

## Genomic Perl

This introduction to computational molecular biology will help programmers and biologists learn the skills they need to start work in this important, expanding field. The author explains many of the basic computational problems and gives concise, working programs to solve them in the Perl programming language. With minimal prerequisites, the author explains the biological background for each problem, develops a model for the solution, and then introduces the Perl concepts needed to implement the solution.

The book covers pairwise and multiple sequence alignment, fast database searches for homologous sequences, protein motif identification, genome rearrangement, physical mapping, phylogeny reconstruction, satellite identification, sequence assembly, gene finding, and RNA secondary structure. The author focuses on one or two practical approaches for each problem rather than an exhaustive catalog of ideas. His concrete examples and step-by-step approach make it easy to grasp the computational and statistical methods, including dynamic programming, branch-and-bound optimization, greedy methods, maximum likelihood methods, substitution matrices, BLAST searching, and Karlin–Altschul statistics.

Rex A. Dwyer founded Genomic Perl Consultancy, Inc. in July 2001. He was formerly an Associate Professor of Computer Science at North Carolina State University, where he taught data structures, algorithms, and formal language theory and demonstrated his skill as both theoretician and practitioner. He has published more than a dozen papers in academic journals such as *Advances in Applied Probability*, *Algorithmica*, and *Discrete and Computational Geometry*. His accomplishments as a Perl software developer include a proprietary gene-finding system for the Novartis Agribusiness Biotechnology Research Institute (now part of Syngenta, Inc.) and a web-accessible student records database for faculty colleagues at NCSU.

Cambridge University Press  
052180177X - Genomic Perl: From Bioinformatics Basics to Working Code  
Rex A. Dwyer  
Frontmatter  
[More information](#)

---

# Genomic Perl

## From Bioinformatics Basics to Working Code

**REX A. DWYER**

Genomic Perl Consultancy, Inc.



**CAMBRIDGE**  
UNIVERSITY PRESS

Cambridge University Press  
052180177X - Genomic Perl: From Bioinformatics Basics to Working Code  
Rex A. Dwyer  
Frontmatter  
[More information](#)

---

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE  
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS  
The Edinburgh Building, Cambridge CB2 2RU, UK  
40 West 20th Street, New York, NY 10011-4211, USA  
477 Williamstown Road, Port Melbourne, VIC 3207, Australia  
Ruiz de Alarcón 13, 28014 Madrid, Spain  
Dock House, The Waterfront, Cape Town 8001, South Africa  
<http://www.cambridge.org>

© Cambridge University Press 2002

This book is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2002

Printed in the United States of America

*Typeface* Times 10/13 pt. *System* AMS- $\text{T}_{\text{E}}\text{X}$  [FH]

*A catalog record for this book is available from the British Library.*

*Library of Congress Cataloging in Publication data available*

ISBN 0 521 80177 X hardback

Cambridge University Press  
052180177X - Genomic Perl: From Bioinformatics Basics to Working Code  
Rex A. Dwyer  
Frontmatter  
[More information](#)

---

*A mi chapinita y a mi chapinito.*

Contents

Preface	<i>page</i> xiii
Acknowledgments	xvii
<b>1 The Central Dogma</b>	<b>1</b>
1.1 DNA and RNA	1
1.2 Chromosomes	2
1.3 Proteins	4
1.4 The Central Dogma	5
1.5 Transcription and Translation in Perl	7
1.6 Exercise	12
1.7 Complete Program Listings	12
1.8 Bibliographic Notes	14
<b>2 RNA Secondary Structure</b>	<b>16</b>
2.1 Messenger and Catalytic RNA	16
2.2 Levels of RNA Structure	17
2.3 Constraints on Secondary Structure	18
2.4 RNA Secondary Structures in Perl	20
2.4.1 Counting Hydrogen Bonds	21
2.4.2 Folding RNA	24
2.5 Exercises	28
2.6 Complete Program Listings	29
2.7 Bibliographic Notes	29
<b>3 Comparing DNA Sequences</b>	<b>30</b>
3.1 DNA Sequencing and Sequence Assembly	30
3.2 Alignments and Similarity	32
3.3 Alignment and Similarity in Perl	36
3.4 Exercises	40
3.5 Complete Program Listings	42
3.6 Bibliographic Notes	43
<b>4 Predicting Species: Statistical Models</b>	<b>44</b>
4.1 Perl Subroutine Libraries	49
4.2 Species Prediction in Perl	51
	<b>vii</b>

