Biological Sequence Analysis (Spring 2015)

Exercise 5 (Thu 12.2, 10-12, B222, Veli Mäkinen)

Do any 5 assignments from below.

1. **Gene prediction in eukaryotes.**

   The flexibility of choosing the states, transitions, emissions, and their probabilities, make HMMs a powerful modeling device. So far we have used *zero-th order Markov model* for emission probabilities (probabilities only depended on the state, not on the sequence context). We could use as well *first-order Markov chains* or, more generally, *k-th order Markov chains*, in which the probability depends on the state and on the last k symbols preceding the current one: $\mathbb{P}(s_i \mid s_{i-k} \cdots s_{i-1}) = \mathbb{P}(s_i \mid s_1 \cdots s_{i-1})$.

   Notice that the states of the HMM are independent, in the sense that each state can choose a different order Markov chain it uses for its emission probabilities. In addition to the use of different order Markov chains, we could adjust how many symbols are emitted in each state. Use these considerations to design a realistic HMM for *eukaryote gene prediction.* Try to take into account intron/exon boundary di-nucleotides, codon adaptation, and other features known about eukaryote genes. Consider also how you can train the HMM.

2. **Profile HMMs I.** *Profile HMMs* are an extension of HMMs to the problem of aligning a sequence with an existing multiple alignment (profile). Consider for example a multiple alignment of a protein family:

   ```
   AVLSLSKTTNNVSPA
   AV-SLSK-TANVSPA
   A-LSLSK-TANV-PA
   A-LSSSK-TNNV-PA
   AS-SSSK-TNNV-PA
   AVLSLSKTTANV-PA
   ```

   We considered the problem of aligning a sequence A against a profile in the context of progressive multiple alignment, and the idea was to consider the multiple alignment as a sequence of columns and apply normal pair-wise alignment with proper extensions of substitution and indel scores. Consider A = `AVTLSLSTAANVSPA` aligned to the our example profile above, for example, as follows:

   ```
   AVTLSLS--TAANVSPA

   AV-LSLSKTTN-NVSPA
   AV--SLSK-TA-NVSPA
   A--LSLSK-TA-NV-PA
   A--LSSSK-TN-NV-PA
   AS--SSSK-TN-NV-PA
   AV-LSLSKTTA-NV-PA
   ```

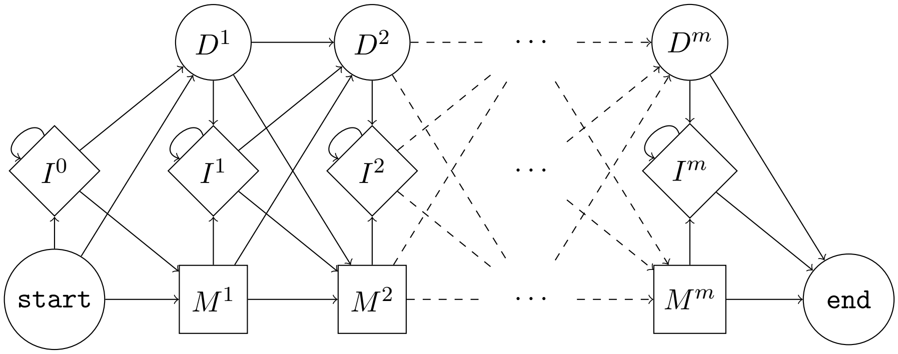

Figure 1: Profile HMM illustration without showing the transition and emission probabilities.

Here we have added two gaps to the sequence and two gap columns to the profile following the ‘once a gap, always a gap’ principle.

Profile HMMs are created using *inhomogeneous* Markov chains, such that each columns form separate match, insertion, and deletions states, and transitions go from left to right, as illustrated in Figure 1. Match and deletion states emit the columns of the profile, so they do not contain self-loops. Insertion states emit symbols from the input sequence, so they contain self-loops to allow any number of symbols emitted between states that emit also columns of the profile.

Since the resulting HMM is reading only one sequence, the Viterbi, forward, and backward algorithms are almost identical to the ones we studied so far. The only difference is that deletion states are *silent* with respect to the input string, as they do not emit any symbol.

a) Modify the Viterbi recurrences to handle both emitting and silent states.

b) Derive the Viterbi recurrences specific to profile HMMs.

3. **Profile HMMs II.**

   Derive a local alignment version of a profile HMM.

4. **Pair HMMs (2 points)**

   Pair HMMs are a variant of HMMs emitting two sequences, such that a path through the HMM can be interpreted as an alignment of the input sequences. Such pair HMMs have a *match* state emitting a symbol from both sequences simultaneously, and symmetric *insertion* and *deletion* states to emit only from one input sequence.

   a) Fix a definition for pair HMMs and derive the corresponding Viterbi, forward, and backward recurrences. *Hint.* The result should look very similar to Gotoh’s algorithm for global alignment with affine gap costs.

   b) Derive the probability of a_i aligning to b_j over all alignments of $A = a_1 \cdots a_m$ and $B = b_1 \cdots b_n$.

c) Let p_{ij} denote the probability derived above to align a_i to b_j. We say that the *most robust alignment* of A and B is the alignment maximizing the sum of values p_{ij} over i, j such that the $a_i \to b_j$ substitution is part of the alignment. Derive a dynamic programming algorithm to compute this most robust alignment.

5. **NP-hardness.**

   Give an alternative proof for the NP-hardness of the longest common subsequence problem on multiple sequences, by using a reduction from the *vertex cover problem. Hint.* From a graph $G = (V = \{1, \ldots, |V|\}, E)$ and integer k, construct $|E| + 1$ sequences having LCS of length $|V| - k$ if and only if there is vertex cover of size k in G. Recall that vertex cover V' is a subset of V such that all edges in E are incident to at least one vertex in V.

6. **DAG-path alignment I.**

   Reformulate the DAG-path alignment problem as a local alignment problem. Can the algorithm be modified to solve this variant?

7. **DAG-path alignment II.**

   Show how to use DAG-path alignment to align a protein sequence to DNA. *Hint.* Represent the protein sequence as a *codon-DAG* replacing each amino acid by a sub-DAG representing its codons.

8. **DAG-path alignment III.**

   Consider a DAG with predicted exons as vertices and arcs formed by pairs of exons predicted to appear inside a transcript: this is called a *splicing graph.* If you apply DAG-path alignment on splicing graph and the codon-DAG of previous assignment, what problem are you trying to solve?