

1. Record Nr.	UNINA9910583478103321
Titolo	Bioinformatics algorithms : design and implementation in python // Miguel Rocha and Pedro G. Ferreira
Pubbl/distr/stampa	London : , : Academic Press, an imprint of Elsevier, , [2018] ©2018
ISBN	0-12-812521-7 0-12-812520-9
Descrizione fisica	1 online resource (402 pages)
Disciplina	572.80285
Soggetti	Bioinformatics - Data processing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Front Cover -- Bioinformatics Algorithms -- Copyright -- Contents -- 1 Introduction -- 1.1 Prelude -- 1.2 What is Bioinformatics -- 1.3 Book's Organization -- 2 An Introduction to the Python Language -- 2.1 Features of the Python Language -- 2.2 Variables and Pre-De ned Functions -- 2.2.1 Variable Types -- 2.2.2 Assigning Values to Variables -- 2.2.3 Numerical and Logical Variables -- 2.2.4 Containers -- 2.2.4.1 Lists -- 2.2.4.2 Strings -- 2.2.4.3 Tuples -- 2.2.4.4 Sets -- 2.2.4.5 Dictionaries -- 2.2.5 Variable Comparison -- 2.2.6 Type Conversion -- 2.3 Developing Python Code -- 2.3.1 Indentation -- 2.3.2 User-De ned Functions -- 2.3.3 Conditional Statements -- 2.3.4 Conditional Loops -- 2.3.5 Iterative Loop Statements -- 2.3.6 List Comprehensions -- 2.3.7 Help -- 2.4 Developing Python Programs -- 2.4.1 Data Input and Output -- 2.4.2 Reading and Writing From Files -- 2.4.3 Handling Exceptions -- 2.4.4 Modules -- 2.4.5 Putting It All Together -- 2.5 Object-Oriented Programming -- 2.5.1 De ning Classes and Creating Objects -- 2.5.2 Special Methods -- 2.5.3 Inheritance -- 2.5.4 Modularity -- 2.6 Pre-De ned Classes and Methods -- 2.6.1 Generic Methods for Containers -- 2.6.2 Methods for Lists -- 2.6.3 Methods for Strings -- 2.6.4 Methods for Sets -- 2.6.5 Methods for Dictionaries -- 2.6.6 Assigning and Copying Variables -- Bibliographical Notes and Further Reading -- Exercises and Programming Projects -- Exercises -- Programming Projects -- 3

Cellular and Molecular Biology Fundamentals -- 3.1 The Cell: The Basic Unit of Life -- 3.2 Genetic Information: Nucleic Acids -- 3.2.1 Transcription: RNA Synthesis -- 3.2.2 Translation: Protein Synthesis -- 3.3 Genes: Discrete Units of Genetic Information -- 3.3.1 Gene Structure -- 3.3.2 Regulation of Gene Expression -- 3.4 Human Genome -- 3.5 Biological Resources and Databases. Bibliographic References and Further Reading -- Exercises -- 4 Basic Processing of Biological Sequences -- 4.1 Biological Sequences: Representations and Basic Algorithms -- 4.2 Transcription and Reverse Complement -- 4.3 Translation -- 4.4 Seeking Putative Genes: Open Reading Frames -- 4.5 Putting It All Together -- 4.6 A Class for Biological Sequences -- 4.7 Processing Sequences With BioPython -- 4.8 Sequence Annotation Objects in BioPython -- Exercises and Programming Projects -- Exercises -- Programming Projects -- 5 Finding Patterns in Sequences -- 5.1 Introduction: Importance of Pattern Finding in Bioinformatics -- 5.2 Naive Algorithm for Fixed Pattern Finding -- 5.3 Heuristic Algorithm: Boyer-Moore -- 5.4 Deterministic Finite Automata -- 5.5 Finding Flexible Patterns: Regular Expressions -- 5.5.1 Definitions and Regular Expressions in Python -- 5.5.2 Examples in Biological Sequence Analysis -- 5.5.3 Finding Protein Motifs -- 5.5.4 An Application to Restriction Enzymes -- Bibliographic Notes and Further Reading -- Exercises and Programming Projects -- Exercises -- Programming Projects -- 6 Pairwise Sequence Alignment -- 6.1 Introduction: Comparing Sequences and Sequence Alignment -- 6.2 Visual Alignments: Dot Plots -- 6.3 Sequence Alignment as an Optimization Problem -- 6.3.1 Problem Definition and Complexity -- 6.3.2 Objective Function: Substitution Matrices and Gap Penalties -- 6.3.3 Implementing the Calculation of the Objective Function -- 6.4 Dynamic Programming Algorithms for Global Alignment -- 6.4.1 The Needleman-Wunsch Algorithm -- 6.4.2 Implementing the Needleman-Wunsch Algorithm -- 6.5 Dynamic Programming Algorithms for Local Alignment -- 6.5.1 The Smith-Waterman Algorithm -- 6.5.2 Implementing the Smith-Waterman Algorithm -- 6.6 Special Cases of Sequence Alignment -- 6.7 Pairwise Sequence Alignment in BioPython. Bibliographical Notes and Further Reading -- Exercises and Programming Projects -- Exercises -- Programming Projects -- 7 Searching Similar Sequences in Databases -- 7.1 Introduction -- 7.2 BLAST Algorithm and Programs -- 7.2.1 Overview of the BLAST Algorithm -- 7.2.2 BLAST Programs -- 7.2.3 Significance of the Alignments -- 7.3 Implementing Our Own BLAST -- 7.4 Using BLAST Through BioPython -- Bibliographical Notes and Further Reading -- Exercises and Programming Projects -- Exercises -- Programming Projects -- 8 Multiple Sequence Alignment -- 8.1 Introduction: Problem Definition and Complexity -- 8.2 Classes of Optimization Algorithms for Multiple Sequence Alignment -- 8.2.1 Dynamic Programming -- 8.2.2 Heuristic Algorithms -- 8.3 Implementing Progressive Alignments in Python -- 8.3.1 Representing Alignments: Class MyAlign -- 8.3.2 Pairwise Alignment: Class AlignSeq -- 8.3.3 Implementing Multiple Sequence Alignment: Class MultipleAlign -- 8.4 Handling Alignments in BioPython -- Bibliographical Notes and Further Reading -- Exercises and Programming Projects -- Exercises -- Programming Projects -- 9 Phylogenetic Analysis -- 9.1 Introduction: Problem Definition and Relevance -- 9.2 Classes of Algorithms for Phylogenetic Analysis -- 9.2.1 Distance-Based Methods -- 9.2.2 Maximum Parsimony -- 9.2.3 Statistical Methods -- 9.3 Implementing Distance-Based Algorithms in Python -- 9.3.1 Implementing Binary Trees -- 9.3.2 Implementing the UPGMA Algorithm -- 9.4 BioPython Functions for Phylogenetic Analysis -- Bibliographical Notes and Further Reading -- Exercises and

