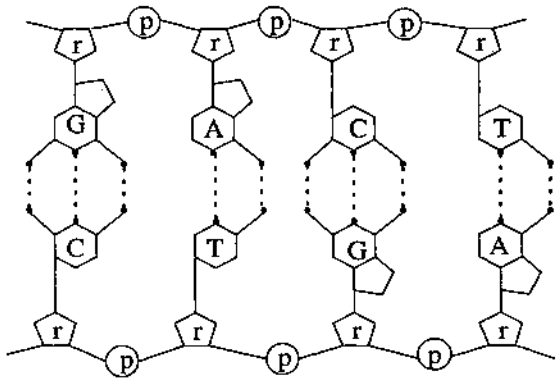

FIGURE 1.7

A schematic molecular structure view of a double strand of DNA.

```
5′  ···  TACTGAA  ···  3′
3′  ···  ATGACTT  ···  5′
```

FIGURE 1.8

A double-stranded DNA sequence represented by strings of letters.

1.3.2 RNA

RNA molecules are much like DNA molecules, with the following basic compositional and structural differences:

- In RNA the sugar is ribose instead of 2′-deoxyribose (see Figure 1.4).
- In RNA we do not find thymine (T); instead, uracil (U) is present. Uracil also binds with adenine like thymine does.
- RNA does not form a double helix. Sometimes we see RNA-DNA hybrid helices; also, parts of an RNA molecule may bind to other parts of the same molecule by complementarity. The three-dimensional structure of RNA is far more varied than that of DNA.

Another difference between DNA and RNA is that while DNA performs essentially one function (that of encoding information), we will see shortly that there are different kinds of RNAs in the cell, performing different functions.

1.4 THE MECHANISMS OF MOLECULAR GENETICS

The importance of DNA molecules is that the information necessary to build each protein or RNA found in an organism is encoded in DNA molecules. For this reason, DNA is sometimes referred to as "the blueprint of life." In this section we will describe this encoding and how a protein is built out of DNA (the process of *protein synthesis*). We will see also how the information in DNA, or genetic information, is passed along from a parent to its offspring.

1.4.1 GENES AND THE GENETIC CODE

Each cell of an organism has a few very long DNA molecules. Each such molecule is called a **chromosome**. We will have more to say about chromosomes later, so for the moment let us examine the encoding of genetic information from the point of view of only one very long DNA molecule, which we will simply call "the DNA." The first important thing to know about this DNA is that certain contiguous stretches along it encode information for building proteins, but others do not. The second important thing is that to each different kind of protein in an organism there usually corresponds one and only one contiguous stretch along the DNA. This stretch is known as a **gene**. Because some genes originate RNA products, it is more correct to say that a gene is a contiguous stretch of DNA that contains the information necessary to build a protein or an RNA molecule. Gene lengths vary, but in the case of humans a gene may have something like 10,000 bp. Certain cell mechanisms are capable of recognizing in the DNA the precise points at which a gene starts and at which it ends.

A protein, as we have seen, is a chain of amino acids. Therefore, to "specify" a protein all you have to do is to specify each amino acid it contains. And that is precisely what the DNA in a gene does, using triplets of nucleotides to specify each amino acid. Each nucleotide triplet is called a **codon**. The table that gives the correspondence between each possible triplet and each amino acid is the so-called **genetic code**, seen in Table 1.2. In the table you will notice that nucleotide triplets are given using RNA bases rather than DNA bases. The reason is that it is RNA molecules that provide the link between the DNA and actual protein synthesis, in a process to be detailed shortly. Before that let us study the genetic code in more detail.

Notice that there are 64 possible nucleotide triplets, but there are only 20 amino acids to specify. The consequence is that different triplets correspond to the same amino acid. For example, both AAG and AAA code for lysine. On the other hand, three of the possible codons do not code for any amino acid and are used instead to signal the end of a gene. These special termination codons are identified in Table 1.2 with the word STOP written in the corresponding entry. Finally, we remark that the genetic code shown above is used by the vast majority of living organisms, but some organisms use a slightly modified code.

TABLE 1.2

The genetic code mapping codons to amino acids.

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

1.4.2 TRANSCRIPTION, TRANSLATION, AND PROTEIN SYNTHESIS

Now let us describe in some detail how the information in the DNA results in proteins. A cell mechanism recognizes the beginning of a gene or gene cluster thanks to a *promoter*. The promoter is a region before each gene in the DNA that serves as an indication to the cellular mechanism that a gene is ahead. The codon AUG (which codes for methionine) also signals the start of a gene. Having recognized the beginning of a gene or gene cluster, a copy of the gene is made on an RNA molecule. This resulting RNA is the *messenger RNA*, or mRNA for short, and will have exactly the same sequence as one of the strands of the gene but substituting U for T. This process is called **transcription**. The mRNA will then be used in cellular structures called ribosomes to manufacture a protein.

Because RNA is single-stranded and DNA is double-stranded, the mRNA produced is identical in sequence to only one of the gene strands, being complementary to the other strand — keeping in mind that T is replaced by U in RNA. The strand that looks like the mRNA product is called the *antisense* or *coding* strand, and the other one is the *sense* or *anticoding* or else *template* strand. The template strand is the one that is actually transcribed, because the mRNA is composed by binding together ribonucleotides complementary to this strand. The process always builds mRNA molecules from their 5′ end to their 3′ end, whereas the template strand is read from 3′ to 5′. Notice also that it is

not the case that the template strand for genes is always the same; for example, the template strand for a certain gene *A* may be one of the strands, and the template strand for another gene *B* may be the other strand. For a given gene, the cell can recognize the corresponding template strand thanks to a promoter. Even though the reverse complement of the promoter appears in the other strand, this reverse complement is *not* a promoter and thus will not be recognized as such. One important consequence of this fact is that genes from the same chromosome have an *orientation* with respect to each other: Given two genes, if they appear in the same strand they have the same orientation; otherwise they have opposite orientation. This is a fundamental fact for Chapter 7. We finally note that the terms *upstream* and *downstream* are used to indicate positions in the DNA in reference to the orientation of the coding strand, with the promoter being upstream from its gene.

Transcription as described is valid for organisms categorized as *prokaryotes*. These organisms have their DNA free in the cell, as they lack a nuclear membrane. Examples of prokaryotes are bacteria and blue algae. All other organisms, categorized as *eukaryotes*, have a nucleus separated from the rest of the cell by a nuclear membrane, and their DNA is kept inside the nucleus. In these organisms genetic transcription is more complex. Many eukaryotic genes are composed of alternating parts called *introns* and *exons*. After transcription, the introns are spliced out from the mRNA. This means that introns are parts of a gene that are *not* used in protein synthesis. An example of exon-intron distribution is given by the gene for bovine atrial naturietric peptide, which has 1082 base pairs. Exons are located at positions 1 to 120, 219 to 545, and 1071 to 1082. Introns occupy positions 121 to 218 and 546 to 1070. Thus, the mRNA coding regions has just 459 bases, and the corresponding protein has 153 residues. After introns are spliced out, the shortened mRNA, containing copies of only the exons plus regulatory regions in the beginning and end, leaves the nucleus, because ribosomes are outside the nucleus.

Because of the intron/exon phenomenon, we use different names to refer to the entire gene as found in the chromosome and to the spliced sequence consisting of exons only. The former is called *genomic* DNA and the latter *complementary* DNA or cDNA. Scientists can manufacture cDNA without knowing its genomic counterpart. They first capture the mRNA outside the nucleus on its way to the ribosomes. Then, in a process called *reverse transcription*, they produce DNA molecules using the mRNA as a template. Because the mRNA contains only exons, this is also the composition of the DNA produced. Thus, they can obtain cDNA without even looking at the chromosomes. Both transcription and reverse transcription are complex processes that need the help of enzymes. *Transcriptase* and *reverse transcriptase* are the enzymes that catalyze these processes in the cell. There is also a phenomenon called *alternative splicing*. This occurs when the same genomic DNA can give rise to two or more different mRNA molecules, by choosing the introns and exons in different ways. They will in general produce different proteins.

Now let us go back to mRNA and protein synthesis. In this process two other kinds of RNA molecules play very important roles. As we have already mentioned, protein synthesis takes place inside cellular structures called ribosomes. Ribosomes are made of proteins and a form of RNA called *ribosomal* RNA, or rRNA. The ribosome functions like an assembly line in a factory using as "inputs" an mRNA molecule and another kind of RNA molecule called *transfer* RNA, or tRNA.

Transfer RNAs are the molecules that actually implement the genetic code in a pro-

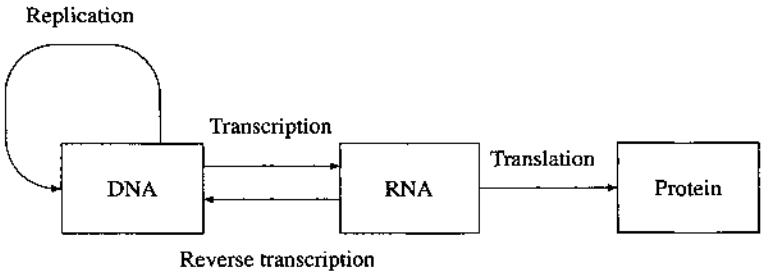

FIGURE 1.9

Genetic information flow in a cell: the so-called central dogma of molecular biology.

cess called **translation**. They make the connection between a codon and the specific amino acid this codon codes for. Each tRNA molecule has, on one side, a conformation that has high affinity for a specific codon and, on the other side, a conformation that binds easily to the corresponding amino acid. As the messenger RNA passes through the interior of the ribosome, a tRNA matching the current codon — the codon in the mRNA currently inside the ribosome — binds to it, bringing along the corresponding amino acid (a generous supply of amino acids is always "floating around" in the cell). The three-dimensional position of all these molecules in this moment is such that, as the tRNA binds to its codon, its attached amino acid falls in place just next to the previous amino acid in the protein chain being formed. A suitable enzyme then catalyzes the addition of this current amino acid to the protein chain, releasing it from the tRNA. A protein is constructed residue by residue in this fashion. When a STOP codon appears, no tRNA associates with it, and the synthesis ends. The messenger RNA is released and degraded by cell mechanisms into ribonucleotides, which will be then recycled to make other RNA.

One might think that there are as many tRNAs as there are codons, but this is not true. The actual number of tRNAs varies among species. The bacterium *E. coli*, for instance, has about 40 tRNAs. Some codons are not represented, and some tRNAs can bind to more than one codon.

Figure 1.9 summarizes the processes we have just described. The expression *central dogma* is generally used to denote our current synthetic view of genetic information transfer in cells.

