国外数学名著系列

（影印版）76

Alexander Isaev

Introduction to Mathematical Methods in Bioinformatics

生物信息学中的数学方法引论

科学出版社

国外数学名著系列(影印版)　76

Introduction to Mathematical Methods in Bioinformatics

生物信息学中的数学方法引论

Alexander Isaev

科学出版社
北京

图字：**01-2011-3329**

Alexander Isaev: Introduction to Mathematical Methods in Bioinformatics

图书在版编目 (CIP) 数据

生物信息学中的数学方法引论= Introduction to Mathematical Methods in Bioinformatics /（澳）伊萨耶夫(Isaev, A.)编著. —影印版. —北京：科学出版社, 2011

(国外数学名著系列; 76)

ISBN 978-7-03-031381-2

I. ①生… II. ①伊… III. ①生物分析-序列-数据模型-生物信息论-研究 IV. ①Q5

中国版本图书馆 CIP 数据核字(2011) 第 105031 号

责任编辑:赵彦超　徐园园/责任印刷:钱玉芬/封面设计:陈　敬

科学出版社 出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

双青印刷厂 印刷

科学出版社发行　各地新华书店经销

*

2011 年 6 月第　一　版　开本：B5(720 × 1000)
2011 年 6 月第一次印刷　印张：20
印数：1—2 000　字数：380 000

定价：78.00 元

(如有印装质量问题，我社负责调换)

《国外数学名著系列》(影印版)序

要使我国的数学事业更好地发展起来，需要数学家淡泊名利并付出更艰苦地努力。另一方面，我们也要从客观上为数学家创造更有利的发展数学事业的外部环境，这主要是加强对数学事业的支持与投资力度，使数学家有较好的工作与生活条件，其中也包括改善与加强数学的出版工作。

从出版方面来讲，除了较好较快地出版我们自己的成果外，引进国外的先进出版物无疑也是十分重要与必不可少的。从数学来说，施普林格(Springer)出版社至今仍然是世界上最具权威的出版社。科学出版社影印一批他们出版的好的新书，使我国广大数学家能以较低的价格购买，特别是在边远地区工作的数学家能普遍见到这些书，无疑是对推动我国数学的科研与教学十分有益的事。

这次科学出版社购买了版权，一次影印了 23 本施普林格出版社出版的数学书，就是一件好事，也是值得继续做下去的事情。大体上分一下，这 23 本书中，包括基础数学书 5 本，应用数学书 6 本与计算数学书 12 本，其中有些书也具有交叉性质。 这些书都是很新的，2000 年以后出版的占绝大部分，共计 16 本，其余的也是 1990 年以后出版的。这些书可以使读者较快地了解数学某方面的前沿，例如基础数学中的数论、代数与拓扑三本，都是由该领域大数学家编著的“数学百科全书”的分册。对从事这方面研究的数学家了解该领域的前沿与全貌很有帮助。按照学科的特点，基础数学类的书以“经典”为主，应用和计算数学类的书以“前沿”为主。这些书的作者多数是国际知名的大数学家，例如《拓扑学》一书的作者诺维科夫是俄罗斯科学院的院士，曾获“菲尔兹奖”和“沃尔夫数学奖”。这些大数学家的著作无疑将会对我国的科研人员起到非常好的指导作用。

当然，23 本书只能涵盖数学的一部分，所以，这项工作还应该继续做下去。更进一步，有些读者面较广的好书还应该翻译成中文出版，使之有更大的读者群。

总之，我对科学出版社影印施普林格出版社的部分数学著作这一举措表示热烈的支持，并盼望这一工作取得更大的成绩。

王　元

2005 年 12 月 3 日

Preface

Broadly speaking, Bioinformatics can be defined as a collection of mathematical, statistical and computational methods for analyzing biological sequences, that is, DNA, RNA and amino acid (protein) sequences. Numerous projects for sequencing the DNA of particular organisms constantly supply new amounts of data on an astronomical scale, and it would not be realistic to expect that biologists will ever be able to make sense of it without resorting to help from more quantitative disciplines.

