Fall 9-1-2018

CSCI 558.00: Introduction to Bioinformatics

Oliver Serang
University of Montana, Missoula

CSCI 451 / 558: Computational Biology / Bioinformatics

Basic Information
Semester: Autumn 2018
Lectures: M/W/F 2pm-2:50pm in Social Science 362
Final Exam: TBD

Instructor
Instructor: Oliver Serang, Social Science 408, oliver.serang@umontana.edu
Office Hours: M/W 3pm-4pm (SS 408)
Course webpage: alg.cs.umt.edu/lectures.html

Course Description
This course introduces algorithms and tools for computational biology (both classic and new). These include sequence alignment, finding identical regions of genomes, fundamental mass spectrometry, phylogenetics, *etc*. It's a great field with limitless options!

Course Objectives
After taking this course, students will have a good understanding of the origins of modern biological data, algorithms for analyzing this data, and techniques for implementing these algorithms.

Learning Outcomes
Students will write descriptions of how to solve problems and write programs from those descriptions. Students will then learn to solve biological problems in Python and C++ and to understand and use classic algorithms.

Textbook
No textbook is required! All relevant reading will be provided or free resources will be used.

Lanugages Used
Python (primary) and C++ (secondary).

Course Prerequisites
Programming ability in Python and some understanding of C++

Grading Scheme
Graduate (CSCI 558):

40% Active participation
(Includes attendance, participation in discussion, required reading)

40% Out-of-class programming assignments
(Must be turned in before deadline-- do them early-- they are trickier than they seem)

20% Final exam

Undergraduate (CSCI 451):

80% Active participation
(Includes attendance, participation in discussion, required reading)
20% Final exam

Difference Between Undergraduate and Graduate Version
The graduate version will occasionally have separate, more complex programming tasks that are more open ended.

Grading Scale
Grades will be determined by the instructor using the performance of students at the boundary.

Exam Material
The exam will be drawn from topics discussed in the in-class lectures.

Grading Out-of-class Programming Assignments
These will be graded by verifying the program produces the correct output (using diff). Accuracy will be very important, and partial credit will only be given on a case-by-case basis as decided by the instructor. (For completeness: the code must also be correct, and not simply produce the correct output for one example.)

Students' code will be run in a Linux environment using the g++ compiler; other compilers (*e.g.*, Visual Studio) may be used privately by the students, but it is their responsibility to ensure the code compiles and runs effectively in Linux via g++.

Attendance Policy
Come to class. On-time attendance is necessary for active participation.

Cheating
Feel free to discuss broad strategies for the assignments together (*e.g.*, in front of a white board); however, do not write code together, copy someone else's code, or copy code from a foreign source (*e.g.*, the internet)-- protect yourself, this will be seen as cheating. Academic dishonesty (including plagiarism and cheating) will not be tolerated as specified by the university conduct code. If you have questions, please ask in advance.

Disabilities
Requests from students can be made with advance notice by emailing the instructor. For information on official university policy, please see http://www.umt.edu/disability.

Electronic Devices
Set your phone to silent (or off) during class and take calls outside the classroom. Students texting during class will be asked to leave.

We are a team and I'm really looking forward to playing with these fun ideas together!