Using Hidden Markov Models to Align Multiple Sequences

David W. Mount

Cold Spring Harb Protoc; doi: 10.1101/pdb.top41

Topic Introduction

Using Hidden Markov Models to Align Multiple Sequences

David W. Mount

INTRODUCTION

A hidden Markov model (HMM) is a probabilistic model of a multiple sequence alignment (msa) of proteins. In the model, each column of symbols in the alignment is represented by a frequency distribution of the symbols (called a "state"), and insertions and deletions are represented by other states. One moves through the model along a particular path from state to state in a Markov chain (i.e., random choice of next move), trying to match a given sequence. The next matching symbol is chosen from each state, recording its probability (frequency) and also the probability of going to that state from a previous one (the transition probability). State and transition probabilities are multiplied to obtain a probability of the given sequence. The hidden nature of the HMM is due to the lack of information about the value of a specific state, which is instead represented by a probability distribution over all possible values. This article discusses the advantages and disadvantages of HMMs in msa and presents algorithms for calculating an HMM and the conditions for producing the best HMM.

RELATED INFORMATION

HMMs have been used very successfully for speech recognition, and an excellent review of the methodology is available (Rabiner 1989). In addition to their use in producing msas (Baldi et al. 1994; Krogh et al. 1994; Eddy 1995, 1996), HMMs are used extensively in sequence analysis to produce an HMM that represents a sequence profile (a profile HMM) to analyze sequence composition and patterns (Churchill 1989), to locate genes by predicting open reading frames, and to produce protein structure predictions. Pfam, a database of profiles that represent protein families, is based on profile HMMs (Sonnhammer et al. 1997).

Other approximate methods for global alignment of multiple sequences are discussed in **Using Iterative Methods for Global Multiple Sequence Alignment** (Mount 2008a) and **Using Progressive Methods for Global Multiple Sequence Alignment** (Mount 2008b). In **Comparing Programs and Methods to Use for Global Multiple Sequence Alignment** (Mount 2008c), additional alignment methods are introduced, and the utility of different methods is compared under various conditions. Programs that format and edit msas are presented in **Using Multiple Sequence Alignment Editors and Formatters** (Mount 2008d).

HIDDEN MARKOV MODELS

The HMM is a statistical model that considers all possible combinations of matches, mismatches, and gaps to generate an alignment of a set of sequences (Fig. 1). These models are primarily used for protein sequences to represent protein families or sequence domains, but they are also used to represent patterns in DNA sequences such as RNA splice junctions. Using a computer program designed for producing HMMs, a model of a sequence family, which takes into account the lengths of the sequences and accommodates insertions and deletions, is produced and initialized with prior information; that is, a guess of the expected variation in each position of the msa. The program then uses the sequences of a set of 20-100 sequences or more as data to train the model. The trained model may then be used to produce the most probable alignment of the sequences as posterior information. Alternatively,

Adapted from *Bioinformatics: Sequence and Genome Analysis*, 2nd edition, by David W. Mount. CSHL Press, Cold Spring Harbor, NY, USA, 2004.
Cite as: Cold Spring Harb Protoc; 2009; doi:10.1101/pdb.top41

