

CS 124 Spring 2005, Final Exam. Open books and notes, but only those books and notes used in this class. The first number in the two numbers in the parenthesis is the number of points for the problem. The second number is the maximum time I estimate is needed for the question if you are prepared. Write clearly and to the point. Use the paper provided, number each page, and put your name on each page. Use both sides of the paper, but DON'T put any answers for problems 1 through 5 on the same sheet as answers for problems 6 through 9.

1. (10,10) Briefly explain the critical difference between the original BLAST and BLAST 2. What are the advantages and disadvantages of using BLAST 2 compared to the original BLAST?

2. (5,5) In the context of database search, what is jumbling and why is it used?

3. (15,15) Given the following labeled tree T, find a multiple alignment of the sequences that is consistent with T. Then display the PSSM for that alignment. Remember that the pairwise alignment is based on minimizing the distance between the two sequences. We use a cost of 1 for each mismatch and each space opposite a non-space, and a cost of 0 for each match and each space opposite a space.

4. (5,5) Briefly, why do we need to multiply align sequences? That is, why isn't two sequence alignment always sufficient to compare biological

sequences?

5. (10,5) Clustal basically computes a multiple alignment that is consistent with a guide-tree computed from the sequences. Where does this guide-tree come from? That is, what is the idea behind the guide-tree?

REMEMBER NOT TO PUT ANSWERS TO PROBLEMS 6 THROUGH 9 ON THE SAME SHEET AS ANSWERS TO PROBLEMS 1 THROUGH 5.

6. (10,10) Let Π_x denote a Markov chain (path) in an HMM H that generates a particular sequence x , and let \mathcal{P}_x denote the set of all chains in H that generate x . Let $p(\Pi_x)$ denote the probability of the specific chain Π_x in H . Given H , how is the number $p(\Pi_x)$ obtained?

For a given input string x , the Viterbi algorithm first computes the Maximum $p(\Pi_x)$ over all Markov chains in \mathcal{P}_x . A chain in \mathcal{P}_x with that maximum probability is called a “most-likely” chain for x .

After computing $p(\Pi_x)$, how does the Viterbi algorithm find an actual most-likely chain for x ?

7. Suppose that instead of computing the Maximum $p(\Pi_x)$ over all Markov chains in \mathcal{P}_x , we want to compute $p(x)$, which is defined as the $\sum p(\Pi_x)$ over all Markov chains in \mathcal{P}_x .

7a. (10,5) Intuitively, what would $p(x)$ tell us about H and x ? That is, why would we want to know that sum? In contrast, give an example where we specifically want one most-likely chain for x rather than $S(x)$.

We want to compute $p(x)$ by a DP approach. We first define our DP notation: Given an HMM H and a sequence x , we let $f_l(i)$ denote the sum of the probabilities of all the paths that end at state l (state “ell”) in H having generated the first i characters of x .

7b. (15,15) Below are the recurrences for a DP algorithm to compute $p(x)$. Explain the logic behind those recurrences - i.e., how do they work and roughly why are they correct. Which DP value gives us $p(x)$? Roughly, how many operations does it take to compute $p(x)$ if H has m states (nodes) and x has n characters?

The basis of the DP is: $f_0(0) = 1$ and $f_k(0) = 0$ for every $k > 0$. State 0 is the start state of H .

The general DP recurrence is: $f_l(i + 1) = e_l(x_{i+1}) \sum_k f_k(i) a_{kl}$, where $e_l(x_{i+1})$ is the probability that character x_{i+1} gets generated when the chain

enters state l , and a_{kl} is the probability of a transition from state k to state l .

8. (10, 10) When we discussed (in class and notes) deriving and using alignment scores from a multiple alignment, we used the following approach.

We start with a given a multiple alignment of a set of k “related” sequences. For concreteness, lets say that the sequences are all globins. Then in every column i of the multiple alignment, and for every character X , we count how many times character X appears in column i ; Call that $c(i,X)$. Then we define $f(i, X) \equiv c(i, X)/k$, i.e. the frequency of character X in column i . Now let $f(X)$ be the frequency of character X in the k sequences, and define the score of character X in column i as:

$$S(i, X) \equiv \log_2 \left[\frac{f(i, X)}{f(X)} \right]$$

When we have a new sequence S and want to know if it might be a globin, we use these scores to evaluate S , i.e. if in S we have a character X in column i , we use $S(i, X)$ for that position-character pair. The score for the entire sequence is the sum over all the scores for each position. A high score is more suggestive that S is a globin, and if the score is above some threshold, we declare S to be a “likely globin”.

In explaining this approach, we said that the score for the sequence is really a log-odds ratio comparing the probability that S is generated by a Markov Model of globins, to the probability that S is generated by a Markov model of random strings. Declaring S to be a “likely globin” is equivalent to rejecting the null-hypothesis that S is generated by Markov model of random strings, in favor of the hypothesis that S is generated by a Markov model of globins. That lead to the question of whether it was best to take $f(X)$ as the frequency of character X in the mulitple alignment (as stated above), or whether it would be better instead to take $f(X)$ to be equal to the frequency of X in all known sequences in nature (this is called the “background frequency”). There is no correct answer for this, but it partly depends on what is already known about the new sequence S , and partly on ones tolerance for false positive and false negative results (declarations).

Suppose S really is a globin. In which of these two choices for $f(X)$ is the null hypothesis more likely to be rejected? Explain. Which choice of $f(X)$ is best if you want to reduce the chance of a false-positive result? Explain.

9. (10,5) What is the molecular clock thesis? Suppose we collect rough “time since divergence” data for pairs of sequences, and we notice that those data are roughly ultrametric. Does that support or contradict the molecular clock thesis? Briefly explain.