viii	Contents
4.3 Exercises	53
4.4 Complete Program Listings	53
4.5 Bibliographic Note	54
<b>5 Substitution Matrices for Amino Acids</b>	<b>55</b>
5.1 More on Homology	57
5.2 Deriving Substitution Matrices from Alignments	57
5.3 Substitution Matrices in Perl	60
5.4 The PAM Matrices	65
5.5 PAM Matrices in Perl	68
5.6 Exercises	70
5.7 Complete Program Listings	71
5.8 Bibliographic Notes	71
<b>6 Sequence Databases</b>	<b>72</b>
6.1 FASTA Format	73
6.2 GenBank Format	73
6.3 GenBank's Feature Locations	75
6.4 Reading Sequence Files in Perl	79
6.4.1 Object-Oriented Programming in Perl	80
6.4.2 The SimpleReader Class	81
6.4.3 Hiding File Formats with Method Inheritance	85
6.5 Exercises	89
6.6 Complete Program Listings	91
6.7 Bibliographic Notes	92
<b>7 Local Alignment and the BLAST Heuristic</b>	<b>93</b>
7.1 The Smith–Waterman Algorithm	94
7.2 The BLAST Heuristic	96
7.2.1 Preprocessing the Query String	98
7.2.2 Scanning the Target String	99
7.3 Implementing BLAST in Perl	100
7.4 Exercises	106
7.5 Complete Program Listings	108
7.6 Bibliographic Notes	108
<b>8 Statistics of BLAST Database Searches</b>	<b>109</b>
8.1 BLAST Scores for Random DNA	109
8.2 BLAST Scores for Random Residues	114
8.3 BLAST Statistics in Perl	116
8.4 Interpreting BLAST Output	123
8.5 Exercise	125
8.6 Complete Program Listings	126
8.7 Bibliographic Notes	126
<b>9 Multiple Sequence Alignment I</b>	<b>127</b>
9.1 Extending the Needleman–Wunsch Algorithm	128
9.2 NP-Completeness	131

<b>Contents</b>	<b>ix</b>
9.3 Alignment Merging: A Building Block for Heuristics	132
9.4 Merging Alignments in Perl	133
9.5 Finding a Good Merge Order	137
9.6 Exercises	139
9.7 Complete Program Listings	139
9.8 Bibliographic Notes	139
<b>10 Multiple Sequence Alignment II</b>	<b>141</b>
10.1 Pushing through the Matrix by Layers	141
10.2 Tunnel Alignments	147
10.3 A Branch-and-Bound Method	149
10.4 The Branch-and-Bound Method in Perl	152
10.5 Exercises	153
10.6 Complete Program Listings	154
10.7 Bibliographic Notes	154
<b>11 Phylogeny Reconstruction</b>	<b>155</b>
11.1 Parsimonious Phylogenies	155
11.2 Assigning Sequences to Branch Nodes	157
11.3 Pruning the Trees	160
11.4 Implementing Phylogenies in Perl	162
11.5 Building the Trees in Perl	168
11.6 Exercise	171
11.7 Complete Program Listings	171
11.8 Bibliographic Notes	171
<b>12 Protein Motifs and PROSITE</b>	<b>173</b>
12.1 The PROSITE Database Format	174
12.2 Patterns in PROSITE and Perl	175
12.3 Suffix Trees	177
12.3.1 Suffix Links	184
12.3.2 The Efficiency of Adding	188
12.4 Suffix Trees for PROSITE Searching	189
12.5 Exercises	193
12.6 Complete Program Listings	195
12.7 Bibliographic Notes	195
<b>13 Fragment Assembly</b>	<b>196</b>
13.1 Shortest Common Superstrings	196
13.2 Practical Issues and the PHRAP Program	202
13.3 Reading Inputs for Assembly	204
13.4 Aligning Reads	207
13.5 Adjusting Qualities	212
13.6 Assigning Reads to Contigs	217
13.7 Developing Consensus Sequences	222
13.8 Exercises	227
13.9 Complete Program Listings	230
13.10 Bibliographic Notes	230

