Multiple alignment using hidden Markov models

Sean R. Eddy
Dept. of Genetics, Washington University School of Medicine
660 S. Euclid Box 8232, St. Louis, MO 63110
eddy@genetics.wustl.edu

Abstract

A simulated annealing method is described for training hidden Markov models and producing multiple sequence alignments from initially unaligned protein or DNA sequences. Simulated annealing in turn uses a dynamic programming algorithm for correctly sampling suboptimal multiple alignments according to their probability and a Boltzmann temperature factor. The quality of simulated annealing alignments is evaluated on structural alignments of ten different protein families, and compared to the performance of other HMM training methods and the ClustalW program. Simulated annealing is better able to find near-global optima in the multiple alignment probability landscape than the other tested HMM training methods. Neither ClustalW nor simulated annealing produce consistently better alignments compared to each other. Examination of the specific cases in which ClustalW outperforms simulated annealing, and vice versa, provides insight into the strengths and weaknesses of current hidden Markov model approaches.

Introduction

Hidden Markov models (HMMs) are useful as a formal, fully probabilistic form of profiles (Baldi *et al.* 1994; Eddy, Mitchison, & Durbin 1995; Krogh *et al.* 1994; Stultz, White, & Smith 1993). Profiles are statistical models of protein structure consensus (Barton 1990; Bashford, Chothia, & Lesk 1987; Bowie, Luthy, & Eisenberg 1991; Gribskov, McLachlan, & Eisenberg 1987; Taylor 1986). They have been applied to the protein fold recognition problem with encouraging success (Shortle 1995). Successful fold recognition could go a long way towards a pragmatic solution of the protein folding problem at low resolution, since about 80% of new protein structures with no easily recognized sequence similarity to previous structures adopt an already known fold (Orengo, Jones, & Thornton 1994).

Profiles are complicated models involving thousands of free parameters. An attraction of HMM-based approaches is the possibility of exploiting probability theory in choosing optimal model parameters. HMMs, for instance, wield a mathematically consistent description of insertions and deletions, a traditionally problematic area. HMMs also offer theoretical insight into the difficulties of combining disparate forms of information, such as sequences and three-dimensional structures.

The prerequisite for building a sensitive profile of a protein family is a multiple alignment of a large number of evolutionarily divergent sequences, in which all structurally homologous positions have been identified and aligned. However, accurate alignment is only possible for proteins of known structure – at least for an identifiable core of residues that comprises the secondary structure elements and active site of the molecule (Chothia & Lesk 1986) – and far more sequences than structures are known. It is therefore routinely necessary to infer accurate multiple alignments from sequence information alone. Interestingly, one of the features of HMMs is that it is possible to train models from initially unaligned sequences, thus producing HMM-based multiple alignments (Baldi *et al.* 1994; Krogh *et al.* 1994). Since accurate sequence-based alignment is essentially a limited structure prediction problem, there would seem to be some benefit in applying powerful fold recognition methods during the multiple alignment process, as well as afterwards.

Existing algorithms for training hidden Markov models from initially unaligned example sequences are hill-climbing algorithms, such as gradient descent (Baldi *et al.* 1994) or expectation maximization (Krogh *et al.* 1994). These methods seek a local optimum in a probability landscape of possible multiple alignments by iteratively refining an initial guess at the alignment or the model parameters. Like most hill-climbing algorithms, they are prone to getting stuck in incorrect local optima far from the global optimum. Early results with HMMs indicated that this was a serious problem. Much better optima and much better alignments could be found by starting training off with a good (manual) alignment. A partly successful attempt at avoiding unsatisfactory local optima was to

slightly randomize model parameters at each iteration ("noise injection") (Krogh *et al.* 1994). In this paper, I have pursued this idea and developed it into a fuller realization of simulated annealing (Kirkpatrick, Gelatt, & Vecchi 1983).

Methods and algorithms

HMMs

An HMM is composed of a number of interconnected *states*, each of which emits an observable output symbol (such as a single amino acid). Each state has two kinds of parameters. *Symbol emission probabilities* are the probabilities of emitting each possible symbol from a state. *State transition probabilities* are the probabilities of moving from the current state to a new state. A sequence is generated by starting at an initial state and moving from state to state until a terminal state is reached, emitting symbols from each state that is passed through. The state sequence is a first-order Markov chain. This state sequence is "hidden" and only the symbol sequence it emits is observable – hence the term hidden Markov model (Rabiner 1989).

