Hidden Markov Models in Computational Biology: Applications to Protein Modeling UCSC-CRL-93-32

Anders Krogh*†, Michael Brown†, I. Saira Mian§, Kimmen Sjölander†, David Haussler†

† Computer and Information Sciences

§ Sinsheimer Laboratories

University of California, Santa Cruz, CA 95064, USA.

email: krogh@nordig.ei.dth.dk, haussler@cse.ucsc.edu

August 17, 1993

Keywords: Hidden Markov Models, Multiple Sequence Alignments, Database Searching, Globin, Kinase, EF-hand, EM algorithm.

Abstract

Hidden Markov Models (HMMs) are applied to the problems of statistical modeling, database searching and multiple sequence alignment of protein families and protein domains. These methods are demonstrated on the globin family, the protein kinase catalytic domain, and the EF-hand calcium binding motif. In each case the parameters of an HMM are estimated from a training set of unaligned sequences. After the HMM is built, it is used to obtain a multiple alignment of all the training sequences. It is also used to search the SWISS-PROT 22 database for other sequences that are members of the given protein family, or contain the given domain. The HMM produces multiple alignments of good quality that agree closely with the alignments produced by programs that incorporate three-dimensional structural information. When employed in discrimination tests (by examining how closely the sequences in a database fit the globin, kinase and EF-hand HMMs), the HMM is able to distinguish members of these families from non-members with a high degree of accuracy. Both the HMM and PROFILESEARCH (a technique used to search for relationships between a protein sequence and multiply aligned sequences) perform better in these tests than PROSITE (a dictionary of sites and patterns in proteins). The HMM appears to have a slight advantage

*Present address: Electronics Institute, Build 349, Technical University of Denmark, 2800 Lyngby, Denmark

over PROFILESEARCH in terms of lower rates of false negatives and false positives, even though the HMM is trained using only unaligned sequences, whereas PROFILESEARCH requires aligned training sequences. Our results suggest the presence of an EF-hand calcium binding motif in a highly conserved and evolutionarily preserved putative intracellular region of 155 residues in the α-1 subunit of L-type calcium channels which play an important role in excitation-contraction coupling. This region has been suggested to contain the functional domains that are typical or essential for all L-type calcium channels regardless of whether they couple to ryanodine receptors, conduct ions or both.

1 Introduction

The rate of generation of sequence data in recent years provides abundant opportunities for the development of new approaches to problems in computational biology. In this paper, we apply *hidden Markov models* (HMMs) to the problems of statistical modeling, database searching, and multiple alignment of protein families and protein domains. To demonstrate the method, we examine three protein families. Each family consists of a set of proteins that have the same overall three-dimensional structure but widely divergent sequences. Features of the sequences that are determinants of folding, structure and function should be present as conserved elements in the family of sequences. We consider the globins, whole proteins ranging in length from 130 to 170 residues (with few exceptions) and two domains, the protein kinase catalytic domain (250-300 residues) and the EF-hand calcium-binding motif (29 residues). The same approach can be used to model families of nucleic acid sequences as well (Krogh *et al.*, 1993).

A hidden Markov Model (Rabiner, 1989) describes a series of observations by a "hidden" stochastic process — a Markov process. In speech recognition, where HMMs have been used extensively, the observations are sounds forming a word, and a model is one that by its "hidden" random process generates these sounds with high probability. Every possible sound sequence can be generated by the model with some probability. Thus, the model defines a probability distribution over possible sound sequences. A good word model would assign high probability to all sound sequences that are likely utterances of the word it models, and low probability to any other sequence. In this paper we propose an HMM similar to the ones used in speech recognition to model *protein families* such as globins and kinases. In speech recognition, the "alphabet" from which words are constructed could be the set of phonemes valid for a particular language; in protein modeling, the alphabet we use is the 20 amino acids from which protein molecules are constructed. Where the observations in speech recognition are words, or strings of phonemes, in protein modeling the observations are strings of amino acids forming the primary sequence of a protein. A model for a set of proteins is one that assigns high probability to the sequences in that particular set.

