COM S 567 : Bioinformatics I (Fundamentals of Genome Informatics)

Fall Semester, 2016
Assignment #4

Due Date : 5pm, Tuesday, Oct 11

Homework Guidelines

Provided in the attached zip file is a collection of skeleton, source code files to be used in completing the following problems. Your grading will be based on the functionality of the methods specified; they must have the exact signatures provided, though you are free to create your own classes/methods/constructs as you see fit. Please submit the completed template files along with all other source files you create. Assert that your submission is zipped and that all contents are located in the proper directory/package.

Problem 1 : Enumeration

Write a Java class called EnumerationScheme which represents a general, tree-based, pre-order enumeration sceheme similar to that mentioned in class, but with the caveat that each digit in the enumeration can have a different value range. Your task is to implement the "nextVertex" method, which takes as input a vertex (represented as a java.util.List of Integers), an integer L, representing the total number of digits, and a List of Integers k, where the ith element of k is total number of values the ith digit can take. Your class should be located in the package cs567. Please see EnumerationScheme.java for more detailed information.

Problem 2 : Solving Using Enumeration

Using your newly crafted enumeration scheme from problem 1, design solutions to the motif finding problem and the median string problem. However, instead of dealing with input as a fixed t by n matrix of DNA sequences, each sequence may be of differing length (i.e. n might not always be the same). Your task is to complete the methods contained within "SolutionsByEnumeration.java": "findMotif", "score", "medianString", "hd", "totalDistance". Please see SolutionsByEnumeration.java for more examples and illustrations as to how DNA input Strings will be structured. Example input is also provided. This class should be located in the cs567.hw4 package.

Problem 3 : Analysis

Evaluate the runtime of your algorithms from problem 2. Fix an n and an l, where n is the length of each DNA sequence in an input matrix, and l is the length of an l-mer to find. What is the largest n that your algorithm/computer can handle in an hour?