Many studies in molecular biology require performing specific computational procedures on given sequence data, for example, simply organizing the data conveniently, and therefore analysis of biological sequences is often viewed as part of computational science. As a result, bioinformatics is frequently confused with *computational sequence analysis*, which is somewhat narrower. However, understanding biological sequences now increasingly requires profound ideas beyond computational science, specifically, mathematical and statistical ideas. For example, the protein folding problem incorporates serious differential geometry and topology; understanding the evolution of sequences is dependent upon developing better probabilistic evolutionary models; analysis of microarray data requires new statistical approaches. Generally, when one goes beyond algorithms and starts either looking for *first principles* behind certain biological processes (which is an extremely difficult task) or paying more attention to *modeling* (which is a more standard approach), one crosses the boundary between computational science and mathematics/statistics. These days many mathematicians are becoming interested in bioinformatics and beginning to contribute to research in biology. This is also my personal story: I am a pure mathematician specializing in several complex variables, but now I work in bioinformatics as well.

Despite this mathematical trend, bioinformatics is still largely taught by computational groups and computer science departments, and partly by engineering and biology (in particular, genetics) departments. Naturally, in courses developed by these departments emphasis is placed on algorithms and their implementation in software. Although it is useful to know how a particular piece of software works, this software-oriented education does not always reveal the mathematical principles on which the algorithms are based. Such incompleteness may lead to certain problems for the graduates. Suppose, for example, that a commonly used model is implemented in software and the students are taught how to use it. Of course, the model makes some simplifying *assumptions* about the biological processes it attempts to describe, and these assumptions are buried in the mathematical core of the model. If the students are taught only how to use the software and are not taught the mathematical foundations of the model, they will know nothing about the assumptions and therefore *limitations* of the model; this in turn means that they will not be able to interpret correctly the results of applying the software to biological data.

This situation with education in bioinformatics is now beginning to change as mathematics departments around the world are starting to teach this subject. I have been teaching two bioinformatics courses at the Department of Mathematics of the Australian National University (ANU) in Canberra, for two years now. When I started teaching them I quickly realized that all the textbooks that I found on the subject were skewed towards computational issues, reflecting, of course, the dominant teaching culture at the time. Those textbooks were not very satisfying from a mathematician's point of view and were unacceptable for my purposes. What I needed was a clear and mathematically rigorous exposition of procedures, algorithms and models commonly used in bioinformatics. As a result, I began writing my own lecture notes, and eventually they formed the basis for this book.

The book has two parts corresponding to the two courses. The first course is for second-year students and requires two medium-level first year mathematics courses as prerequisites. It concerns four important topics in bioinformatics (sequence alignment, profile hidden Markov models, protein folding and phylogenetic reconstruction) and covers them in considerable detail. Many mathematical issues related to these topics are discussed, but their probabilistic and statistical aspects are not covered in much depth there, as the students are not required to have a background in these areas. The second course (intended for third-year students) includes elements of probability and statistics; this allows one both to explore additional topics in sequence alignment, and to go back to some of the issues left unexplained in the first course, treating them from the general probabilistic and statistical point of view.

The second course is much more demanding mathematically because of its probabilistic and statistical component. At the same time, the chapters on probability and statistics (Chaps. 6 and 8) contain very few proofs. The path taken in these chapters is to give the reader all the main constructions (for instance, the construction of probability measure) and to illustrate them by many examples. Such a style is more gentle on students who only have taken a couple of mathematical courses and do not possess the mathematical maturity of a student majoring in mathematics. In fact, this is the general approach taken in the book: I give very few proofs, but a lot of discussions and examples.

Nevertheless, the book is quite mathematical in its logical approach, rigor and paying attention to subtle details.

Thus, for someone who wants to get a mathematical overview of some of the important topics in bioinformatics but does not want to go too deeply into the associated probabilistic and statistical issues, Part I of the book is quite sufficient. But it should be stressed that without reading Part II, one's understanding of various procedures from Part I will be incomplete.

Although Parts I and II together cover a substantial amount of material, none of the topics discussed in the book is treated comprehensively. For example, the chapter on protein folding (Chap. 4) and the one on phylogenetic reconstruction (Chap. 5) could each easily be expanded into a separate book. The amount of material included in the book is what realistically can be taught as two one-semester courses. Certainly, if the probability and statistics components of the book are taught separately in a different course, one can fit in more genuine bioinformatics topics, for example, the analysis of microarray data, currently not represented in the book at all.

The book concentrates on the mathematical basics of bioinformatics rather than on recent progress in the area. Even the material included in the book is found quite demanding by many students, and this is why I decided to select for it only a few topics in bioinformatics. Thus, this book is by no means a comprehensive guide to bioinformatics.

