BTRY 4840 / 6840 / CS 4775
Computational Genetics and Genomics

Problem Set 4

Problem 1: Phylogeny Reconstruction

In this problem, you will estimate a phylogeny for the *APOE* gene by maximum likelihood, using a computer program of your own creation. You will then reconstruct the version of this gene that belonged to the most recent common ancestor of humans and mice (which lived roughly 80 MYA) and you will compare this ancestral gene with the modern human version.

APOE stands for "apolipoprotein E precursor." This gene produces a lipid-binding protein that helps to break down certain lipoprotein constituents. (Lipoproteins are biochemical assemblies of proteins and lipids which, among other things, act as enzymes and carry fats around the body.) Mutations in *APOE* are associated with abnormalities of blood lipids, with cardiovascular disease, and most famously, with late onset Alzheimer disease.

You will base your analysis on a multiple alignment of the coding regions of the human, mouse, rat, and dog copies of *APOE*, which is available for download from the course website. This alignment consists of 927 columns of coding DNA. It begins with a start codon, ends with a stop codon, and contains no alignment gaps. (For simplicity, 60 columns that had gaps in one or more species have been removed.)

(a) (45 points total) Find the maximum likelihood tree for *APOE* based on the Jukes-Cantor model, using your own implementation of Felsenstein's algorithm. Your program should consider the three alternative tree topologies in Fig. 1. These are the only three possible unrooted trees for the four species in question, so you will be performing an exhaustive search over tree topologies. Ordinarily, you would have to estimate the branch lengths for each topology, but we have done this for you, to simplify your task. The maximum likelihood branch lengths are given in Table 1.

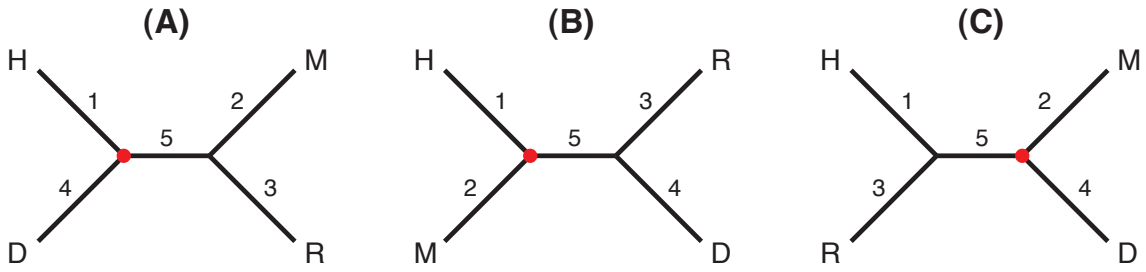

Figure 1: The three possible unrooted tree topologies for human (H), mouse (M), rat (R), and dog (D). The ancestral node of interest, to be selected as the root of the tree, is indicated in red.

As presented in class, Felsenstein's algorithm requires a rooted binary tree, even though, with the Jukes-Cantor model, the likelihood will be the same for all rootings. Root each

unrooted tree from Fig. 1 at the node indicated in red. Convert the tree to a binary tree by introducing a zero-length branch beneath the root, as shown in Fig. 2.

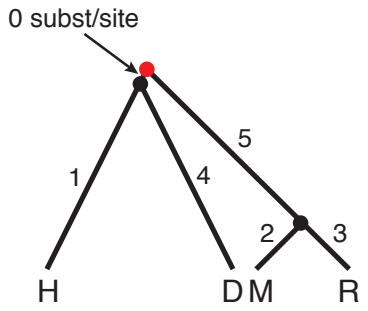

Figure 2: The rooted binary tree corresponding to topology (A).

