

Assignment 4: Hidden Markov Models: Protein Secondary Structure Analysis

CS397-CXZ: Algorithms in Bioinformatics - Spring 2004

University of Illinois at Urbana-Champaign

Due at 12:30pm on April 7, 2004

1 Overview

In this assignment, you are asked to use the HMM program you implemented in the previous assignment to predict the secondary structure of proteins. A common way of using an HMM is to train an HMM and then find the most likely state transition path for a given sequence of observed symbols. Such an optimal path can be interpreted as providing a sequence of *tags* corresponding to each of the symbols. The tags can also be interpreted as specifying a way of *segmenting* the observed sequence with a segment corresponding to a consecutive sequence of identical tags (i.e. states). In this assignment, we will use a two-state HMM to segment protein sequences to identify regions with simple secondary structures.

2 Prediction of Protein Secondary Structure

The problem of predicting protein structures is one of the many possible applications of HMM in the exciting emerging field of bioinformatics.

A protein can be represented as a sequence of amino acid symbols, and this is called the *primary (linear) structure* of the protein. In the three-dimensional space, a protein would *fold* into certain shape, and the shape affects its function. Interestingly, the three-dimensional structure of a protein is also determined by its primary structure, in the sense that one type of proteins generally prefers certain shape. However, how a sequence of amino acid symbols exactly determines a particular 3-D shape is largely unknown. There are two types of “basic” shapes that occur frequently in the 3-D structure of proteins: the α -helix and the β -sheet. Thus, at a level above the primary structure, we can tag the amino acid sequence with three different tags: “**in- α -helix**”, “**in- β -sheet**”, and “**other**”. This is called the *secondary structure* of a protein. A possible 3-D structure and the corresponding primary and secondary structures are shown in Figure 2.

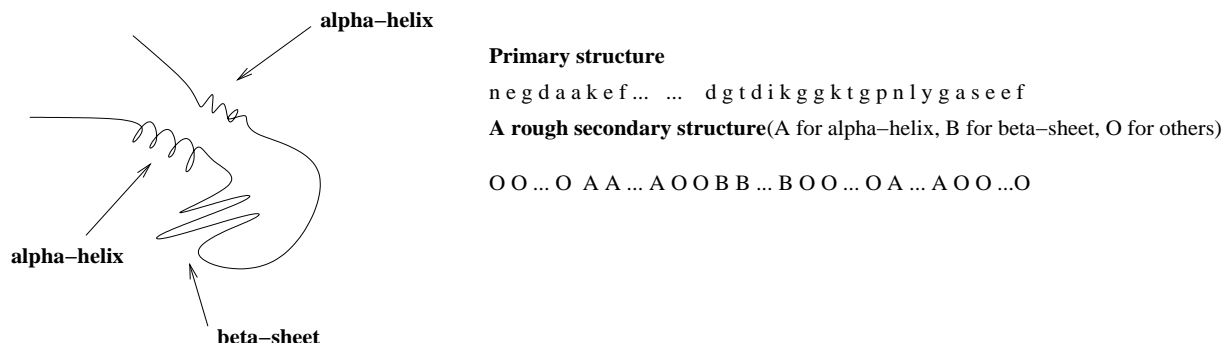


Figure 1: A make-up example of the 3-D structure and corresponding primary and secondary structures of a protein

Understanding the structure of a protein is essential to understanding how the function of a protein is encoded in the amino acid sequence. In this assignment, we will apply HMM to predict the secondary structure of a protein, but we will only consider some specific proteins that are known to have no β - *sheet* in their secondary structures¹, so the secondary structure only involves two tags: either **in- α** or **other**. More specifically, we will use the Golem dataset that Muggleton and Sternberg used in their research [1]. It is available at <http://www.doc.ic.ac.uk/~shm/proteins.html>.

You can visit the website to know more about this dataset. You can also easily find a lot of tutorial-like material on the Internet about bioinformatics in general.

The original data involves 12 training proteins and 4 testing proteins. Each protein sequence has a known segmentation that indicates where the α -helix is. We suspect that these proteins are known to have only *alpha*-helix, so the whole sequence can be seen as containing two types of alternating regions: either within an α -helix or outside one. To make it easy to finish this assignment, we concatenated all the training (testing) sequences into one long training (testing) sequence. They are stored in two files: `traintag` and `testtag`. Each file is a sequence of pairs. Each pair is on a separate line with the first character denoting the amino acid and the second the region tag (0 for within α -helix and 1 for outside). The file `testseq` has only the sequence of amino acids without the tag information. This is the file we will use for testing. We will run the Viterbi algorithm with an HMM to “decode” this sequence, i.e., to identify which part belongs to an α -helix and which does not. The predicted segmentation can be compared with the true tags in `testtag` to compute the prediction accuracy. For this assignment, we will use the simplest measure of prediction accuracy, which is defined as the percentage of tags that are correct. A perl script (`eval.pl`) is included in the code package to compute this accuracy for you. The usage is

```
% eval.pl testtag result
```

¹This is our understanding of the data set. No verification was made.

3 Tasks

0. Download the HMM code and the data from the assignment web page.

1. (25 points) Supervised training with the same data

Train an HMM using the *tagged* testing protein data and test the HMM on the raw testing sequence. That is, do the following

```
% hmm -c -n 2 -m mod1.test -s testtag
% hmm -d -m mod1.test -s testseq > result1
```

Evaluate result1 with eval.pl. What is the prediction accuracy?

2. (30 points) Supervised training with different data Now train an HMM using the *training* protein data and test the HMM on the raw *testing* sequence. That is, do the following

```
% hmm -c -n 2 -m mod1.train -s traintag
% hmm -d -m mod1.train -s testseq > result2
```

Evaluate result2 with eval.pl. What is the prediction accuracy? How is result2 compared with result1? Which one is better? Which one should we expect to be better? Why?

3. (45 points) Unsupervised training

Now, train the HMM on the protein testing data (i.e., assuming no labelled training data).

```
% hmm -t -n 2 -m mod2.test -s testseq
% hmm -d -m mod2.test -s testseq>result3
```

Note that since this is unsupervised training, the two states (0 and 1) may pick up their roles fairly arbitrarily, i.e., either state 0 or state 1 can mean within an α -helix. So, you need to evaluate the results in two different ways: assuming either 0 or 1 to be within an α -helix. For such evaluation, we provided a different perl script evalflip.pl in the code package. You should use evalflip.pl to evaluate result4. The usage is exactly the same as eval.pl.

Repeat this process at least 10 times. Do you get the same likelihood, the same model, and the same accuracy every time? Record the accuracy each time. What is the mean and standard deviation of all the recorded accuracy numbers? How is it compared with result2 and result1? Among the three results (i.e., result1, result2, and the average result3), which one should we expect to be the best? Which one should we expect to be the worst? Are the actual results consistent with what we expected?

4 What to hand in

Please turn in a hardcopy of your written answers at the class. Computer print-out is strongly recommended. Please also copy the three results files (i.e., result1, result2, and the best result3) to your handin directory on machines of CSIL, which is `/home/class/cs397cxz/handin/assign4/YOURID`.

References

- [1] Muggleton S., King R.D., and Sternberg M.J.E. (1992). Predicting protein secondary structure using inductive logic programming. in *Protein Engineering*, 5:647–657.