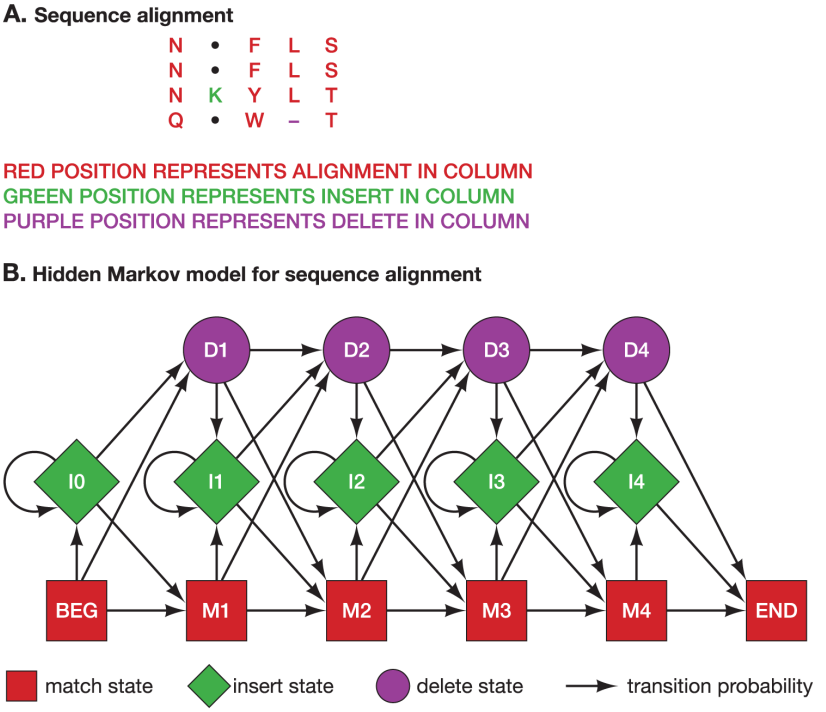

FIGURE 1. Relationship between the sequence alignment and the hidden Markov model of the alignment (for details, see Krogh et al. [1994]). This particular form for the HMM was chosen to represent the sequence, structural, and functional variation expected in proteins. The model accommodates the identities, mismatches, insertions, and deletions expected in a group of related proteins. (*A*) A section of an msa. The illustration shows the columns generated in an msa. Each column may include matches and mismatches (red positions), insertions (green positions), and deletions (purple positions). (*B*) The HMM. Each column in the model represents the possibility of a match, insert, or delete in each column of the alignment in *A*. The HMM is a probabilistic representation of a section of the msa. Sequences can be generated from the HMM by starting at the beginning state labeled BEG and then by following any one of many pathways from one type of sequence variation to another (states) along the state transition arrows and terminating in the ending state labeled END. Any sequence can be generated by the model, and each pathway has a probability associated with it. Each square match state stores an amino acid distribution such that the probability of finding an amino acid depends on the frequency of that amino acid within that match state. Each diamond-shaped insert state produces random amino acid letters for insertions between aligned columns, and each circular delete state produces a deletion in the alignment with probability 1. For example, one of many ways of generating the sequence **N K Y L T** in the above profile is by the sequence BEG → M1 → I1 → M2 → M3 → M4 → END. Each transition has an associated probability, and the sum of the probabilities of transitions leaving each state is 1. The average value of a transition would thus be 0.33, since there are three transitions from most states (there are only two from M4 and D4; hence, the average from them is 0.5). For example, if a match state contains a uniform distribution across the 20 amino acids, the probability of any amino acid is 0.05. Using these average values of 0.33 or 0.5 for the transition values and 0.05 for the probability of each amino acid in each state, the probability of the above sequence **N K Y L T** is the product of all of the transition probabilities in the path BEG → M → I1 → M2 → M3 → M4 → END, and the probability that each state will produce the corresponding amino acid in the sequences, or $0.33 \times 0.05 \times 0.33 \times 0.05 \times 0.33 \times 0.05 \times 0.33 \times 0.05 \times 0.33 \times 0.05 \times 0.5 = 6.1 \times 10^{-10}$. Since these probabilities are very small numbers, amino acid distributions and transition probabilities are converted to log odds scores, as done in other statistical methods, and the logarithms are added to give the overall probability score. The secret of the HMM is to adjust the transition values and the distributions in each state by training the model with the sequences. The training involves finding every possible pathway through the model that can produce the sequences, counting the number of times each transition is used and which amino acids were required by each match and insert state to produce the sequences. This training procedure leaves a memory of the sequences in the model. As a consequence, the model will be able to give a better prediction of the sequences. Once the model has been adequately trained, of all the possible paths through the model that can generate the sequence **N K Y L T**, the most probable should be the match-insert-3 match combination (as opposed to any other combination of matches, inserts, and deletions). Likewise, the other sequences in the alignment would also be predicted with highest probability as they appear in the alignment; that is, the last sequence would be predicted with highest probability by the path match-match-delete-match. In this fashion, the trained HMM provides an msa, such as shown in *A*. For each sequence, the objective is to infer the sequence of states in the model that generate the sequences. The generated sequence is a Markov chain because the next state is dependent on the current one. Because the actual sequence information is hidden within the model, the model is described as an HMM. (For color figure, see doi: 10.1101/pdb.top41 online at www.cshprotocols.org.)

