axis is the joint probability, $P(O^d, Q^b|\lambda)$, that the HMM will visit the states on path Q^b and emit sequence O^d. In the three-state TM model example, the set of all possible sequences, $O^1, O^2, O^3, \ldots$ corresponds to `H, L, HH, HL, LH, LL, HHH, ...` and the set of all possible state paths, $Q^1, Q^2, Q^3, \ldots$ corresponds to *C, M, E, CC, CM, CE, MC, ...* Note that $P(O^d, Q^b) = 0$ for many (O^d, Q^b) pairs. For example, $P(O^d, Q^b) = 0$ when O^d and Q^b have different lengths. In our three-state model, $P(O^d, Q^b) = 0$ for any state path that contains C adjacent to E, because $a_{CE} = 0$.

An HMM emits each sequence $O^d \in \Sigma^*$ with probability $P(O^d) \geq 0$. Since a sequence can, potentially, be emitted from more than one state path, in order to obtain the total probability of a sequence, O, we must sum over the all possible paths:

$$P(O) = \sum_b P(O|Q^b, \lambda) \cdot P(Q^b) = \sum_b P(O, Q^b|\lambda).$$

Fig. 5.6b shows a cartoon representation of $P(O, Q)$ for a single sequence, O^5, for the set of all possible state paths, Q. The area under the curve is equal to $P(O)$, the total probability of sequence O.

When all possible sequences and all possible paths are considered, the probability distribution shown in Fig. 5.6a sums to one:

$$\sum_d \sum_b P(O^d, Q^b|\lambda) = 1.$$

5.3 Using HMMs for recognition

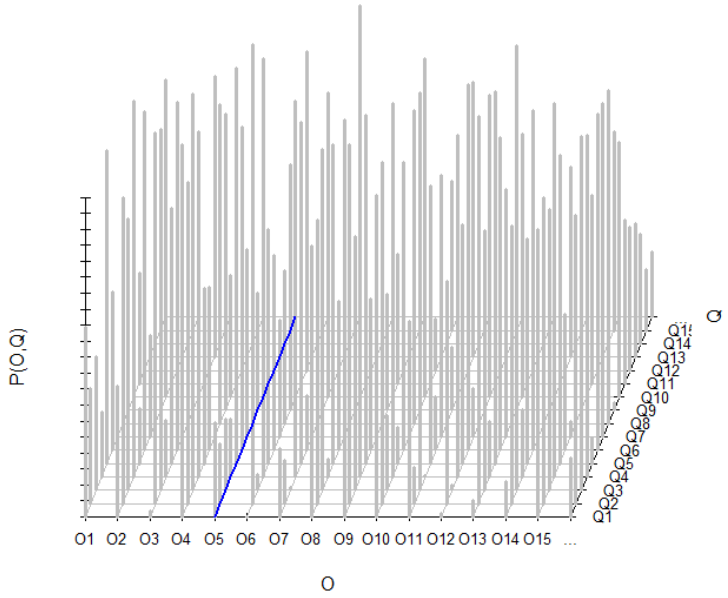
In this section, we focus on motif recognition using HMMs. We will discuss parameter estimation, motif discovery, and modeling using HMMs in future sections. Here, we assume that we are given an HMM with known parameter values.

Our goal is to use the HMM to answer the various recognition questions, including:

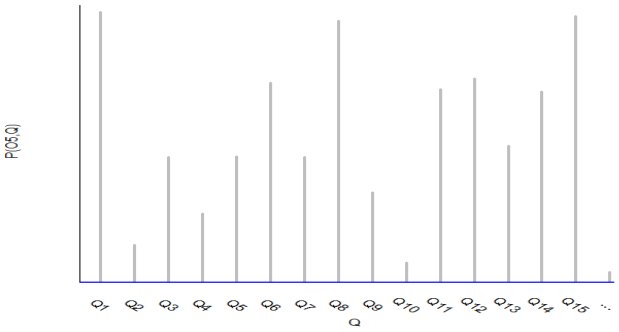
1. What is the probability that a given sequence, O, was generated by the HMM?
   Example: Is sequence O a transmembrane protein?

