Homework 4

CS 4390/5390
Fall 2019

Due: 13 November 2019

This homework is worth 6 points out of the total 25 points of homework in the class.

1. **(1 Point)** Assume that we compare the sequences below by counting the shared number of 3-mers. For the following sequences, which pair is more similar.
   - S_1 = `ACGTCGATC`
   - S_2 = `CCGGCGTCA`
   - S_3 = `ACGCTCGAT`

2. **(2 points)** Given a set of sequences $S_1, S_2, S_3, ..., S_k$, we would like to find k substrings $T_1, T_2, T_3, ..., T_k$ of $S_1, S_2, S_3, ..., S_k$ respectively, such that the optimal SP score of the multiple sequence alignment of $T_1, T_2, T_3, ..., T_k$ is maximized. (a) Design a dynamic programming algorithm to solve the problem with match, mismatch, and indel penalties of α, β, γ respectively (i.e. not affine gap scoring). (b) What is the time complexity? (Note that when $k = 2$, this problem is the same as pairwise local alignment).

3. **(1 point)** Given an additive tree $T = (E, V)$ for n species. (a) Describe an algorithm for reconstructing the distance matrix between all of the species. (b) What is the time complexity of this algorithm?

4. **(2 point)** When constructing a neighbor-joining tree from a set of taxa, we select the pair to merge that is equally balances the desire to find the pair that is closest in distance and the desire to find the pair that is *on average* furthest from everything else. We could also look at the minimum distance between a group and the other groups. (a) Create an algorithm that merges pairs of groups that:
   - unequally tries to (weighted using parameter α)
   - the minimize distance between the two groups, and

- the maximize the minimum distance between either of the two groups and any other group not in the pair.

(b) What is the time complexity of your new algorithm?