the model may be used to search sequence databases to identify additional members of a sequence family. A different HMM is produced for each set of sequences.

Advantages and Disadvantages of HMMs

HMMs often provide an msa as good as, if not better than, other methods such as global alignment and local alignment methods, including profiles and scoring matrices. The approach also has several other strong features: It is well grounded in probability theory; no sequence ordering is required; guesses of insertion/deletion penalties are not needed; and experimentally derived information can be used. The disadvantage to using HMMs is that at least 20 sequences and sometimes many more are required to accommodate the evolutionary history of the sequences (see Mitchison and Durbin 1995). The HMM can be used to improve an existing heuristic alignment. The two HMM programs in common use are Sequence Alignment and Modeling Software System (SAM) (Krogh et al. 1994; Hughey and Krogh 1996) and HMMER (see Eddy 1998). The software is available at http://www.cse.ucsc.edu/research/compbio/sam.html and http://hmmer.janelia.org/, respectively. The algorithms used for producing HMMs are discussed extensively in Durbin et al. (1998).

The HMM representation of a section of msa that includes deletions and insertions was devised by Krogh et al. (1994) and is shown in Figure 1. This HMM generates sequences with various combinations of matches, mismatches, insertions, and deletions, and gives these a probability, depending on the values of the various parameters in the model. The object is to adjust these parameters so that the model represents the observed variation in a group of related protein sequences. A model trained in this manner will provide a statistically probable msa of the sequences.

One problem with HMMs is that the training set has to be quite large (50 or more sequences) to produce a useful model for the sequences. A difficulty in training the HMM residues is that many different parameters must be found (the amino acid distributions, the number and positions of insert and delete states, and the state transition frequencies add up to thousands of parameters) to obtain a suitable model, and the purpose of the prior and training data is to find a suitable estimate for all these parameters. When trying to make an alignment of short sequence fragments to produce a profile HMM, this problem is worsened because the amount of data for training the model is even further reduced.

Algorithms for Calculation of an HMM

As illustrated in Figure 1, the goal is to calculate the best HMM for a group of sequences by optimizing the transition probabilities between states and the amino acid compositions of each match state in the model. The sequences do not have to be aligned to use the method. Once a reasonable model length reflecting the expected length of the sequence alignment is chosen, the model is adjusted incrementally to predict the sequences. Several methods for training the model in this fashion have been described (Baldi et al. 1994; Krogh et al. 1994; Eddy et al. 1995; Eddy 1996; Hughey and Krogh 1996; Durbin et al. 1998). For example, the Baum-Welch algorithm, previously used in speech recognition methods, adjusts the parameters of HMMs for optimal matching to sequences, as discussed below.

This HMM is developed as follows:

1. The model is initialized with estimates of transition probabilities, the probability of moving from one state to another particular state in the model (e.g., the probability of moving from one match state to the next), and the amino acid composition for each match and insert state. If an initial alignment of the sequences is known or some other kinds of data suggest which sequence positions are the same, these data may be used in the model. For other cases, the initial distribution of amino acids to be used in each state is described below. The initial transition probabilities that are chosen generally favor transitions from one match state, a part of the model that represents one column in an msa, to the next match state, representing the next column. The alternative of using transitions to insert and delete states, which would delete a position or add another sequence character, is less favored because this builds more uncertainty into the HMM sequence model.