This is primarily a textbook for students with some mathematical background. At the same time, it is suitable for any mathematician, or, indeed, anyone who appreciates quantitative thinking and mathematical rigor, and who wants to learn about bioinformatics. It took me a substantial effort to explain various bioinformatics procedures in a way suitable for a general mathematical audience, and hence this book can be thought of, at least to some extent, as a translation and adaptation of some topics in bioinformatics for mathematicians. On top of this, the book contains a mathematical introduction to statistics that I have tried to keep as rigorous as possible.

I would like to thank my colleagues Prof. Sue Wilson and Prof. Simon Easteal of the Mathematical Sciences Institute (MSI) and the Centre for Bioinformation Science (CBiS) at the ANU who first suggested that I should turn my lecture notes into a book and encouraged me during the course of writing. I would like to thank Prof. Peter Hall of the MSI for patiently answering my many questions on the theory of statistics. Finally, I am grateful to Prof. John Hutchinson of the Department of Mathematics for encouragement and general discussions.

Canberra,
March 2004

Alexander Isaev

Contents

Part I Sequence Analysis

Part II Mathematical Background for Sequence Analysis

Part I

Sequence Analysis

1

Introduction: Biological Sequences

This book is about analyzing sequences of letters from a finite alphabet $\mathcal{Q}$. Although most of what follows can be applied to sequences derived from arbitrary alphabets, our primary interest will be in *biological sequences*, that is, DNA, RNA and protein sequences.

DNA (deoxyribonucleic acid) sequences are associated with the four-letter *DNA alphabet* $\{A, C, G, T\}$, where A, C, G and T stand for the *nucleic acids* or *nucleotides* adenine, cytosine, guanine and thymine respectively. Most DNA sequences currently being studied come from DNA molecules found in chromosomes that are located in the nuclei of the cells of living organisms. In fact, a DNA molecule consists of *two strands* of nucleotides (attached to a sugar-phosphate backbone) twisted into the well-known double-helical arrangement. The two-strand structure is important for the replication of DNA molecules. There is a pairing (called *hybridization*) of nucleotides across the two strands: A is bonded to T, and C is bonded to G. Therefore, if one knows the sequence of one strand of a DNA molecule, that of the other strand can easily be reconstructed, and DNA sequences are always given as sequences of single, not paired nucleotides. The chemistry of the backbone of each strand of a DNA molecule determines a particular *orientation* of the strand, the so-called $5'$ *to* $3'$ *orientation*. This is the orientation in which DNA sequences are written. It should be noted that the orientations of the two strands in a DNA molecule are opposite (for this reason the strands are said to be *antiparallel*), and therefore, although the sequences of the strands determine one another, they are read in opposite directions.

Traditionally, DNA research has been focused on special stretches of the strands of DNA molecules called *protein-coding genes*; they are found on both strands, and rarely overlap across the strands. Protein-coding genes are used to produce proteins which are linear polymers of 20 different *amino acids* linked by *peptide bonds*. The single-letter amino acid notation is given in Table 1.1.

Table 1.1.

Single letter code	Amino acid
A	Alanine
R	Arginine
N	Asparagine
D	Aspartic acid
C	Cysteine
Q	Glutamine
E	Glutamic acid
G	Glycine
H	Histidine
I	Isoleucine
L	Leucine
K	Lysine
M	Methionine
F	Phenylalanine
P	Proline
S	Serine
T	Threonine
W	Tryptophan
Y	Tyrosine
V	Valine

Thus, protein sequences are associated with the 20-letter *amino-acid alphabet* $\{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$. The three-dimensional structure of a protein molecule results from the folding of the polypeptide chain and is much more complicated than that of a DNA molecule. A protein can only function properly if it is correctly folded. Deriving the correct three-dimensional structure from a given protein sequence is the famous *protein folding problem* that is still largely unsolved (see Chap. 4).

The main components of a protein-coding gene are *codons*. Each codon is a triplet of nucleotides coding for a single amino acid. The process of producing proteins from genes is quite complex. Every gene begins with one of the standard *start codons* indicating the beginning of the process and ends with one of the standard *stop codons* indicating the end of it. A particular way codons code for amino acids is called a *genetic code*. Several genetic codes are known, and different ones apply to different DNA molecules depending on their origin (see, e.g., [Kan]). When the sequence of a gene is read from the start to the stop codon, a growing chain of amino acids is made which, once the stop codon has been reached, becomes a complete protein molecule. It has a natural orientation inherited from that of the gene used to produce it, and this is the orientation in which protein sequences are written. The start of a protein chain is called the *amino end* and the end of it the *carboxy end*.