1.4.3 JUNK DNA AND READING FRAMES

In this section we provide some additional details regarding the processes described in previous sections.

As mentioned, genes are certain contiguous regions of the chromosome, but they do

not cover the entire molecule. Each gene, or group of related genes, is flanked by regulatory regions that play a role in controlling gene transcription and other related processes, but otherwise intergenic regions have no known function. They are called "junk DNA" because they appear to be there for no particular reason. Moreover, they accumulate mutations, as a change not affecting genes or their regulatory regions is often not lethal and is therefore propagated to the progeny. Recent research has shown, however, that junk DNA has more information content than previously believed. The amount of junk DNA varies from species to species. Prokaryotes tend to have little of it — their chromosomes are almost all covered by genes. In contrast, eukaryotes have plenty of junk DNA. In human beings it is estimated that as much as 90% of the DNA in chromosomes is composed of junk DNA.

An aspect of the transcription process that is important to know is the concept of *reading frame*. A reading frame is one of the three possible ways of grouping bases to form codons in a DNA or RNA sequence. For instance, consider the sequence

TAATCGAATGGGC.

One reading frame would be to take as codons TAA, TCG, AAT, GGG, leaving out the last C. Another reading frame would be to ignore the first T and get codons AAT, CGA, ATG, GGC. Yet another reading frame would yield codons ATC, GAA, TGG, leaving out two bases at the beginning (TA) and two bases at the end (GC).

Notice that the three reading frames start at positions 1, 2, and 3 in the given sequence, respectively. If we were to consider a reading frame starting at position 4, the codons obtained would be a subset of the ones for starting position 1, so this is actually the same reading frame starting at a different position. In general, if we take starting positions i and j where the difference $j - i$ is a multiple of three, we are in fact considering the same reading frame.

Sometimes we talk about six, not three, different reading frames in a sequence. In this case, what we have is a DNA sequence and we are looking at the opposite strand as well. We have three reading frames in one strand plus another three in the complementary strand, giving a total of six. It is common to do that when we have newly sequenced DNA and want to compare it to a protein database. We have to translate the DNA sequence into a protein sequence, but there are six ways of doing that, each one taking a different reading frame. The fact that we lose one or two bases at the extremities of the sequence is not important; these sequences are long enough to yield a meaningful comparison even with a few missing residues.

An *open reading frame*, or ORF, in a DNA sequence is a contiguous stretch of this sequence beginning at the start codon, having an integral number of codons (its length is a multiple of three), and such that none of its codons is a STOP codon. The presence of additional regulatory regions upstream from the start codon is also used to characterize an ORF.

1.4.4 CHROMOSOMES

In this section we briefly describe the process of genetic information transmission at the chromosome level. First we note that the complete set of chromosomes inside a cell is

called a **genome**. The number of chromosomes in a genome is characteristic of a species. For instance, every cell in a human being has 46 chromosomes, whereas in mice this number is 40. Table 1.3 gives the number of chromosomes and genome size in base pairs for selected species.

TABLE 1.3

Genome sizes of certain species. Apart from our own species, the organisms listed are important in molecular biology and genetics research.

Species	*Number of Chromosomes (diploid)*	*Genome Size (base pairs)*
Bacteriophage λ (virus)	1	5×10^4
Escherichia coli (bacterium)	1	5×10^6
Saccharomyces cerevisiae (yeast)	32	1×10^7
Caenorhabditis elegans (worm)	12	1×10^8
Drosophila melanogaster (fruit fly)	8	2×10^8
Homo sapiens (human)	46	3×10^9

Prokaryotes generally have just one chromosome, which is sometimes a circular DNA molecule. On the other hand, in eukaryotes chromosomes appear in pairs (and for this reason the cells that carry them are called *diploid*). In humans, for example, there are 23 pairs. Each member of a pair was inherited from each parent. The two chromosomes that form a pair are called *homologous* and a gene in one member of the pair corresponds to a gene in the other. Certain genes are exactly the same in both the paternal and the maternal copies. One example is the gene that codes for hemoglobin, a protein that carries oxygen in the blood. Other genes may appear in different forms, which are called *alleles*. A typical example is the gene that codes for blood type in humans. It appears in three forms: A, B, and O. As is well known, if a person receives, say, the A allele from the mother and the B allele from the father, this person's blood type will be AB.

Cells that carry only one member of each pair of chromosomes are called *haploid*. These are the cells that are used in sexual reproduction. When a haploid cell from the mother is merged with a haploid cell from the father we have an *egg cell*, which is again diploid. Haploid cells are formed through a process called *meiosis*, in which a cell divides in two and each daughter cell gets one member of each pair of chromosomes.

It is interesting to note that despite the fact that all genes are present in all cells, only a portion of the genes are normally used (*expressed*, in biological jargon) by any specific cell. For instance, liver cells express a different set of genes than do skin cells. The mechanisms through which the cells in an organism differentiate into liver cells, skin cells, and so on, are still largely unclear.

1.4.5 IS THE GENOME LIKE A COMPUTER PROGRAM?

Having reviewed the basic mechanisms through which proteins are produced, it is tempting to view them in light of the so-called "genetic program metaphor." In this metaphor, the genome of an organism is seen as a computer program that completely specifies the organism, and the cell machinery is simply an interpreter of this program. The biological functions performed by proteins would be the execution of this "program."

This metaphor is overly simplistic, due in part to the following two facts:

- The "DNA program" in fact undergoes changes during transcription and translation, so that one cannot simply apply the genetic code to a stretch of DNA known to contain a gene in order to know what protein corresponds to that gene.
- Gene expression is a complex process that may depend on spatial and temporal context. For example, not all genes in a genome are expressed in an organism's lifetime, whereas others are expressed over and over again; some genes are expressed only when the organism is subject to certain outside phenomena, such as a virus invasion. The opposite is also true: Genes that are normally expressed may be repressed because of outside stimuli. It is true that the expression of certain genes is essentially context-free, and this is what makes biotechnology possible. But this is by no means true for all genes. If we view gene expression inside a cell as a "computing process," we can say that, in the case of the human genome, there are more than 10^{18} such processes occurring and interacting simultaneously.

In view of these observations, it seems better to view an organism not as determined by its genome but rather as the result of a very complex network of simultaneous interactions, in which the genome sequence is one of several contributing factors.

1.5 HOW THE GENOME IS STUDIED

In the scientific study of a genome, the first thing to notice is the different orders of magnitude that we have to deal with. Let us use the human genome as an example. The basic information we want to extract from any piece of DNA is its base-pair sequence. The process of obtaining this information is called **sequencing**. A human chromosome has around 10^8 base pairs. On the other hand, the largest pieces of DNA that can be sequenced in the laboratory are 700 bp long. This means that there is a gap of some 10^5 between the scale of what we can actually sequence and a chromosome size. This gap is at the heart of many problems in computational biology, in particular those studied in Chapters 4 (fragment assembly) and 5 (physical mapping). In this section we briefly describe some of the lab techniques that underlie those problems.

1.5.1 MAPS AND SEQUENCES

One particular piece of information that is very important is the location of genes in chromosomes. The term *locus* (plural *loci*) is used to denote the location of a gene in a chromosome. (Sometimes the word locus is used as a synonym for the word gene.) The simplest question in this context is, given two genes, are they in the same homologous pair? This can be answered without resorting to molecular techniques, as long as the genes in question are ones that affect visible characteristics, such as eye color or wing shape. We have to test whether the characteristics are being inherited independently. We will say that they are, or more technically, that they *assort* or *segregate* independently if an offspring has about a 50% chance of inheriting both characteristics from the same parent. If the characteristics do assort independently, then the genes are probably not linked; that is, they probably belong to different chromosomes. Two genes carried in the same pair should segregate together, and the offspring will probably inherit both characteristics from the same parent.

In fact, as is usually the case in biology, things are not so clear-cut: 100% or 50% segregation does not always happen. Any percentage in between can occur, due to *crossing over*. When cells divide to produce other cells that will form the progeny, new gene arrangements can form. We say that *recombination* occurs. Recombination can happen because homologous chromosomes can "cross" and exchange their end parts before segregating. There are an enormous number of recombination possibilities, so what we really see is that rates of recombination vary a great deal. These rates in turn give information on how far apart the genes are in the chromosome. If they are close together, there is a small chance of separation due to crossing over. If they are far apart, the chances of separation increase, to the point that they appear to assort independently.

The first genetic maps were constructed by producing successive generations of certain organisms and analyzing the observed segregation percentages of certain characteristics. A *genetic linkage map* of a chromosome is a picture showing the order and relative distance among genes using such information. Genetic maps constructed from recombination percentages are important, but they have two drawbacks: (1) They do not tell the actual distance in base pairs or other linear unit of length along the chromosome; and (2) If genes are very close, one cannot resolve their order, because the probability of separation is so small that observed recombinant frequencies are all zero.

Maps that reflect the actual distance in base pairs are called **physical maps**. To construct physical maps, we need completely different techniques. In particular we have to work with pieces of DNA much smaller than a chromosome but still too large to be sequenced directly. A physical map can tell the location of certain *markers*, which are precisely known small sequences, within 10^4 base pairs or so. Computational problems associated with physical map construction are studied in Chapter 5.

Finally, for pieces of DNA that are on the order of 10^3 base pairs, we can use still other techniques and obtain the whole sequence. We have mentioned that current lab techniques can sequence DNA pieces of at most 700 bp; to sequence a 20,000-bp piece (in what is known as **large-scale sequencing**), the basic idea is to break apart several copies of the piece in different ways, sequence the (small) fragments directly, and then put the fragments together again by using computational techniques that are studied in Chapter 4.

Figure 1.10 illustrates the different scales at which the human genome is studied.

