Contents

| Li | st of Code Challenges | xviii |
|------------|--|-------|
| A l | bout the Textbook | xxi |
| | Meet the Authors | xxi |
| | Meet the Development Team | xxii |
| | Acknowledgments | xxiii |
| 1 | Where in the Genome Does DNA Replication Begin? | 2 |
| | A Journey of a Thousand Miles | 3 |
| | Hidden Messages in the Replication Origin | 5 |
| | DnaA boxes | 5 |
| | Hidden messages in "The Gold-Bug" | 6 |
| | Counting words | 7 |
| | The Frequent Words Problem | 8 |
| | Frequent words in Vibrio cholerae | 10 |
| | Some Hidden Messages are More Surprising than Others | 11 |
| | An Explosion of Hidden Messages | 13 |
| | Looking for hidden messages in multiple genomes | 13 |
| | The Clump Finding Problem | 14 |
| | The Simplest Way to Replicate DNA | 16 |
| | Asymmetry of Replication | 18 |
| | Peculiar Statistics of the Forward and Reverse Half-Strands | 22 |
| | Deamination | 22 |
| | The skew diagram | 23 |
| | Some Hidden Messages are More Elusive than Others | 26 |
| | A Final Attempt at Finding <i>DnaA</i> Boxes in <i>E. coli</i> | |
| | Epilogue: Complications in <i>oriC</i> Predictions | |

| Open Problems | 33 |
|---|----|
| Multiple replication origins in a bacterial genome | 33 |
| Finding replication origins in archaea | 35 |
| Finding replication origins in yeast | 36 |
| Computing probabilities of patterns in a string | 37 |
| Charging Stations | 39 |
| The frequency array | 39 |
| Converting patterns to numbers and vice-versa | 41 |
| Finding frequent words by sorting | 43 |
| Solving the Clump Finding Problem | 44 |
| Solving the Frequent Words with Mismatches Problem | 47 |
| Generating the neighborhood of a string | 49 |
| Finding frequent words with mismatches by sorting | 51 |
| Detours | 52 |
| Big-O notation | 52 |
| Probabilities of patterns in a string | 52 |
| The most beautiful experiment in biology | 57 |
| Directionality of DNA strands | 59 |
| The Towers of Hanoi | 60 |
| The overlapping words paradox | 62 |
| Bibliography Notes | 64 |
| | |
| Which DNA Patterns Play the Role of Molecular Clocks? | 66 |
| Do We Have a "Clock" Gene? | 67 |
| Motif Finding Is More Difficult Than You Think | 68 |
| Identifying the evening element | 68 |
| Hide and seek with motifs | 69 |
| A brute force algorithm for motif finding | 71 |
| Scoring Motifs | 72 |
| From motifs to profile matrices and consensus strings | 72 |
| Towards a more adequate motif scoring function | 75 |
| Entropy and the motif logo | 76 |
| From Motif Finding to Finding a Median String | 77 |
| The Motif Finding Problem | 77 |
| Reformulating the Motif Finding Problem | 77 |
| The Median String Problem | 80 |
| Why have we reformulated the Motif Finding Problem? | 82 |

2

| Greedy Motif Search | 83 |
|---|-----|
| Using the profile matrix to roll dice | 83 |
| Analyzing greedy motif finding | 85 |
| Motif Finding Meets Oliver Cromwell | 86 |
| What is the probability that the sun will not rise tomorrow? | 86 |
| Laplace's Rule of Succession | 87 |
| An improved greedy motif search | 88 |
| Randomized Motif Search | 91 |
| Rolling dice to find motifs | 91 |
| Why randomized motif search works | 93 |
| How Can a Randomized Algorithm Perform So Well? | 96 |
| Gibbs Sampling | 98 |
| Gibbs Sampling in Action | 100 |
| Epilogue: How Does Tuberculosis Hibernate to Hide from Antibiotics? | |
| Charging Stations | |
| Solving the Median String Problem | |
| Detours | |
| Gene expression | 108 |
| DNA arrays | |
| Buffon's needle | |
| Complications in motif finding | |
| Relative entropy | |
| Bibliography Notes | |
| | |
| How Do We Assemble Genomes? | 115 |
| Exploding Newspapers | |
| The String Reconstruction Problem | |
| Genome assembly is more difficult than you think | |
| Reconstructing strings from <i>k</i> -mers | |
| Repeats complicate genome assembly | |
| String Reconstruction as a Walk in the Overlap Graph | |
| From a string to a graph | |
| The genome vanishes | |
| Two graph representations | |
| Hamiltonian paths and universal strings | |
| Another Graph for String Reconstruction | |
| Gluing nodes and de Bruijn graphs | 131 |

