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 ($\star$) 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 Uniform 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