2. All possible paths through the model for generating each sequence in turn are examined. There are many possible such paths for each sequence. This procedure would normally require a huge amount of time computationally. Fortunately, an algorithm, the forward-backward algorithm,

reduces the number of computations to the number of steps in the model times the total length of the training sequences. This calculation provides a probability of the sequence, given all possible paths through the model, and, from this value, the probability of any particular path may be found. The Baum-Welch algorithm, referred to above, then counts the number of times a particular state-to-state transition is used and a particular amino acid is required by a particular match state to generate the corresponding sequence position.

3. A new version of the HMM is produced that uses the results found in Step 2 to generate new transition probabilities and match-insert state compositions.

4. Steps 3 and 4 are repeated up to 10 more times to train the model until the parameters do not change significantly.

5. The trained model is used to provide the most likely path for each sequence, as described in Figure 1. The algorithm used for this purpose, the Viterbi algorithm, does not have to go through all of the possible alignments of a given sequence to the HMM to find the most probable alignment, but instead can find the alignment by a dynamic programming technique very much like that used for the alignment of two sequences. The collection of paths for the sequences provides an msa of the sequences with the corresponding match, insert, and delete states for each sequence. The columns in the msa are defined by the match states in the HMM such that amino acids from a particular match state are placed in the same column. For columns that do not correspond to a match state, a gap is added.

6. The HMM may be used to search a sequence database for additional sequences that share the same sequence variation. In this case, the sum of the probabilities of all possible sequence alignments to the model is obtained. This probability is calculated by the forward component of the forward-backward algorithm described above in Step 2. This analysis gives a type of distance score of the sequence from the model, thus providing an indication of how well a new sequence fits the model and whether the sequence may be related to the sequences used to train the model. In later derivations of HMMs, the score was divided by the length of the sequence because it was found to be length-dependent. A *z*-score giving the number of standard deviations of the sequence length-corrected score from the mean length-corrected score is therefore used (Durbin et al. 1998).

Recall that for the Bayes block aligner, the initial or prior conditions were amino acid substitution matrices, block numbers, and alignments of the sequences. The sequences were then used as new data to examine the model by producing scores for every possible combination of prior conditions. By using Bayes' rule, these data provided posterior probability distributions for all combinations of prior information. Similarly, the prior conditions of the HMM are the initial values given to the transition values and amino acid compositions. The sequences then provide new data for improving the model. Finally, the model provides a posterior probability distribution for the sequences and the maximum posterior probability for each sequence represented by a particular path through the model. This path provides the alignment of the sequence in the msa; that is, the sequence plus matches, inserts, and deletes, as described in Figure 1.

Prior Conditions for an HMM

The success of the HMM method depends on having appropriate initial or prior conditions, that is, a good model for the sequences and a sufficient number of sequences to train the model. The prior model should attempt to capture, for example, the expected amino acid frequencies found in various types of structural and functional domains in proteins. As the distributions are modified by adding amino acid counts from the training sequences, new distributions should begin to reflect common patterns as one moves through the model and along the sequences. It is important that the model reflect not only the patterns in the training sequences, but also pattern variations that might be present in other members of the same protein family. Otherwise, the model will be overtrained and only recognize the training sequences but not other family members. Thus, some smoothing of the amino acid frequencies is desirable, but not to the extent of suppressing highly conserved pattern information from the training sequences. Such problems are avoided by using a method called "regularization" to avoid overfitting the data to the model. Basically, the method involves using a carefully designed amino acid distribution as the prior condition and then modifying this distribution in a manner that uses sequence information in the training sequences in a complementary manner to

build a model that represents the sequences but also includes a reasonable degree of variation as found in related sequences.

Other prior conditions may be used for match states in an HMM. For example, rather than using simple amino acid composition as a prior condition for the match states in the HMM, amino acid patterns that capture some of the important features of protein structure and function have been used with considerable success (Sjölander et al. 1996). Other prior conditions include using Dayhoff PAM or BLOSUM amino acid substitution matrices modified by adding additional counts (pseudocounts) to smooth the distributions (Tatusov et al. 1994; Eddy 1996; Henikoff and Henikoff 1996; Sonnhammer et al. 1997).