The HMM we build identifies a set of positions that describe the (more or less) conserved first-order structure in the sequences from a given family of proteins. In biological terms, this corresponds to identifying the core elements of homologous molecules. The model provides additional information, such as the probability of initiating an insertion at any position in the model and the probability of extending it. The structure of the model is similar to that of a *profile* (Waterman & Perlwitz, 1986; Barton & Sternberg, 1990; Gribskov *et al.*, 1990; Bowie *et al.*, 1991; Lüthy *et al.*, 1991), but slightly more general. Once we have built the model from unaligned sequences, we can generate a multiple alignment of the sequences using a dynamic programming method. By employing it for database searching, the model can be used to discriminate sequences that belong to a given family from non-members. Finally, we can study the model we have found directly, and see what it reveals about the common structure underlying the various sequences in the family.

Our method of multiple alignment differs quite markedly from conventional techniques, which are usually based on pairwise alignments generated by dynamic programming schemes (Waterman, 1989; Feng & Doolittle, 1987; Barton, 1990; Subbiah & Harrison, 1989)). The alignments produced by these methods often depend strongly on the particular values of the parameters required by the method, in particular the gap penalties (Vingron & Argos, 1991). Furthermore, a given set of sequences is likely to possess both fairly conserved regions and highly variable regions, yet conventional global methods assign identical penalties for all regions of the sequences. Substitutions, insertions, or deletions in a region of high conservation should ideally be penalized more than in a variable region, and some kinds of substitutions should be penalized differently in one position compared to another. That is one of the motivations for the present work. The statistical model we propose corresponds to multiple alignment with *variable, position-dependent* gap penalties. Furthermore, these penalties are in large part learned from the data itself. Essentially, we build a statistical model during the process of multiple alignment, rather than leaving this as a separate task to be done after the alignment is completed. We believe the model should guide the alignment as much as the alignment determines the model.

We are not the first group to employ hidden Markov models in computational biology. Lander and Green (1987) used hidden Markov models in the construction of genetic linkage maps. Other work employed HMMs to distinguish coding from non-coding regions in DNA (Churchill, 1989). Later, simple HMMs were used in conjunction with the EM algorithm to model certain protein-binding sites in DNA (Lawrence & Reilly, 1990; Cardon & Stormo, 1992) and, more recently, to model the N-caps and C-caps of alpha helices in proteins (D. Morris, unpublished). These applications of HMMs and the EM (Expectation-Maximization) algorithm, including our own, presage a more widespread use of this technique in computational biology. During the time that we have been developing this approach, several related efforts have come to our attention. One is that of White, Stultz and Smith (1991) (1993), who use HMMs to model protein superfamilies. This work is more ambitious than our own, since superfamilies are harder to characterize than families. It is not yet clear how successful their work has been since no results are reported for sequences not in the training set. If there are weaknesses in their method, it is possible that these are due to the use of handcrafted models and reliance on prealigned data for parameter estimation. In contrast, our models have a simple regular structure, and we are able to estimate all the parameters of these models, including the size of the model directly from unaligned training sequences. Interestingly enough, they independently propose an alternate HMM state structure similar to ours in section 6.3 of their paper,[1] where they discuss the relationship of their work to (1991), but they do not pursue this further. It is possible that the type of models we use may work better for characterizing superfamilies than those investigated by White *et al.* However, it is more likely that they are too simple, and that richer and more varied state structure along the lines they propose is required for this problem. We recently found that Asai, Hayamizu and Onizuka have applied HMMs to the problem of predicting the secondary structure of proteins, obtaining prediction rates that are competitive with previous methods in some cases (Asai, K. and Hayamizu, S. and Onizuka, K., 1993). In addition, Tanaka, Ishikawa, Asai, and Konagaya also discuss the relationship between the HMM method for obtaining multiple alignments and previous methods (Tanaka *et al.*, 1993). Finally, in work

[1]Instead of using delete states, they have direct transitions between each pair of match states m_i and m_j with $i < j$.

