PROJECT 3 - MACHINE LEARNING IN BIOINFORMATICS Q3/2014

This project is about using a hidden Markov model for gene finding in prokaryotes as explained in the lecture on Feb 25.

We suggest that you do the project in groups of 2-3 students. The project is mandatory and each group must give a 10 minutes presentation about their work and results on **Tuesday, Mar 11**, as outlined in section Presentation below.

Data set

You are give a data set containing 11 Staphylococcus genomes, each containing several genes (i.e. substring) obeying the "gene syntax" explained in the lecture on Feb 25. You also considered this data set in project 1. The genomes are between 1.8 million and 2.8 million nucleotides. For 5 of the genomes, you are also given the location of the genes. For the remaining 6 genomes, you only know that they contain genes according to the "gene syntax". The genomes and their annontations are given inFASTA format. There are several libraries for reading sequences in this format (use Google to find one for your language of choice), but it is also easy to read it directly. The data is:

- The files genome1.fa, genome2.fa, genome3.fa, genome4.fa, genome5.fa, genome6.fa, genome7.fa, genome8.fa,genome9.fa, genome10.fa, and genome11.fa contain the 11 genomes.
- The files annotation1.fa, annotation2.fa, annotation3.fa, annotation4.fa, and annotation5.fa, contain the annontation of the first 5 genomes. The annotation is given in FASTA format as a sequence over the symbols N, C, and R. The symbol N, C, or R at position *i* in annotation*k*.fa gives the "state" of the nucleotide at position *i* in genome*k*.fa. N means that the nucleotide is non-coding. C means that the nucleotide is coding and part of a gene in the direction from left to right. R means that the nucleotide is coding and part of gene in the reverse direction from right to left.

By simple reading of the annotations and the corresponding genomes, it is clear that fx several start-codons are possible (and not only atg as modeled in the lecture on Feb 25). The genome files might also contain symbols other that A, C, G, and T, due to ambiguity in the sequencing process, see this Wikipedia entry. In this project, you can safely substitute such symbols with an A, C, G, or T.

Problem

Your task is to predict the gene structure of the 6 unannotated genomes, i.e. genome6.fa, genome7.fa, genome8.fa, genome9.fa,genome10.fa, and genome11.fa.

As explained in the lectures, predicting gene structure using HMM involves:

- Deciding on an initial model structure, i.e. the number of hidden states and which transitions and emission should have a fixed probability (e.g. 0 for "not possible", or 1 for "always the case"). You might get inspiration from, or even use, one of the model structures discussed in the lecture on Feb 25. However, be aware that the models presented in the lecture assume start-codon atg. This is not the case in the presented data, where multiple start-codons are possible.

  To predict the full gene structure, you must decide how to handle to fact that the genomes have genes in both directions (i.e. a nucleotide can be C or R). You can e.g. choose to make a HMM which only models genes in one direction and then use it twice to predict the genes in each direction (beware that start- and stop-codons look different in the reversed direction, unless you make the reverse-complement of the input string), or you can make a model which models genes in both directions.

- Tune model parameters by training, i.e. set the non-fixed emission and transition probabilities. This can be done by several methods as explained in the lecture on February 12: (1) "By counting" using the 5 genomes with know gene structure because you know the underlying hidden state for each observation (i.e. symbol) in these sequences, or (2) by Viterbi- or EM-training using all the 11 genomes, including the 6 genomes with unknown gene structure. You can of course also combine the two approaches. You can evaluate the performance of your gene finder by computing the approximate correlation between your predictions and the true structure using this small python programcompare_anns.py (see this paper for details).

  In this project, training "by counting" is mandatory, while Viterbi- or EM-training is voluntarily. If you implement EM-training you should of course be aware of the precision problems of the forward- and backward-algorithms.

- Use your best model to predict the gene structure for the 6 unannotated genomes using the Viterbi algorithm with subsequent backtracking. I.e. for each unannotated genome you must find the most likely sequence of states in your model generating it, and convert this sequence of states into a FASTA file giving the annotation of each nucleotide as N, C, or R. You should make files, annotation6.fa, ..., annotation11.fa, giving the annotation for the 6 unannotated genomes.

  Note that the genomes are long, i.e. running Viterbi takes a lot of memory. If you run into problems, you might consider implementing one of the space-saving techniques discussed in the lecture on Feb 18.

Presentation

You must present your work and results in 10-minutes presentation in class on **Tuesday, Mar 11**. Please e-mail me, your presentation before class. Your presentation should contain:

- The status of the work, i.e. does it work, if not, then why.
- An explanation and illustration of your model structure (i.e. a transition diagram). This should include an explanation of how you model that there can be several start and stop codons, and an explanaition of how you have trained your model. For training by counting you should comment on how you have translated the given annotations of C, N, and R's into corresponding sequences of hidden states. The transition-diagram should contain all non-0 probabilities as computed by your training.
- An explanation of how you have predicted the gene structure for the unannotated sequences. If you use a different implemention of the Viterbi algorithm than the one you developed in project 2, you should explain why.
- An evaluation of the performance of your gene predictor. In this project you must make a 5-fold cross validation on the 5 genomes with known structure:

  For each genome 1 to 5, you train your model by training-by-counting on the remaining 4 genomes, and predict the gene structure on the genome you picked. You compute and report the approximate correlation coefficient (AC) between your prediction and the true annotation using the small python program compare_anns.py. Include a table with the computed ACCs in your presentation.
- A reference to the files annotation6.fa and annotation7.fa,..., annotation11.fa containing your predictions on the 6 genomes with unknown structure. The files *must* be in FASTA format similar to the annotation of the other genomes.