Figure 1 diagrams the structure of a hidden Markov model for modeling primary sequence consensus information derived from multiple sequence alignments, as introduced by Krogh *et al.* (1994). One *match* state is assigned to each consensus column of the multiple alignment. *Insert* states insert extra symbols relative to the consensus match states, and *delete* states allow skipping consensus positions. Standard dynamic programming algorithms are used to align HMMs to sequences and calculate a likelihood for the alignment.

Sampling suboptimal HMM alignments

Simulated annealing HMM training relies on sampling suboptimal multiple alignments. The degree of suboptimality in the sampled multiple alignment is controlled through a Boltzmann temperature factor, kT. The higher kT is, the more random the alignment. At the limit $kT = \infty$, all possible alignments are sampled equiprobably. At the limit $kT = 0$, only the most likely alignment (Viterbi path) is sampled. An assumption is made that all the example sequences are independent emissions from the HMM. Then a suboptimal multiple alignment is correctly sampled by picking a suboptimal alignment for each individual sequence.

For notational simplicity, this sampling algorithm is given in terms of a general HMM alignment case. A sequence consensus HMM has a special case of non-emitting delete states. Delete states create the possibility of more than one state transition between two symbols. With minor obvious rearrangements, the algorithm also applies to sequence consensus HMMs.

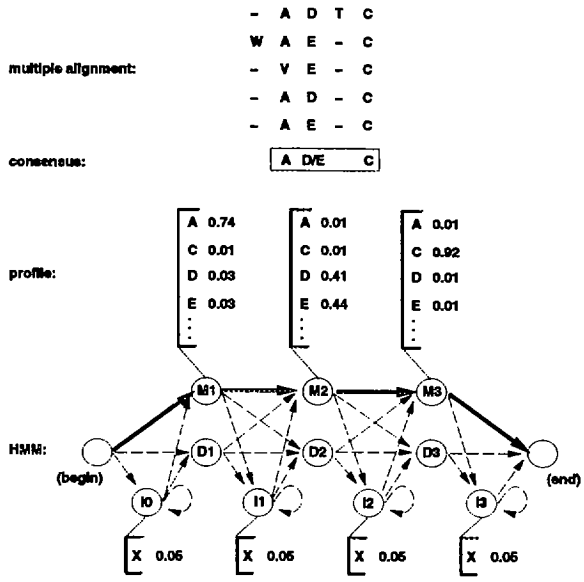

Figure 1: Comparison of consensus modeling methods, from simple patterns to an HMM. At the top is a toy multiple sequence alignment showing both conserved columns and insertions. A consensus sequence derived from this alignment is shown. A sequence profile extends a simple pattern by the use of variable amino acid scores (and variable gap penalties) at each consensus position. An HMM extends a profile by replacing arbitrary scores with probabilities, yielding a full probabilistic model of mismatches, insertions, and deletions relative to a consensus.

The model M consists of N states denoted $S_1 ... S_N$. An initial probability distribution of occupying state S_i is given by π_i; typically, this is 1.0 for a special dummy begin state and zero elsewhere. An observable sequence O consists of T symbols denoted $O_1 ... O_T$, from an alphabet consisting of symbols x. The parameters of the model consist of an NxN matrix of state transition probabilities a_{ij} for a transition from S_i to S_j, and an array of symbol emission probabilities $b_j(x)$ for emission of a symbol x from each state S_j. The state occupied at time t in an alignment is denoted as q_t. A complete alignment Q of a sequence O is given by $Q = q_1, q_2 ... q_T$.

Formally stated, the goal is to sample an alignment Q from a Boltzmann distribution according to the actual likelihood of the alignment Q given the current model, $P(Q, O \mid M)$, and the Boltzmann temperature factor kT:

$$\text{Prob}(Q) = \frac{P(Q, O \mid M)^{\frac{1}{kT}}}{\sum_{\text{all } Q^\dagger} P(Q^\dagger, O \mid M)^{\frac{1}{kT}}}$$

By analogy to statistical mechanics, the summation

over all possible alignments in the denominator is Z, the *partition function*. Z is calculated using the forward algorithm (Rabiner 1989), using *exponentiated* parameters $\hat{\pi}_i = \pi_i^{\frac{1}{kT}}$, $\hat{a}_{ij} = a_{ij}^{\frac{1}{kT}}$, and $\hat{b}_j(x) = b_j(x)^{\frac{1}{kT}}$ in place of the unmodified probability parameters:

$$\begin{aligned}
&\text{Initialization:} \\
&\quad F_1(i) = \hat{\pi}_i \hat{b}_i(O_1) \\
&\text{Induction:} \qquad \text{for } 2 \leq t \leq T, 1 \leq j \leq N \\
&\quad F_t(i) = \left[\sum_{j=1}^{N} F_{t-1}(j)\hat{a}_{ji}\right] \hat{b}_i(O_t) \\
&\text{Termination:} \\
&\quad Z = \sum_{i=1}^{N} F_T(i)
\end{aligned}$$

