COMP 462 / 561 – Fall 2019
Homework #2

DUE DATE: October 18 2019, 23:59, on MyCourses

Question 1. (30 points)
To answer this question, you will need to use some online tools implementing some of the algorithms seen in class. These might include:
Blast (http://www.ncbi.nlm.nih.gov/BLAST)
and
the LIRMM Phylogenetic inference package
http://phylogeny.lirmm.fr/phylo_cgi/index.cgi

If you have never used Blast before, you may find this tutorial useful: http://www.ncbi.nlm.nih.gov/books/NBK1734/ . In particular, learn what an E-value is, and what the different Blast versions do (Blastn vs MegaBlast vs Blastp vs TBlastN, etc.)

Context: You are doctor and you have a patient suffering from a mysterious infection. You've extracted DNA from the infected area, sequenced it, and obtained the DNA sequence at http://www.cs.mcgill.ca/~blanchem/561/mystery.fa .

a) (15 points) What do you think is the cause of the infection? What is the name of the disease?
Hint: Default parameters for blastn may not result in the identification of very useful hits. Consider changing some of these parameters in order to try to identify hits with good E-values.

b) (15 points) Suppose there exist treatments for various strains of the pathogen. Five known strains exist, with sequences given at:
http://www.cs.mcgill.ca/~blanchem/561/strains.fa
Which treatment (i.e. that for which strain) is the most likely to be appropriate for your patient? Use a phylogenetic inference tool such as http://phylogeny.lirmm.fr/phylo_cgi/index.cgi to figure it out. Explain your answer.

Question 2. (30 points)
The first algorithm to calculate the parsimony score of a given set of nucleotides at the leaves of a given rooted binary tree was invented by Fitch and Wagner in 1971. The algorithm works as follows. For each node *u* of the tree *T*, define a set X_u as follows:

```
If $u$ is a leaf then
    $X_u = \{x\}$, where $x$ is the nucleotide at leaf u.
    Score(u) = 0

If $u$ is an internal node with children $v$ and $w$, then

    If ($X_v \cap X_w = \emptyset$ ) then
            $X_u = X_v \cup X_w$
            Score(u) = Score(v) + Score(w) + 1
    Else
            $X_u = X_v \cap X_w$
            Score(u) = Score(v) + Score(w)
```

After all the *X* sets have been computed, starting from the leaves back to the root, Score(root) is the desired parsimony score.

Question: Prove that the algorithm always yields the correct (minimal) parsimony score. Perhaps the easiest (but not the only) way to do this is to show that the Fitch algorithm will always produce the same answer as Sankoff's algorithm, which we can assume is a correct algorithm. Try to be as formal as you can in your proof, but if you don't have experience writing mathematical/algorithmic proofs, just explain your reasoning as clearly as possible.

Question 3. (40 points)
Phylogenetic trees can be built from non-genetic data, and in fact this was the only type of information available prior to the 1970's, when DNA sequence became possible. Here, you will design and implement an algorithm similar to Sankoff's algorithm, but that will work on quantitative traits (things about a species that can be measured with integers) rather than genetic data. Suppose that you have information about *k* traits for each species. These traits are measured as non-negative integers. For example, for mammals, trait1 might be the length of the forearm (in cm), trait2 might the volume of the skull (in cm^3), trait3 might be the lifespan (in years), etc. As species evolve, their traits change but those changes are generally slow (although there are exceptions). Thus, a parsimony-based phylogenetic inference approach makes sense. The problem we want to solve is the following:

Input:

- A set of *n* species, with, for each species *i*, a vector of *k* integers $D_i = (D_{i,1}, D_{i,2}, ..., D_{i,k})$ representing the measurements made for the *k* traits

- A phylogenetic tree *T* with leaves labeled with the *n* species.

Goal:
Assign a vector D_u to each internal node $u = n+1, \ldots, 2n-1$ such that $\sum_{(u,v)\in E(T)} |D_u - D_v|_1$ is minimized, where $|D_u - D_v|_1 = \sum_{i=1}^{k} |D_u(i) - D_v(i)|$.

a) (25 points) Write the pseudocode of algorithm to solve this problem. You can assume that no trait value will ever exceed a given maximum value *M*.
b) (5 points) What is the running time of your algorithm (using big-O notation)?
c) (10 points) The problem formulation above is not ideal in cases where the range of values of different traits differs significantly (e.g. the lifespan ranges between 0 and 50 years, whereas the volume of the brain ranges from 0 to 1500 cm^3). **Explain why and propose a modification to the problem formulation that would make it more biologically relevant. You do not need to propose a revised algorithm to go with your revised problem formulation.**

Good luck!