CSC 2417 Algorithms in Molecular Biology
PS3: Due December 8

Don't Panic

This is an individual assignment. While you may discuss this assignment with classmates, please do not give away answers. You are NOT allowed to use the internet, besides to look up things not directly related to the assignment, such as a generic formula or a well-known algorithm. This homework may have bugs. If you spot something that looks wrong or is not clear please contact me.

1. Motifs & Rearrangements

a) The Gibbs Sampling algorithm we described in class (and many other related approaches) assume independence of adjacent positions. This is not always a valid assumption; for example the bias against CpG di-nucleotides makes adjacent nucleotides non-independent. Develop an HMM representation of a profile that incorporates non-independence of adjacent nucleotides. What is the number of parameters in such a model? What is the complexity (number of parameters) for a full model that incorporates dependencies between all positions in a motif of length k?

b) In the Gibbs Sampling examples discussed in class we were searching for ungapped motifs. It is actually not very well known how many – if any gaps transcription binding sites can tolerate. You are asked to come up with a Gibbs Sampling algorithm that will search for gapped motifs. Instead of returning a position weight matrix (PWM) of length K, your algorithm should return an alignment of length K. For simplicity we will treat the gap as a 5th DNA character. Describe how one may search for these gapped motifs in the one sequence we are currently leaving out.

2. Gene finding

(a) The sequence below is the RNA of a fake eukaryotic gene with exactly one intron. Where is the intron and what is the coding sequence?

ATGCAGTCTAGGTAA

A simple strategy for locating genes in compact genomes not containing introns is to look for long open reading frames (ORFs). An ORF is defined as a sequence of DNA beginning with a start codon (ATG) and containing no in-frame stop codons (TAA, TAG, or TGA). ORF scanning works because genes contain long open reading frames which are unlikely to occur by chance. For these sections you can use any programming language. You should submit a printout of your code with the solution or by e-mail.

(b) Write a program that given a genomic sequence searches for ORFs of length > K amino acids, where K is a parameter. Your program should take a genome in FASTA format (an example will be posted on the website) and find all these ORFs in all 6 frames.

Report the length of the longest ORF, as well as the mean size of all ORFs ≥ 600 nucleotides.

(c) Using the ORFS ≥ 600 bp long, determine the frequency of the various codons (DNA triplets) in ORFs. Map the codons to the amino acids (so it is a 21 element frequency table) and include with your writeup. Determine and submit the same table for DNA sequences outside of these ORFs (putatively non- coding DNA).

(d) Compare this table with the table posted in the lecture slides. Do you observe the same CpG bias as in the table from lecture? While an answer as to why is not required, you may want to do some digging to find out on your own, if you are curious.