HMM Lecture Notes-Part 2

Dannie Durand

Designing HMMs: Motif discovery and modeling

Position Specific Scoring Matrices capture the distribution of residues observed in each position in a conserved motif, but are not a good model for variable length motifs, recognition of new instances with insertions and deletions, and positional dependencies. Moreover, PSSMs can be used to search for instances of an ungapped motif in an unlabeled sequence, but do not lend themselves to precise boundary detection. We turned to Hidden Markov models to address these limitations. HMMs provide a flexible and expressive formalism for modeling conserved sequence motifs. In addition to modeling precise conserved motifs, like the `WEIRD` motif, HMMs can also be used to model biologically distinct regions that are characterized by a change in underlying sequence composition, rather than a precise pattern. Examples of these include transmembrane regions, which are enriched for hydrophobic residues, and CpG islands, which have higher GC content.

There are three major computational tasks associated with conserved motifs found in multiple sequences: Discovery, modeling, and recognition. In the past two weeks, we have discussed the recognition problem: Given an HMM, how do we use it to ask questions about patterns in a new, unlabeled sequences? In this lecture, we consider modeling and discovery. For HMMs, modeling and discovery are closely coupled. There are two major issues to consider: designing the HMM topology and estimating the parameters of the model.

A fundamental tradeoff drives HMM design: On the one hand, more complex models, with more parameters, can yield more accurate and biologically realistic models. On the other hand, as the number of parameters increases, so does the amount of data needed to estimate parameters without overfitting.

HMM topology

Here, we discuss how to design the topology of an HMM. This includes the set of states, $S_1, \ldots, S_N$, and how they are connected; in other words, we must specify which pairs of states will be connected by edges with non-zero transition probabilities. We could just choose a fully connected graph, but typically this has too many parameters to estimate. Instead, we can exploit biological knowledge. The goal is to choose a topology that limits the number of states and edges, while still being expressive enough to represent the structure of the biological pattern of interest.

Model topology strongly influences the properties of the patterns we will discover. Recall the limitations of the PSSM motif model that HMMs can address: boundary detection, dependencies between positions in the motif, variable length patterns, and recognition of pattern instances that

contain indels. As we discussed in the lectures on motif recognition, the generative aspect of HMMs provides an improved model for boundary detection: HMMs map symbols to states; changes in state define boundaries. The other problems can all be addressed by designing a topology that captures inter-site dependencies and flexibility in pattern length.

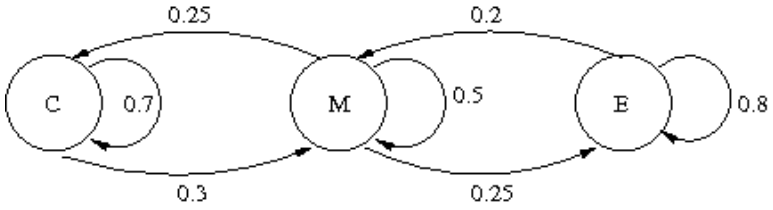

Figure 1:

The transmembrane models we discussed in class illustrate some of the issues to consider. For example, the three state model, Fig. 1, is not sufficiently restrictive to emit only transmembrane protein sequences. It can emit sequences that are entirely cytosolic or sequences that pass from the cytosol into the membrane and back to the cytosol, without ever passing through the extracellular region.

We can use topology to impose order dependencies on the model. Suppose the goal is to model a sequence that always starts and ends in the cytosol, with multiple passes through the cell membrane into the extracellular matrix, and back through membrane to the cytosol. By adding additional states, we can obtain a model that only emits sequences that satisfy these conditions (Fig. 2). Note that this model has silent *Start* and *End* states, which we have not encountered before. These states do not emit symbols. They serve to ensure that the entry and exit from the model occur in specific states.

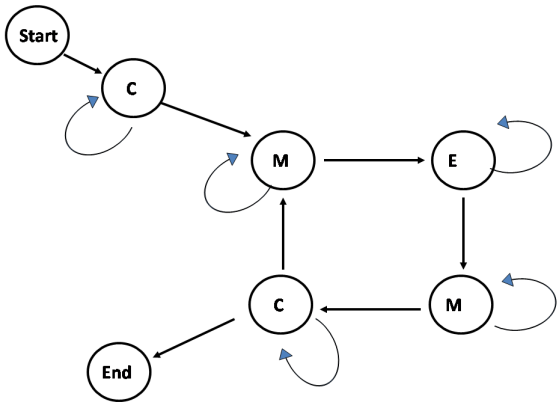

Figure 2:

The topology of the model also influences the distribution of the lengths of sequences that the

model will emit. For example, the model in Fig. 2 emits sequences of symbols with state paths that pass through subcellular compartments in the appropriate order, but it does not constrain these segments to be lengths typical for the associated cellular compartments.