x	Contents
<b>14 Coding Sequence Prediction with Dicodons</b>	231
14.1 A Simple Trigram Model	232
14.2 A Hexagram Model	235
14.3 Predicting All Genes	237
14.4 Gene Finding in Perl	237
14.5 Exercises	244
14.6 Complete Program Listings	244
14.7 Bibliographic Notes	244
<b>15 Satellite Identification</b>	245
15.1 Finding Satellites Efficiently	246
15.1.1 Suffix Testing	247
15.1.2 Satellite Testing	249
15.2 Finding Satellites in Perl	251
15.3 Exercises	255
15.4 Complete Program Listings	256
15.5 Bibliographic Notes	256
<b>16 Restriction Mapping</b>	257
16.1 A Backtracking Algorithm for Partial Digests	258
16.2 Partial Digests in Perl	260
16.3 Uncertain Measurement and Interval Arithmetic	262
16.3.1 Backtracking with Intervals	263
16.3.2 Interval Arithmetic in Perl	265
16.3.3 Partial Digests with Uncertainty in Perl	267
16.3.4 A Final Check for Interval Consistency	269
16.4 Exercises	271
16.5 Complete Program Listings	273
16.6 Bibliographic Notes	274
<b>17 Rearranging Genomes: Gates and Hurdles</b>	275
17.1 Sorting by Reversals	276
17.2 Making a Wish List	278
17.3 Analyzing the Interaction Relation	279
17.4 Clearing the Hurdles	280
17.5 Happy Cliques	284
17.6 Sorting by Reversals in Perl	287
17.7 Exercise	297
17.8 Appendix: Correctness of Choice of Wish from Happy Clique	298
17.9 Complete Program Listings	298
17.10 Bibliographic Notes	298
<b>A Drawing RNA Cloverleaves</b>	300
A.1 Exercises	304
A.2 Complete Program Listings	306
A.3 Bibliographic Notes	306

<b>Contents</b>	<b>xi</b>
<b>B Space-Saving Strategies for Alignment</b>	307
B.1 Finding Similarity Scores Compactly	307
B.2 Finding Alignments Compactly	309
B.3 Exercises	312
B.4 Complete Program Listings	312
B.5 Bibliographic Note	312
<b>C A Data Structure for Disjoint Sets</b>	313
C.1 Union by Rank	314
C.2 Path Compression	315
C.3 Complete Program Listings	315
C.4 Bibliographic Note	317
<b>D Suggestions for Further Reading</b>	318
Bibliography	319
Index	325

## Preface

This book is designed to be a concrete, digestible introduction to the area that has come to be known as “bioinformatics” or “computational molecular biology”. My own teaching in this area has been directed toward a mixture of graduate and advanced undergraduate students in computer science and graduate students from the biological sciences, including biomathematics, genetics, forestry, and entomology. Although a number of books on this subject have appeared in the recent past – and one or two are quite well written – I have found none to be especially suitable for the widely varying backgrounds of this audience.

My experience with this audience has led me to conclude that its needs can be met effectively by a book with the following features.

- To meet the needs of computer scientists, the book must teach basic aspects of the structure of DNA, RNA, and proteins, and it must also explain the salient features of the laboratory procedures that give rise to the sorts of data processed by the algorithms selected for the book.
- To meet the needs of biologists, the book must (to some degree) teach programming and include *working programs* rather than abstract, high-level descriptions of algorithms – yet computer scientists must not become bored with material more appropriate for a basic course in computer programming.
- Justice to the field demands that its statistical aspects be addressed, but the background of the audience demands that these aspects be addressed in a concrete and relatively elementary fashion.

To meet these criteria, a typical chapter of this book focuses on a single problem that arises in the processing of biological sequence data: pairwise sequence alignment, multiple alignment, sequence database searching, phylogeny reconstruction, genome rearrangement, and so on. I outline both the biological origins of the input and the interpretation of the desired output; then I develop a single algorithmic approach to the problem. Finally, I show how to implement the algorithm as a working Perl program. Variations on the problem and/or improvements to the program are

presented in exercises at the end of each chapter. In a few cases, I develop a straightforward but inefficient algorithm in one chapter and devote the following chapter to a more sophisticated approach. Bibliographic notes in each chapter are limited to a half dozen of the most accessible references; I assume that a serious student can use these hooks into the literature to track down other relevant resources.

The choice of the Perl language as the medium for presenting algorithms might surprise some, but it has many advantages.

