Homework assignment 1

October 1, 2019

Solve the following problems, using programming language of your choice.

1. A string s is a supersequence of another string t if s contains t as a subsequence (if you don’t know exact mathematical definition, consult with Google). A common supersequence of strings s and t is a string that serves as a supersequence of both s and t. For example, ”GACCTAGGAACTC” serves as a common supersequence of ”ACGTC” and ”ATAT”. A shortest common supersequence of s and t is a supersequence for which there does not exist a shorter common supersequence. Continuing our example, ”ACGTACT” is a shortest common supersequence of ”ACGTC” and ”ATAT”.

Given: Two DNA strings s and t.
Return: A shortest common supersequence of s and t. If multiple solutions exist, you may output any one.

2. An overlap alignment between two strings s and t is a local alignment of a suffix of s with a prefix of t. An optimal overlap alignment will therefore maximize an alignment score over all such substrings of s and t.
Given: Two DNA strings s and t, each having length at most 10 kbp.
Return: The score of an optimal overlap alignment of s and t, followed by an alignment of a suffix s of s and a prefix t of t achieving this optimal score. Use an alignment score in which matching symbols count +1, substitutions count -2, and there is a linear gap penalty of 2. If multiple optimal alignments exist, then you may return any one.

3. An affine gap penalty is written as a+b(L1), where L is the length of the gap, a is a positive constant called the gap opening penalty, and b is a positive constant called the gap extension penalty.

We can view the gap opening penalty as charging for the first gap symbol, and the gap extension penalty as charging for each subsequent symbol added to the gap.

For example, if a=11 and b=1, then a gap of length 1 would be penalized by 11 (for an average cost of 11 per gap symbol), whereas a gap of length 100 would have a score of 110 (for an average cost of 1.10 per gap symbol).

Given: Two protein strings s and t in FASTA format (each of length at most 100 amino acids).

Return: The maximum alignment score between s and t, followed by two augmented strings s and t representing an optimal alignment of s and t. Use:

The BLOSUM62 scoring matrix. Gap opening penalty equal to 11. Gap extension penalty equal to 1.