The choice of topology can impose a probability distribution on the length of the sequences that the HMM generates. For example, a simple self loop with probability p results in sequences with lengths that follow an exponentially decaying (geometric) distribution (Fig. 3). The probability that this model will emit a sequence of length l is $(1-p)p^{l-1}$. This is not a realistic description of the length of amino acid sequences. More complex topologies can be used to obtain more realistic length distributions. Some of these are described in Durbin, 3.4.

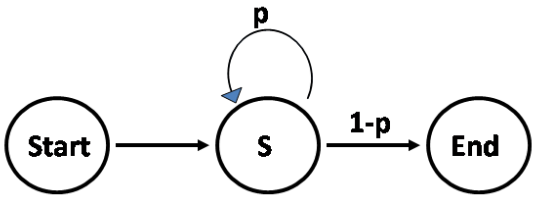

Figure 3:

Finally, we must also choose the alphabet and decide which states will emit which symbols. The larger the alphabet, the greater the number of emission probabilities must be estimated. The transmembrane models we discussed in class used a two letter alphabet of hydrophobic (`H`) and hydrophilic (`L`) residues to represent sequences, instead of the full 20 letter alphabet for amino acids. This not only gives a simpler representation, it also requires fewer training sequences to learn the parameters since all hydrophobic (resp. hydrophilic) residues contribute to the estimation of a single parameter.

Parameter estimation

Once the states and connectivity have been chosen, the parameters of an HMM are estimated from training data. We are given observed sequences, $O^1, O^2, ..., O^k$, and wish to construct an HMM with parameters, λ, to model these sequences. If the sequences are labeled, the transition and emission probabilities can be estimated easily from the observed transition and emission frequencies. If the sequences are unlabeled, we must first discover the conserved pattern using a machine learning algorithm.

Labeled sequences: We are given labeled sequences in which every symbol O_t^d is associated with a state, $q_t^d = S_i$. The transition probabilities are calculated by tabulating the number of observed state changes in the data:

$$a_{ij} = \frac{\sum_{d=1}^{k} A_{ij}^{d}}{\sum_{d=1}^{k} \sum_{j'} A_{ij'}^{d}},$$

where A_{ij}^{d} is the number of pairs of adjacent symbols, $O_t^d \, O_{t+1}^d$, that are labeled $S_i S_j$. The emission probabilities are given by

$$e_i(\sigma) = \frac{\sum_{d=1}^{k} E_i^d(\sigma)}{\sum_{d=1}^{k} \sum_{\alpha=\Sigma} E_i^d(\alpha)}$$

where $E_i^d(\sigma)$ is the number of instances in O^d where the symbol σ is labeled with state S_i. Finally, the initial probability π_i is given by

$$\pi_i = \frac{1}{k} \sum_{d=1}^{k} I^d(i)$$

where

$$I^d(i) = \begin{cases} 1, & \text{if } q_1^d = S_i \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

is an indicator variable that is equal to one when the first symbol in O^d is labeled with state S_i and zero otherwise.

We may wish to include pseudocounts to account for cases not observed in the training data. Pseudocounts are incorporated into the emission probabilities in the same way that we used pseudocounts in the definition of the frequency matrix for PSSMs. The probability of emitting σ from state S_i is

$$e_i(\sigma) = \frac{E_i(\sigma) + b}{\sum_{\alpha \in \Sigma} E_i^d(\alpha) + |\Sigma| b} \, ,$$

where b is a pseudocount. We can also use pseudocounts to account for state transitions that are allowed, but not observed in the training data. Let $\mathcal{N}(i)$, the neighborhood of state S_i, be the set of states that can be reached from S_i in a single transition. In other words, $\mathcal{N}(i)$, is the set of states, S_j such that a_{ij} has not been explicitly defined to be zero in the design of the topology. Then,

$$a_{ij} = \frac{A_{ij} + b}{\sum_{j' \in \mathcal{N}(i)} [A_{ij'} + b]} \, .$$

Note that inferring parameters from counts in labeled data is a form of maximum likelihood estimation - we are assuming that the emission and transition probabilities that best model the motif of

interest are those that maximize the probability of the observed symbols and states in the training data.