3

| Walking in the de Bruijn Graph |
|---|
| Eulerian paths |
| Another way to construct de Bruijn graphs |
| Constructing de Bruijn graphs from <i>k</i> -mer composition |
| De Bruijn graphs versus overlap graphs |
| The Seven Bridges of Königsberg |
| Euler's Theorem |
| From Euler's Theorem to an Algorithm for Finding Eulerian Cycles 146 |
| Constructing Eulerian cycles |
| From Eulerian cycles to Eulerian paths |
| Constructing universal strings |
| Assembling Genomes from Read-Pairs |
| From reads to read-pairs |
| Transforming read-pairs into long reads |
| From composition to paired composition |
| Paired de Bruijn graphs |
| Complications of paired de Bruijn graphs |
| Epilogue: Genome Assembly Faces Real Sequencing Data |
| Breaking reads into <i>k</i> -mers |
| Splitting the genome into contigs |
| Assembling error-prone reads |
| Inferring multiplicities of edges in de Bruijn graphs |
| Charging Stations |
| The effect of gluing on the adjacency matrix |
| Generating all Eulerian cycles |
| Reconstructing a string spelled by a path in the paired de Bruijn graph . 166 |
| Maximal non-branching paths in a graph |
| Detours |
| A short history of DNA sequencing technologies |
| Repeats in the human genome |
| Graphs |
| The icosian game |
| Tractable and intractable problems |
| From Euler to Hamilton to de Bruijn |
| The seven bridges of Kaliningrad |
| The BEST Theorem |
| Bibliography Notes |

| 4 | How Do We Sequence Antibiotics? | 182 |
|---|---|-----|
| | The Discovery of Antibiotics | 183 |
| | How Do Bacteria Make Antibiotics? | 184 |
| | How peptides are encoded by the genome | 184 |
| | Where is Tyrocidine encoded in the <i>Bacillus brevis</i> genome? | 186 |
| | From linear to cyclic peptides | 188 |
| | Dodging the Central Dogma of Molecular Biology | 188 |
| | Sequencing Antibiotics by Shattering Them into Pieces | 190 |
| | Introduction to mass spectrometry | 190 |
| | The Cyclopeptide Sequencing Problem | 191 |
| | A Brute Force Algorithm for Cyclopeptide Sequencing | 193 |
| | A Branch-and-Bound Algorithm for Cyclopeptide Sequencing | 194 |
| | Mass Spectrometry Meets Golf | |
| | From theoretical to real spectra | 197 |
| | Adapting cyclopeptide sequencing for spectra with errors | 198 |
| | From 20 to More than 100 Amino Acids | 201 |
| | The Spectral Convolution Saves the Day | 203 |
| | Epilogue: From Simulated to Real Spectra | |
| | Open Problems | |
| | The Beltway and Turnpike Problems | |
| | Sequencing cyclic peptides in primates | |
| | Charging Stations | |
| | Generating the theoretical spectrum of a peptide | |
| | How fast is CyclopeptideSequencing? | |
| | Trimming the peptide leaderboard | |
| | Detours | |
| | Gause and Lysenkoism | |
| | Discovery of codons | |
| | Quorum sensing | |
| | Molecular mass | |
| | Selenocysteine and pyrrolysine | |
| | Pseudo-polynomial algorithm for the Turnpike Problem | |
| | Split genes | |
| | Bibliography Notes | 221 |
| 5 | How Do We Compare Biological Sequences? | 222 |
| - | Cracking the Non-Ribosomal Code | |

| The RNA Tie Club | .3 |
|--|----|
| From protein comparison to the non-ribosomal code | 4 |
| What do oncogenes and growth factors have in common? | :5 |
| Introduction to Sequence Alignment | 6 |
| Sequence alignment as a game | 6 |
| Sequence alignment and the longest common subsequence | 27 |
| The Manhattan Tourist Problem | 9 |
| What is the best sightseeing strategy? | 9 |
| Sightseeing in an arbitrary directed graph | 2 |
| Sequence Alignment is the Manhattan Tourist Problem in Disguise 23 | 3 |
| An Introduction to Dynamic Programming: The Change Problem 23 | 6 |
| Changing money greedily | 6 |
| Changing money recursively | 37 |
| Changing money using dynamic programming | 9 |
| The Manhattan Tourist Problem Revisited | 1 |
| From Manhattan to an Arbitrary Directed Acyclic Graph | :5 |
| Sequence alignment as building a Manhattan-like graph 24 | :5 |
| Dynamic programming in an arbitrary DAG | :6 |
| Topological orderings | -7 |
| Backtracking in the Alignment Graph | 51 |
| Scoring Alignments | 3 |
| What is wrong with the LCS scoring model? | 3 |
| Scoring matrices | 4 |
| From Global to Local Alignment | 5 |
| Global alignment | 5 |
| Limitations of global alignment | 7 |
| Free taxi rides in the alignment graph | 9 |
| The Changing Faces of Sequence Alignment | 51 |
| Edit distance | 51 |
| Fitting alignment | 3 |
| Overlap alignment | 3 |
| Penalizing Insertions and Deletions in Sequence Alignment | 4 |
| Affine gap penalties | 4 |
| Building Manhattan on three levels | 6 |
| Space-Efficient Sequence Alignment | 9 |
| Computing alignment score using linear memory | 9 |
| The Middle Node Problem 27 | 'n |

