

Comp 555 - Bioalgorithms

Bulletin Description

Bioinformatics algorithms. Topics include DNA restriction mapping, finding regulatory motifs, genome rearrangements, sequence alignments, gene prediction, graph algorithms, DNA sequencing, protein sequencing, combinatorial pattern matching, approximate pattern matching, clustering and evolution, tree construction, Hidden Markov Models, randomized algorithms.

General Course Info

Term: SPRING 2019
Department: COMP
Course Number: 555
Section Number: 001

Time: T Th, 9:30 - 10:45
Location: SN 014
Website: <http://www.csbio.unc.edu/mcmillan/?run=Comp555S19>

Instructor Info

Name: Prof. Leonard McMillan
Office: SN 316
Email: mcmillan@cs.unc.edu
Phone: 919-590-6078
Web: <http://www.cs.unc.edu/~mcmillan>
Office Hours: W 2pm-4pm

Textbooks and Resources

There will be no textbook semester. In the past, I have used the following, which you might find useful as supplement reading material:

Bioinformatics Algorithms: An Active Learning Approach
by Phillip Compeau and Pavel Pevzner
Active Learning Publishers © 2014, ISBN: 978-0-9903746-0-2.

Course Description

Computational methods are fueling a revolution in the biological sciences. Computers are already nearly as indispensable as microscopes for analyzing and interpreting biological data. As a result, two new multidisciplinary fields, bioinformatics and computational biology, have emerged. This course will explore the computational methods and algorithmic principles driving this revolution. It will cover basic topics in molecular biology, genetics, and proteomics. The course also addresses basic computational theory and algorithms including asymptotic notation, recursion, divide-and-conquer approaches, graph algorithms, dynamic

programming, and greedy algorithms. These fundamental concepts from computer science will be taught within the context of motivating problems drawn from contemporary biology. Example biological topics include sequence alignment, motif finding, gene rearrangement, DNA sequencing, protein peptide sequencing, phylogeny, and gene expression analysis.

This course is suitable for both computer science and biology students at both undergraduate and graduate levels. Students taking this course should have some programming experience in a modern language.

Target Audience

This course is intended for advanced undergraduate computer science majors and graduate students who are interested in exploring the challenges of bioinformatics and computational biology. It is also well suited for graduate students from Biostatistics and the Curriculum in Computational Biology who would like to better understand the algorithms underlying the common bioinformatics analysis tools.

Prerequisites

Previous offerings listed COMP 410 and one of COMP 283 or MATH 381 as prerequisites. This was to allow COMP 555 to perhaps someday substitute for COMP 550. However, as of this semester I am retooling the course to address a more multidisciplinary audience. Thus, I am allowing students, with my permission, to take COMP 555 without the standard prerequisites. However, every student is expected to have taken a first and second course in programming equivalent to COMP 110/116 and COMP 401 offered in our department. It will take time for this to percolate to the UNC Undergraduate Bulletin.

Goals and Key Learning Objectives

Comp 555 emphasizes the design, performance, and application of algorithms. All algorithms are motivated by problems from modern biology. A major objective of Comp 555 is to develop a student's intuition for what is the appropriate algorithm to use in a given setting, what is a correct algorithm, and how to analyze the performance and scalability of an algorithm. Comp 555 also exposes the methods underlying many of the common tools used in modern quantitative biology.

Course Requirements

Students must complete assigned reading from the textbook specified in each lecture. Student will be assigned problem sets, with associated programming assignments to be completed on their own computers using a programming language designated by the instructor (Python in this offering). There will be five homework assignments, a midterm, and a final exam.

Key Dates

Midterm:	March 5 or 7	(during class)
Final Exam:	May 3, 2018	(Friday, 8am-11am)

Grading Criteria

The final grade will be based on the follow weighting factors:

5 – Problem Sets w/Programming Exercises	40% (8% each)
1 – Midterm Exam	20%
1 – Final Exam	20%
Many unannounced in-class exercises/quizzes	20% (lowest 2 dropped)

Course Policies

This section should address the following:

- Attendance is expected, but no roll will be taken
- Late problem sets will be penalized by a factor of 0.7071 for each class meeting after the assigned due date. Submission of problem sets is on-line, but they will be considered “due” just before midnight of their due date.

The course final is given in compliance with UNC final exam regulations and according to the UNC Final Exam calendar.

Cheating and the UNC Honor Code

Collaboration on assignments is encouraged. However, what you hand in must be your own work. Good scholarship requires that all collaboration must be acknowledged. Thus, if you collaborate on the solution of a problem set, I expect that you list your collaborators at the top of the page. Turning in code (any function or fragment) written by some other person or found on the Internet is not collaboration. Discussing algorithmic design approaches and sharing test cases are good examples collaboration.

Collaboration on tests (midterms, final) in any form is, of course, a violation of the Honor Code. This includes discussion of questions on a midterm, or final with students that have not yet taken the test.

Using any unauthorized information source, such as social media (Facebook, WeChat, Slack, Shared Google Docs, etc.), texting, or any other interactive means of contacting another person during an exam is a violation of the honor code.

Course Schedule

A course schedule and handouts from each lecture will be posted on the course website.

Disclaimer

“The professor reserves to right to make changes to the syllabus, including due dates and test dates. These changes will be announced as early as possible.”