As an example, consider the three-state transmembrane model. For this model, seven transition probabilities must be inferred: a_{CC}, a_{CM}, a_{MC}, a_{MM}, a_{ME}, a_{EM}, and a_{EE}. Given the following labeled sequence:

H	H	H	L	L	H	L	H	L	L	H	H	H	H	H
C	C	C	C	C	C	C	C	C	C	M	M	M	M	M

we count the number of CC pairs to determine A_{CC} and normalize by the number of pairs of the form $C*$. Since there are nine pairs of adjacent symbols labeled CC and one pair labeled CM, for this sequence $a_{CC} = 0.9$ without pseudocounts. With pseudocounts,

$$a_{CC} = \frac{A_{CC} + b}{\sum_{j' \in \mathcal{N}(C)} [A_{Cj'} + b]} ,$$

where $\mathcal{N}(C) = \{C, M\}$. With a pseudocount of $b = 1$, we obtain

$$\begin{aligned} a_{CC} &= \frac{9+1}{(9+1)+(1+1)} , \\ &= \frac{10}{12}. \end{aligned}$$

To obtain the emission probabilities from state C, note that C is associated with five hydrophobic and five hydrophilic residues. Thus,

$$\begin{aligned} e_C(\texttt{H}) &= \frac{E_C(\texttt{H}) + b}{\sum_{\alpha \in \{\texttt{H},\texttt{L}\}} E_C(\alpha) + 2b} , \\ e_C &= \frac{5+1}{10+2} \end{aligned}$$

or $e_C(\texttt{H}) = 0.5$, assuming a pseudocount of $b = 1$.

The other transition and emission probabilities are estimated similarly. We estimate the initial probability π_C by counting the number of sequences that begin in the cytosol, and normalizing by the total number of sequences. Once we have learned the parameters, we can use the model to recognize new transmembrane sequences.

Unlabeled sequences: If the sequences are *unlabeled*, then it is necessary to both discover the motif and learn the model parameters. The motif is discovered automatically, but implicitly, through the process of parameter inference. Once the parameters have inferred, the parameters are used to obtain an explicit model of the motif via Viterbi or posterior decoding.

Parameters are inferred by maximizing the likelihood of the data. Given sequences $O^1, O^2, \ldots$, we seek

$$\begin{aligned} \underset{\lambda_l}{\operatorname{argmax}} \mathcal{L}(\lambda_l) &= \underset{\lambda_l}{\operatorname{argmax}} \sum_d P(O^d|\lambda_l) \\ &= \underset{\lambda_l}{\operatorname{argmax}} \sum_d \sum_Q P(O^d|\lambda_l, Q) \ . \end{aligned}$$

Except for very small problem instances, finding a global maximum is intractable. We would have to calculate $\mathcal{L}(\lambda_l)$ for all possible combinations of parameters, λ_l, to find the parameters that maximize $P(O^1 \ldots O^k|\lambda_l)$. Instead, heuristics are used. These are typically guaranteed to find at least a local maximum. Since these are heuristics, evaluation is usually done empirically by withholding some of the training data for testing, but we will not discuss this further.

The *Baum-Welch* algorithm is used to estimate the parameters of a Hidden Markov model from unlabeled training data. Baum-Welch belongs to a family of algorithms, called Expectation Maximization (EM) algorithms, that work by alternating between estimating the likelihood of the data, given the current estimate of the parameters and re-estimating the parameters from the current likelihoods. Baum-Welch is based on algorithms that we have already encountered: If we have labeled data, we can estimate the model parameters, as described in the previous section. If we have a model with parameters, we can infer a motif in unlabeled sequences using decoding. Informally, Baum-Welch is an iterative algorithm that alternately applies these two procedures. First, an initial estimate of the parameter values is required, for example, based on prior knowledge of the biology underlying the model or on a uniform prior. With this initial estimate, the model is used to label the training data, typically using posterior decoding with the Forward and Backward algorithm. Once the sequences have been labeled, the parameters are re-estimated from the labeled data. The training sequences are then re-labeled using this new estimate of the parameters. The algorithm iterates, alternately labeling the data with the current estimate of the parameter values and then re-estimating the parameters from the labeled data. At each iteration, the likelihood is guaranteed to remain unchanged or increase. This iterative process terminates when the likelihood ceases to improve.