most closely related to our own, since the time we presented a preliminary report on this work (Haussler & Krogh, 1992) (see also (Haussler *et al.*, 1992)), Baldi *et al.* have further demonstrated the usefulness of this technique by producing multiple alignments for immunoglobins and protease as well as globins and kinases (Baldi *et al.*, 1993).[2]

2 Methods

2.1 HMM Architecture

Consider a family of protein sequences that all have a common three-dimensional structure, for example the globins. The common structure in these sequences can be defined as a sequence of positions in space where amino acids occur. In the case of globins, whose structure contains principally α-helices, the 150 or so helical positions have been named A1, A2, ..., A16, B1, ... etc., where the letter denotes the α-helix, and the number indicates the location within that α-helix (see for example (Bashford *et al.*, 1987)). For each of these positions there is a (distinct) probability distribution over the 20 amino acids that measures the likelihood of each amino acid occurring in that position in a typical globin, as well as the probability that there is no amino acid in that position (*i.e.*, that a sequence belonging to this family may have a gap at that position in a multiple alignment). These have been called *profiles* (Waterman & Perlwitz, 1986; Barton & Sternberg, 1990; Gribskov *et al.*, 1990; Bowie *et al.*, 1991; Lüthy *et al.*, 1991). A profile of globins can be thought of as a statistical model for the family of globins, in that for any sequence of amino acids, it defines a probability for that sequence, in such a way that globin sequences tend to have much higher probabilities than non-globin sequences.

The type of hidden Markov model we use as a statistical model for a protein family can be viewed as a generalized profile. However, instead of describing the HMM directly in terms of the probability it assigns to each protein sequence, we find that it is easier to first think of an HMM as a structure that generates protein sequences by a random process. This structure and corresponding random process is illustrated in Figure 1 and can be described as follows.

The main line of the HMM contains a sequence of M states, which we call match states, corresponding to positions in a protein or columns in a multiple alignment (M equals 4 in figure 1). Each of these states can generate a letter x from the 20-letter amino acid alphabet according to a distribution $\mathcal{P}(x|\mathtt{m}_k)$, $k = 1 \ldots M$. The notation $\mathcal{P}(x|\mathtt{m}_k)$ means that each of the match states $\mathtt{m}_k$, $1 \le k \le M$, have distinct distributions. For each match state $\mathtt{m}_k$, there is a delete state $\mathtt{d}_k$ that does not produce any amino acid but is a "dummy" state used to skip $\mathtt{m}_k$. Finally, there are a total of $M+1$ insert states to either side of the match states which generate amino acids in exactly the same way as the match states, but use probability distributions $\mathcal{P}(x|\mathtt{i}_k)$. In Figure 1, match, delete and insert states are shown as boxes, circles and diamonds respectively. For convenience, we have added a dummy "BEGIN" state and a dummy "END" state, denoted $\mathtt{m}_0$ and $\mathtt{m}_{M+1}$ respectively, which do not produce any amino

[2]They have developed a variant of the method described here that employs a gradient descent training algorithm in place of the EM algorithm.

x	Amino acid.
s	Sequence of amino acids ($s = x_1 \ldots x_2$).
L	Length of sequence.
q, r	State in HMM.
path	A sequence of states, $q_1 \ldots q_N$.
N	Number of states in a path.
M	Length of model.
$\mathtt{m}, \mathtt{i}, \mathtt{d}$	Match, insert, and delete states.
$\mathtt{m}_0, \mathtt{m}_{M+1}$	Begin and end states.
$\mathcal{P}(x\|q)$	Probability distribution of amino acids in state q.
$\mathcal{T}(r\|q)$	Probability of a transition from state q to r.

Table 1: Notation.

acid.

From each state, there are three possible transitions to other states, also shown in Figure 1. Transitions into match or delete states always move forward in the model, whereas transitions into insert states do not. Note that multiple insertions between match states can occur, since the self-loop on the insert state allows a transition from the insert state to itself. The transition probability from state q to state r is called $\mathcal{T}(r|q)$. Our notation is summarized in Table 1.