| | A surprisingly fast and memory-efficient alignment algorithm | . 273 |
|---|--|-------|
| | The Middle Edge Problem | . 275 |
| | Epilogue: Multiple Sequence Alignment | . 277 |
| | Building a three-dimensional Manhattan | . 277 |
| | A greedy multiple alignment algorithm | . 280 |
| | Detours | . 282 |
| | Fireflies and the non-ribosomal code | . 282 |
| | Finding an LCS without constructing a city | . 283 |
| | Constructing a topological ordering | . 284 |
| | PAM scoring matrices | . 285 |
| | Divide-and-conquer algorithms | . 287 |
| | Scoring multiple alignments | . 289 |
| | Bibliography Notes | . 291 |
| 6 | Are There Fragile Regions in the Human Genome? | 292 |
| | Of Mice and Men | . 293 |
| | How different are the human and mouse genomes? | . 293 |
| | Synteny blocks | . 294 |
| | Reversals | . 294 |
| | Rearrangement hotspots | . 295 |
| | The Random Breakage Model of Chromosome Evolution | . 297 |
| | Sorting by Reversals | . 299 |
| | A Greedy Heuristic for Sorting by Reversals | . 304 |
| | Breakpoints | . 306 |
| | What are breakpoints? | . 306 |
| | Counting breakpoints | . 307 |
| | Sorting by reversals as breakpoint elimination | . 308 |
| | Rearrangements in Tumor Genomes | |
| | From Unichromosomal to Multichromosomal Genomes | |
| | Translocations, fusions, and fissions | . 311 |
| | From a genome to a graph | . 313 |
| | 2-breaks | . 314 |
| | Breakpoint Graphs | . 316 |
| | Computing the 2-Break Distance | . 320 |
| | Rearrangement Hotspots in the Human Genome | . 323 |
| | The Random Breakage Model meets the 2-Break Distance Theorem | . 323 |
| | The Fraoile Breakage Model | 324 |

| Epilogue: Synteny Block Construction | 325 |
|---|-----|
| Genomic dot-plots | 325 |
| Finding shared k-mers | |
| Constructing synteny blocks from shared <i>k</i> -mers | 329 |
| Synteny blocks as connected components in graphs | 331 |
| Open Problem: Can Rearrangements Shed Light on Bacterial Evolution? | 333 |
| Charging Stations | 335 |
| From genomes to the breakpoint graph | |
| Solving the 2-Break Sorting Problem | 338 |
| Detours | 340 |
| Why is the gene content of mammalian X chromosomes so conserved? . | 340 |
| Discovery of genome rearrangements | 340 |
| The exponential distribution | 341 |
| Bill Gates and David X. Cohen flip pancakes | 342 |
| Sorting linear permutations by reversals | 343 |
| Bibliography Notes | 346 |
| Bibliography | 349 |
| Image Courtesies | 355 |

List of Code Challenges

| Chapter 1 | 2 |
|---|---------|
| (1A) Compute the Number of Times a Pattern Appears in a Text | 8 |
| (1B) Find the Most Frequent Words in a String | 8 |
| (1C) Find the Reverse Complement of a DNA String | 12 |
| (1D) Find All Occurrences of a Pattern in a String | 13 |
| (1E) Find Patterns Forming Clumps in a String | 15 |
| (1F) Find a Position in a Genome Minimizing the Skew | 25 |
| (1G) Compute the Hamming Distance Between Two Strings | 27 |
| (1H) Find All Approximate Occurrences of a Pattern in a String | 27 |
| (1I) Find the Most Frequent Words with Mismatches in a String | 28 |
| (1J) Find Frequent Words with Mismatches and Reverse Complements . | 29 |
| (1K) Generate the Frequency Array of a String | 40 |
| (1L) Implement PATTERNTONUMBER | 42 |
| (1M) Implement NUMBERTOPATTERN | 43 |
| (1N) Generate the <i>d</i> -Neighborhood of a String | 50 |
| Chapter 2 | 66 |
| (2A) Implement MOTIFENUMERATION | 71 |
| (2B) Find a Median String | 81 |
| (2C) Find a <i>Profile</i> -most Probable <i>k</i> -mer in a String | 85 |
| (2D) Implement GreedyMotifSearch | 85 |
| (2E) Implement GreedyMotifSearch with Pseudocounts | 91 |
| (2F) Implement RANDOMIZEDMOTIFSEARCH | 93 |
| (2G) Implement GIBBSSAMPLER | 100 |
| (2H) Implement DISTANCEBETWEENPATTERNANDSTRINGS | 107 |