A suboptimal path Q is then selected from this dynamic programming lattice by a probabilistic traceback. The alignment consists of a series of states q_t which are recursively chosen such that:

$$\begin{aligned}
&\text{Initialization:} \\
&\quad \text{Prob}(q_T) = \frac{F_T(q_T)}{Z} \\
&\text{Induction:} \qquad \text{for } T \geq t \geq 2 \\
&\quad \text{Prob}(q_{t-1} \mid q_t) = \frac{F_{t-1}(q_{t-1})\hat{a}_{q_{t-1},q_t}}{\sum_{i=1}^{N} F_{t-1}(i)\hat{a}_{i,q_t}}
\end{aligned}$$

Exponentiated parameters are pre-calculated for computational efficiency. Machine precision difficulties arise from the multiplication of small values. Scaling parameters are employed to keep matrix values within the dynamic range of the computer (Rabiner 1989). A viable alternative would be to store the forward matrix scores F as logarithms and compute sums in log space.

A similar procedure has been described by Allison and Wallace (1993). Allison and Wallace exponentiate by $1/kT$ during the traceback instead of during the lattice fill stage. This is not strictly correct. Consider the high temperature case where one wants to choose between all possible paths equiprobably. The Allison/Wallace algorithm chooses between all possible next *states* in a probabilistic traceback equiprobably. The number of paths passing through a state varies. For instance, many more paths pass through the states in the center of the dynamic programming lattice than through the states down its edges. Choosing equiprobably among possible states during the traceback greatly over-favors more unique paths down the edges. An initial HMM simulated annealing implementation using the Allison/Wallace algorithm had a noticeable problem with alignments being biased towards insert- and delete-rich (edge) paths.

Simulated annealing

This suboptimal alignment algorithm is used as the heart of an HMM training algorithm based on simulated annealing, a common strategy for complex optimization problems (Kirkpatrick, Gelatt, & Vecchi 1983). The procedure is a variant of a standard HMM training procedure, the Viterbi approximation to Baum-Welch expectation maximization. Instead of determining an optimal multiple alignment with respect to the current model at the expectation step of each iteration, a suboptimal multiple alignment is sampled. Training is started at high kT with very randomized alignments and slowly cooled. Well-determined motifs in the alignment "freeze" first, followed by finer details. The procedure should be able to jump out of obviously bad local optima. Note, however, that a new alignment is chosen from the probability distribution over alignments given the *previous* model, *a la* the expectation step of expectation maximization, not by the probability of the alignment according to its optimal model. The procedure is therefore not wholly faithful to the formal statistical mechanical basis of simulated annealing (Kirkpatrick, Gelatt, & Vecchi 1983).

In all experiments, a simple annealing schedule was used. kT was started at 5 and reduced by 5% at each iteration. Convergence was monitored by measuring the relative overall change in parameter values between a previous and a new model, and stopping below an arbitrary threshold. If kT cooled below 0.1, training was reverted to the standard Viterbi training procedure (i.e. $kT = 0$) because serious numerical precision difficulties arise at very low temperature. I have not systematically explored different annealing schedules, and I suspect that more efficient schedules can be devised. For instance, the rate of change of $\log Z$ with respect to temperature might be used to monitor "phase changes", adaptively slowing the temperature ramp whenever the model begins to "freeze" (Kirkpatrick, Gelatt, & Vecchi 1983).

Availability

Source code and documentation for HMMER, the HMM package used here, is available by anonymous ftp from `ftp://genome.wustl.edu/pub/eddy` or from `http://genome.wustl.edu/eddy/hmm.html` on the World Wide Web. The code is ANSI C and is portable among various UNIX platforms.