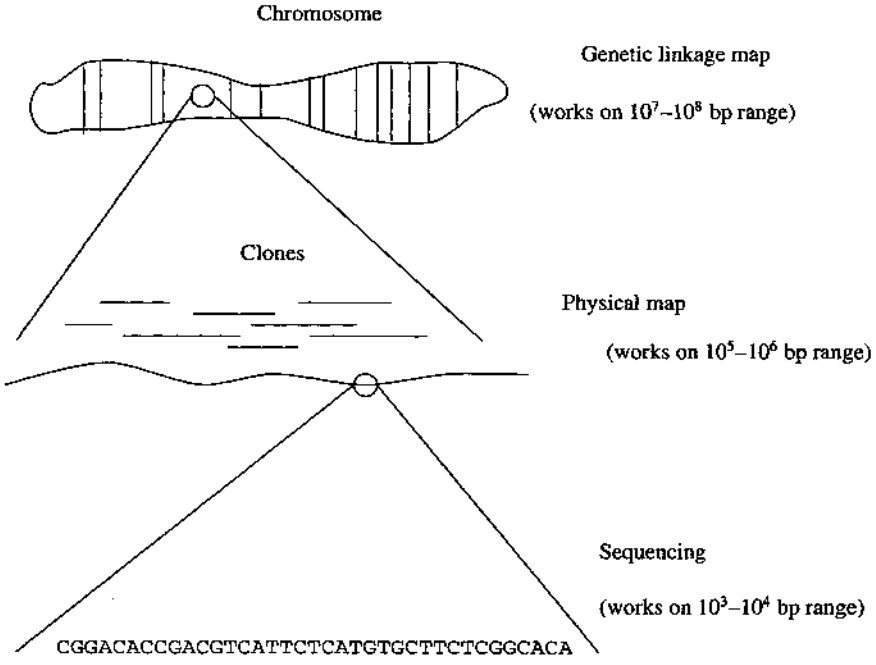

FIGURE 1.10

The different levels at which a genome is studied. (Based on a figure from [19].)

1.5.2 SPECIFIC TECHNIQUES

Some special laboratory techniques must be used to obtain maps and sequences. In this section we give an overview of such techniques.

It is important to realize that lab techniques in molecular biology nearly always produce data that have errors. For this reason, most algorithms for problems in computational biology are useful only to the extent that they can handle errors. Such concern will appear throughout this book.

Viruses and Bacteria

We begin by briefly describing the organisms most used in genetics research: viruses and bacteria.

Viruses are parasites at the molecular level. Viruses can hardly be considered a life form, yet they can reproduce when infecting suitable cells, called *hosts*. Viruses do not exhibit any metabolism — no biochemical reactions occur in them. Instead, viruses rely on the host's metabolism to replicate, and it is this fact that is exploited in laboratory experiments.

Most viruses consist of a protein cap (a *capsid*) with genetic material (either DNA or RNA) inside. Viral DNA is much smaller than the DNA in chromosomes, and therefore is much easier to manipulate. When viruses infect a cell, the genetic material is introduced in the cytoplasm. This genetic material is mistakenly interpreted by the cell mechanism as its own DNA, and for this reason the cell starts producing virus-coded proteins as though they were the cell's own. These proteins promote viral DNA replication and formation of new capsids, so that a large number of virus particles are assembled inside the infected cell. Still other viral proteins break the cell membrane and release all the new virus particles in the environment, where they can attack other cells. Certain viruses do not kill their hosts at once. Instead, the viral DNA gets inserted into the host genome and can stay there for a long time without any noticeable change in cell life. Under certain conditions the dormant virus can be activated, whereupon it detaches itself from the host genome and starts its replication activity.

Viruses are highly specific; they are capable of infecting one type of cell only. Thus, for instance, virus T2 infects only *E. coli* cells; HIV, the human immunodeficiency virus, infects only a certain kind of human cell involved in defending the organism against intruders in general; TMV, the tobacco mosaic virus, infects only tobacco leaf cells; and so on. *Bacteriophages*, or just *phages*, are viruses that infect bacteria.

A bacterium is a single-cell organism having just one chromosome. Bacteria can multiply by simple DNA replication, and they can do this in a very short period of time, which makes them very useful in genetic research. The bacterium most commonly used in labs is the already mentioned *Escherichia coli*, which can divide in 20 minutes. As a result of their size and speed of reproduction, millions of bacteria can be easily generated and handled in a laboratory.

Cutting and Breaking DNA

Because a DNA molecule is so long, some tool to cut it at specific points (like a pair of scissors) or to break it apart in some other way is needed. We review basic techniques for these processes in this section.

The pair of scissors is represented by **restriction enzymes**. They are proteins that catalyze the hydrolysis of DNA (molecule breaking by adding water) at certain specific points called **restriction sites** that are determined by their local base sequence. In other words, they cut DNA molecules in all places where a certain sequence appears. For instance, *Eco*RI is a restriction enzyme that cuts DNA wherever the sequence GAATTC is found. Notice that this sequence is its own reverse complement, that is, $\overline{\text{GAATTC}} = \text{GAATTC}$. Sequences that are equal to their reverse complement are called *palindromes*. So, every time this sequence appears in one strand it appears in the other strand as well. The cuts are made in both strands between the G and the first A. Therefore, the remaining DNA pieces will have "sticky" ends, that is, their 5′ end in the cut point will be four bases shorter than the 3′ end; see Figure 1.11. This favors relinking with another DNA

piece cut with the same enzyme, providing a kind of "DNA cut and paste" technique very useful in genetic engineering for the production of recombinant DNA.

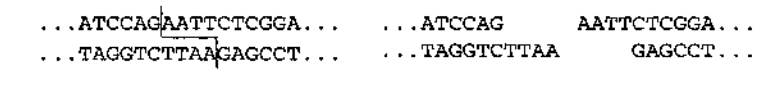

FIGURE 1.11

Enzyme that cuts DNA, leaving "sticky" ends.

Some common types of restriction enzymes are 4-cutters, 6-cutters, and 8-cutters. It is rare to see an odd-cutter because sequences of odd length cannot be palindromes. Restriction enzymes are also called *endonucleases* because they break DNA in an internal point. *Exonucleases* are enzymes that degrade DNA from the ends inwards.

In bacteria, restriction enzymes may act as a defense against virus attacks. These enzymes can cut the viral DNA before it starts its damage. Bacterial DNA on the other hand protects itself from restriction enzymes by adding a methyl group to some of its bases.

DNA molecules can be broken apart by the *shotgun method,* which is sometimes used in DNA sequencing. A solution containing purified DNA — a large quantity of identical molecules — is subjected to some breaking process, such as submitting it to high vibration levels. Each individual molecule breaks down at several random places, and then some of the fragments are filtered and selected for further processing, in particular for copying or *cloning* (see below). We then get a collection of cloned fragments that correspond to random contiguous pieces of the purified DNA sequence. Sequencing these pieces and assembling the resulting sequences is a standard way of determining the purified DNA's sequence. This method can also be used to construct a cloning library — a collection of clones covering a certain long DNA molecule.

Copying DNA

Another very important tool needed in molecular biology research is a DNA copying process (also called DNA *amplification*). We are dealing here with molecules, which are microscopic objects. The more copies we have of a molecule, the easier it is to study it. Fortunately, several techniques have been developed for this purpose.

DNA Cloning: To do any experiment with DNA in the lab, one needs a minimum quantity of material; one molecule is clearly not enough. Yet one molecule is sometimes all we have to start with. In addition, the material should be stored in a way that permits essentially unlimited production of new material for repetition of the experiment or for new experiments as needed. DNA cloning is the name of the general technique that allows these goals to be attained.

Given a piece of DNA, one way of obtaining further copies is to use nature itself:

We insert this piece into the genome of an organism, a *host* or *vector*, and then let the organism multiply itself. Upon host multiplication, the inserted piece (the *insert*) gets multiplied along with the original DNA. We can then kill the host and dispose of the rest, keeping only the inserts in the desired quantity. DNA produced in this way is called *recombinant*. Popular vectors include *plasmids, cosmids, phages,* and *YACs*.

A plasmid is a piece of circular DNA that exists in bacteria. It is separated from and much smaller than the bacterial chromosome. It does get replicated when the cell divides, though, and each daughter cell keeps one copy of the plasmid. Plasmids make good vectors, but they place a limitation on insert size: about 15 kbp. Inserts much larger than this limit will make very big plasmids, and big plasmids tend to get shortened when replicated.

Phages are viruses often used as vectors. One example is phage λ, which infects the bacteria *E. coli*. Inserts in phage DNA get replicated when the virus infects a host colony. Phage λ normally has a DNA of 48 kbp, and inserts of up to 25 kbp are well tolerated. Inserts larger than this limit are impossible, though, because the resulting DNA will not fit in the phage protein capsule. However, if the entire phage DNA is replaced by an insert plus some minimum replicative apparatus, inserts of up to 50 kbp are feasible. These are called *cosmids*.

For very large inserts, on the range of a million base pairs, a YAC (yeast artificial chromosome) can be used. A YAC is an extra, artificially-made chromosome that can be built by adding yeast control chromosomal regions to the insert and making it look like an additional chromosome to the yeast replication mechanism.

Polymerase Chain Reaction: A way of producing many copies of a DNA molecule without cloning it is afforded by the polymerase chain reaction (PCR). DNA Polymerase is an enzyme that catalyzes elongation of a single strand of DNA, provided there is a template DNA to which this single strand is attached. Nucleotides complementary to the ones in the template strand are added until both strands have the same size and form a normal double strand of DNA. The small stretch of double stranded DNA at the beginning needed for polymerase to start its job is called a **primer**.

PCR consists basically of an alternating repetition of two phases: A phase in which double stranded DNA is separated into two single strands by heat and a phase in which each single strand thus obtained is converted into a double strand by addition of a primer and polymerase action. Each repetition doubles the number of molecules. After enough repetitions, the quantity of material produced is large enough to perform further experiments, thanks to the exponential growth of this doubling procedure.

A curious footnote here is that Kary B. Mullis, the inventor of PCR (in 1983), realized it was a good idea because he "had been spending a lot of time writing computer programs" and was thus familiar with iterative functions and thereby with exponential growth processes.

Reading and Measuring DNA

How do we actually "read" the base pairs of a DNA sequence? Reading is done with a technique known as *gel electrophoresis*, which is based on separation of molecules by their size. The process involves a gel medium and a strong electric field. DNA or

RNA molecules are charged in aqueous solution and will move to a definite direction by action of the electric field. The gel medium makes them move slowly, with speed inversely proportional to their size. All molecules are initially placed at one extremity in a gel block. After a few hours, the smaller molecules will have migrated to the other end of the block, whereas larger molecules will stay behind, near the starting place. By interpolation, the relative sizes of molecules can be calculated with good approximation.

In this kind of experiment, DNA molecules can be labeled with radioactive isotopes so that the gel can be photographed, producing a graphic record of the positions at the end of a run. This process is used in DNA sequencing, determination of restriction fragment lengths, and so on. An alternative is to use fluorescent dyes instead of radioactive isotopes. A laser beam can trace the dyes and send the information directly to a computer, avoiding the photographic process altogether. Sequencing machines have been built using this technique.

