Profile Hidden Markov Models: Specialized HMMs for sequence similarity

Based on Chapter 5 from Durbin, Eddy, Krogh, and Mitchison's book

Martin Tompa

CSE 427: Computational Biology

March 2, 2011

Suppose you had a family of related proteins that were aligned in a multiple sequence alignment Q. Suppose that you wanted to search a database of protein sequences to find additional proteins similar to those in Q. This is just like the task that BLAST performs, but the query is a multiple sequence alignment rather than a single sequence. The goal of this section is to show how hidden Markov models can be specialized for this task.

The specialized version is called a *profile HMM*. In Section 1 we will show how to construct a profile HMM from a multiple sequence alignment. In Section 2 we will show how to use the profile HMM to search a database of sequences.

1 Constructing a profile HMM from a multiple sequence alignment

If the multiple sequence alignment Q were ungapped, we could use the very simple HMM

$$\text{Begin} \longrightarrow M_1 \longrightarrow \cdots \longrightarrow M_j \longrightarrow \cdots \longrightarrow M_L \longrightarrow \text{End}$$

where L is the number of columns in the alignment Q, all transition probabilities are 1, and M_j has emission probability $e_{M_j}(a)$ equal to the frequency of residue a in column j of Q.

To accommodate gaps in Q, we generalize this simple HMM to the profile HMM H shown in Figure 5.2 of Durbin *et al.*. Squares (other than Begin and End) are called "match states", diamonds are "insert states", and circles are "delete states". There are no emissions from delete states, nor from Begin or End.

In order to parameterize the HMM H according to Q, first choose the number L of match states. A simple rule here is to let L be the number of alignment columns that contain fewer

gap characters than nongap characters. See Figure 5.3 of Durbin *et al.* for an example, where the starred columns are chosen to correspond to the match states.

Having now fixed the topology of H to have L match states, $L+1$ insert states, and L delete states configured as in Figure 5.2, the next step is to train the parameters of H from Q. Each row of Q determines a path and emissions of H:

1. A nongap character r in a starred column corresponds to going to the next match state and emitting r.
2. A nongap character r in a nonstarred column corresponds to going to the next insert state and emitting r.
3. A gap character in a starred column corresponds to going to the next delete state.
4. A gap character in a nonstarred column is ignored.

Let A_{jk} be the total number of times the rows of Q cause H to traverse the directed edge (j,k), and let $E_j(b)$ be the total number of times the rows of Q cause H to emit b when in state j. Add pseudocounts as described in Section 3.4 of the Markov Models handout. Then set

$$\begin{aligned} a_{jk} &= \frac{A_{jk}}{\sum_{k'} A_{jk'}} \\ e_j(b) &= \frac{E_j(b)}{\sum_{b'} E_j(b')} \end{aligned}$$

exactly as described in that section.

2 Searching a sequence database with a profile HMM

There are two possible algorithms for evaluating how well a sequence X from the database matches the HMM H. We could use the Viterbi algorithm to find the most probable alignment of X with H together with its probability, or we could use the forward algorithm to calculate the posterior probability $\Pr(X \mid H)$ of X summed over all paths.

In either case, what we really want to calculate is the log likelihood ratio of this probability with respect to the background probability of the sequence X, which is $q_{X_1} q_{X_2} \ldots q_{X_L}$, where q_r is the background frequency of residue r in the database. The recurrences for either the Viterbi or the forward algorithm can be modified to calculate this log likelihood ratio directly. For example, Equations (5.1) in Durbin *et al.* provide the recurrences to compute the Viterbi log likelihood ratios. You can derive these similarly to the derivation of the Viterbi recurrence given in Section 3.2 of the handout on Markov Models.

An important benefit of modifying the recurrence to directly compute the log likelihood ratio is that this avoids the underflow problem of computing the Viterbi probability, which is the product of $2L$ probabilities and is likely to be an exponentially small number.

Figure 5.5 from Durbin *et al.* provides an example. A profile HMM was constructed from 300 globin proteins, and then the SWISS-PROT database of proteins was searched with the forward algorithm. The vertical axis shows the log likelihood ratio score, with a point in the graph for each protein in the database. The graph shows a clear separation of globins from non-globins.

Finally, for every good matching sequence X, we would like to add X to the multiple sequence alignment Q of the protein family. The simplest way is to use the Viterbi path of X in H, which dictates how to align X to Q.

References

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G., *Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.