Dirichlet Mixtures Used as Prior Information

A particular set of amino acid substitutions, called "Dirichlet mixtures," have been prepared by Sjölander et al. (1996) to use as prior information in the match states of the HMM. These mixtures provide amino acid compositions that have proven to be useful for the detection of weak but significant sequence similarity. They are a method for weighting the prior distributions expected for several different multinomial distributions into a combined frequency distribution. Calculation of these mixtures is a complex mathematical procedure (Sjölander et al. 1996). Dirichlet mixtures recommended for use in aligning proteins by the HMM method have been described previously (Karplus 1995) and are available from http://www.cse.ucsc.edu/research/compbio/dirichlets/ in the open source libraries.

As an example of this approach, the amino acid frequencies that are characteristic of a particular set of nine blocks in the BLOCKS database have been determined as Dirichlet mixtures. These blocks represent amino acid frequencies that are favored in certain chemical environments such as aromatic, neutral, and polar residues and are useful for detecting such environments in test sequences. This nine-component system has been used successfully for producing an HMM for globin sequences (Hughey and Krogh 1996). To use these frequencies as prior information, they are treated as possible posterior distributions that could have generated the given amino acid frequencies as posterior probabilities. The probability of a particular amino acid distribution given a known frequency distribution (i.e., 100A, 67G, 5C, ... , where pA is the probability of A given by the frequency of A, pG the probability of G, ..., and n is the total number of amino acids given by the multinomial distribution) is

$$P(100\text{A}, 67\text{G}, 5\text{C}, \ldots) = n!\ p\text{A}^{100}\ p\text{G}^{67}\ p\text{C}^{5}\ \ldots/100!\ 67!\ 5!\ \ldots.$$

The prior distribution for the multinomial distribution is the Dirichlet distribution (Carlin and Louis 1996), whose formulation is similar to that given in the equation above with a similar set of parameters but with factorials and powers reduced by 1. The idea behind using this particular distribution is that if additional sequence data with a related pattern are added, then by the Bayesian procedure of multiplying prior probabilities with the likelihood of the new data to obtain the posterior distribution, the probability of finding the correct frequency of amino acids is favored statistically. Because the amino acid frequencies in the test sequences could be any one of several alternatives, a prior distribution that reflects these several choices is necessary. After the prior amino acid frequencies are in place in the match states of the model, these are modified by training the HMM with the sequences, as described in Steps 2 and 3 above. For each match state in the model, a new frequency for each amino acid is calculated by dividing the sum of all new and prior counts for that amino acid by the new total of all amino acids. In this fashion, the new HMM (Step 4 above) reflects a combination of expected distributions averaged over patterns in the Dirichlet mixture and patterns exhibited in the training sequences. A similar method is used to refashion the transition probabilities in the HMM during training following manual insertion of initial values.

Number of Sequences Used for Training the HMM

Another consideration in using HMMs for msas is the number of sequences being aligned. If a good prior model such as the above Dirichlet distribution is used, it should be possible to train the HMM with as few as 20 sequences (SAM manual; see http://www.cse.ucsc.edu/research/compbio/sam.html) (Eddy 1996; Hughey and Krogh 1996). In general, the smaller the sequence number, the more important the prior conditions. If the number of sequences is ~50, the initial conditions play a lesser role because the training step is more effective. As with any msa method, the more sequence diversity, the more challenging the task of aligning sequences with HMMs. HMMs are also more effective if methods to inject statistical noise, for example, simulated annealing described below, into the model are used during the training procedure. As the model is refashioned to fit the sequence data,