Results

The ability of HMM simulated annealing to produce accurate multiple alignments was tested on ten protein families. HMM simulated annealing performance was compared to the performance of two other HMM train-

Family	len	ali	%id	homo	%id
alpha amylase	516	3	36%	106	15%
(multi) EF hand	150	5	30%	228	26%
cytochrome C	109	9	42%	114	56%
EGF domain	48	4	31%	432	26%
globin	152	18	27%	656	39%
homeodomain	66	3	23%	302	42%
Ig light chain V	121	23	53%	738	24%
insulin	84	4	39%	125	37%
protease inhibitors	56	6	40%	140	41%
kringle modules	85	4	36%	108	51%

Table 1: Ten protein families selected as test cases. For each family is given the consensus length of the alignment (len), the number of structurally aligned sequences (ali), the percentage sequence identity averaged over all sequence pairs in the structural alignment (first %id column), the number of homologues identified in and extracted from SwissProt 30 by an automated search (homo), and the average percentage sequence identity in the set of homologues (second %id column).

ing algorithms (Baum-Welch expectation maximization, and the Viterbi approximation to Baum-Welch) and to one of the best heuristic multiple alignment programs, ClustalW (Thompson, Higgins, & Gibson 1994). These experiments address the following questions:

- Is there a problem with local optima in standard HMM training?
- If so, can simulated annealing find better optima?
- Do mathematically better optima correlate with better alignments?
- How do HMM methods compare to the current state of the art in multiple alignment?

An objective operational definition of alignment accuracy was used in interpreting the results. An overall alignment identity between a test alignment and a trusted alignment was calculated by counting the number of symbol pairs (or symbol/gap pairs) in the test alignment which are aligned identically in the trusted alignment, divided by the total number of aligned symbol pairs in the trusted alignment. It must be emphasized that even structure-based alignments are somewhat ambiguous, so any single measure of accuracy is necessarily crude. The disadvantage of an overall measure is that, for distantly related proteins, only a specific core of residues is meaningfully alignable (Chothia & Lesk 1986). For this reason, a second approach was

Family	Vit	B-W	SA	SA10	CW
aa	297.04	379.06	453.30	464.58	388.03
	37%	42%	69%	70%	81%
cbp	196.39	211.94	213.74	214.74	193.58
	50%	80%	78%	81%	61%
cytc	261.65	260.28	264.36	265.07	262.58
	85%	88%	88%	89%	87%
egf	53.07	54.96	55.46	57.17	47.36
	83%	74%	80%	88%	75%
glob	203.98	237.33	250.82	255.93	253.31
	38%	69%	81%	75%	93%
hom	132.74	131.71	133.58	134.31	133.91
	68%	91%	50%	61%	96%
igvar-l	105.33	118.84	120.13	122.37	108.60
	97%	96%	93%	94%	95%
ins	113.36	123.57	132.79	133.37	123.67
	72%	72%	91%	90%	89%
kazal	94.61	110.28	109.69	110.56	107.37
	34%	85%	81%	88%	92%
kringle	217.25	216.78	217.89	218.59	216.01
	83%	80%	82%	80%	86%

Table 2: Alignment accuracy and HMM log likelihood scores achieved by various alignment methods. The log-odds score in bits (log base 2) is given for an HMM built from the final alignment of 100 sequences. Alignment accuracy is calculated relative to the trusted Sali/Overington structural alignment for the subset of sequences with known 3D structures. Methods used are Viterbi (Vit) or Baum-Welch (B-W) expectation maximization, a single simulated annealing run (SA), the best-scoring run of ten simulated annealing runs (SA10), or ClustalW (CW).

also used in which only those columns in the trusted alignment that corresponded to consensus secondary structure elements were counted towards the alignment identity score. In these experiments, the key column approach gave measures which were strongly correlated with the overall measures. Because of this, and because definition of key columns was somewhat arbitrary, only the overall alignment identity measure is reported here.

A collection of protein sequence alignments based on three-dimensional structural information was obtained from John Overington (Sali & Overington 1994). A few (globins and immunoglobulin fold subfamilies) were checked against expert manual structural alignments provided by Cyrus Chothia. These alignment identity scores generally ranged from 90% to 98%. This gives some idea of the inherent ambiguity between different structural alignments. In one case, an alignment of a pair of very distantly related immunoglobulin C2

domains (14% sequence identity) was only 60% identical to the expert manual alignment, even in consensus structure regions. Therefore there is danger in trusting any one structural alignment absolutely. Nonetheless, the Overington structural alignments were trusted as correct for the purposes of comparison. Care was taken to monitor alignments subjectively by eye as well.

Hidden Markov models are a statistical modeling technique, and as such their performance increases as the number of training sequences increases. The structural alignments in the collection each contain few sequences (2 to 23). Therefore, additional sequence homologues from the SwissProt 30 sequence database (Bairoch & Boeckmann 1993) were isolated. Homologues of the sequences in the Sali and Overington alignments were identified with BLASTP(Altschul *et al.* 1990). The first ten families with three or more aligned structures and 100 or more known homologues were selected as test cases (Table 1).