2. Given a sequence, O, what is the true path? Otherwise stated, we wish to assign labels to an unlabeled sequence.
   Example: Identify the cytosolic, transmembrane, and extracellular regions in sequence O. In this case, we wish to assign the labels E, M, or C to each amino acid residue in the sequence.

3. What is the probability of being in state E_i when symbol O_t is emitted?
   Example: Is a given residue localized to the membrane?



(a) The joint probability, $P(O^d, Q^b)$.



(b) $P(O^5, Q^b)$

Figure 5.6: **(a)** The joint probability $P(O^d, Q^b)$ for every sequence O^d and state path Q^b. The volume under this curve is one. **(b)**The probability of sequence $O = O^5$ for every state path $Q^1, Q^2, Q^3 \ldots$. This curve corresponds to the intersection of the probability distribution in Fig. 5.6a and the vertical plane at $O = O^5$ (shown as a blue line in Fig. 5.6a). The area under this curve is $P(O^5|\lambda)$, the probability of O^5. The maximum point on the curve is the most probable path, $Q^* = \text{argmax}_Q P(Q|O, \lambda)$.

Although this is not a focus of this class, we should point out that, since HMMs are generative models, an HMM can also be used for simulation; for example, to generate sequences with properties similar to real transmembrane sequences.

5.4 Calculating the total probability of sequence O

In order to answer the first question,

1. What is the probability that a given sequence, O, was generated by the HMM?

we must calculate $P(O|\lambda)$, the total probability of the sequence given the model. The total probability is the probability of visiting states in path Q and emitting sequence O, summed over all possible state paths:

$$P(O|\lambda) = \sum_b P(O|Q^b, \lambda) \cdot P(Q^b|\lambda) = \sum_b P(O, Q^b|\lambda).$$

We could calculate this quantity by enumerating all paths, Q^b, and calculating $P(O, Q^b|\lambda)$ for each one, but this brute force approach becomes intractable as the number of states gets large, since the number of state paths grows as $O(N^T)$. Instead, we use a dynamic programming algorithm called the *Forward* algorithm, which recursively calculates the probability of emitting prefixes of O. At each step, the Forward algorithm calculates the probability of emitting the first t symbols, $O_1, O_2, \ldots O_t$, summing over all possible paths that end in state E_i. We designate this quantity

$$\alpha(t, i) = P(O_1, O_2, O_3, ...O_t, q_t = E_i).$$

The variable α is an $T \times N$ matrix, where the rows correspond to prefixes of O and the columns correspond to states. At the t^{th} iteration, the algorithm calculates the entries in row t of the matrix, based on the entries in row $t-1$ and the parameters of the model. The entries in the final row contain the probability of emitting the entire sequence and ending in state E_i, for $i = 1$ to N. The probability of emitting the entire sequence, independent of the final state, is obtained by the summing the entries in the last row. The algorithm to calculate $\alpha(t, i)$ for all $t \in (1, T)$ proceeds as follows:

Algorithm: Forward

Initialization:

$$\alpha(1, i) = \pi_i e_i(O_1)$$

Recursion:

$$\alpha(t, i) = \sum_{j=1}^{N} \alpha(t-1, j) \cdot a_{ji} \cdot e_i(O_t)$$

Final:

$$P(O) = \sum_{i=1}^{N} \alpha(T, i)$$

The computational complexity of the Forward algorithm is $O(TN^2)$: There are $T \times N$ cells in the α matrix and the computational cost of computing each cell is $O(N)$.

In class, we worked an example based on the three-state transmembrane model shown in Fig. 5.5. A worksheet for this exercise is linked to the class syllabus page. The solution is also available. I recommend that you try to work through the Forward algorithm before looking at the solution.