it sometimes goes into a form that provides locally optimal instead of globally optimal alignments of the sequences. One of several noise injection methods (Baldi et al. 1994; Krogh et al. 1994; Eddy et al. 1995; Eddy 1996; Hughey and Krogh 1996) may be used in the training procedure. One method, called simulated annealing, is used by SAM (Hughey and Krogh 1996). A user-defined number of sequences are generated from the model at each cycle, and the counts so generated are added to those from the training sequences. The noise generated in this way is reduced as the cycle number is increased. Finally, the HMM program SAM has a built-in feature of "model surgery" during training. If a match state is used by fewer than half of the sequences, it is deleted. These same sequences then have to use an insert state in the revised model. Similarly, if an insert state is used by more than half of the sequences, a number of additional match states equal to the average number of insertions is added, and the model has to be revised accordingly. These fractions may be varied in SAM to test the effect on the type of HMM model produced (Hughey and Krogh 1996).

PRODUCING THE BEST HMM

In trying to produce an HMM for a set of related sequences (see Fig. 2), the recommended procedure is to produce several models by varying the prior conditions. Using regularization by adding prior

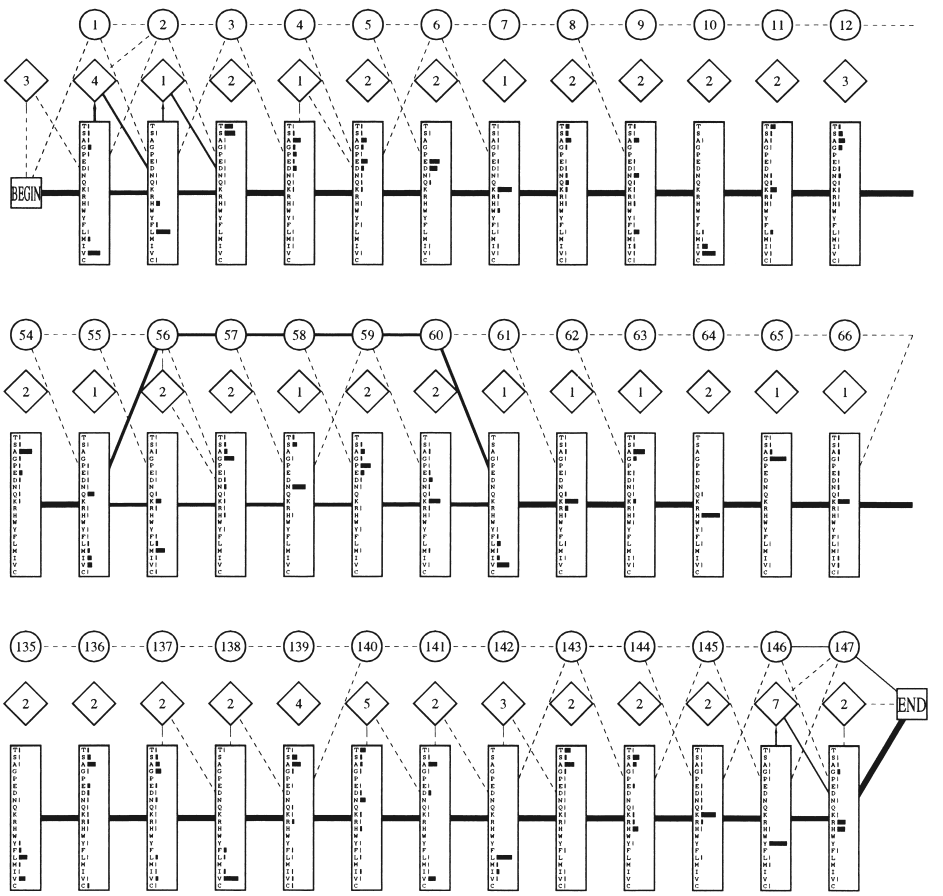

FIGURE 2. HMM trained for recognition of globin sequences. Circles in the top row are delete states that include the position in the alignment; the diamonds in the second row are insert states showing the average length of the insertion; and the rectangles in the bottom row show the amino acid distribution in the match states: V is common at match position 1, L at 2, and so on. The width of each transition line joining these various states indicates the extent of use of that path in the training procedure, and dotted lines indicate a rarely used path. The most used paths are between the match states, but about one-half of the sequences use the delete states at model positions 56-60. Thus, for most of the sequences, the msa or profile will show the first two columns aligned with a V followed by an L, but at 56-60, about one-half of the sequences will have a 5-amino-acid deletion. (Reprinted from Krogh et al. 1994, with permission from Elsevier © 1994.)