In fact, proteins are derived from genes in two steps. Firstly, RNA (ribonucleic acid) is made (this step is called *transcription*) and, secondly, the RNA is used to produce a protein (this step is called *translation*). RNA is another linear macromolecule, it is closely related to DNA. RNA is single-stranded, its backbone is slightly different from that of DNA, and instead of the nucleic acid thymine the nucleic acid uracil denoted by U is used. Thus, RNA sequences are associated with the four-letter *RNA alphabet* $\{A, C, G, U\}$. An RNA molecule inherits its orientation from that of the DNA strand used to produce it, and this is the orientation in which RNA sequences are written. Since RNA is single-stranded, parts of it can hybridize with its other parts which gives rise to non-trivial three-dimensional structures essential for the normal functioning of the RNA. There are in fact many types of RNA produced from not necessarily protein-coding genes, but from *RNA-coding genes*. The RNA derived from a protein-coding gene is called *messenger RNA* or *mRNA*. As an example of RNA of another type we mention *transfer RNA* or *tRNA* that takes part in translating mRNA into protein.

In this book we concentrate on DNA and protein sequences although everything that follows can be applied, at least in principle, to RNA sequences as well (subject to the availability of RNA sequence data). In fact, most procedures are so general that they work for sequences of letters from any finite alphabet, and for illustration purposes we often use the artificial two- and three-letter alphabets $\{A, B\}$ and $\{A, B, C\}$.

Table 1.2.

Database	Principal function	Organization	Address
MEDLINE	Bibliographic	National Library of Medicine	www.nlm.nih.gov
GenBank	Nucleotide sequences	National Center for Biotechnology Information	www.ncbi.nlm.nih.gov
EMBL	Nucleotide sequences	European Bioinformatics Institute	www.ebi.ac.uk
DDBJ	Nucleotide sequences	National Institute of Genetics, Japan	www.ddbj.nig.ac.jp
SWISS-PROT	Amino acid sequences	Swiss Institute of Bioinformatics	www.expasy.ch
PIR	Amino acid sequences	National Biomedical Research Foundation	www-nbrf.georgetown.edu
PRF	Amino acid sequences	Protein Research Foundation, Japan	www.prf.or.jp
PDB	Protein structures	Research Collaboratory for Structural Bioinformatics	www.rcsb.org
CSD	Protein structures	Cambridge Crystallographic Data Centre	www.ccdc.cam.ac.uk

Biological sequences are organized in databases, many of which are public. In Table 1.2 we list all major public molecular biology databases. Detailed information on them can be found in [Kan].

2

Sequence Alignment

2.1 Sequence Similarity

New DNA, RNA and protein sequences develop from pre-existing sequences rather than get invented by nature from scratch. This fact is the foundation of any sequence analysis. If we manage to relate a newly discovered sequence to a sequence about which something (e.g., structure or function) is already known, then chances are that the known information applies, at least to some extent, to the new sequence as well. We will think of any two related sequences as sequences that arose from a common ancestral sequence during the course of evolution and say that they are *homologous*. It is *sequence homology* that will be of interest to us during much of the book. Of course, if we believe that all life forms on earth came from the same origin and apply the above definition directly, then all sequences are ultimately homologous. In practice, two sequences are called homologous, if one can establish their relatedness by *currently available methods*, and it is the sensitivity of the methods that produces a borderline between sequences called homologous and ones that are not called homologous. For example, two protein sequences can be called homologous, if one can show experimentally that their functions in the respective organisms are related. Thus, sequence homology is a dynamic concept, and families of homologous sequences known at the moment may change as the sensitivity of the methods improves.

The first step towards inferring homology is to look for *sequence similarity*. If two given sequences are very long, it is not easy to decide whether or not they are similar. To see if they are similar, one has to properly *align* them. When sequences evolve starting from a common ancestor, their residues can undergo *substitutions* (when residues are replaced by some other residues). Apart from substitutions, during the course of evolution sequences can accumulate a number of events of two more types: *insertions* (when new residues appear in a sequence in addition to the existing ones) and *deletions* (when some residues disappear). Therefore, when one is trying to produce the best possible *alignment* between two sequences, residues must be allowed to be