A sequence can be generated by a "random walk" through the model as follows: Commencing at state $\mathtt{m}_0$ (BEGIN), choose a transition to $\mathtt{m}_1$, $\mathtt{d}_1$, or $\mathtt{i}_0$ randomly according to the probabilities $\mathcal{T}(\mathtt{m}_1|\mathtt{m}_0)$, $\mathcal{T}(\mathtt{d}_1|\mathtt{m}_0)$, and $\mathcal{T}(\mathtt{i}_0|\mathtt{m}_0)$. If $\mathtt{m}_1$ is chosen, generate the first amino acid x_1 from the probability distribution $\mathcal{P}(x|\mathtt{m}_1)$, and choose a transition to the next state according to probabilities $\mathcal{T}(\cdot|\mathtt{m}_1)$, where "$\cdot$" indicates any possible next state. If this next state is the insert state $\mathtt{i}_1$, then generate amino acid x_2 from $\mathcal{P}(x|\mathtt{i}_1)$ and select the next state from $\mathcal{T}(\cdot|\mathtt{i}_1)$. If delete ($\mathtt{d}_2$) is chosen next, generate no amino acid, and choose the next state from $\mathcal{T}(\cdot|\mathtt{d}_2)$. Continue in this manner all the way to the "END" state, generating a sequence of amino acids $x_1, x_2 \ldots x_L$ by following a *path* of states $q_0, q_1 \ldots q_N, q_{N+1}$ through the model, where $q_0 = \mathtt{m}_0$ (the BEGIN state) and $q_{N+1} = \mathtt{m}_{M+1}$ (the END state). Because the delete states do not produce any amino acid, N is larger than or equal to L. If q_i is a match or insert state, we define $l(i)$ to be the index in the sequence $x_1 \ldots x_L$ of the amino acid produced in state q_i. The probability of the event that the path $q_0 \ldots q_{N+1}$ is taken and the sequence $x_1 \ldots x_L$ is generated is

$$\mathrm{Prob}(x_1 \ldots x_L, q_0 \ldots q_{N+1}|\mathrm{model}) = \mathcal{T}(\mathtt{m}_{N+1}|q_N) \times \prod_{i=1}^{N} \mathcal{T}(q_i|q_{i-1}) \mathcal{P}(x_{l(i)}|q_i), \tag{1}$$

where we set $\mathcal{P}(x_{l(i)}|q_i) = 1$ if q_i is a delete state. The probability of any sequence $x_1 \ldots x_L$ of amino acids is a sum over all possible paths that could produce that sequence, which we write as follows:

$$\mathrm{Prob}(x_1 \ldots x_L|\mathrm{model}) = \sum_{\text{paths } q_0, \ldots, q_{N+1}} \mathrm{Prob}(x_1 \ldots x_L, q_0 \ldots q_{N+1}|\mathrm{model}). \tag{2}$$

In this way a probability distribution on the space of sequences is defined. The goal is to find a model (*i.e.*, a proper model length and probability parameters) that accurately describes a family of proteins by assigning large probabilities to sequences in that family.

This particular structure for the HMM was chosen because it is the simplest model that captures the structural intuition of a protein: a) a sequence of positions, each with its own distribution over the amino acids; b) the possibility for either skipping a position or inserting extra amino acids between consecutive positions; and c) allowing for the possibility that continuing an insertion or deletion is more likely than starting one. This choice appears to have worked well for modeling the protein families that we have examined, but other types of HMMs may be better at other tasks, *e.g.*, the more elaborate models for protein superfamilies used in (White *et al.*, 1991; Stultz *et al.*, 1993). The important feature of the HMM method is its generality. One can choose any structure for the states and transitions that is appropriate for the problem at hand. Examples of more general HMM architectures are given in Sections 2.4 and 2.5 below.

2.2 Estimating the parameters of an HMM from training sequences

All the parameters in the HMM (*i.e.*, the transition probabilities and the amino acid distributions) could in principle be chosen by hand from an existing alignment of protein sequences, as in (Gribskov *et al.*, 1990; White *et al.*, 1991; Stultz *et al.*, 1993), or from information about the three-dimensional structure of proteins, as in (Bowie *et al.*, 1991; White *et al.*, 1991; Stultz *et al.*, 1993). The novel approach we take is to "learn" the parameters entirely automatically from a set of *unaligned* primary sequences, using an EM algorithm. This approach can in principle find the model that best describes a given set of sequences.