DNA or RNA bases can be read using this process with the following technique. Given a DNA molecule, it is possible to obtain all fragments from it that end at every position where an A appears. Similarly all fragments that end in T, in G, or in C can be obtained. We thus get four different test tubes, one for each base. The fragments in each tube will have differing lengths. For example, suppose the original DNA piece is the following:

GACTTAGATCAGGAAACT

The fragments that end in T are GACT, GACTT, GACTTAGAT, and the whole sequence itself. Thus if we separate these fragments by size, and we do it simultaneously but separately for all four test tubes, we will know the precise base composition of the original DNA sequence. This is illustrated in Figure 1.12.

Some errors may occur in reading a gel film, because sometimes marks are blurred, especially near the film borders. One other limitation is the size of DNA fragments that can be read in this way, which is about 700 bp.

1.6 THE HUMAN GENOME PROJECT

The Human Genome Project is a multinational effort, begun in 1988, whose aim is to produce a complete physical map of all human chromosomes, as well as the entire human DNA sequence. As part of the project, genomes of other organisms such as bacteria, yeast, flies, and mice are also being studied.

This is no easy task, since the human genome is so large. So far, many virus genomes have been entirely sequenced, but their sizes are generally in the 1 kbp to 10 kbp range. The first free-living organism to be totally sequenced was the bacterium *Haemophilus influenzae*, containing a 1800 kbp genome. In 1996 the whole sequence of the yeast genome — a 10 million bp sequence — was also determined. This was an important milestone, given that yeast is a free-living eukaryote. Mapping the human genome still seems remote, as it is 100 times bigger than the largest genome sequenced so far.

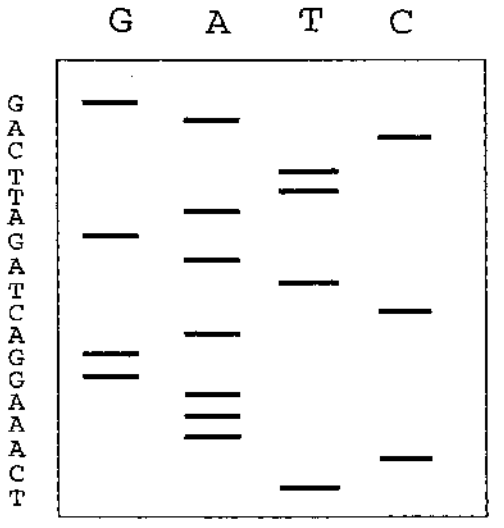

FIGURE 1.12

Schematic view of film produced by gel electrophoresis. Individual DNA bases can be identified in each of the four columns. Shorter fragments leave their mark near the top, whereas longer fragments leave their mark near the bottom.

One of the reasons why other organisms are part of the project is to perfect sequencing methods so that they can then be applied to the human genome. The cost of sequencing is also expected to drop as a result of new, improved technology. Another reason lies in research benefits. All species targeted are largely employed in genetic and molecular research.

Some lessons have been learned. A large effort like this cannot be entertained by a single lab. Rather, a consortium of first-rate labs working together is a more efficient way — and perhaps the only way — of getting the work done. Coordinating these tasks is in itself a challenge. On the computer science side, databases with updated and consistent information have to be maintained, and fast access to the data has to be provided. A major concern is with possible errors in the sequences obtained. The current target is to obtain DNA sequences with at most one error for every 10,000 bases.

Another problem is that of the "average genome." Different individuals have different genomes (this is what makes *DNA fingerprinting* possible). A gene may have many alleles in the population, and the sequence of intergenic regions (which do not code for products) may vary from person to person. It is estimated that the genomes of two distinct humans differ on average in one in every 500 bases. So the question is: Whose genome is going to be sequenced? Even if one individual is somehow chosen and her or his DNA is taken as the standard, there is the problem of transposable elements. It is now known that certain parts of the genome keep moving from one place to another, so that at best what we will get after sequencing is a "snapshot" of the genome in a given instant.

Once the entire sequence is obtained, we will face the difficult task of analyzing it.

We have to recognize the genes in it and determine the function of the proteins produced. But gene recognition is still in its infancy, and protein function determination is still a very laborious procedure. Treatment of genetic diseases based on data produced by the Human Genome Project is still far ahead, although encouraging pioneering efforts have already yielded results.

1.7 SEQUENCE DATABASES

Thanks in part to the techniques described in previous sections, large numbers of DNA, RNA, and protein sequences have been determined in the past decades. Some institutional sequence databases have been set up to harbor these sequences as well as a wealth of associated data. The rate at which new sequences are being added to these databases is exponential. Computational techniques have been developed to allow fast search on these databases, some of which are described in Chapter 3. Here we give a brief description of some representative sequence databases.

GenBank: Maintained by the National Center for Biotechnology Information (NCBI), USA, GenBank contains hundreds of thousands of DNA sequences. It is divided into several sections with sequences grouped according to species, including:

- ***PLN:*** Plant sequences
- ***PRI:*** Primate sequences
- ***ROD:*** Rodent sequences
- ***MAM:*** Other mammalian sequences
- ***VRT:*** Other vertebrate sequences
- ***IVN:*** Invertebrate sequences
- ***BCT:*** Bacterial sequences
- ***PHG:*** Phage sequences
- ***VRL:*** Other viral sequences
- ***SYN:*** Synthetic sequences
- ***UNA:*** Unannotated sequences
- ***PAT:*** Patent sequences
- ***NEW:*** New sequences

Searches can be made by keywords or by sequence. A typical entry is shown in Figure 1.13. The entry is divided into *fields,* with each field composed by a field identifier, which is a word describing the contents of the field, and the information per se. The entries are just plain text. One important field is *accession no.*, the accession number, which is a code that is unique to this entry and can be used for faster access to it. In the example, the accession number is `M12174`. Some entries have several accession numbers as a

result of combination of several related but once distinct entries into one comprehensive entry. Other fields are self-explanatory for the most part.

This database is part of an international collaboration effort, which also includes the DNA DataBank of Japan (DDBJ) and the European Molecular Biology Laboratory (EMBL). GenBank can be reached through the following locator:

http://www.ncbi.nlm.nih.gov/

EMBL: The European Molecular Biology Laboratory is an institution that maintains several sequence repositories, including a DNA database called the Nucleotide Sequence Database. Its organization is similar to that of GenBank, with the entries having roughly the same fields. In the EMBL database, entries are identified by two-letter codes (see Figure 1.14). The accession number, for instance, is identified by the letters AC. Code XX indicates blank lines. Both GenBank and EMBL separate the sequences in blocks of ten characters, with six blocks per line. This scheme makes it easy to find specific positions within the sequence. EMBL can be reached through the following locator:

http://www.embl-heidelberg.de/

PIR: The Protein Identification Resource (PIR) is a database of protein sequences cooperatively maintained and distributed by three institutions: the National Biomedical Research Foundation (in the USA), the Martinsried Institute for Protein Sequences (in Europe), and the Japan International Protein Information Database (in Japan). A typical entry appears in Figure 1.15. Several web sites provide interfaces to this database, including the following:

http://www.gdb.org/
http://www.mips.biochem.mpg.de/

PDB: The Protein Data Bank is a repository of three-dimensional structures of proteins. For each protein represented, a general information header is provided followed by a list of all atoms present in the structure, with three spatial coordinates for each atom that indicate their position with three decimal places. An example is given in Figures 1.16 and 1.17. This repository is maintained in Brookhaven, USA. Access to it can be gained through

http://www.pdb.bnl.gov/

Other Databases: Apart from those mentioned above, many other databases have been created to keep molecular biology information. Among them we cite ACEDB, a powerful platform prepared for the *C. elegans* sequencing project but adaptable to similar projects; Flybase, a database of fly sequences; and databases for restriction enzymes, codon preferential usage, and so on.

```
LOCUS       HUMRHOA     539 bp   mRNA          PRI     04-AUG-1986
DEFINITION  Human ras-related rho mRNA (clone 6), partial cds.
ACCESSION   M12174
KEYWORDS    c-myc proto-oncogene; ras oncogene; rho gene.
SOURCE      Human peripheral T-cell, cDNA to mRNA, clone 6.
  ORGANISM  Homo sapiens
            Eukaryota; Animalia; Chordata; Vertebrata; Mammalia;
            Theria; Eutheria; Primates; Haplorhini; Catarrhini;
            Hominidae.
REFERENCE   1  (bases 1 to 539)
  AUTHORS   Madaule,P.
  JOURNAL   Unpublished (1985) Columbia U, 701 W 168th St,
            New York, NY 10032
REFERENCE   2  (bases 1 to 539)
  AUTHORS   Madaule,P. and Axel,R.
  TITLE     A novel ras-related gene family
  JOURNAL   Cell 41, 31-40 (1985)
  MEDLINE   85201682
COMMENT     [2] has found and sequenced a family of highly
            evolutionarily conserved genes with homology to the
            ras family (H-ras, K-ras, N-ras) of oncogenes.
            [2] named this family rho (for ras homology).
            In humans at least three distinct rho genes are
            present. A draft entry and computer-readable copy of
            this sequence were kindly
            provided by P.Madaule (07-OCT-1985).

            NCBI gi: 337392
FEATURES             Location/Qualifiers
     source          1..539
                     /organism="Homo sapiens"
     CDS             <1..509
                     /note="rho protein;  NCBI gi: 337393"
                     /codon_start=2
BASE COUNT      105 a    180 c    172 g     82 t
ORIGIN      185 bp upstream of HinfI site.
        1 cgagttcccc gaggtgtacg tgc .... tatgtggccg acattgaggt
       61 ggacggcaag caggtggagc tgg .... ggccaggagg actacgaccg
      121 cctgcggccg ctctcctacc cgg .... atgtgcttct cggtggacag
      181 cccggactcg ctggagaaca tcc .... gaggtgaagc acttctgtcc
      ...
      421 cgaggtcttc gagacggcca cgc .... cgctacggct cccagaacgg
      481 ctgcatcaac tgctgcaagg tgc .... cgcgcctgcc cctgccggc
//
```

FIGURE 1.13

Typical GenBank entry. The actual file was edited to fit the page. In a real entry each sequence line has 60 characters, with the possible exception of the last line.

```
ID   ECTRGA      standard; RNA; PRO; 75 BP.
XX
AC   M24860;
XX
DT   24-APR-1990 (Rel. 23, Created)
DT   31-MAR-1992 (Rel. 31, Last updated, Version 3)
XX
DE   E.coli Gly-tRNA.
XX
KW   transfer RNA-Gly.
XX
OS   Escherichia coli
OC   Prokaryota; Bacteria; Gracilicutes; Scotobacteria;
OC   Facultatively anaerobic rods; Enterobacteriaceae;
OC   Escherichia.
XX
RN   [1]
RP   1-75
RA   Carbon J., Chang S., Kirk L.L.;
RT   "Clustered tRNA genes in Escherichia coli: Transcription
RT   and processing";
RL   Brookhaven Symp. Biol. 26:26-36(1975).
XX
FH   Key             Location/Qualifiers
FH
FT   tRNA            1..75
FT                   /note="Gly-tRNA"
XX
SQ   Sequence 75 BP; 13 A; 24 C; 19 G; 19 T; 0 other;
     gcgggcatcg tataatggct attacctcag ... tgatgatgcg ggttcgattc
     ccgctgcccg ctcca
//
```

