

CMSC 423 Homework #4:

Due: Dec. 9th at the start of class

You may discuss these problems with other students in this class, but you **must write up your solutions independently**, without using common notes or worksheets. You must indicate at the top of your homework who you worked with. Your write up should be clear, concise, and neat. You are trying to convince a skeptical reader that your algorithms or answers are correct. Messy or hard-to-read homeworks will not be graded.

1. Suppose Alice has two coins, one that is fair (probability heads = probability tails = 0.5) and one that is biased (probability heads = 0.25, probabilities of tails = 0.75). She chooses one of the coins (fair and biased are chosen with equal probability), and starts flipping the coin. Subject to the constraints below, she has a 0.1 probability of switching which coin she uses and 0.9 of continuing to use the same coin.

(a) Suppose to avoid detection Alice always makes **at least 5** flips with a coin before switching. Draw an HMM that can model this situation.

(b) Suppose instead of requiring ≥ 5 flips before switching, we require Alice flips a given coin **no more than k** times before switching, where k is some parameter $\leq n$.

Give an algorithm that, given k and a sequence x of n results of coin flips (H or T), finds the most probable sequence of fair/biased coin usages under this restriction. In other words, modify the Viterbi algorithm to find the best path that never stays in any state for longer than k consecutive steps. The output should be a list of n instances of “Fair” and “Biased”. Your algorithm should use $O(n)$ space and $O(nk)$ time.

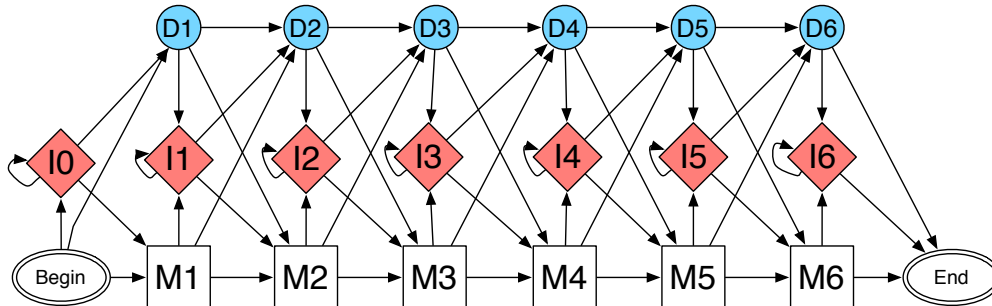
2. How many open reading frames are present in the following DNA sequence? Assume the start codon is ATG and the stop codon is TAA. Draw lines under where all these ORFs (open reading frames) occur.

ATGCATCATGGATGTTAATGTAACCGTCTAACTAA

3. Suppose we modify the affine-gap pairwise sequence alignment problem to use probabilities in the following ways: instead of a `gap_start` and `gap_extend` scores, you are given probabilities p_{start} and p_{extend} for the probability of starting and extending a gap, respectively; and instead of a match score $m(a, b)$ between characters a and b , you are given a probability $p_{align}(a, b)$ that a character a would align to a character b and you are given $p_{gap}(a)$ as the probability that character a would be aligned with a gap. Generating random alignments according to this model is a way to generate test cases for algorithms and can help explore how different parameter choices affect the alignments.

Give an HMM that will output a random pairwise alignment according to the above parameters. (Hint: your states will output pairs of characters.)

4. **Multiple sequence alignment with “motif-search” HMMs.** Recall the motif search HMM described in class:



where the I, D, M states are *insertion*, *deletion*, and *match* states, respectively. In lecture, we used this type of HMM to search for short sequences. Such an HMM can also be used for a multiple sequence alignment algorithm alternative to the Star algorithm you implemented in the project.

Suppose you have two functions defined for you: `Viterbi(M, x)` that returns the most probable path through an HMM `M` for an observed sequence `x`, and `Train(x1, x2, ..., xk)` that returns an HMM of the form above with all the parameters set so that the sequences `x1, ..., xk` are likely to be output by the HMM.

(a) Design a MSA algorithm that uses these two functions and takes as input m sequences s_1, \dots, s_m and outputs a multiple sequence alignment of them. (b) Describe why your algorithm will generate a plausible multiple sequence alignment, and justify your algorithm design decisions. (c) Give an estimate of the running time and space usage of your algorithm.