Problem 1: Database Search
The purpose of this problem is to make you experiment “a little bit” with BLAST.

(a) Go to http://www.ncbi.nlm.nih.gov/ and click on the human genome under Genomic Biology in the left margin.

(b) Click on map viewer in the middle of the page. This will display a the set of human chromosomes. Choose your favorite chromosome.

(c) Click on a gene under symbol (you can scroll up and down on the chromosome and you can zoom in).

(d) Scroll down to NCBI Reference Sequences (RefSeq) Database and look at mRNA sequence. This is the accession number that uniquely identifies that sequence. Click on it. This displays information about that gene like name, references, authors, etc...

(e) Scroll down until you see translation. This is the amino acid sequence that is produced by the mRNA. Scroll further down until you see the sequence itself. Cut and paste the sequence in any text editor and remove the spaces and the numbers.

(f) Pick a substring of that sequence of 20-30 nucleotides. Go the http://www.ncbi.nlm.nih.gov/BLAST. Click on nucleotide-nucleotide BLAST (blastn). In the search box enter your 20-30 nucleotide sequence then click BLAST! and then Format!.

(g) BLAST retrieves a number of sequences that contain segments that score high with your query. Verify that your original sequence is one of them (by the accession number). For several sequences, click on the score and see how the query sequence aligns with the sequence in the database. Some sequences might be in the mouse or rat or other genomes; for instance, you might see Mus Musculus (for mouse) and Homo Spapiens (for human), etc...

(h) Pick two sequences among the sequences you found, obtain their nucleotide sequences as before and then go to http://www.ncbi.nlm.nih.gov/BLAST and click on Align two sequences (bl2seq). Input your two sequences in the corresponding boxes and click Align. See the similarity between the two.
Problem 2: HMM Learning

(a) What is the computational complexity of a single iteration for the HMM estimation of paramaters $a_{kl}$ and $e_k(b)$ during

- Baum-Welch training
- Viterbi training

in terms of $k$: the number of states, $n$: the number of training sequences, and $l$: the length of each training sequence.

(b) Suppose we have a large number of training sequences emitted by an HMM that has a particular transitional probability $a_{kl} = 0$, for some $k$ and $l$. Say that we now use these sequence to train (using Baum-Welch) a new HMM with the same architecture, one that happens to start with $a_{kl} = 0$. Show that the parameter $a_{kl}$ remains 0 after the training.

Initial parameters play an important role on the result of training the HMM. Both Baum-Welch and Viterbi training are local optimization algorithms and their end result depends on the starting point.

(c) Construct an HMM architecture of your choice representing some random process (die, coin, nucleotides, etc...), with some parameters for all $a_{kl}$ and all $e_k(b)$.

(d) Generate a set of sequences from the above model. The number $n$ and the length of the sequences is up to you. Make sure that the counts of each transition $A_{kl}$ and emission $E_k(b)$ in all the sequences is consistent with your model, i.e. $a_{kl} = \frac{A_{kl}}{\sum_{ij} A_{ij}}$ and $e_k(b) = \frac{E_k(b)}{\sum_{ix} E_i(x)}$.

In what follows you can use either Baum-Welch or Viterbi

(e) Choose a set of initial non-zero parameters for a new HMM with the same architecture such that your chosen learning algorithm converges to the true model (that of part (c)).

(f) Choose another set of initial non-zero parameters such that your chosen learning algorithm converges to something different that the true model (local maximum, not global one).

(g) Calculate $p(x_1...x_n|\theta)$ of your training sequences $x_1, ..., x_n$, given the resulting models: $\theta_1$: the true model, and $\theta_2$: the wrong local maximum model.
Problem 3: Pairwise alignment with HMMs

(a) The pair HMM for alignments that we described in class does not allow a gap in $x$ followed immediately by a gap in $y$. Modify the state diagram to allow for this. How should you choose the second gap open penalty (the one in $y$) in terms of $\epsilon$ and $\delta$ for this model to make sense? Write down the modified Viterbi algorithm using the log-odds equations.

(b) Construct a pair HMM for semi-global alignments. Show the state diagram with the transitional probabilities.

(c) Construct a pair HMM for k-bounded global alignment (like in bounded Needleman-Wunsch). How many states does your HMM have? Show the state diagram with the trasitional probabilities.

Problem 4: Gene Prediction

Genes are considerably more conserved than the rest of the DNA. This problem illustrates a simplified version of gene prediction using comparative genomics.

You are given the full genomes of two animals $X$ and $Y$.

Now... you wil be facing too many parameters...

The genes in these animals are conserved 80% on average. Insertions and deletions in gene regions are very rare (i.e. gaps), occurring roughly every 50 nucleotides. Moreover, because insertions and deletions should not interrupt the codon structure of a gene (recall that 3 nucleotides translate to one amino acid) they almost ALWAYS happen in multiples of 3. The mean gap length is 4 but most gaps are 3 long in gene regions). Genes are $3k$ long where $k$ has a mean of 200.

The non-gene regions are conserved on average at 40%, gaps open on average every 12 nucleotides and have a mean length of 2. In addition, a different kind of gaps, long gaps, open on average every 1000 nucleotides, and have a mean length of 100.

You wish to be able to compare the genome of $X$ and $Y$, and find the optimal way to divide it into conserved gene regions, and less conserved non-genen regions. Therefore, you have to construct an HMM for this problem.

The HMM will consist of two sets of states for gene and non-gene regions with small transitional probabilities between them (like we did with CG islands and non-CG islands). Each set of states will represent a pair HMM for alignment.

Build an HMM for this problem.