FIGURE 1.14

A typical EMBL entry, edited to fit the page. Actual entries have 60 characters per line in the sequence section, with the possible exception of the last line.

```
PIR1:CCHP
cytochrome c - hippopotamus

Species: Hippopotamus amphibius (hippopotamus)

Date: 19-Feb-1984 #sequence_revision 19-Feb-1984 #text_change
      05-Aug-1994

Accession: A00008

Thompson, R.B.; Borden, D.; Tarr, G.E.; Margoliash, E.
    J. Biol. Chem. 253, 8957-8961, 1978
    Title: Heterogeneity of amino acid sequence in hippopotamus
      cytochrome c.
    Reference number: A00008; MUID:79067782
    Accession: A00008
    Molecule type: protein
    Residues: 1-104 <THO>
    Note: 3-Ile was also found

Superfamily: cytochrome c; cytochrome c homology

Keywords: acetylated amino end; electron transfer; heme;
    mitochondrion; oxidative phosphorylation; respiratory chain

Residues            Feature
1                   Modified site: acetylated amino end (Gly)
                      #status predicted
14,17               Binding site: heme (Cys) (covalent) #status
                      predicted
18,80               Binding site: heme iron (His, Met) (axial
                      ligands) #status predicted

                        Composition
          6 Ala  A      4 Gln  Q      6 Leu  L      2 Ser  S
          2 Arg  R      8 Glu  E     17 Lys  K      8 Thr  T
          5 Asn  N     14 Gly  G      2 Met  M      1 Trp  W
          3 Asp  D      3 His  H      4 Phe  F      4 Tyr  Y
          2 Cys  C      6 Ile  I      4 Pro  P      3 Val  V
    Mol. wt. unmod. chain = 11,530   Number of residues = 104

           5        10        15        20        25        30
  1 G D V E K G K K I F V Q K C A Q C H T V E K G G K H K T G P
 31 N L H G L F G R K T G Q S P G F S Y T D A N K N K G I T W G
 61 E E T L M E Y L E N P K K Y I P G T K M I F A G I K K K G E
 91 R A D L I A Y L K Q A T N E
```

FIGURE 1.15

A typical PIR entry, edited to fit the page.

```
HEADER    MYOGLOBIN (CARBONMONOXY)                19-JUL-95   1MCY
TITLE     SPERM WHALE MYOGLOBIN (MUTANT WITH INITIATOR MET AND
TITLE    2 WITH HIS 64 REPLACED BY GLN, LEU 29 REPLACED BY PHE
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: MYOGLOBIN (CARBONMONOXY);
COMPND   3 CHAIN: NULL;
COMPND   4 ENGINEERED: YES;
COMPND   5 MUTATION: INS(MET 0), F29L, Q64H, N122D
SOURCE    MOL_ID: 1;
SOURCE   2 SYNTHETIC: SYNTHETIC GENE;
SOURCE   3 ORGANISM_SCIENTIFIC: PHYSETER CATODON;
SOURCE   4 ORGANISM_COMMON: SPERM WHALE;
SOURCE   5 EXPRESSION_SYSTEM: ESCHERICHIA COLI
KEYWDS    HEME, OXYGEN TRANSPORT, RESPIRATORY PROTEIN
EXPDTA    X-RAY DIFFRACTION
AUTHOR    T.LI,G.N.PHILLIPS JUNIOR
REVDAT   1   07-DEC-95 1MCY     0
JRNL        AUTH   X.ZHAO,K.VYAS,B.D.NGUYEN,K.RAJARATHNAM,
JRNL        AUTH 2 G.N.LAMAR,T.LI,G.N.PHILLIPS JUNIOR,R.EICH,
JRNL        AUTH 3 J.S.OLSON,J.LING,D.F.BOCIAN
JRNL        TITL   A DOUBLE MUTANT OF SPERM WHALE MYOGLOBIN
JRNL        TITL 2 MIMICS THE STRUCTURE AND FUNCTION OF
JRNL        TITL 3 ELEPHANT MYOGLOBIN
JRNL        REF    J.BIOL.CHEM.                V. 270 20763 1995
JRNL        REFN   ASTM JBCHA3  US ISSN 0021-9258          0071
REMARK   1
REMARK   2
REMARK   2 RESOLUTION. 1.7  ANGSTROMS.
REMARK   3
REMARK   3 REFINEMENT.
REMARK   3   PROGRAM                      X-PLOR
REMARK   3   AUTHORS                      BRUNGER
REMARK   3   R VALUE                      0.182
REMARK   3   RMSD BOND DISTANCES          0.020  ANGSTROMS
REMARK   3   RMSD BOND ANGLES             1.82   DEGREES
REMARK   3
REMARK   3   NUMBER OF REFLECTIONS        23187
REMARK   3   RESOLUTION RANGE        5.0 - 1.7  ANGSTROMS
REMARK   3   DATA CUTOFF                 0.0    SIGMA(F)
REMARK   3
REMARK   3  DATA COLLECTION.
REMARK   3   NUMBER OF UNIQUE REFLECTIONS       25787
REMARK   3   RESOLUTION RANGE  INFINITY - 1.7  ANGSTROMS
REMARK   3   COMPLETENESS OF DATA               95.8   %
```

FIGURE 1.16

A typical PDB entry, partial header. The last columns containing the characters 1MCY *i, where i is the line number, are omitted. Some other editing was done to fit the page.*

```
ATOM      1  N   MET  0      24.486   8.308  -9.406  1.00 37.00
ATOM      2  CA  MET  0      24.542   9.777  -9.621  1.00 36.40
ATOM      3  C   MET  0      25.882  10.156 -10.209  1.00 34.30
ATOM      4  O   MET  0      26.833   9.391 -10.078  1.00 34.80
ATOM      5  CB  MET  0      24.399  10.484  -8.303  1.00 39.00
ATOM      6  CG  MET  0      24.756   9.581  -7.138  1.00 41.80
ATOM      7  SD  MET  0      24.017  10.289  -5.719  1.00 44.80
ATOM      8  CE  MET  0      24.761  12.009  -5.824  1.00 41.00
ATOM      9  N   VAL  1      25.951  11.334 -10.816  1.00 31.10
ATOM     10  CA  VAL  1      27.185  11.834 -11.382  1.00 28.40
ATOM     11  C   VAL  1      27.330  13.341 -11.124  1.00 26.30
ATOM     12  O   VAL  1      26.444  14.135 -11.452  1.00 26.60
ATOM     13  CB  VAL  1      27.270  11.547 -12.912  1.00 29.30
ATOM     14  CG1 VAL  1      28.532  12.207 -13.526  1.00 28.60
ATOM     15  CG2 VAL  1      27.275  10.038 -13.163  1.00 29.70
ATOM     16  N   LEU  2      28.435  13.739 -10.500  1.00 23.00
ATOM     17  CA  LEU  2      28.691  15.142 -10.318  1.00 20.80
ATOM     18  C   LEU  2      29.289  15.737 -11.599  1.00 20.10
ATOM     19  O   LEU  2      30.129  15.110 -12.276  1.00 19.50
ATOM     20  CB  LEU  2      29.661  15.356  -9.134  1.00 20.90
ATOM     21  CG  LEU  2      29.036  15.387  -7.726  1.00 20.40
ATOM     22  CD1 LEU  2      28.556  13.983  -7.402  1.00 19.60
ATOM     23  CD2 LEU  2      30.058  15.904  -6.689  1.00 19.50
ATOM     24  N   SER  3      28.996  17.003 -11.826  1.00 18.80
ATOM     25  CA  SER  3      29.696  17.733 -12.852  1.00 19.40
ATOM     26  C   SER  3      31.096  18.141 -12.385  1.00 19.50
ATOM     27  O   SER  3      31.397  18.121 -11.174  1.00 19.10
ATOM     28  CB  SER  3      28.861  18.954 -13.223  1.00 20.60
ATOM     29  OG  SER  3      29.019  19.969 -12.261  1.00 22.10
ATOM     30  N   GLU  4      31.947  18.561 -13.310  1.00 18.40
ATOM     31  CA  GLU  4      33.293  18.956 -12.937  1.00 19.00
ATOM     32  C   GLU  4      33.173  20.215 -12.047  1.00 19.20
ATOM     33  O   GLU  4      34.026  20.457 -11.206  1.00 19.10
ATOM     34  CB  GLU  4      34.135  19.270 -14.198  1.00 19.60
ATOM     35  CG  GLU  4      35.491  19.937 -13.932  1.00 21.10
ATOM     36  CD  GLU  4      36.537  19.020 -13.295  1.00 22.60
ATOM     37  OE1 GLU  4      36.355  17.787 -13.230  1.00 23.90
ATOM     38  OE2 GLU  4      37.569  19.550 -12.840  1.00 24.60
ATOM     39  N   GLY  5      32.182  21.062 -12.313  1.00 18.30
ATOM     40  CA  GLY  5      32.039  22.287 -11.532  1.00 19.10
ATOM     41  C   GLY  5      31.669  22.025 -10.056  1.00 18.20
ATOM     42  O   GLY  5      32.251  22.638  -9.140  1.00 18.60
```

FIGURE 1.17

A typical PDB entry, coordinates. The last columns, containing the characters 1MCY *i, where i is the line number, are omitted. A few blank characters in each line were removed.*

EXERCISES

1. Consider the sequence TAATCGAATGGGC. Derive the six possible protein sequences translated from it.
2. Given the fictitious "gene" below, find
   a. the sequence of the corresponding mRNA.
   b. the sequence of the resulting protein (use the standard genetic code).

   ATGATACCGACGTACGGCATTTAA
   TACTATGGCTGCATGCCGTAAATT
3. Given the protein sequence LMK, how many DNA sequences could possibly have given rise to it? *Hint:* One of them is CTGATGAAG.
4. Find the cut patterns of restriction enzymes *Bam*HI and *Hind*III.
5. Suppose we have a DNA molecule of length 40,000 and digest it with a 4-cutter restriction enzyme. Assuming a random distribution of bases in the molecule, how many pieces can we expect to get?
6. Browse through the GenBank viral sequences section and find a disease-causing agent you have never heard of before.

