Task 9: Biological applications.

Data

This practical explores the application of Hidden Markov models to biological data. To start, download the bio dataset and look at a file. The format is different to the dice dataset you worked with before. The observed sequence starts with a hash (which you should otherwise ignore) and the corresponding hidden sequence follows in a row below. Each pair of sequences is separated by an empty line.

The observed sequence consists of a series of amino acids. As you may remember from biology classes, amino acids typically have three letter abbreviations, like Leu for leucine. In our dataset these more common names have been replaced with single letters. You can find the mapping in `AminoAcid` class if you are curious.

The sequences of amino acids correspond to **transmembrane proteins**: i.e., protein which are located so that they cross the membrane. Each of the amino acids is associated with a hidden `Feature` state, which indicates whether the amino acid is inside the cell, outside the cell, or in the membrane. From the application point of view, the membrane state is the most interesting one. It is actually rather difficult to distinguish between the inside and outside states, so low overall accuracy is to be expected.

This data comes from http://www.cbs.dtu.dk/~krogh/TMHMM/ specifically http://www.cbs.dtu.dk/~krogh/TMHMM/data/set160.labels

Task

In this task you should adapt your code from Tasks 7 and 8 to build a Hidden Markov Model of the amino acid sequences and to predict hidden states using the Viterbi algorithm. Instead of `DiceRoll` and `DiceType`, the relevant classes are now `AminoAcid` and `Feature`. The functions in the `IExercise9` interface have slightly different signatures because of the differences in the data format, however you should be able to adjust for that with minor refactoring of your code.

As in Task 8, you can test your code with the train-dev-test split provided. What results do you get if you vary the random seed? After varying the seed, you should understand why you should implement cross-validation. You are required to do this for the tick.

Starred tick (optional!)

If you think about this task, you will realize that our first-order HMM cannot completely capture the structure of the data. What is missing, in terms of state sequence information? What's more, if you analyse the dataset, you will see that the length of the 'within membrane' sequence is relatively constrained. Find the minimum and maximum length. Why does this make sense in biological terms?

How could we adapt the simple HMM approach to take account of this structure? This is quite challenging: probably the easiest approach (though not the most efficient) is to generate the n-best sequence of hidden states using Viterbi (instead of just the most probable sequence) and to filter accoring to the other constraints. If you do have a go at this exercise, please let us know what you did and what results you get. Because of the small size of the dataset, you will have to be careful about methodology!

If you look at the papers written by the creators of the dataset we're using (https://www.ncbi.nlm.nih.gov/pubmed/11152613 - full paper requires that you login in via Raven/Shibboleth) you will see that the way HMMs are used there is considerably more elaborate than the experiments we're carrying out.