It is instructive to note the similarities and differences between the Baum-Welch algorithm and the Gibbs sampler. Like Baum-Welch, the Gibbs sampler alternates between re-estimating parameters (i.e., a PSSM) from the current estimate of the motif and inferring new instance of the motif from the updated parameters. However, unlike Baum-Welch, where every training sequence is relabeled at each iteration, in the Gibbs sampler, only one sequence is relabeled at each iteration. A second major difference between the two methods is that the Gibbs sampler is guaranteed to converge to a global optimum (given enough time). This is because at each iteration the Gibbs sampler is allowed to select a suboptimal instance of the motif with non-zero probability. In contrast, the Baum-Welch algorithm is only guaranteed to find a local optimum.

A formal statement of the Baum-Welch algorithm is given on page 7. Given the observed, unlabeled sequences (denoted $O^d = O_1^d, O_2^d, ...$), the parameters are re-estimated in the inner loop of the

algorithm. A_{ij} is the expected number of transitions from S_i to S_j. For a given sequence, O^d, probability of transiting from state i to j at time t is $P(q_t^d = i, q_{t+1}^d = j|O^d, \lambda)$. To facilitate the calculation, we again use the trick of converting the conditional probability into a joint probability:

$$\begin{aligned} P(q_t^d = i, q_{t+1}^d = j|O^d, \lambda) &= \frac{P(q_t^d = i, q_{t+1}^d = j, O^d)}{P(O^d)} \\ &= \frac{\alpha(t,i)a_{ij}e_j(O_{t+1}^d)\beta(t+2,j)}{P(O^d)} . \end{aligned}$$

The term $\alpha_t(i)$ is the probability that the model has emitted symbols $O_1^d \ldots O_t^d$ and is in state S_i at time t. This probability can be obtained using the Forward algorithm. The term in the denominator, $P(O^d)$, is also calculated with the Forward Algorithm. The terms a_{ij} and $e_j(O_{t+1}^d)$ give the probability of making the transition from S_i to S_j and emitting O_{t+1}. The Backward algorithm yields $\beta_{t+2}(j)$, the probability of emitting the rest of the sequence if the model was in state S_j at time $t+1$. From this we can estimate

$$A_{ij} = \sum_d \frac{\sum_t \alpha(t,i)\, a_{ij}\, e_j(O_{t+1}^d)\, \beta(t+2,j)}{P(O^d)} \tag{2}$$

Note that instead of explicitly labeling the data and then counting state transitions as we do with labeled data, the association of symbols and states is implicit in the re-estimation process in the inner loop of the algorithm.

$E_i(\sigma)$ is the expected number of times that σ is emitted from state S_i:

$$\begin{aligned} E_i(\sigma) &= \sum_d P(q_t^d = S_i, O_t^d = \sigma|O^d, \lambda) &(3)\\ &= \sum_d \frac{\sum_{\{t|O_t^d=\sigma\}} \alpha(t,i)\, \beta(t+1,i)}{P(O^d)} . &(4) \end{aligned}$$

Again, the quantities on the right hand side can be calculated using the Forward and Backward algorithms. Finally, the initial probability π_i is given by

$$\begin{aligned} \pi_i &= \sum_{d=1}^{k} p(q_1^d = S_i|O_d, \lambda) &(5)\\ &= \sum_{d=1}^{k} \frac{I^d(i)}{P(O^d)} &(6) \end{aligned}$$

Algorithm 1: Baum-Welch

Input:

`A set of observed sequences,` $O^1, O^2, \ldots O^k$

Initialization:

`Select arbitrary model parameters,` $\lambda = \pi_i a_{ij}, e_i(\cdot)$.

Iteration:

`Repeat`

{

`For each sequence,` O^d,

{

`Calculate` $\alpha(t,i)$, $\beta(t,i)$ `and` $P(O_d)$ `using Forward and Backward algorithms.`

$A^d_{ij} = \frac{1}{P(O^d)} \cdot \sum_t \alpha(t,i) a_{ij} e_j(O^d_{t+1}) \beta(t+2, j)$

$E^d_i(\sigma) = \frac{1}{P(O^d)} \sum_{\{t | O^d_t = \sigma\}} \alpha(t,i)\, \beta(t+1, i).$

}

$a_{ij} = \frac{\sum_d A^d_{ij}}{\sum_d \sum_l A^d_{il}}$

$e_i(\sigma) = \frac{\sum_d E^d_i(\sigma)}{\sum_d \sum_\alpha E^d_i(\alpha)}$

$\pi_i = \sum_{d=1}^k \frac{I^d(i)}{P(O^d)}$

$\lambda = (\pi_i, a_{ij}, e_i())$

$\mathcal{L}(\lambda) = \prod_d P(O^d | \lambda)$

}

`Until (`$\mathcal{L}(\lambda)$ `stops changing.)`

Convergence: It can be proven that if current estimate is replaced by these new estimates then the likelihood of the data will not decrease (i.e. will increase unless already at a local maximum/critical point). See Durbin, Section 11.6 for discussion of avoiding local maxima and other typical pitfalls with this algorithm.