BIBLIOGRAPHIC NOTES

Readers completely new to molecular biology may wish to take a look at the introductory "comic-book" by Rosenfeld, Ziff, and van Loon [165]. Readers who wish to go beyond the overview we have given should look at the standard references in molecular biology, some of which are:

- Watson, Hopkins, Roberts, Steitz, and Weiner [203, 204].
- Alberts, Bray, Lewis, Raff, Roberts, and Watson [7].
- Lewin [124].

The reader should keep in mind that such books are constantly being updated, due to the rapid progress in molecular biology research.

Other textbooks that we found useful in preparing this chapter were as follows. For biochemistry, Mathews and van Holde [131]; for genetics, Tamarin [183]; for protein structures, Branden and Tooze [28].

The quote from R. F. Doolittle on page 2 is from [50], which is a good review of proteins for the "educated layperson," and appears on a special issue dedicated to the "molecules of life." An entertaining account of the discovery of PCR by Mullis appears in [141]. The quote on page 20 was taken from that article.

Robbins [162] and Frenkel [66] provide articles discussing the difficulties of the Human Genome Project from the point of view of computer science. Robbins has emphasized the view of an organism as the result of a complex interaction of many "cell processes." Lewontin [125] presents a sobering critique of the Human Genome Project.

The problems with the "genetic program metaphor" were pointed out by many people; our source was the paper by Atlan and Koppel [17]. On the other hand, the interested reader may wish to take a look at the book by Hofstadter [96], which contains an interesting discussion of parallels between molecular biology and mathematical logic.

REFERENCES

1. M. A. Adams, C. Fields, and J. C. Venter, editors. *Automated DNA Sequencing and Analysis*. New York: Academic Press, 1994.
2. L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.
3. R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 365–372, 1996.
4. R. Agarwala and D. Fernández-Baca. A polynomial-time algorithm for the phylogeny problem when the number of character states is fixed. *SIAM Journal on Computing*, 23(6):1216–1224, 1994.
5. R. Agarwala, D. Fernandez-Baca, and G. Slutzki. Fast algorithms for inferring evolutionary trees. In *Proceedings of the 30th Allerton Conference on Communication, Control, and Computation*, pages 594–603, 1992.
6. A. V. Aho. Algorithms for finding patterns in strings. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, pages 255–300. Amsterdam/Cambridge, MA: Elsevier/MIT Press, 1990.
7. B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular Biology of the Cell*. New York & London: Garland Publishing, 1994.
8. F. Alizadeh, R. M. Karp, L. A. Newberg, and D. K. Weisser. Physical mapping of chromosomes: A combinatorial problem in molecular biology. *Algorithmica*, 13(1/2):52–76, 1995.
9. F. Alizadeh, R. M. Karp, D. K. Weisser, and G. Zweig. Physical mapping of chromossomes using unique probes. *Journal of Computational Biology*, 2(2):159–184, 1995.
10. S. F. Altschul. Amino acid substitution matrices from an information theoretical perspective. *Journal of Molecular Biology*, 219:555–565, 1991.
11. S. F. Altschul, M. S. Boguski, W. Gish, and J. C. Wootton. Issues in searching molecular sequence databases. *Nature Genetics*, 6:119–129, Feb. 1994.
12. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.
13. S. F. Altschul and D. J. Lipman. Trees, stars, and multiple biological sequence alignment. *SIAM Journal on Applied Mathematics*, 49(1):197–209, 1989.
14. A. Amir and D. Keselman. Maximum agreement subtrees in multiple evolutionary trees. In

Proceedings of the IEEE Thirty-Fifth Annual Symposium on Foundations of Computer Science, pages 758–769, 1994.

15. C. Armen and C. Stein. Short superstrings and the structure of overlapping strings. *Journal of Computational Biology*, 2(2):307–332, 1995.

16. C. Armen and C. Stein. A 2 2/3-approximation algorithm for the shortest superstring problem. In *Proceedings of the Seventh Symposium on Combinatorial Pattern Matching*, volume 1075 of *Lecture Notes in Computer Science*, pages 87–103. Berlin: Springer-Verlag, 1996.

17. H. Atlan and M. Koppel. The cellular computer DNA: program or data? *Bulletin of Mathematical Biology*, 52(3):335–348, 1990.

18. V. Bafna and P. A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.

19. G. I. Bell. The human genome: an introduction. In Bell and Marr [20], pages 3–12.

20. G. I. Bell and T. M. Marr, editors. *Computers and DNA*. Reading, MA: Addison-Wesley, 1990.

21. B. Berger. Algorithms for protein structural motif recognition. *Journal of Computational Biology*, 2(1):125–138, 1995.

22. B. Berger and D. B. Wilson. Improved algorithms for protein motif recognition. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 58–67, 1995.

23. P. Berman and S. Hannenhalli. Fast sorting by reversal. In *Proceedings of the Seventh Symposium on Combinatorial Pattern Matching*, volume 1075 of *Lecture Notes in Computer Science*, pages 168–185. Berlin: Springer-Verlag, 1996.

24. A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis. Linear approximation of shortest superstrings. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, pages 328–336, 1991.

25. H. L. Bodlaender, M. R. Fellows, and T. J. Warnow. Two strikes against perfect phylogeny. In *Proceedings of the International Colloquium on Automata, Languages and Programming*, volume 623 of *Lecture Notes in Computer Science*, pages 273–283. Berlin: Springer-Verlag, 1993.

26. H. L. Bodlaender and T. Kloks. A simple linear time algorithm for triangulating three-colored graphs. *Journal of Algorithms*, 15:160–172, 1993.

27. K. S. Booth and G. S. Lueker. Testing of the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976.

28. C. Branden and J. Tooze. *Introduction to protein structure*. New York & London: Garland Publishing, 1991.

29. P. Buneman. A characterisation of rigid circuit graphs. *Discrete Mathematics*, 9:205–212, 1974.

30. A. Caprara. Sorting by reversals is difficult. In *Proceedings of the First Annual International Conference on Computational Molecular Biology*, 1997.

31. A. Caprara, G. Lancia, and S.-K. Ng. A column-generation-based branch-and-bound algorithm for sorting by reversals. Presented at the 4th DIMACS Implementation Challenge Workshop, 1995.

32. H. Carrillo and D. Lipman. The multiple sequence alignment problem in biology. *SIAM Journal on Applied Mathematics*, 48(5):1073–1082, Oct. 1988.

33. S. C. Chan, A. K. C. Wong, and D. K. Y. Chiu. A survey of multiple sequence comparison methods. *Bulletin of Mathematical Biology*, 54(4):563–598, 1992.

34. W. I. Chang and E. L. Lawler. Approximate string matching in sublinear expected time. In *Proceedings of the IEEE Thirty-First Annual Symposium on Foundations of Computer Science*, pages 116–124, 1990.

35. K.-M. Chao and W. Miller. Linear-space algorithms that build local alignments from fragments. *Algorithmica*, 13:106–134, 1995.

36. K.-M. Chao, J. Zhang, J. Ostell, and W. Miller. A local alignment tool for very long DNA sequences. *Computer Applications in the Biosciences*, 11(2):147–153, 1995.

37. T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. Cambridge, MA/New York: MIT Press/McGraw-Hill, 1990.

38. T. E. Creighton. Protein folding. *Biochemistry Journal*, 270:1–16, 1990.

39. M. Crochemore and W. Rytter. *Text Algorithms*. Oxford: Oxford University Press, 1994.

40. J. C. Culberson and P. Rudnicki. A fast algorithm for constructing trees from distance matrices. *Information Processing Letters*, 30:215–220, 1989.

41. A. Czumaj, L. Gasienec, M. Piotrow, and W. Rytter. Parallel and sequential approximations of shortest superstrings. In *Proceedings of the Fourth Scandinavian Workshop on Algorithm Theory*, pages 95–106, 1994.

42. W. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49(4):461–467, 1987.

43. W. E. Day, D. S. Johnson, and D. Sankoff. The computational complexity of inferring rooted phylogenies by parsimony. *Mathematical Biosciences*, 81:33–42, 1986.

44. W. E. Day and D. Sankoff. Computational complexity of inferring phylogenies by compatibility. *Systematic Zoology*, 35(2):224–229, 1986.

45. M. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. In M. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, volume 5, pages 345–352. National Biomedical Research Foundation, Silver Spring, MD, 1978. Supplement 3.

46. S. Dean and R. Staden. A sequence assembly and editing program for efficient management of large projects. *Nucleic Acids Research*, 19(14):3907–3911, 1991.

47. J. Devereux, P. Haeberli, and D. Smithies. A comprehensive set of sequence analysis programs for the VAX. *Nucleic Acids Research*, 12:387–395, 1984.

48. K. A. Dill. Dominant forces in protein folding. *Biochemistry*, 29(31):7133–7155, 1990.

49. A. J. Dobson. Unrooted trees for numerical taxonomy. *Journal of Applied Probability*, 11:32–42, 1974.

50. R. F. Doolittle. Proteins. *Scientific American*, 253(4):74–83, Oct. 1985.

51. R. F. Doolittle, editor. *Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences*, volume 183 of *Methods in Enzymology*. New York: Academic Press, 1990.

52. A. Dress and M. Steel. Convex tree realizations of partitions. *Applied Mathematics Letters*, 5(3):3–6, 1992.

53. M. L. Engle and C. Burks. Artificially generated data sets for testing DNA fragment assembly algorithms. *Genomics*, 16:286–288, 1993.

54. M. L. Engle and C. Burks. Genfrag 2.1: New features for more robust fragment assembly benchmarks. *Computer Applications in the Biosciences*, 10:567–568, 1994.

55. D. Eppstein, Z. Galil, and R. Giancarlo. Speeding up dynamic programming. In *Proceedings of the IEEE Twenty-Ninth Annual Symposium on Foundations of Computer Science*, pages 488–495, 1988.

56. M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13:155–179, 1995.

57. M. Farach, T. M. Przytycka, and M. Thorup. On the agreement of many trees. Unpublished manuscript, 1995.

58. C. Fauron and M. Havlik. The maize mitochondrial genome of the normal type and the cytoplasmic male sterile type have very different organization. *Current Genetics*, 15:149–154, 1989.

59. M. R. Fellows, M. T. Hallett, and H. T. Wareham. DNA physical mapping: Three ways difficult. In *Proceedings of the First Annual European Symposium on Algorithms*, volume 726 of *Lecture Notes in Computer Science*, pages 157–168. Berlin: Springer-Verlag, 1993.