| Chapter 3 | 115 |
|---|-------|
| (3A) Generate the <i>k</i> -mer Composition of a String | 120 |
| (3B) Reconstruct a String from its Genome Path | 125 |
| (3C) Construct the Overlap Graph of a Collection of <i>k</i> -mers | 128 |
| (3D) Construct the de Bruijn Graph of a String | 132 |
| (3E) Construct the de Bruijn Graph of a Collection of k -mers | |
| (3F) Find an Eulerian Cycle in a Graph | 146 |
| (3G) Find an Eulerian Path in a Graph | 147 |
| (3H) Reconstruct a String from its <i>k</i> -mer Composition | 147 |
| (3I) Find a k-Universal Circular String | . 148 |
| (3J) Reconstruct a String from its Paired Composition | 157 |
| (3K) Generate the Contigs from a Collection of Reads | 160 |
| (3L) Construct a String Spelled by a Gapped Genome Path | 169 |
| (3M) Generate All Maximal Non-Branching Paths in a Graph | 169 |
| Chapter 4 | 182 |
| (4A) Translate an RNA String into an Amino Acid String | 186 |
| (4B) Find Substrings of a Genome Encoding a Given Amino Acid String | 187 |
| (4C) Generate the Theoretical Spectrum of a Cyclic Peptide | 191 |
| (4D) Compute the Number of Peptides of Given Total Mass | 193 |
| (4E) Find a Cyclic Peptide with Theoretical Spectrum Matching an Idea | ıl |
| Spectrum | |
| (4F) Compute the Score of a Cyclic Peptide Against a Spectrum | 198 |
| (4G) Implement Leaderboard Cyclopeptide Sequencing | 200 |
| (4H) Generate the Convolution of a Spectrum | 203 |
| (4I) Implement CONVOLUTION CYCLOPEPTIDE SEQUENCING | 205 |
| (4J) Generate the Theoretical Spectrum of a Linear Peptide | 211 |
| (4K) Compute the Score of a Linear Peptide | 214 |
| (4L) Implement TRIM to Trim a Peptide Leaderboard | 215 |
| (4M) Solve the Turnpike Problem | 219 |
| Chapter 5 | 222 |
| (5A) Find the Minimum Number of Coins Needed to Make Change | 240 |
| (5B) Find the Length of a Longest Path in a Manhattan-like Grid | 245 |
| (5C) Find a Longest Common Subsequence of Two Strings | 252 |
| (5D) Find the Longest Path in a DAG | . 253 |
| (5E) Find a Highest-Scoring Alignment of Two Strings | 255 |

| (5F) Find a Highest-Scoring Local Alignment of Two S | trings 260 |
|--|----------------------------|
| (5G) Compute the Edit Distance Between Two Strings | 262 |
| (5H) Find a Highest-Scoring Fitting Alignment of Two | Strings 263 |
| (51) Find a Highest-Scoring Overlap Alignment of Two | Strings 264 |
| (5J) Align Two Strings Using Affine Gap Penalties | 268 |
| (5K) Find a Middle Edge in an Alignment Graph in Lir | near Space 275 |
| (5L) Align Two Strings Using Linear Space | 276 |
| (5M) Find a Highest-Scoring Alignment of a Collection | of Strings 279 |
| (5N) Find a Topological Ordering of a DAG | 285 |
| Chapter 6 | 292 |
| (6A) Implement GREEDYSORTING to Sort a Permutation | on by Reversals 305 |
| (6B) Compute the Number of Breakpoints in a Permuta | ation 308 |
| (6C) Compute the 2-Break Distance Between a Pair of C | Genomes 321 |
| (6D) Find a Shortest Transformation of One Genome in | to Another via 2-Breaks322 |
| (6E) Find All Shared <i>k</i> -mers of a Pair of Strings | |
| (6F) Implement CHROMOSOMETOCYCLE | |
| (6G) Implement CYCLETOCHROMOSOME | |
| (6H) Implement COLOREDEDGES | |
| (6I) Implement GRAPHTOGENOME | |
| (6J) Implement 2-BreakOnGenomeGraph | |
| (6K) Implement 2-BreakOnGenome | |