The *Baum-Welch* algorithm will learn the parameters from the data and implicitly, also discovers the motif. To determine the motif explicitly, we use the Viterbi algorithm on the new HMM to label the states of each input sequence.

Profile HMMs

In 1994, Krogh, Haussler [1] and colleagues introduced a flexible HMM topology specifically to model conserved sequence motifs. It captures the propensity to observe specific amino acids or nucleotides at each position in a pattern and allows for insertions and deletions. This topology, called a *Profile HMM*, can be customized for a broad range of conserved motifs by selecting the appropriate length for a given motif and initializing the parameters to capture the specific properties of the motif.

Here, we introduce the features of the Profile HMM model by showing how it could be used to model the `WEIRD` motif based on the following alignment, which has no gaps and no positional dependencies:

```
WEIRD
WEIRD
WEIRE
WEIQH
```

We can recognize the `WEIRD` motif using an HMM with the simple topology shown in Fig. 4,

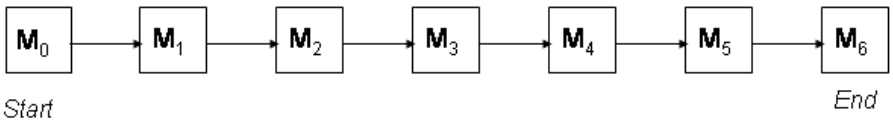

Figure 4:

where the transitions probabilities are $a_{i,j} = 1$, if $j = i + 1$, and zero, otherwise. Given labeled training sequences $O^1 \dots O^k$, the emission probabilities are

$$e_j(\sigma) = \frac{E_j(\sigma) + b}{\sum_{j\prime} E_{j\prime}(\sigma) + |\Sigma| b} , \tag{7}$$

where $E_j(\sigma)$ is the number of instances of σ at position j in the motif and b is a pseudocount. Note that according to this definition, $e_j(\sigma) = F[\sigma, j]$, where $F[\sigma, j]$ is the same frequency matrix that we derived for the PSSM example, using pseudocounts. The Start and End states (M_0 and M_6) are silent.

[1] Krogh, A., Brown, M., Mian, I. S., Sjolander, K., and Haussler, D. (1994). Hidden Markov models in computational biology: Applications to protein modeling. J. Mol. Biol., 235:1501-1531.

In order to score a new sequence, O, possibly containing an instance of the `WEIRD` motif, we calculate a likelihood ratio.

$$\frac{P(O|H_A)}{P(O|H_O)}$$

The above model is our alternate hypothesis, H_A. In order to obtain a likelihood ratio, we also need a background model (the null hypothesis, H_0):

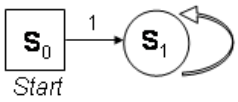

Figure 5:

In this model, all transition probabilities are equal to one. The emission probabilities are $e_j(\sigma) = p(\sigma)$, where $p(\sigma)$ is the background frequency of residue σ. We can then score a new sequence, O, by calculating the probability that O was emitted by the Profile HMM in Fig. 4 and comparing it with the probability that O was emitted by the background model (Fig. 5). For example, if $O = O_1O_2O_3O_4O_5$ is a sequence of length five, then $P(O|H_A)$ is equal to

$$\pi_{M_O} \cdot e_{M_1}(O_1) \cdot a_{M_O M_1} \cdot e_{M_2}(O_2) \cdot a_{M_2 M_3} \cdot e_{M_3}(O_3) \cdot a_{M_3 M_4} \cdot e_{M_4}(O_4) \cdot a_{M_4 M_5} \cdot e_{M_5}(O_5) \cdot a_{M_5 M_6}.$$

Since the initial and transition probabilities are equal to one ($\pi_{M_O} = 1; a_{M_i M_{i+j}} = 1, 0 \leq i \leq 5$), this reduces to

$$P(O|H_A) = e_{M_1}(O_1) \cdot e_{M_2}(O_2) \cdot e_{M_3}(O_3) \cdot e_{M_4}(O_4) \cdot e_{M_5}(O_5)$$

or

$$P(O|H_A) = \prod_{j=1}^{5} e_{M_j}(O_j) \ .$$

The probability that O was emitted by the background model is $\prod_j^5 p(O_j)$. Putting it all together, the score of sequence O is the log likelihood ratio

$$\sum_{j=1}^{5} \log \frac{e_{M_j}(O_j)}{p(O_j)} \ ,$$

which is equivalent to $\sum_{i=1}^{5} S[O_i, i]$, the score we would have obtained with the PSSM for the `WEIRD` motif.

