COMP 462 / 561 – Computational Biology Methods
Assignment #1
Due date: October 5th 2016, in class

Question 1. (10 points) Needleman-Wunsch algorithm
What is the optimal global alignment for APPLE and HAPE? Show all optimal solutions and the corresponding paths under the match score of +1, mismatch score of -1, and indel penalty -1 (linear gap penalty).

Question 2. (5 points) Affine gap penalty
Give an example of a pair of short sequences for which the optimal pairwise alignment under the linear gap penalty scheme described in question 1 is different from the optimal pairwise alignment under an affine gap penalty with cost(l) = 2 + 0.5*l.

Question 3. (40 points) Sequence alignment with arbitrary gap penalty
When aligning two sequences, it is important to choose an alignment scoring scheme (substitution matrix cost and gap penalty function) that best captures the types of evolutionary events that are common in the type of sequences being aligned. For example, if the two sequences being aligned are protein-coding genes, then insertions or deletions of length 3, 6, 9, etc. nucleotides are common, whereas those whose length is not a multiple of three are rare, because they would result in a frameshift, i.e in the complete change of the amino acid encoded by the portion of the gene that follows the indel point.

a) (20 points) Give the pseudocode of an algorithm for pairwise global alignment that would work for the following gap penalty function, where L is the length of the indel:

cost(L) = a * L, if L is a multiple of 3
b * L, if L is not a multiple of 3

Parameters a and b would be chosen so that a < b. Your algorithm should run in time O(m n), where m and n are the lengths of the two sequenes.

b) (15 points) Implement your algorithm, in the language of your choice. Your algorithm should take six arguments: (1) File containing the two sequences to be aligned (FASTA format). (2) the value of a in the equation above. (3) the value of b in the equation above (4) the score for matches, (5) score for transition mistmatches, (6) score for transversion mismatches. Use your program to compute the optimal alignment and alignment score for the two coding sequences available at http://www.cs.mcgill.ca/~blanchem/561/BRCA1.fa Those are the coding sequences for the BRCA1 gene, an important gene involved in breast cancer, in human and mouse.
Report the score of the alignment, and submit your code on MyCourses. Your code will not be marked, but submitting an answer (alignment score) that would not have been produced by your code would be considered cheating.

c) (5 points) Use the alignment you have obtained in (b) to answer the following question: Suppose that the subsequence TTGGTATCTGACACTGACTACGACACTCAGAA, which is found in the mouse BRCA1 gene, has been found to be associated to breast cancer in mouse. What is the position of the corresponding sequence in human? Again,

the answer must be obtained by manually looking at the alignment produced by your program.

Question 4. (5 points) Longest common subsequence problem
Suppose that you have a very fast machine to execute the Smith-Waterman algorithm with any user-specific substitution cost matrix and linear gap penalty (such special-purpose machines exist and were fashionable in the 90's). How would you use it to solve the longest common subsequence: Given two DNA sequences S and T, find the longest sequence X that is a subsequence of both S and T.

Note: Subsequences and substrings are two different things. A subsequence is made of characters that occur in the right order in the input spring, but not necessarily consecutively. A substring is a set of consecutive characters of the input string. So every substring is a subsequence, but not vice-versa. For example, BOICS is a subsequence of BIOINFORMATICS, but it is not a substring. IRO is neither a subsequence nor a substring of BIOINFORMATICS.

Question 5. (10 points) NoDeletion alignment problem
Consider the NoDeletion global alignment problem, which consists of optimally aligning sequences S and T under a linear gap penalty for insertions but where deletions from S to T are not allowed. Let m and n be the lengths of S and T respectively, and let k = n – m (of course, the problem only makes sense if m <= n). Give an algorithm to solve the problem in time O(m*k).

Question 6. (20 points) Linear space pairwise alignment
The standard definition of the Needleman-Wunsch algorithm requires O(m*n) space to align sequences of length m and n respectively. In many cases, this is prohibitive. Of course, one can compute the score of the optimal alignment in linear space trivially by only keeping in memory the last two rows of the dynamic programming table. Let's call this the ForgetThePast-NW algorithm. However, this prevents us from recovering the alignment that achieves that score, since trace-back is impossible.

Describe a O(m+n) space algorithm to compute both the score and the alignment itself. Hint: Think Divide-and-Conquer! Find out where the path corresponding to the optimal alignment will cross the row m/2, using two calls to a modified ForgetThePast-NW. Then, recurse.

Question 7. (10 points) Progressive multiple sequence alignment
The progressive alignment algorithm we have seen in class to solve the multiple sequence alignment is not guaranteed to produce an optimal solution. Assume a sum-of-pairs scoring scheme with a linear gap penalty of c = -2 and the cost of transition mismatches is -1 and the cost of transversion mismatches is -2.

Give a simple example of specific short DNA strings where the algorithm fails to produce an optimal result. Give the score of both the alignments produced by the algorithm and the optimal alignment.

Question 8. (5 points) Blast algorithm
Give an example of the two most similar DNA sequences of length 20 that Blast using word length w=5 will fail to align.