60. J. Felsenstein. Numerical methods for inferring evolutionary trees. *The Quarterly Review of Biology*, 57(4):379–404, 1982.

61. J. Felsenstein. PHYLIP — Phylogeny Inference Package (Version 3.2). *Cladistics*, 5:164–166, 1989.

62. V. Ferreti, J. H. Nadeau, and D. Sankoff. Original synteny. In *Proceedings of the Seventh Symposium on Combinatorial Pattern Matching*, number 1075 in Lecture Notes on Computer Science, pages 159–167. Berlin: Springer-Verlag, 1996.

63. J. Fickett. Fast optimal alignment. *Nucleic Acids Research*, 12(1):175–179, 1984.

64. L. R. Foulds and R. L. Graham. The Steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics*, 3:43–49, 1982.

65. A. S. Fraenkel. Complexity of protein folding. *Bulletin of Mathematical Biology*, 55(6):1199–1210, 1993.

66. K. A. Frenkel. The human genome project and informatics. *Communications of the ACM*, 34(11), Nov. 1991.

67. D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15(3):835–855, 1965.

68. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.

69. W. H. Gates and C. H. Papadimitriou. Bounds for sorting by prefix reversal. *Discrete Mathematics*, 27:47–57, 1979.

70. D. G. George, W. C. Barker, and L. T. Hunt. Mutation data matrix and its uses. In Doolittle [51], pages 333–351.

71. T. Gingerias, J. Milazzo, D. Sciaky, and R. Roberts. Computer programs for assembly of DNA sequences. *Nucleic Acids Research*, 7:529–545, 1979.

72. P. W. Goldberg, M. C. Golumbic, H. Kaplan, and R. Shamir. Three strikes against physical mapping of DNA. Unpublished manuscript, 1993.

73. L. Goldstein and M. S. Waterman. Mapping DNA by stochastic relaxation. *Advances in Applied Mathematics*, 8:194–207, 1987.

74. M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. New York: Academic Press, 1980.

75. M. C. Golumbic, H. Kaplan, and R. Shamir. On the complexity of physical mapping. *Advances in Applied Mathematics*, 15:251–261, 1994.

76. G. H. Gonnet and R. Baeza-Yates. *Handbook of Algorithms and Data Structures*, 2nd ed. Reading, MA: Addison-Wesley, 1991.

77. O. Gotoh. Optimal alignments between groups of sequences and its application to multiple sequence alignment. *Computer Applications in the Biosciences*, 9(3):361–370, 1993.

78. D. Greenberg and S. Istrail. The chimeric mapping problem: Algorithmic strategies and performance evaluation on synthetic genomic data. *Computers and Chemistry*, 18(3):207–220, 1994.

79. D. Greenberg and S. Istrail. Physical mapping by STS hybrdization: Algorithmic strategies and the challenge of software evaluation. *Journal of Computational Biology*, 2(2):219–274, 1995.

80. A. Grigoriev, R. Mott, and H. Lehrach. An algorithm to detect chimeric clones and random noise in genomic mapping. *Genomics*, 22:482–486, 1994.

81. S. K. Gupta, J. Kececioglu, and A. A. Schäffer. Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *Journal of Computational Biology*, 2(3):459–472, 1995.

82. D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21:19–28, 1991.

83. D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds. *Bulletin of Mathematical Biology*, 55(1):141–154, 1993.

84. D. Gusfield. Faster implementation of a shortest superstring approximation. *Information Processing Letters*, 51:271–274, 1994.

85. D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge, UK: Cambridge University Press, 1997. Forthcoming.

86. D. Gusfield, G. M. Landau, and B. Schieber. An efficient algorithm for the all pairs suffix–prefix problem. *Information Processing Letters*, 41:181–185, 1992.

87. S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 178–189, 1995.

88. S. Hannenhalli and P. A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the IEEE Thirty-Sixth Annual Symposium on Foundations of Computer Science*, pages 581–592, 1995.

89. J. Hein. A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given. *Molecular Biology and Evolution*, 6(6):649–668, 1989.

90. J. Hein. An optimal algorithm to reconstruct trees from additive distance data. *Bulletin of Mathematical Biology*, 51(5):597–603, 1989.

91. J. Hein. A tree reconstruction method that is economical in the number of pairwise comparisons used. *Molecular Biology and Evolution*, 6(6):669–684, 1989.

92. J. Hein. Unified approach to alignment and phylogenies. In Doolittle [51], pages 626–645.

93. S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the U.S.A.*, 89:10915–10919, 1992.

94. D. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18:341–343, 1975.

95. R. J. Hoffmann, J. L. Boore, and W. M. Brown. A novel mitochondrial genome organization for the blue mussel, *mytilus edulis*. *Genetics*, 131:397–412, 1992.

96. D. Hofstadter. *Gödel, Escher, Bach*. New York: Basic Books, 1979.

97. W.-L. Hsu. A simple test for the consecutive ones property. In *Proceedings of the International Symposium on Algorithms & Computation* (ISAAC), 1992.

98. X. Huang. A contig assembly program based on sensitive detection of fragment overlaps. *Genomics*, 14:18–25, 1992.

99. X. Huang. An improved sequence assembly program. *Genomics*, 33:21–31, 1996.

100. X. Huang, R. C. Hardison, and W. Miller. A space-efficient algorithm for local similarities. *Computer Applications in the Biosciences*, 6(4):373–381, 1990.

101. R. Idury and A. Schäffer. Triangulating three-colored graphs in linear time and linear space. *SIAM Journal on Discrete Mathematics*, 6(2), 1993.

102. T. Jiang, E. Lawler, and L. Wang. Aligning sequences via an evolutionary tree: complexity and approximation. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 760–769, 1994.

103. R. Jones, W. Taylor, IV, X. Zhang, J. P. Mesirov, and E. Lander. Protein sequence comparison on the connection machine CM-2. In Bell and Marr [20], pages 99–108.

104. D. Joseph, J. Meidanis, and P. Tiwari. Determining DNA sequence similarity using maximum independent set algorithms for interval graphs. In *Proceedings of the Third Scandinavian Workshop on Algorithm Theory*, volume 621 of *Lecture Notes in Computer Science*, pages 326–337. Berlin: Springer-Verlag, 1992.

105. M. Kanehisa and C. DeLisi. The prediction of a protein and nucleic acid structure: problems and prospects. In G. Koch and M. Hazewinkel, editors, *Mathematics of Biology*, pages 115–137. Dordrecht: D. Reidel, 1985.

106. S. K. Kannan and T. J. Warnow. Triangulating 3-colored graphs. *SIAM Journal on Discrete Mathematics*, 5(2):249–258, 1992.

107. S. K. Kannan and T. J. Warnow. Inferring evolutionary history from DNA sequences. *SIAM Journal on Computing*, 23(4):713–737, 1994.

108. H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping. In *Proceedings of the IEEE Thirty-Fifth Annual Symposium on Foundations of Computer Science*, pages 780–791, 1994.

109. S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the U.S.A.*, 87:2264–2268, 1990.

110. S. Karlin, A. Dembo, and T. Kawabata. Statistical composition of high-scoring segments from molecular sequences. *Annals of Statistics*, 18(2):571–581, 1990.

111. R. M. Karp. Mapping the genome: some combinatorial problems arising in molecular biology. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 278–285, 1993.

112. R. M. Karp, C. Kenyon, and O. Waarts. Error-resilient DNA computation. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 458–467, 1996.

113. J. Kececioglu. The maximum weight trace problem in multiple sequence alignment. In *Proceedings of the Fourth Symposium on Combinatorial Pattern Matching*, volume 684 of *Lecture Notes in Computer Science*, pages 106–119. Berlin: Springer-Verlag, 1993.

114. J. D. Kececioglu. *Exact and approximation algorithms for DNA sequence reconstruction*. Ph.D. thesis, University of Arizona, 1991.

115. J. D. Kececioglu and E. W. Myers. Combinatorial algorithms for DNA sequence assembly. *Algorithmica*, 13:7–51, 1995.

116. J. D. Kececioglu and D. Sankoff. Exact and approximate algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13:180–210, 1995.

117. E. V. Koonin and V. V. Dolja. Evolution and taxonomy of positive-strand RNA viruses: implications of comparative analysis of amino acid sequences. *Critical Reviews in Biochemistry and Molecular Biology*, 28(5):375–430, 1993.

118. R. Kosaraju, J. Park, and C. Stein. Long tours and short superstrings. In *Proceedings of the IEEE Thirty-Fifth Annual Symposium on Foundations of Computer Science*, pages 166–177, 1994.

119. E. S. Lander. Analysis with restriction enzymes. In Waterman [196], pages 35–51.

120. E. S. Lander and M. S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2:231–239, 1988.

121. L. Larmore and B. Schieber. On-line dynamic programming with applications to the prediction of RNA secondary structure. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 503–512, 1990.

122. R. H. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Engineering*, 7(9):1059–1068, 1994.

123. R. H. Lathrop and T. F. Smith. Global optimum protein threading with gapped alignment and empirical pair score functions. *Journal of Molecular Biology*, 255(4):641–665, 1996.

124. B. Lewin. *Genes V*. Oxford: Oxford University Press, 1994.

125. R. Lewontin. *Biology as Ideology*. New York: HarperPerennial, 1993.

126. M. Li. Towards a DNA sequencing theory (learning a string). In *Proceedings of the IEEE Thirty-First Annual Symposium on Foundations of Computer Science*, pages 125–134, 1990.

127. D. J. Lipman and W. R. Pearson. Rapid and sensitive protein similarity search. *Science*, 227:1435–1441, 1985.

128. R. J. Lipton. Using DNA to solve NP-complete problems. *Science*, 268:542–545, 1995.

129. U. Manber. *Introduction to Algorithms*. Reading, MA: Addison-Wesley, 1989.

130. U. Manber and E. W. Myers. Suffix arrays: A new method for on-line string searches. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 319–327, 1990.

131. C. K. Mathews and K. E. van Holde. *Biochemistry*. Redwood City, CA: Benjamin/Cummings, 1990.

132. F. R. McMorris. On the compatibility of binary qualitative taxonomic characters. *Bulletin of Mathematical Biology*, 39:133–138, 1977.

133. F. R. McMorris, T. Warnow, and T. Wimer. Triangulating vertex-colored graphs. *SIAM Journal on Discrete Mathematics*, 7(2), May 1994.

134. J. Meidanis. Distance and similarity in the presence of nonincreasing gap-weighting functions. In *Proceedings of the Second South American Workshop on String Processing*, pages 27–37, Valparaíso, Chile, Apr. 1995.