Programming Projects -- Exercises -- Programming Projects -- 10
Motif Discovery Algorithms -- 10.1 Introduction: Problem Definition and
Relevance -- 10.2 Brute-Force Algorithms: Exhaustive Search -- 10.3
Branch-and-Bound Algorithms -- 10.4 Heuristic Algorithms.
Bibliographic Notes and Further Reading -- Exercises and Programming
Projects -- Exercises -- Programming Projects -- 11 Probabilistic
Motifs and Stochastic Algorithms -- 11.1 Representing and Searching
Probabilistic Motifs -- 11.2 Stochastic Algorithms: Expectation-
Maximization -- 11.3 Gibbs Sampling for Motif Discovery -- 11.4
Probabilistic Motifs in BioPython -- Bibliographic Notes and Further
Reading -- Exercises and Programming Projects -- Exercises --
Programming Projects -- 12 Hidden Markov Models -- 12.1
Introduction: What Are Hidden Markov Models? -- 12.2 Algorithms and
Python Implementation -- 12.2.1 Joint Probability of an Observed
Sequence and State Path -- 12.2.2 Probability of an Observed Sequence
Over All State Paths -- 12.2.3 Probability of the Remainder of an
Observed Sequence -- 12.2.4 Finding the Optimal State Path -- 12.2.5
Learning the Parameters of an HMM Model -- 12.3 HMMs for Database
Search -- Bibliographic Notes and Further Reading -- Exercises and
Programming Projects -- 13 Graphs: Concepts and Algorithms -- 13.1
Graphs: Definitions and Representations -- 13.2 A Python Class for
Graphs -- 13.3 Adjacent Nodes and Degrees -- 13.4 Paths, Searches,
and Distances -- 13.5 Cycles -- Bibliographic Notes and Further
Reading -- Exercises and Programming Projects -- Exercises --
Programming Projects -- 14 Graphs and Biological Networks -- 14.1
Introduction -- 14.2 Representing Networks With Graphs -- 14.2.1 A
Python Class for Metabolic Networks -- 14.2.2 An Example Metabolic
Network for a Real Organism -- 14.3 Network Topological Analysis --
14.3.1 Degree Distribution -- 14.3.2 Shortest Path Analysis -- 14.3.3
Clustering Coefficients -- 14.3.4 Hubs and Centrality Measures -- 14.4
Assessing the Metabolic Potential -- Bibliographic Notes and Further
Reading -- Exercises and Programming Projects -- Exercises --
Programming Projects.
15 Assembling Reads Into Genomes: Graph-Based Algorithms -- 15.1
Introduction to Genome Assembly and Related Challenges -- 15.2
Overlap Graphs and Hamiltonian Cycles -- 15.2.1 Problem Definition
and Exhaustive Search -- 15.2.2 Overlap Graphs -- 15.2.3 Hamiltonian
Circuits -- 15.3 DeBruijn Graphs and Eulerian Paths -- 15.3.1 DeBruijn
Graphs for Genome Assembly -- 15.3.2 Eulerian Paths -- 15.4 Genome
Assembly in Practice -- Bibliographic Notes and Further Reading --
Exercises and Programming Projects -- Exercises -- Programming
Projects -- 16 Matching Reads to Reference Sequences -- 16.1
Introduction: Problem Definition and Applications -- 16.2 Pre-
Processing the Patterns: Tries -- 16.2.1 Definitions and Algorithms --
16.2.2 Implementing Tries in Python -- 16.3 Pre-Processing the
Sequence: Suffix Trees -- 16.3.1 Definitions and Algorithms -- 16.3.2
Implementing Suffix Trees in Python -- 16.4 Burrows-Wheeler
Transforms -- 16.4.1 Definitions and Algorithms -- 16.4.2
Implementation in Python -- 16.4.3 Aligning References to Genomes in
Practice -- Bibliographic Notes and Further Reading -- Exercises and
Programming Projects -- Exercises -- Programming Projects -- 17
Further Reading and Resources -- 17.1 Complementary Books -- 17.2
Journals and Conferences -- 17.3 Formal Education -- 17.4 Online
Resources -- Final Words -- Bibliography -- Index -- Back Cover.
