Due Date: November 6, 2017, Monday (23:55)

CENG 465
Introduction to Bioinformatics
Fall 2016-2017

Assignment #1

Programming Assignment

Local Alignment of Whole Genomes using a Reference Sequence

Given two genomes, G1 and G2, of size ***M*** and ***N*** respectively, finding the best local alignment with dynamic programming (i.e., Smith-Waterman algorithm) takes O(***MN***) time and space. In this assignment, with the help of a reference sequence, RS, of size ***K***, you will implement a heuristic which perform local alignment of these two genomes in O(***M***+***N***+$\boldsymbol{K}^2$) time and O($\boldsymbol{K}^2$) space. The reference sequence is expected to be a much smaller sequence than the input genomes (possibly a sequence of one or more genes) and it may be from a different organism then these genomes. The idea is to quickly find regions in G1 and G2 which are potentially similar to RS and perform the local alignment of these regions only. Potentially similar regions to RS will be found by using counts of nucleotide triplets.

Formally, you will be given three input strings in three separate files, G1.txt, G2.txt, RS.txt (each sequence in a single line in these files). For the reference sequence, you will construct a 64 dimensional nucleotide triplet count vector by scanning the reference sequence from left to right. Let's call this vector VRS. Each of the integer entries (0 to ***K***-2) in VRS will store the number of occurrence of a particular nucleotide triplet, e.g., entry 1 may correspond to the number of AAAs in RS. You are free to order the triplets in this vector in any way you want. The second stage of the algorithm will scan both G1 and G2, one after the other, to find a subsequence of size ***K*** in each genome with a count vector most similar to VRS. For computing the distance between two 64 dimensional vectors you will use the L1-norm (i.e. Manhattan distance) which is given as:

$$d(VRS, VG1) \ = \sum_{i=1}^{64} |VRS_i - VG1_i|$$

where $|n|$ represents the absolute value of n. In order to find the best subsequence of size ***K*** in both genomes, you will slide a window of size ***K*** over the genome and compute nucleotide triplet counts. Note that when you slide the window one base to the right, the new count vector can easily be found by updating the previous one: by incrementing the count of the new triplet that occurs at the right and by decrementing the leftmost triplet that dissapears. After scanning the two genomes (in O(***M***+***K***) time) you will have found the two window locations with minimum distance to the VRS vector in both genomes (i.e., two subsequences of G1 and G2 potentially most similart to RS). In the final stage, you will locally align these subsequences of G1 and G2 using Smith-Waterman and report both the alignment score and the alignment itself (in O($\boldsymbol{K}^2$) time and space). Use a match score of +4, a mismatch score of -3 and a linear gap penalty of -2 in your alignment. You may assume that ***K*** is at least 50, and $\boldsymbol{M},\boldsymbol{N} \geq 10\boldsymbol{K}$ and $\boldsymbol{M},\boldsymbol{N} \leq 10^8$.

Notes:

You may write your code in any programming language of your choice.

Testing, verifying, and debugging your code is part of the assignment. I will not provide any test cases and their outputs. Imagine you are working in a Bioinformatics company and you are the first one to implement this proposed solution. How would you convince your bosses that what you implemented works correctly?

You should take the input sequences preferably as command line arguments because many bioinformatics tools are command line programs that take their inputs from the command line.
Example usage: `> align g1.txt g2.txt rs.txt`

This is an individual assignment. Be extra careful about not sharing any code fragment with your friends. If such shared code is detected, both parties will get a 0 (zero) from this assignment.

Submission

Submit your source code only as a single file (for example, send only *.c, *.java, *.cpp, *.py) via ODTU-Class before the deadline. Late submission is -20 pts per day.