For each family, a test set of unaligned sequences of 100 sequences was created. Each set contained all the sequences from the structural alignment and enough randomly selected homologues to bring the total number to 100 sequences. These ten test sets were aligned using HMM training by simulated annealing, Baum-Welch expectation maximization, and the Viterbi approximation to Baum-Welch, and also using ClustalW. An additional experiment was done in which ten different simulated annealing runs were done and the highest-scoring run was saved. The homologues were filtered back out, and the resulting alignments of known structures were compared to the original Sali/Overington structural alignments. These data are summarized in Table 2.

Discussion

Simulated annealing HMM training is able to find mathematically better optima than other HMM training methods tested here. It should be noted, though, that gradient descent training was not tested (Baldi *et al.* 1994), nor was full Baum-Welch with noise injection tested (Krogh *et al.* 1994). Table 2 indicates that there are clearly local optimum problems, especially for the Viterbi approximation of Baum-Welch expectation maximization, as indicated by low-scoring solutions with poor alignment qualities. (Multiple runs from random start points partially alleviate such problems, but not completely.) Full Baum-Welch is significantly less prone to local optima problems, but ClustalW is sometimes able to find higher-scoring alignments than Baum-Welch training.

In many cases, better optima (in terms of log likelihood) were found to correspond with improved alignments. However, this was not always the case. The slight non-correlation of score with alignment quality was previously obscured by the dominance of the local optimum problem in HMM training. It seems from these results that further research might focus on the log likelihood objective function itself. One promising attack is to incorporate more information about amino acids and protein structure in the form of more sophisticated Bayesian priors (Brown *et al.* 1993). For instance, the implementation of HMMs used here includes no knowledge about amino acid substitution probabilities (e.g. PAM matrices).

Overall, simulated annealing HMM training appears to compare favorably with the well-established multiple alignment program ClustalW (Thompson, Higgins, & Gibson 1994). In particular, two fairly difficult alignments – EGF domains and EF-hands – seem to have matched well with the strengths of HMMs, and were significantly better aligned by HMM simulated annealing than by ClustalW. As an example, the EGF structural alignment is shown in Figure 2 along with the ClustalW and the HMM simulated annealing alignments. These alignments are probably the most insertion and deletion prone families in the test. One advantage of HMMs over other alignment methods is a consistent theory for insertion and deletion penalties. This could be the reason for the relative quality of these two alignments.

On the other hand, the careful heuristics in ClustalW seem to have given it an edge for three of the ten families tested here – alpha amylases, globins, and homeodomains. ClustalW works by progressive alignment of most similar sequences first. In the alpha amylase test, the three test structures were 36% identical, while the homologue set was only 15% identical on average. Because of the progressive alignment strategy, ClustalW aligns the three similar alpha amylase structures without confusion from the other relatively dissimilar TIM-barrel relatives in the test set. The rest of the alignment is in fact quite poor. Since the score is calculated over all 100 test sequences, but alignment identity is evaluated on the subset of known structures, this is probably also the reason for the large non-correlation between score and quality in the alpha amylases. It is interesting that the other two families where ClustalW had a significant edge also have large disparities between the sequence identity of the structures and the sequence identity of the homologues. For the globins and the homeodomains, the homologue set is more similar on average than the structures are. The reason for this is a strong bias in these sequence families towards particular highly similar subfamilies, such as the overwhelming dominant α and β globins

EGF domain structural alignment:

ClustalW alignment:

HMM simulated annealing alignment:

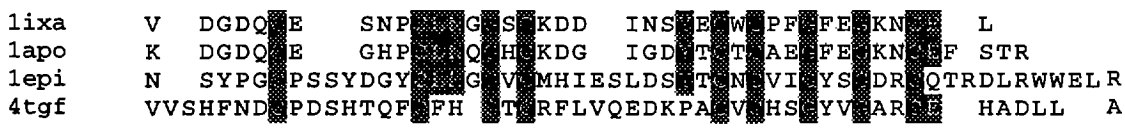

Figure 2: Alignments of four epidermal growth factor (EGF) domains of known three-dimensional structure. The Sali/Overington structure-based alignment is at the top, followed by the ClustalW and HMM simulated annealing alignments (extracted from alignments of 100 total test sequences). Residues in columns with $\geq 75\%$ identity are highlighted.

in the globin family. ClustalW weights the example sequences to compensate for this, and HMM training does not (yet). It seems promising to try to cast some of the careful heuristics of programs like ClustalW into the HMM framework.

