# Biological Sequence Analysis (Spring 2015)

## Exercise 4 (Thu 5.2, 10-12, B222, Veli Mäkinen)

Do any 5 assignments from below.

1. **Log-transform I.**

   Multiplication is the source of numerical errors in HMM algorithms, when the numbers become too large to fit into a computer word. Show how the Viterbi algorithm can be implemented using sum of logarithms to avoid these numerical problems.

2. **Log-transform II.**

   For the forward and backward algorithms the sum of logarithms conversion is not enough for numerical stability. Browse the literature to find a solution for this problem.

3. **Training coding/non-coding HMM.**

   Assume a set of DNA sequences with coding/non-coding labeling. *Implement* a program that trains the emission/transition probabilities for the coding/non-coding HMM considered at lectures, given the training data.

4. **Implementing viterbi from scratch.**

   *Implement* the viterbi algorithm in the special case of the coding/non-coding HMM. Implement also the tracing of optimal path and deduce the labels for some example sequence. If you did the previous assignment, train the HMM with some other example sequence.

5. **HMMs with biopython.**

   Familiarize yourself with the HMM package of biopython: `http://biopython.org/DIST/docs/api/Bio.HMM-module.html`. Explain how the provided functions match the lecture script material.

6. **Implementing viterbi using biopython.**

   Use biopython to run viterbi on the coding/non-coding HMM.

7. **Space improvement I.**

   Argue that $O(|H|)$ space suffices to compute the final numerical value (not the actual solution path) with the Viterbi algorithm.

8. **Space improvement II.**

   To traceback the solution of Viterbi algorithm, the naive solution requires $O(n|H|)$ space. Show that this can be improved to $O(\sqrt{n}|H|)$ by sampling every $\sqrt{n}$-th position in $S$, without asymptotically affecting the running time of the traceback.

9. **Baum-Welch.**

   Derive the Baum-Welch formula for the expected number of times that transition $(h', h)$ is used given the training data, $\texttt{TC}(h', h)$. The derivation is analogous to the one given at the lecture script for $\texttt{EC}(h, c)$.

10. **Dynamic programming variations.**

    Given an HMM, derive an $O(n|H|)$ time algorithm to compute

$$\arg \max_{P = p_0 p_1 \cdots p_n p_{n+1} \in \mathcal{P}(n) \,:\, p_i = h} \mathbb{P}(P, s_1 \cdots s_n), \tag{1}$$

    for all $h \in H$.