Your program should read in the sequence data, then compute the log likelihood of each tree topology by iterating over alignment columns and applying Felsenstein's algorithm at each column. For the topologies, note that you will need to use a data structure that allows each node to be associated with its two children, and each branch to be associated with a non-negative real number from Table 1. You will also need to visit the nodes of the tree in a post-order traversal. This can be done with a stack, with recursive procedure calls (generally less efficient), or in this case, to save programming time, by simply hard-coding an ordering over nodes for each tree topology. The conditional probabilities $P(b|a,t)$ can be computed using the closed-form expressions given in lecture for the Jukes-Cantor model. For efficiency, you should compute a 4×4 matrix of conditional probabilities for each branch as a preprocessing step, then simply look probabilities up when you run Felsenstein's algorithm on each alignment column. You do not need to work with logs within Felsenstein's algorithm (this gets a bit messy and underflow at each column should not be a problem) but you should use logs when combining the probabilities across columns. Recall that the equilibrium distribution is assumed to be uniform under the Jukes-Cantor model ($\pi_A = \pi_C = \pi_G = \pi_T = \frac{1}{4}$).

i (35 points) What is the log likelihood of each tree topology? Which is the maximum likelihood tree? Submit the code you wrote to compute the likelihoods.

ii (5 points) Suppose in the true phylogeny branch 5 (the internal branch) were very short compared with branches 1, 2, 3, and 4. What effect would you expect this to have on

Table 1: Maximum likelihood estimates of branch lengths (subst/site)

	branch				
topology	1	2	3	4	5
(A)	0.07517	0.03059	0.03161	0.11761	0.14289
(B)	0.20843	0.03397	0.03497	0.24952	0.00000
(C)	0.20843	0.03397	0.03497	0.24952	0.00000

the differences in likelihood between the candidate trees, and on your ability to identify the true tree?

iii (5 points) How might you quantitatively measure your confidence in the reported phylogeny? Assuming the species set is fixed (i.e., you cannot add data for new species), can you suggest (and explain) possible ways to improve your confidence in your result?

(b) (20 points) Adapt your program to compute the posterior distribution over bases at the root of the tree. Then, using the maximum likelihood tree topology, reconstruct the ancestral sequence at this root node by finding the maximum a posteriori base at each position (i.e., the base with maximum posterior probability). Your program should output this reconstructed sequence along with the human, mouse, rat, and dog sequences, in such a way that the alignment can easily be examined by eye (i.e., in a constant width font and in blocks that do not include line-wraps, as for problem set 2). Note that, despite the rooting assumed for the analysis, the real root of the tree is on the branch leading to dog—i.e., dog is an outgroup to the other three species. Therefore, the sequence you have reconstructed belonged to the most recent common ancestor of human, mouse, and rat.

(c) (10 points) Compare this reconstructed ancestral sequence with the human sequence. Perform this comparison at the protein level, by writing a short program to convert each DNA sequence to the corresponding sequence of amino acids, as determined by the "universal" genetic code. Display the amino acid sequences for the human and ancestral sequence in an easy-to-read alignment, as above, and highlight the differences between them. These represent changes that have occurred during the past 80 million years, since humans and mice diverged. Some of them may have important phenotypic consequences. These changes and others like them are what make humans different from mice!

Problem 2: Phylogenetic inference: Neighbor joining

(a) (20 points) Implement the neighbor joining algorithm (as described in chapter 7 of Durbin) for an arbitrary number of species in either Python, R, C, or C++. It should take as input a matrix representing the distance between all pairs of species (the matrix is symmetric, with zero on the diagonal). The output should be a tree in Newick format (described on Joe Felsenstein's webpage at http://evolution.genetics.washington.edu/phylip/newicktree.html).

(b) (5 points) The distances between 10 primate species are available in a tab-delimited file called dist10.txt on the course webpage. Use your neighbor joining algorithm to estimate the tree for this species set. Report the tree, with estimated branch lengths, in Newick format. Plot the tree and turn in the figure as well. You may use external software to create the plot; suggestions for doing this include the plot.phylo() function in the "ape" package for R, or the UCSC phyloPng tool available at https://genome.ucsc.edu/cgi-bin/phyloPng. (For the phyloPng tool, select the "Use branch lengths?" option and note that you can change the width and/or height of the tree to make the output more legible. You may wish to use more than the default output of 2 decimal places.)