We can modify the basic HMM to recognize query sequences with insertions as shown in Fig. 6, such as $O =$ `WECIRD`, by adding an insertion state between any two match states, M_i and M_{i+1}. The emission probabilities for the insertion state are the background frequencies.

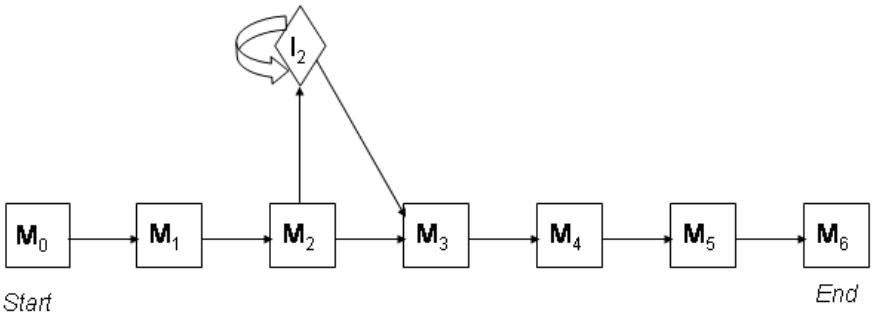

Figure 6:

Suppose our query sequences has a deletion, e.g., O = `WERD`. One approach to capturing such deletions would be to add edges allowing us to jump over any set of match states (Fig. 7a):

The disadvantage to this approach is that the number of transitions grows rapidly as the number of match states increases. To infer the transition probabilities, we would need a very large set of training data, one in which all deletions of all possible sizes were represented. Instead, we can model long deletions as sequences of short ones, as seen in the HMM in Fig. 7b.

Putting all of these features together, we obtain the canonical Profile HMM model, shown in Fig. 8. A Profile HMM has a column containing a Match, an Insertion and a Deletion state for each position in the conserved pattern. States M_j, I_j, and D_j, correspond to the jth position in the pattern. Insertion and Match states emit the 20 amino acids (for protein motifs), or the four nucleotides (for DNA and RNA motifs). Delete states emit "-". The emission and transition probabilities must be estimated from data. A leading insertion state, I_0, allows for patterns that occur in the middle of a sequence. If the pattern ends before the end of the sequence, the remaining sequence is emitted by the insertion state I_n, where n is the last position in the pattern. A Profile HMM of length 5 is shown as an example.

Note that there is a path from the Start state, M_0, to the End state, M_n, that passes only through Insertion and Deletion states. Thus, a Profile HMM can emit a sequence that does not contain an instance of the pattern. Such a sequence would have a low probability, compared with a sequence generated by the Match states. One can also determine whether a sequence has the pattern by inspecting the state path inferred using Viterbi or Posterior decoding.

Parameter estimation from labeled data: Given labeled training data (i.e., we are given the state path for each sequence, O^d), we use maximum likelihood to estimate the parameters. In

