CSE3800: Bioinformatics – Fall 2025

TuTh 3:30-4:45pm, ITE 125

This course is an introduction to the fundamental mathematical models and algorithmic techniques used in bioinformatics. Emphasis will be placed on modeling biological applications as graph-theoretic or mathematical optimization problems, and on designing, analyzing, and implementing efficient algorithms for solving these formulations. Covered algorithmic techniques will include exhaustive search, greedy algorithms, dynamic programming, divide-and-conquer, graph algorithms, combinatorial pattern matching, clustering, and randomized algorithms. Biological applications will include motif finding, sequence assembly, pairwise and multiple sequence alignment, genome rearrangement analysis, gene expression analysis, evolutionary tree reconstruction, and single-cell sequencing data analysis.

Prerequisites: CSE 1729 or CSE2050; and one of STAT 3025, STAT3345, STAT3375, or MATH 3160.
Recommended preparation: BIOL 1107.

Instructor

Ion Măndoiu
ion@engr.uconn.edu
Office: ITE 261
Office Hours:
Tu/W/Th 12:30-1:30pm
or by appointment

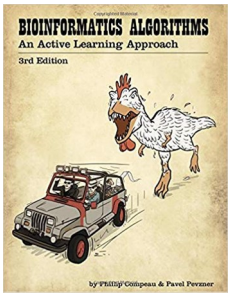
Teaching Assistant

Rye Howard-Stone
rye.howard-stone@uconn.edu
Office Hours:
M/W 4:30-6pm or by appointment
uconn-cmr.webex.com/meet/rhs19002

Course objectives

Upon completion of the course you will be able to:

- Translate biological questions into computational problems using mathematical models
- Design computer algorithms for answering relevant biological questions
- Use high-level programming languages to implement bioinformatics algorithms
- Evaluate efficiency and accuracy of existing bioinformatics software tools



Textbook

Phillip Compeau and Pavel Pevzner, *Bioinformatics Algorithms*, 3rd Edition.
Book website: https://www.bioinformaticsalgorithms.org

Additional materials will be distributed on Moodle.

Course website

We will use a course website hosted using Moodle at https://edx.engr.uconn.edu/. Please check this site regularly to access assignments, grades, and course materials including videos, handouts, etc. The Moodle site also integrates a discussion forum to ask class-related questions and communicate with the instructor and your peers. Please use this forum for general questions about the covered material and clarifications on the assignments. For questions that are specific to your own work you should contact the instructor directly.

Assessment and grading

Grading for the course will be based on interactive videos (10%), online quizzes (10%), theoretical homework assignments (20%), programming assignments (20%), and a final project (40%). Interactive videos with embedded questions will be linked on Moodle for each topic; it is essential that you watch these videos prior to the corresponding class meetings to ensure you are prepared to actively participate in class discussions. Short online quizzes based on the covered material will be scheduled throughout the semester. In addition to the interactive videos and quizzes, course assessments will include bi-weekly theoretical homeworks and programming assignments along with a final group project. Solutions to both theoretical homeworks and programming assignments must be submitted in electronic format via Moodle. The recommended language for solving programming assignments is Python, but common programming language including Java, C, C++, and R can also be used. Solutions to the programming assignments will be automatically checked for correctness on a set of standard test cases, allowing you to receive immediate feedback and fix potential problems before the due date. The final group project is designed to give you the opportunity to study a bioinformatics application in more depth. Project requirements will include submitting several written reports and short presentations. Suitable final project topics include surveys of bioinformatics topics not covered in lectures, design and implementation of novel algorithms, and empirical comparisons of existing bioinformatics tools. Full final project details including a list of potential topics will be provided towards the middle of the semester.

Late policy

Unless otherwise specified, quizzes, theoretical homeworks, and programming assignments are due at midnight on the specified due date. Late submissions are allowed for up to three days with a 10% penalty for each late day. To accommodate unforeseen circumstances that may prevent timely submission, the lowest quiz, theoretical homework, and programming assignment scores will be dropped from the final grade calculation.

Academic integrity

You may discuss ideas and concepts with others, but must not share quiz questions and answers, homework solutions, or code. The use of generative artificial intelligence (GAI) tools such as ChatGPT is disallowed unless otherwise stated in the assignment. When allowed, GAI use must be ap-

propriately acknowledged and cited.

Students with disabilities

If you have a documented disability for which you are or may be requesting an accommodation, you are encouraged to contact the instructor and the Center for Students with Disabilities or the University Program for College Students with Learning Disabilities as soon as possible to ensure that such accommodations are implemented in a timely fashion.

Tentative Schedule

Meeting dates	Topic	Textbook chapter
Aug 26 & 28	Course intro; Finding DNA replication origins	Ch. 1
Sept 2 & 4	Motif finding	Ch. 2
Sept 9 & 11	Genome assembly	Ch. 3
Sept 16 & 18	Sequence alignment	Ch. 5
Sept 23 & 25	Sequence alignment	Ch. 5
Sept 30 & Oct 2	Pattern matching	Ch. 9
Oct 7 & 9	Pattern matching	Ch. 9
Oct 14 & 16	Antibiotics sequencing	Ch. 4
Oct 21 & 23	Computational proteomics	Ch. 11
Oct 28 & 30	Genome rearrangements	Ch. 6
Nov 4 & 6	Evolutionary tree reconstruction	Ch. 7
Nov 11 & 13	Gene expression analysis	Ch. 8
Nov 18 & 20	Randomized Algorithms	Ch. 10
Dec 2 & 4	Hidden Markov Models	Ch. 10