Course Title: BINF8500 "Algorithms for Bioinformatics"

Instructor: Jan Mrázek (mrazek@uga.edu)

Class meets: Tuesdays & Thursdays 2:00-3:15, room C128 Life Sciences

Textbook: No textbook is required for this course. Relevant material will be distributed by the instructor.

Approximate topical outline:
Week 1: Introduction, technical issues.
Weeks 2-3: Sorting algorithms (quicksort)
Weeks 4-5: Clustering algorithms (k-means clustering)
Weeks 6-7: Paiwise sequence alignment (Needleman-Wunsch algorithm)
Weeks 8-10: Supervised motif finding (Position-specific score matrix)
Weeks 11-15: Unsupervised motif finding (Gibbs sampler)

The time allotted to each subject is approximate and the topics themselves may be modified depending on the overall class progress. Each topic will include explanation how the algorithm works, alternative algorithms, biological background and applications, discussion of problems one may encounter during implementation and what one can do to improve efficiency of the algorithm, writing the program, and in-class presentations of your code and creative solutions.

Grading: Students will write computer programs implementing several algorithms (listed above). Each program will be graded by up to 40 points. Additional points may be earned in the classroom or for special achievements (e.g., writing the fastest program in the class).

Class attendance: mandatory. Penalties will be imposed for absences without a valid excuse.

Notes on programming languages:
        Every student is expected to have mastered a certain level of programming skills such that the course can focus on implementation of algorithms as opposed to programming basics such as loops, arrays, or file manipulations. No single programming language is prescribed that you have to use in the class. C/C++ or Java are strongly recommended. Python may be acceptable in some situations per the instructor's permission.

Notes on what is cheating and what is not:
        Sometimes there is a thin line between what is cheating and what is desirable collaboration. The bottom line is that **each project is your individual work and you should be the one who wrote all of the final code**. You cannot copy or modify someone else's code and you cannot allow others to copy or modify yours. You also cannot look at your classmates' or former students' completed projects. That said, you can talk to your friends about their solutions and look at code you find on the internet if it helps you to understand how the method works. However, then you have to put any such code designed by others aside and write your own without looking at (or having memorized) how exactly the code you saw was written.

You can use someone else's code in your program if you disclose it clearly in comments. You can make a comment at the start of your code such as "This program uses parts of the code from [sources], which are marked below" and mark clearly the parts that are not your own. **You will be penalized by point deductions based on how significant part of someone else's code you use but this will not be considered cheating for the purposes of the academic honesty policy** as long as you clearly distinguish what is your own work and what is not and give credit where it is due.

Notes on what you can use in your code:
Specialized libraries like BioJava implement some of the algorithms we are going to design and there is a number of additional functions that would make the assignments far too easy and eliminate their main purpose – understanding even the small nuances of the algorithm and its practical implementation. You are therefore not allowed to use any of the following:
- Absolutely no Biojava, Biopython, or anything similar.
- No functions or operators that are applied to whole arrays or other data structures. For example, you cannot use a min() function to find the smallest number in an array; you have to write the appropriate loop or write your own equivalent of the min() function. Exception: you can use functions or operators to copy, concatenate, or compare strings, which are technically arrays of characters, as well as standard input and output functions. Naturally, you can also use your own functions.
- Anything else that violates the spirit of the limitations above (if not sure, ask)

Like with using someone else's code, you can violate these rules if you disclose it clearly in comments at the start of the program at the cost of point deductions. Undisclosed violations of the rules will incur bigger deduction.

PLEASE NOTE:

As a University of Georgia student, you have agreed to abide by the University's academic honesty policy, "A Culture of Honesty", and the Student Honor Code. All academic work must meet the standards described in "A Culture of Honesty" found at: http://www.uga.edu/honesty. Lack of knowledge of the academic honesty policy is not a reasonable explanation for a violation. Questions related to the course assignments and the academic honesty policy should be directed to the instructor. The link to more detailed information about academic honesty can be found at: http://www.uga.edu/ovpi/honesty/acadhon.htm

The course syllabus is a general plan for the course; deviations announced to the class by the instructor may be necessary.

Additional office hours may be arranged with the instructors as needed. Please use email to set up appointment times.