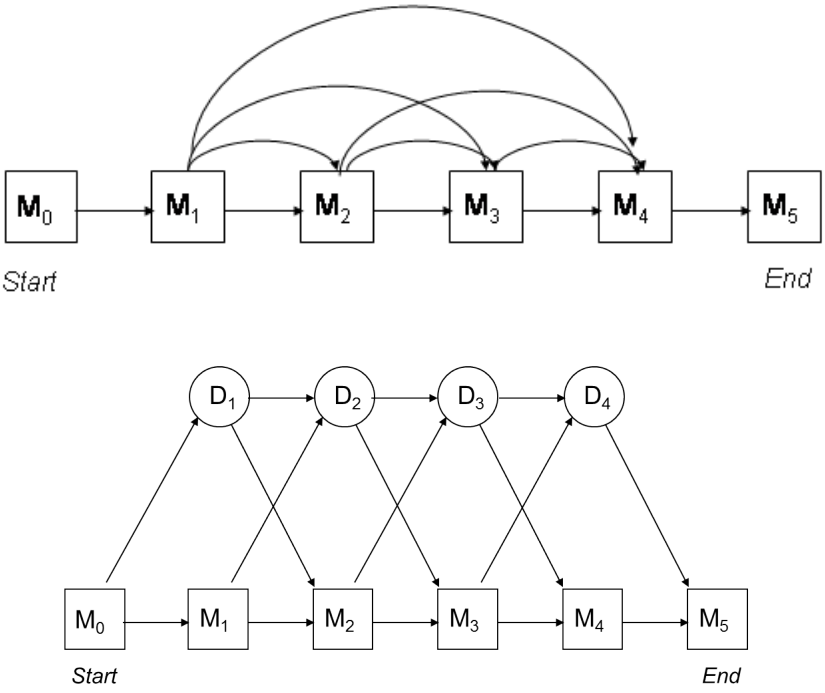

Figure 7: (a) top, (b) bottom

general,

$$a_{i,j} = \frac{A'_{i,j}}{\sum_l A'_{i,l}}$$

where $A'_{i,j}$ is the number of transitions from i to j in the training data, plus a pseudocount to reflect transitions that are not observed in the training data (see page 13 for an example). The emission probabilities for the match states are defined in equation 7.

For our Profile HMM, the estimation of the emission probabilities might look like this:

$$e_{M_0}(\sigma) = e_{M_6}(\sigma) = 0, \ \forall \sigma$$

$$e_{I_j}(\sigma) = p(\sigma), \ \forall I_j, \sigma \in \Sigma$$

$$e_{D_j}(\sigma) = 0 \qquad e_{D_j}(\text{“}-\text{”}) = 1$$

$$e_{M_j}(\sigma) = \frac{E_j(\sigma) + b}{\sum_\alpha E_j(\alpha) + 20b}$$

where $p(\sigma)$ is the background probability of residue σ.

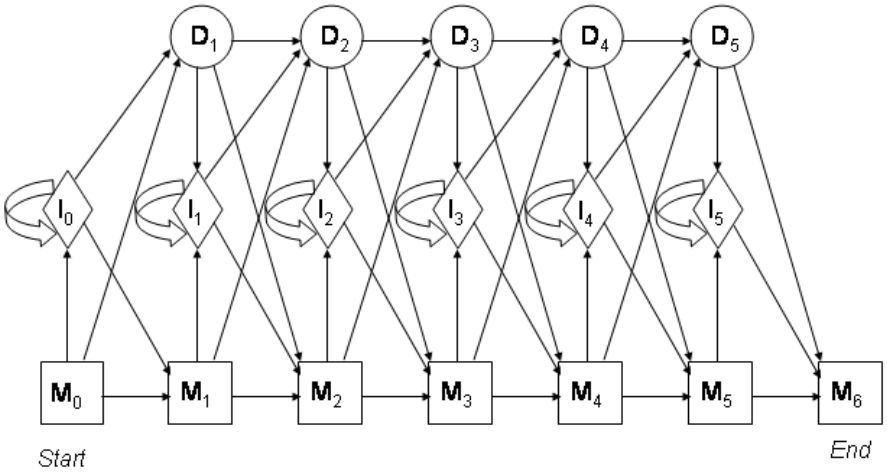

Figure 8: A profile HMM of length 5

Constructing a Profile HMM

How do we construct a Profile HMM to model a specific conserved pattern in biomolecular sequences? If the sequences are already aligned, then we have labeled training data. In other words, we can determine from the alignment which state is associated with each symbol in each sequence. In that case, all we need to do is determine the number of Match states in the Profile HMM, set up the topology, and determine the parameters from the labeled data. We will refer to the number Match states, not including the silent Start and End states, as the *lengths* of a Profile HMM.

Given unlabeled sequences that are known to have a common pattern, we can use the Profile HMM to discover the pattern, infer the parameters, label the data, and construct a multiple sequence alignment. We give an example of each case below.

An example of a profile HMM for a variable length motif with labeled data: Profile HMM's like the one above can be used to model variable length motifs, such as this one:

```
VG--H
V---N
VE--D
IAADN
```

The length of the Profile HMM should be the average of the length of the sequences. The above sequences are of lengths 3, 2, 3, and 5 (before the gaps were inserted), respectively, yielding an

average of 3.25. Our HMM will have a silent start state M_0, Match states M_1, M_2, M_3, Insertion states I_0, I_1, I_2, I_3, Deletion states D_1, D_2, D_3 and a silent end state M_4.