Dirichlet mixtures to the match states produces models more representative of the protein family from which the training sequences are derived. Varying the noise and model surgery levels is another way to vary the training procedure and the HMM model. The best HMM model is the one that predicts a family of related sequences with the narrowest distribution of probability scores. An example of a portion of an HMM trained on a set of globin sequences is shown in Figure 2.

REFERENCES

Baldi P, Chauvin Y, Hunkapillar T, McClure MA. 1994. Hidden Markov models of biological primary sequence information. *Proc Natl Acad Sci* **91:** 1059–1063.

Carlin BP, Louis TA. 1996. In *Bayes and empirical Bayes methods for data analysis (Monographs on statistics and applied probability)* (eds. DR Cox et al.), Chapman and Hall, New York.

Churchill GA. 1989. Stochastic models for heterogeneous DNA sequences. *Bull Math. Biol* **51:** 79–94.

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

Eddy SR. 1995. Multiple alignment using hidden Markov models. *ISMB* **3:** 114–120.

Eddy SR. 1996. Hidden Markov models. *Curr Opin Struct Biol* **6:** 361–365.

Eddy SR. 1998. Profile hidden Markov models. *Bioinformatics* **14:** 755–763.

Eddy SR, Mitchison G, Durbin R. 1995. Maximum discrimination hidden Markov models of sequence consensus. *J Comput Biol* **2:** 9–23.

Henikoff JG, Henikoff S. 1996. Using substitution probabilities to improve position-specific scoring matrices. *Comput Appl Biosci* **12:** 135–143.

Hughey R, Krogh A. 1996. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *Comput Appl Biosci* **12:** 95–107.

Karplus K. 1995. Regularizers for estimating the distributions of amino acids from small samples. *UCSC Technical Report (UCSC-CRL-95-11)*. University of California, Santa Cruz.

Krogh A, Brown M, Mian IS, Sjölander K, Haussler D. 1994. Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol* **235:** 1501–1531.

Mitchison GJ, Durbin RM. 1995. Tree-based maximal likelihood substitution matrices and hidden Markov models. *J Mol Evol* **41:** 1139–1151.

Mount DW. 2008a. Using iterative methods for global multiple sequence alignment. *Cold Spring Harb Protoc* (this issue). doi: 10.1101/pdb.top44.

Mount DW. 2008b. Using progressive methods for global multiple sequence alignment. *Cold Spring Harb Protoc* (this issue). doi: 10.1101/pdb.top43.

Mount DW. 2008c. Comparing programs and methods to use for global multiple sequence alignment. *Cold Spring Harb Protoc* (this issue). doi: 10.1101/pdb.ip61.

Mount DW. 2008d. Using multiple sequence alignment editors and formatters. *Cold Spring Harb Protoc* (this issue). doi: 10.1101/pdb.top45.

Rabiner LR. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE* **77:** 257–286.

Sjölander K, Karplus K, Brown M, Hughey R, Krogh A, Mian IS, Haussler D. 1996. Dirichlet mixtures: A method for improved detection of weak but significant protein sequence homology. *Comput Appl Biosci* **12:** 327–345.

Sonnhammer EL, Eddy SR, Durbin R. 1997. Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins* **28:** 405–420.

Tatusov RL, Altschul SF, Koonin EV. 1994. Detection of conserved segments in proteins: Iterative scanning of sequence databases with alignment blocks. *Proc Natl Acad Sci* **91:** 12091–12095.