CS5566 Fall 2016 Assignment II. 100 pts. NAME(S):

OBJECTIVES: k-mer and motif analysis; Learning to program in a scripting language
RESOURCES: Textbook, Class website, Internet

Electronic submission on Blackboard is due by 11 pm on Friday, Oct 12th, 2016. Submissions received after the deadline will be graded only for effort and receive a maximum of only 70% of the grade (Refer to class syllabus for detailed grading policy). If you are submitting this jointly as a 2-member team, include all names on a single upload and remember that undergrads and grad students cannot partner with each other. Questions marked as G are optional for undergraduate students. All answers should be in your own words with all sources you referred to cited at the appropriate places (Refer to class syllabus for detailed policy on plagiarism). It is highly recommended to use Python for this assignment.

For programming assignments, the submission should include this page as the first page of your assignment, pseudo-code reflecting your understanding of the solution, and examples of actual input and output to validate your implementation. This report should be a standalone summary of your work. A plain text file version of your code should be submitted (include your name as comments at the beginning of the code file), together with test input file(s), and a README file that includes instructions on how to run the program. Combine all files into a single ZIP file before uploading them to Blackboard. State any assumptions you make and show work for maximum credit. In addition to completeness and accuracy, grading will also take into account the modularity and clarity of your implementation. For example, it is necessary to read and write input and output using filenames as arguments rather than hardcoding filenames. It should be possible to run programs from a command prompt or shell by typing "python programName InputFileName OutputFileName." This will help you reuse your code for subsequent assignments. It is necessary to write this from scratch rather than by using pre-existing code. **Make sure your program prints intermediate output to a file so that the underlying steps and reasoning are self-evident.**

1. (50UG/40G) Download the set of partial sequences for any motif from the JASPAR database (http://jaspar.genereg.net/; 'Browse' and click on a sequence logo to obtain a set of sequences for a motif). Write a program that can recover the motif from the sequences using one of the following approaches - brute force, hash function, or CONSENSUS. Discuss to what extent you were able to recover the motif.
2. 50UG/40G) Design and implement a program that computes the similarity between two FASTA format sequences based on their 2-mer composition. Evaluate the accuracy of your program by comparing it with the output of an existing implementation of global pairwise sequence similarity. Compute a correlation coefficient between the scores generated by the two programs.

3G. (20G)

   a. A multiple sequence alignment is necessary to determine the degree of similarity between the sequences. However, one needs to know the degree of similarity to be able to choose the appropriate amino-acid substitution matrix for a multiple-sequence alignment. How can this paradox be resolved?
   b. DNA motifs can sometimes be variable in length. Outline an approach to detect motifs that can vary in length.