135. J. Meidanis and E. G. Munuera. A simple linear time algorithm for binary phylogeny. In *Proceedings of the Fifteenth International Conference of the Chilean Computing Society*, pages 275–283, 1995.

136. J. Meidanis and E. G. Munuera. A theory for the consecutive ones property. In *Proceedings of the Third South American Workshop on String Processing*, volume 4 of *International Informatics Series*, pages 194–202. Carleton University Press, 1996.

137. J. Meidanis and J. C. Setubal. Multiple alignment of biological sequences with gap flexibility. In *Proceedings of Latin American Theoretical Informatics*, volume 911 of *Lecture Notes in Computer Science*, pages 411–426. Berlín: Springer-Verlag, 1995.

138. J. Messing, R. Crea, and P. H. Seeburg. A system for shotgun DNA sequencing. *Nucleic Acids Research*, 9:309–321, 1981.

139. W. Miller. Building multiple alignments from pairwise alignments. *Computer Applications in the Biosciences*, 9(2):169–176, 1993.

140. W. Miller and E. W. Myers. Sequence comparison with concave weighting functions. *Bulletin of Mathematical Biology*, 50(2):97–120, 1988.

141. K. B. Mullis. The unusual origin of the polymerase chain reaction. *Scientific American*, 262(4):56–65, Apr. 1990.

142. E. W. Myers. An $O(ND)$ difference algorithm and its variations. *Algorithmica*, 1:251–266, 1986.

143. E. W. Myers. Advances in sequence assembly. In Adams et al. [1], pages 231–238.

144. E. W. Myers. Toward simplifying and accurately formulating fragment assembly. *Journal of Computational Biology*, 2(2):275–290, 1995.

145. E. W. Myers and W. Miller. Optimal alignments in linear space. *Computer Applications in the Biosciences*, 4(1):11–17, 1988.

146. S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

147. M. Nei. *Molecular Evolutionary Genetics*. New York: Columbia University Press, 1987.

148. J. D. Palmer. Chloroplast DNA evolution and biosystematic uses of chloroplast DNA variation. *The American Naturalist*, 130:S6–S29, 1987. Supplement.

149. J. D. Palmer and L. A. Herbon. Unicircular structure of the *brassica hirta* mitochondrial genome. *Current Genetics*, 11:565–570, 1987.

150. J. D. Palmer, B. Osorio, and W. F. Thompson. Evolutionary significance of inversions in legume chloroplast DNAs. *Current Genetics*, 14:65–74, 1988.

151. C. H. Papadimitriou. *Computational Complexity*. Reading, MA: Addison-Wesley, 1994.

152. C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

153. W. R. Pearson. Rapid and sensitive sequence comparison with FASTP and FASTA. In Doolittle [51], pages 63–98.

154. W. R. Pearson. Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics*, 11:635–650, 1991.

155. W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the U.S.A.*, 85:2444–2448, 1988.

156. W. R. Pearson and W. Miller. Dynamic programming algorithms for biological sequence comparison. In L. Brand and M. L. Johnson, editors, *Numerical Computer Methods*, volume 210 of *Methods in Enzymology*, pages 575–601. New York: Academic Press, 1992.

157. H. Peltola, H. Söderlund, J. Tarhio, and E. Ukkonen. Algorithms for some string matching problems arising in molecular genetics. In *Information Processing 83:Proceedings of the International Federation for Information Processing (IFIP) Ninth World Computer Congress*, pages 53–64. Amsterdam: North Holland, 1983.

158. H. Peltola, H. Söderlund, and E. Ukkonen. SEQAIDS: A DNA sequence assembling program based on a mathematical model. *Nucleic Acids Research*, 12:307–321, 1984.

159. D. Penny, M. D. Hendy, and M. A. Steel. Progress with methods for constructing evolutionary trees. *Trends in Ecology and Evolution*, 7(3):73–79, 1992.

160. P. A. Pevzner. DNA physical mapping and alternating Eulerian cycles in colored graphs. *Algorithmica*, 13(1/2):77–105, 1995.

161. F. M. Richards. The protein folding problem. *Scientific American*, 264(1):54–63, Jan. 1991.

162. R. J. Robbins. Challenges in the human genome project. *IEEE Engineering in Medicine and Biology*, 11(1):25–34, Mar. 1992.

163. K. H. Rosen. *Discrete Mathematics and Its Applications*, 2nd ed. New York: McGraw-Hill, 1991.

164. M. Rosenberg and D. Court. Regulatory sequences involved in the promotion and termination of RNA transcription. *Annual Review of Genetics*, 13:319–353, 1979.

165. I. Rosenfeld, E. Ziff, and V. van Loon. *DNA for beginners*. Writers and Readers, 1984.

166. D. Sankoff. Minimal mutation trees of sequences. *SIAM Journal on Applied Mathematics*, 28:35–42, 1975.

167. D. Sankoff. Analytical approaches to genomic evolution. *Biochimie*, 75(409–413), 1993.

168. D. Sankoff and J. B. Kruskal. *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*. Reading, MA: Addison-Wesley, 1983.

169. W. Schmitt and M. S. Waterman. Multiple solutions of DNA restriction mapping problems. *Advances in Applied Mathematics*, 12:412–427, 1991.

170. R. Sedgewick. *Algorithms*, 2nd ed. Reading, MA: Addison-Wesley, 1988.

171. D. Seto, B. Koop, and L. Hood. An experimentally derived data set constructed for testing large-scale DNA sequence assembly algorithms. *Genomics*, 15:673–676, 1993.

172. I. Simon. Sequence comparison: some theory and some practice. In *Proceedings of the LITP Spring School on Theoretical Computer Science*, volume 377 of *Lecture Notes in Computer Science*, pages 79–92. Berlin: Springer-Verlag, 1987.

173. J. Sims, D. Capon, and D. Dressler. *dna*G (Primase)-dependent origins of DNA replication. *Journal of Biolical Chemistry*, 254:12615–12628, 1979.

174. S. S. Skiena and G. Sundaram. A partial digest approach to restriction site mapping. *Bulletin of Mathematical Biology*, 56(2):275–294, 1994.

175. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

176. T. F. Smith, M. S. Waterman, and W. M. Fitch. Comparative biosequence metrics. *Journal of Molecular Evolution*, 18:38–46, 1981.

177. C. Soderlund and C. Burks. GRAM and genfragII: solving and testing the single-digest, partially ordered restriction map problem. *Computer Applications in the Biosciences*, 10(3):349–358, 1994.

178. R. Staden. A strategy of DNA sequencing employing computer programs. *Nucleic Acids Research*, 6:2601–2610, 1979.

179. M. A. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992.

180. G. A. Stephen. *String Searching Algorithms*. Singapore: World Scientific, 1994.

181. D. L. Swofford and W. P. Maddison. Reconstructing ancestral character states under Wagner parsimony. *Mathematical Biosciences*, 87:199–229, 1987.

182. D. L. Swofford and G. J. Olsen. Phylogeny reconstruction. In D. M. Hillis and C. Moritz, editors, *Molecular Systematics*, pages 411–501. Sunderland, MA: Sinauer Associates, 1990.

183. R. Tamarin. *Principles of Genetics*. Dubuque, IA: Wm. C. Brown, 1991.

184. R. E. Tarjan. *Data Structures and Network Algorithms*. CBMS-NSF Regional conference series in applied mathematics. Society for Industrial and Applied Mathematics, 1983.

185. S.-H. Teng and F. Yao. Approximating shortest superstrings. In *Proceedings of the IEEE Thirty-Fourth Annual Symposium on Foundations of Computer Science*, pages 158–165, 1993.

186. D. H. Turner and N. Sugimoto. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17:167–192, 1988.

187. J. S. Turner. Approximation algorithms for the shortest common superstring problem. *Information and Computation*, 83:1–20, 1989.

188. E. Ukkonen. Algorithms for approximate string matching. *Information and Control*, 64:100–118, 1985.

189. R. Unger and J. Moult. Finding the lowest free energy conformation of a protein is an NP-hard problem: proof and implications. *Bulletin of Mathematical Biology*, 55(6):1183–1198, 1993.

190. M. Vingron and A. von Haeseler. Towards integration of multiple alignment and phylogenetic tree construction. Unpublished manuscript, 1995.

191. G. von Heijne. *Sequence Analysis in Molecular Biology: Treasure Trove or Trivial Pursuit?* New York: Academic Press, 1987.

192. L. Wang and D. Gusfield. Improved approximation algorithms for tree alignment. In *Proceedings of the Seventh Symposium on Combinatorial Pattern Matching*, volume 1075 of *Lecture Notes in Computer Science*, pages 220–233. Berlin: Springer-Verlag, 1996.

193. L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

194. T. J. Warnow. Constructing phylogenetic trees efficiently using compatibility criteria. Unpublished manuscript, 1993.

195. T. J. Warnow. Tree compatibility and inferring evolutionary history. *Journal of Algorithms*, 16:388–407, 1994.

196. M. S. Waterman, editor. *Mathematical Methods for DNA Sequences*. Boca Raton, FL: CRC Press, 1989.

197. M. S. Waterman. Sequence alignments. In Waterman [196], pages 53–92.

198. M. S. Waterman. Parametric and ensemble sequence alignment algorithms. *Bulletin of Mathematical Biology*, 56(4):743–767, 1994.

199. M. S. Waterman. *Introduction to Computational Biology*. London: Chapman & Hall, 1995.

200. M. S. Waterman and J. R. Griggs. Interval graphs and maps of DNA. *Bulletin of Mathematical Biology*, 48(2):189–195, 1986.

201. M. S. Waterman and T. F. Smith. Rapid dynamic programming algorithms for RNA secondary structure. *Advances in Applied Mathematics*, 7:455–464, 1986.

202. M. S. Waterman, T. F. Smith, M. Singh, and W. A. Beyer. Additive evolutionary trees. *Journal of Theoretical Biology*, 64:199–213, 1977.

203. J. D. Watson et al. *Molecular Biology of the Gene*, volume 1. Redwood City, CA: Benjamin/Cummings, 1987.

204. J. D. Watson et al. *Molecular Biology of the Gene*, volume 2. Redwood City, CA: Benjamin/Cummings, 1987.

205. G. A. Watterson, W. J. Ewens, and T. E. Hall. The chromosome inversion problem. *Journal of Theoretical Biology*, 99:1–7, 1982.

206. M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

207. M. Zuker. The use of dynamic programming algorithms in RNA secondary structure prediction. In Waterman [196], pages 159–185.