In order to estimate the parameters, we need to assign labels to the data using the multiple alignment. Columns in the alignment that have gaps in less than 50% of the rows correspond to Match states. Those with more than 50% gaps correspond to Insertion states:

V	G	-	-	H
V	-	-	-	N
V	E	-	-	D
I	A	A	D	N
M_1	M_2	I_2	I_2	M_3

This yields the following labeled sequences:

V	G	H
M_1	M_2	M_3

V	-	H
M_1	D_2	M_3

V	E	D
M_1	M_2	M_3

I	A	A	D	N
M_1	M_2	I_2	I_2	M_3

From these labeled sequences, we can estimate the parameters. For example, for state M_1 using $b = 1$ as a pseudocount, we obtain

$$e_{M_1}(V) = \frac{3+1}{4+20}.$$

The transition probability from M_2 to I_2 is

$$a_{M_2 I_2} = \frac{1+1}{(2+1)+(1+1)+(0+1)}.$$

The three sums in the denominator correspond to all possible transitions out of state M_2, plus a pseudocount for each edge leaving M_2. Specifically, in the training sequences there are two transitions from M_2 to M_3, one transition from M_2 to I_2 and no transitions from M_2 to D_3. The

other emission and transition probabilities are calculated the same way. The model always starts in M_O, so $\pi_{M_j} = 0$, when j>0.

Modeling unlabeled data with a Profile HMM: To discover a pattern in unlabeled data requires the following steps:

1. **Estimating the length:** If you are given a set of unaligned sequences, where each sequence is an instance of the pattern, set the length of the HMM (i.e., the number of non-silent Match states) to L, where L is the average sequence length.. An example of this type of input would be sequences approximately 50 residues long, where each sequence correspond to a different instance of the Ig domain.

   If you are given sequences that contain a pattern, but are much longer than the pattern, then you will need some approach to estimating the length. An example of this type of input would be a set of protein sequences, typically several hundred residues in length, each of which contains an instance of an unknown domain. In this case, you might estimate the length of the pattern to be approximately 100, since that is the length of a typical domain.

2. **The topology:** Construct a Profile HMM with $L + 2$ Match states, $L + 1$ Insertion states, and L Deletion states. M_0 and M_{L+1} are silent states corresponding to the Start state and the End state.

3. **Learn parameters:** Guess "good" initial parameters (e.g., $a_{M_i\, M_j} \gg a_{M_i\, I_j}$ and $a_{M_i\, M_j} \gg a_{M_i D_j}$) and train the model using the Baum Welch algorithm.

4. **Determining the motif:** Use the Viterbi algorithm or posterior decoding to infer the state path that emitted each sequence. The Viterbi recurrence can be greatly simplified and expressed in terms of log odds for the special case of Profile HMMs (Durbin, pp 108-110). The log odds formulation avoids underflow and reduces length effects. This was not covered in class. Note the similarity to the dynamic programming algorithm for pairwise alignment.

5. **Multiple Sequence Alignment:** The most likely paths for each sequence obtained from decoding can be used to obtain a multiple alignment of the input sequences. If symbols O^d_t and O^c_u were emitted by same Match state, then align positions t and u in sequences O^d and O^c, respectively. See Ewens and Grant, p 337 - 339 for a discussion and example of multiple sequence alignment using Profile HMMs.

6. **Model surgery:** The topology of the model can be iteratively refined. If more than half of the sequences enter the Delete state at a particular position, then remove the Match, Insertion and Deletion states at that position from the topology. If more than half of the sequences enter the Insertion state at a given position, then add Match, Insertion and Deletion states

(in number equal to average length of the insertion) at that position.

7. **Re-estimate the parameters:** If the states change due to model surgery, you will need to re-estimate the parameters. Label the multiple alignment with the new states and calculate the transition and emission probabilities as described above for labeled data. If the number of states that changed is a significant percentage of the entire HMM, then you may obtain better results by retraining with the Baum Welch algorithm.

Compared with the exact dynamic programming algorithm for multiple sequence alignment, which runs in exponential time, this approach can align many sequences quickly.

Pattern recognition with profile HMMs: Once you have constructed your Profile HMM, how do you determine whether a new, unlabeled sequence, O, contains the motif?

- If you have a model for a suitable null hypothesis, H_0, you can obtain a log odds ratio, $\log \frac{P(O|H_A)}{P(O|H_0)}$, using the Forward algorithm to determine the probability of the sequence for each model. This gives a score, but does not infer the location of the motif.
- Alternatively, you can find the most likely path using the Viterbi algorithm or posterior decoding. The location of the motif corresponds to the symbols emitted by the Match states. If no symbols were emitted by Match states, then the motif is not present in O.