5.5 Decoding

Next, we tackle the second recognition question:

2. Given a sequence O, what is the true path?

Given an unlabeled sequence, our goal is to classify or *label* each symbol in the sequence by inferring the state path. This process is called "decoding" because we decode the sequence of symbols to determine their meaning. HMMs were developed in the field of speech recognition, where recorded speech is "decoded" into words or phonemes to determine the meaning of the utterance. In our application, we decode an amino acid sequence to infer the functional role of each residue. There are two common approaches to decoding: Viterbi decoding and posterior decoding.

5.5.1 Viterbi decoding

Viterbi decoding is based on the assumption that the *most probable path*,

$$Q^* = \underset{Q}{\operatorname{argmax}}\, P(Q|O, \lambda),$$

is a good estimation of the sequence of states that generated the observed sequence O.[1] In practice, we maximize the joint probability $P(Q, O|\lambda)$, rather than the conditional $P(Q|O, \lambda)$, but this will still give us the most probable path because the path that maximizes $P(Q, O|\lambda)$ also maximizes $P(Q|O, \lambda)$. To see this, note that

$$P(Q|O, \lambda) = \frac{P(Q, O|\lambda)}{P(O|\lambda)}.$$

Since $P(O|\lambda)$ is independent of Q,

$$\operatorname*{argmax}_Q P(Q|O, \lambda) = \operatorname*{argmax}_Q P(Q, O|\lambda).$$

As in the case of the Forward algorithm, the brute approach of enumerating all paths and calculating $P(Q|O, \lambda)$ for each one is intractable, because the number of state paths grows as $O(N^T)$. Instead, we calculate $\operatorname{argmax}_Q P(Q, O|\lambda)$ using a dynamic programming algorithm called the *Viterbi* algorithm. Let $\delta(t, i)$ be the probability of emitting the first t residues via the most probable path that ends in E_i. We calculate $\delta(t, i)$ as follows:

Algorithm: Viterbi

Initialization:

$$\delta(1, i) = \pi_i \cdot e_i(O_1)$$

Recursion:

$$\delta(t, i) = \max_{1 \leq j \leq N} \delta(t-1, j) \cdot a_{ji} \cdot e_i(O_t)$$

$$j^*(t, i) = \operatorname*{argmax}_{1 \leq j \leq N} \delta(t-1, j) \cdot a_{ji} \cdot e_i(O_t)$$

Final:

$$P(Q^*, O|\lambda) = \max_{1 \leq j \leq N} \delta(T, j)$$

$$j^*(T) = \operatorname*{argmax}_{1 \leq j \leq N} \delta(T, j).$$

At each step in the recursion, we save $j^*(t, i)$, the value of j that maximizes $\delta(t-1) \cdot a_{ji} \cdot e_i(O_t)$. These values are used to reconstruct the most probable path. The final state on the most probable path, q_T^*, is the state that maximizes $\delta(T, j)$. The rest of the state path is reconstructed by tracing back through the dynamic programming matrix, a procedure similar to the traceback in pairwise sequence alignment.

The running time of the Viterbi algorithm is $O(TN^2)$. There are TN entries in the dynamic programming matrix. Each entry requires calculating N terms.

[1] Note that the most probable path is not the same as the path that maximizes the likelihood of O.

In class, we used the three-state HMM shown in Fig. 5.5 as an example. As an exercise, try applying the Viterbi algorithm to determine the most probable path through this model for the sequence `HHH`. A worksheet for this exercise is linked to the class syllabus page. The solution is also available. I recommend that you try to work through the Viterbi algorithm before looking at the solution.

5.5.2 The probability that state E_i emitted O.

The third question

3. What is the probability of being in state E_i when O_t is emitted?

is a special case of the decoding problem, where the focus is on classifying one specific residue. The probability of being in state E_i when O_t is emitted is the product of two probabilities: (1) the total probability of emitting $O_1 \dots O_t$ over all paths that end in E_i and (2) the total probability emitting $O_{t+1} \dots O_T$ over all paths, given that the model was in state E_i at time t:

$$P(q_t = E_i, O) = P(O_1, O_2, O_3, ...O_t, q_t = E_i) \cdot P(O_{t+1}, O_{t+2}, ...O_T | q_t = E_i). \tag{5.2}$$