- Perl's built-in strings, lists, and hash tables make it possible to express some algorithms very concisely. (Chapter 12's 80-line program for constructing suffix trees is an outstanding example.)
- Perl is already widely used for server-side scripting (CGI) in web-based applications, and a large library of code (the bioPerl effort described at [www.bioperl.org](http://www.bioperl.org)) is freely available to assist bioinformatic programmers.
- Perl falls outside the standard computer science curriculum. This means that attention to language details will be of some interest even to students with strong computer science backgrounds.
- Perl is portable; it runs on all major operating systems.
- Perl is available without charge as part of Linux distributions and from the World Wide Web at <http://www.perl.org>.
- Rare but legal Perl constructs like `@{$_[$#_]}|[ ]` inspire awe in the uninitiated; this awe sometimes rubs off on Perl programmers.

Perl also has a few disadvantages.

- Perl programs are not compiled to native-mode code, so they often run many times slower than equivalent C or C++ programs in computationally intensive applications. (In CGI applications dominated by disk-access or network-response times, this disadvantage evaporates.)
- Perl's built-in strings, lists, and hash tables sometimes hide potential performance problems that can be overcome only with nonintuitive tricks.
- Perl is profligate in its use of memory. This means that the input size that a Perl program can handle may be many times smaller than for an equivalent C or C++ program.

All in all, the advantages prevail for the purposes of this book, and – although using Perl makes many of the programs in this book teaching toys rather than production-grade tools – they do work. Unlike pseudocode descriptions, they can be modified to print traces of their operation or to experiment with possible improvements. They can also serve as prototypes for more efficient implementations in other languages.

In the interest of clarity, here are a few words about what this book is *not*.

- This book is not a consumer's guide to the many software packages available to assist the biologist with sequence assembly, database searching, phylogeny reconstruction, or other tasks. Such packages will be mentioned in passing from

## Preface

xv

time to time, but the focus of this book is on how these packages solve (or might solve) the problems presented to them and not on how problems must be presented to these packages.

- This book is not an encyclopedic compendium of computational aspects of molecular biology. The problems included are presumed to be significant to biologists, but other equally significant problems have doubtless been omitted.
- Likewise, this book is not a tutorial or reference manual on the Perl language. Features of Perl will be explained and used as needed, and diligent readers should come away with a good knowledge of Perl, but algorithms are the focus, and many widely used Perl features will be left unmentioned.

I embarked upon this writing project not to convey to others my own research contributions in the area (since I have made none) but rather to study the contributions of others and make them more accessible to students. It is certain that my own lack of expertise will show through now and again. Despite this, I hope that my concrete approach will permit a much larger audience to understand and to benefit from the contributions of the real experts.

## Acknowledgments

Special thanks to Cambridge University Press and my editor, Lauren Cowles, for accepting a book proposal from a first-time author with no publications dealing directly with the subject matter of the book. Ms. Cowles's careful comments and other assistance in the preparation of the manuscript have been invaluable.

Phil Green of the University of Washington was kind enough to allow me to use the source of his PHRAP program to develop Chapter 13. Alejandro Schäffer and Steven Altshul of NCBI offered helpful feedback on chapters dealing with multiple alignment and BLAST.

While I was a faculty member at North Carolina State University, the students of CSC 695 in Spring 1997 and Spring 1999 and CSC 630 in Spring 2001 accepted and proofread an ever-growing collection of notes and programs upon which this book was eventually based. Among these, Barry Kesner deserves special mention for giving very helpful feedback on several chapters. Tim Lowman, systems administrator extraordinaire, was most helpful in answering Perl questions in the early weeks of this project.

Financially, this writing project was partially supported by NC State University in the form of sabbatical support. The laboratory now known as Syngenta Biotechnology, Inc. (in Research Triangle Park, NC) complemented this support on several occasions while permitting me to experiment with proprietary data.

Last but not least, Pam and Kevin Bobol, proprietors of the New World Cafe and Larry's Beans, provided the chairs in which a large fraction of this book was written. Staff members Amy, D.J., Heather, James, Jeramy, Kathy, Margaret, Marwa, Patty, Sam, Sue, and others served the hot liquids that kept my fingers moving.

As expected, any remaining errors are rightfully counted against my own account alone and not against that of my editor, colleagues, students, or barristas.