Ultimately, the goal is to actually use these methods, not just to compare them against each other. Large collections of protein structural alignments are now available (Holm & Sander 1994; Sali & Overington 1994), and a complete hierarchical classification of protein structure folds, superfamilies, and families is maintained on the World Wide Web (Murzin *et al.* 1995). The ability of HMMs to do sensitive fold recognition is becoming apparent (Shortle 1995) and HMM methods are rapidly improving. The tools are essentially now in hand to produce a library of HMMs for protein fold recognition based on all the currently known fold families and their SwissProt homologues. It will be interesting to apply these methods and threading methods to the analysis of the thousands of new predicted protein sequences that are coming out of large-scale genome sequencing projects.

Acknowledgments

I thank Richard Durbin, Graeme Mitchison, Cyrus Chothia, David MacKay, Erik Sonnhammer, Tim Hubbard, and Gos Micklem for their many ideas and suggestions that contributed to this work. Thanks also to John Overington for providing the database of structural alignments. I gratefully acknowledge support from a long-term Human Frontier Science Program postdoctoral fellowship, and NIH National Research Service Award 1-F32-GM16932-01 from the NIGMS.

References

Allison, L., and Wallace, C. 1993. The posterior probability distribution of alignments and its application to parameter estimation of evolutionary trees and to optimisation of multiple alignments. Technical Report TR 93/188, Monash University Computer Science.

Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; and Lipman, D. J. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215:403–410.

Bairoch, A., and Boeckmann, B. 1993. The SWISS-PROT protein sequence data bank, recent developments. *Nucl. Acids Res.* 21:3093–3096.

Baldi, P.; Chauvin, Y.; Hunkapiller, T.; and McClure, M. A. 1994. Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA* 91:1059–1063.

Barton, G. J. 1990. Protein multiple sequence alignment and flexible pattern matching. *Meth. Enzymol.* 183:403–427.

Bashford, D.; Chothia, C.; and Lesk, A. M. 1987. Determinants of a protein fold: Unique features of the

globin amino acid sequences. *J. Mol. Biol.* 196:199–216.

Bowie, J. U.; Luthy, R.; and Eisenberg, D. 1991. A method to identify protein sequences that fold into a known three-dimensional structure. *Science* 253:164–170.

Brown, M.; Hughey, R.; Krogh, A.; Mian, I. S.; Sjolander, K.; and Haussler, D. 1993. Using dirichlet mixture priors to derive hidden Markov models for protein families. In Hunter, L.; Searls, D.; and Shavlik, J., eds., *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, 47–55. Menlo Park, CA: AAAI.

Chothia, C., and Lesk, A. 1986. The relation between the divergence of sequence and structure in proteins. *EMBO J.* 5:823–826.

Eddy, S. R.; Mitchison, G.; and Durbin, R. 1995. Maximum discrimination hidden Markov models of sequence consensus. J. Comput. Biol., in press.

Gribskov, M.; McLachlan, A. D.; and Eisenberg, D. 1987. Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA* 84:4355–4358.

Holm, L., and Sander, C. 1994. The FSSP database of structurally aligned protein fold families. *Nucl. Acids Res.* 22:3600–3609.

Kirkpatrick, S.; Gelatt, C.; and Vecchi, M. 1983. Optimization by simulated annealing. *Science* 220:671–680.

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.

Murzin, A.; Brenner, S. E.; Hubbard, T.; and Chothia, C. 1995. SCOP: A structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol., in press.

Orengo, C.; Jones, D. T.; and Thornton, J. M. 1994. Protein superfamilies and domain superfolds. *Nature* 372:631–634.

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

Sali, A., and Overington, J. P. 1994. Derivation of rules for comparative protein modeling from a database of protein structure alignments. *Protein Sci.* 3:1582–1596.

Shortle, D. 1995. Protein fold recognition. *Nature Struct. Biol.* 2:91–93.

Stultz, C. M.; White, J. V.; and Smith, T. F. 1993. Structural analysis based on state-space modeling. *Protein Sci.* 2:305–314.

Taylor, W. R. 1986. Identification of protein sequence homology by consensus template alignment. *J. Mol. Biol.* 188:233–258.

Thompson, J.; Higgins, D.; and Gibson, T. 1994. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties, and weight matrix choice. *Nucl. Acids Res.* 22:4673–4680.