Note that the first term is just $\alpha(t, i)$, as defined in the section on the Forward algorithm. To calculate the second term, we introduce $\beta(t + i)$, the probability of emitting $O_{t+1}, O_{t+2}, ...O_T$ given that O_t was emitted from state E_i:

$$\beta(t + 1, i) = P(O_{t+1}, O_{t+2}, ...O_T | q_t = E_i).$$

Substituting α and β for the first and second terms in Equation 5.2, we obtain the following expression for the probability of emitting O_t from state E_i

$$P(q_t = E_i, O_t) = \alpha(t, i) \cdot \beta(t + 1, i). \tag{5.3}$$

The first term, $\alpha(t, i)$, is calculated using the Forward algorithm. The second term, $\beta(t+1, i)$, is calculated using an algorithm called the Backward algorithm. Like the Forward and Viterbi algorithms, the Backward algorithm is a dynamic programming algorithm. However, the Backward algorithm is different in that we start by calculating the probability of emitting the last symbol, O_T, and then work backwards from O_T to O_{t+1}.

Algorithm: Backward

Initialization:

$$\beta(T, i) = \sum_{j=1}^{N} a_{ij} \cdot e_j(O_T)$$

Recursion:

$$\beta(t,i) = \sum_{j=1}^{N} a_{ij} \cdot e_j(O_t) \cdot \beta(t+1,j)$$

In addition to determining the probability that O_t was emitted from a given state, the Backward algorithm has several other applications. Although the Forward algorithm is usually used to calculate the probability of a sequence, O, the Backward algorithm can also be used for this purpose. To calculate the probability of the entire sequence, we use the Backward algorithm to calculate $\beta(t,i)$ recursively, starting with $\beta(2,i)$. The total probability of O is given by:

$$P(O) = \sum_{j=1}^{N} \pi_j e_j(O_1)\beta(2,j).$$

In motif discovery, both the Forward and the Backward algorithm are needed in order to learn parameters from unlabeled data using the Baum Welch procedure, which is a form of Expectation Maximization. The Backward algorithm is also used in another approach to inferring the true state path, called "Posterior decoding".

5.5.3 Posterior decoding

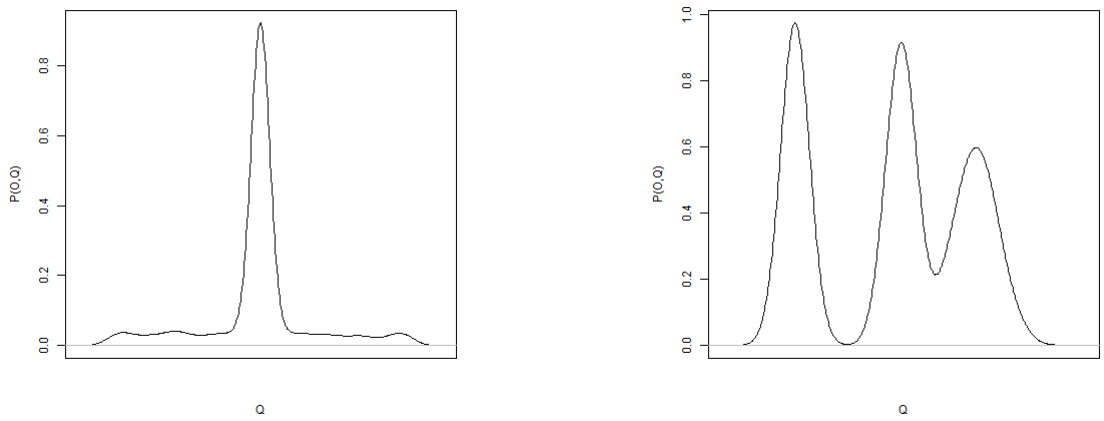
Let us revisit the question of estimating the path through an HMM that corresponds to the true labeling of the data. In Viterbi decoding, the *most probable path* is considered the best estimate of the true path. An alternative is to use $\hat{Q}$, the sequence of *most probable states*, as an estimate of the true path. This approach is referred to as *posterior decoding* because it is based on the posterior probability of emitting O_t from state i, when the emitted sequence is known. The most probable state at time t is the state that has the highest probability of emitting O_t when all possible state paths are considered:

$$\begin{aligned}
\hat{q}_t &= \underset{i}{\operatorname{argmax}}\, P(q_t = E_i, O_t) \\
&= \underset{i}{\operatorname{argmax}}\, P(O_1, O_2, O_3, ...O_t, q_t = E_i) \cdot P(O_{t+1}, O_{t+2}, ...O_T | q_t = E_i) \\
&= \underset{i}{\operatorname{argmax}}\, \alpha(t,i) \cdot \beta(t+1,i).
\end{aligned}$$

Note that the most probable state for emitting O_t is independent of the most probable state for any other symbol in O. In fact, the sequence of most probable states, $\hat{Q} = \hat{q}_1, \hat{q}_2, \ldots\ \hat{q}_T$ may not correspond to any legitimate path through the model.

Posterior decoding may give better results than Viterbi decoding in some cases, such as when suboptimal paths are almost as probable as the most probable path. If there is only one state path with high probability (e.g., Fig. 5.7a), then it is likely that Q^* and $\hat{Q}$ will

represent the same sequence of states. However, when there are two or more state paths with high probability (e.g., Fig. 5.7b), each of those paths contributes some information about the classification of each symbol, O_t. Posterior decoding takes advantage of the information encoded in all state paths, while Viterbi decoding does not.



(a) (b)

Figure 5.7: (a) The probability distribution of paths for a given sequence of symbols, O_1, for a hypothetical hidden Markov model. In this hypothetical case, the probability of the most probable path is much greater than the probability of all other paths. (b) The probability distribution of paths for a given sequence of symbols, O_2, for a hypothetical hidden Markov model. In this hypothetical case, there are several paths with relatively high probability. One of these is almost as probable as the most probable path

5.6 Summary

We started by introducing three recognition questions:

1. What is the probability that a given sequence, O, was generated by the HMM?
   Example: Is sequence O a transmembrane protein?

2. Given a sequence O, what is the true path? Otherwise stated, we wish to assign labels to an unlabeled sequence.
   Example: Identify the cytosolic, transmembrane, and extracellular regions in O. In this case, we wish to assign the labels E, M, or C to each amino acid residue in the

sequence.

3. What is the probability of being in state E_i when O_t is emitted?
   Example: Is a given residue localized to the membrane?

We then discussed several approaches to answering these questions:

- Calculating $P(O|\lambda)$ using the Forward or Backward algorithms
- Inferring the state path that emitted O using Viterbi or Posterior decoding
- Inferring the state that emitted O_t using the Forward and Backward algorithms

These tools can be used to answer biological questions in a variety of ways. For example, one approach to predicting whether O is a transmembrane protein is to calculate $P(O|\lambda_{TM})$, the probability that O was emitted by the transmembrane model. However, the resulting probability can be difficult to interpret. How big must the probability be to convince us that O is in fact a transmembrane sequence? To answer the question, it is useful to construct a model representing a null hypothesis and to calculate $P(O|\lambda_0)$, the probability that O was emitted by this null model. If the resulting likelihood ratio

$$\frac{P(O|\lambda_{TM})}{P(O|\lambda_0)}$$

is much greater than one, then we can infer that O is a transmembrane sequence.

An alternate approach would be to infer the state path that emitted O using the Viterbi or posterior decoding. If the resulting path includes membrane states, then we can conclude that O is a transmembrane sequence. If the entire sequence is labeled with C states or with E states, then we conclude that it is not.

5.7 Designing HMMs: Motif discovery and modeling

There are three major computational tasks associated with conserved motifs found in multiple sequences: Discovery, Modeling, and Recognition. In previous sections, we discussed the recognition problem: Given an HMM, how do we use it to ask questions about patterns in a new, unlabeled sequence? Here, 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.