Given a set of training sequences $s(1), \ldots, s(n)$, one can see how well a model fits them by calculating the probability that it generates them. This probability is simply a product of terms of the form given by (2), *i.e.*

$$\mathrm{Prob}(\mathrm{sequences}|\mathrm{model}) = \prod_{j=1}^{n} \mathrm{Prob}(s(j)|\mathrm{model}), \tag{3}$$

where each term $\mathrm{Prob}(s(j)|\mathrm{model})$ is calculated by substituting $x_1 \ldots x_L = s(j)$ in eq. (2). This is called the *likelihood* of the model. One would like this value to be high. The *maximum likelihood* (ML) method of model estimation is to find the model that maximizes the likelihood (3).

An alternate approach to ML estimation is the *maximum a posteriori* (MAP) approach. Here, we assume a *prior* probability distribution over all possible parameters of the model embodying prior beliefs on what a model should be like. This can then be used to "penalize" models that are known to be bad or uninteresting. We discuss this further in Section 2.8. In MAP estimation, we try to maximize the *posterior* probability of the model given the sequences. Using Bayes rule, the posterior probability can be calculated as

$$\mathrm{Prob}(\mathrm{model}|\mathrm{sequences}) = \frac{\mathrm{Prob}(\mathrm{sequences}|\mathrm{model})\mathrm{Prob}(\mathrm{model})}{\mathrm{Prob}(\mathrm{sequences})}. \tag{4}$$

Here Prob(model) is the prior probability distribution, and Prob(sequences) can be viewed as a normalizing constant. Since this normalizing constant is independent of the model, MAP estimation is equivalent to maximizing

$$\text{Prob}(\text{sequences}|\text{model})\text{Prob}(\text{model}). \tag{5}$$

over all possible models. The MAP approach is closely related to minimum description length (Jurka & Milosavljevic, 1991) and minimum message length (Allison *et al.*, 1992) methods.

There is no known efficient way to directly calculate the best HMM model either in the ML or MAP sense. However, there are algorithms that given an arbitrary starting point find a local maximum by iteratively re-estimating the model in such a way that the likelihood (or the posterior probability) increases in each iteration. The most common one is the Baum-Welch or forward-backward algorithm (Rabiner, 1989; Lawrence & Reilly, 1990), which is a version of the general EM method often used in statistics (Dempster *et al.*, 1977). The process of the EM algorithm can be viewed as an iterative adaptation of the model to fit the training sequences. The steps in this process can be summarized as follows.

1. An initial model is created by assigning values to the transition probability $\mathcal{T}(r|q)$ and the amino acid generation probability $\mathcal{P}(x|q)$ for each x, q and r, where x is one of the 20 amino acids and q and r are states in the HMM connected by a transition arc. If one already knows some features present in the sequences, or constraints on the sequences, these may sometimes be encoded in the initial model. The current model is set to this initial model.

2. Using the current model, all possible paths for each training sequence are considered in order to get a new estimate $\hat{\mathcal{T}}(r|q)$ of the transition probability $\mathcal{T}(r|q)$ and a new estimate $\hat{\mathcal{P}}(x|q)$ of the amino acid generation probability $\mathcal{P}(x|q)$ for each x, q and r. The transition probability estimate $\hat{\mathcal{T}}(r|q)$ is obtained by counting the number of times a transition is made from state q to r, for all paths of all training sequences, weighted by the probability of the path. The estimate $\hat{\mathcal{P}}(x|q)$ is made in a similar manner, by counting the number of times the amino acid x is aligned to the state q.

3. In the next step of maximum likelihood estimation, a new current model is created by simply replacing $\mathcal{T}(r|q)$ by $\hat{\mathcal{T}}(r|q)$ and $\mathcal{P}(x|q)$ by $\hat{\mathcal{P}}(x|q)$ for each x, q and r. In MAP EM estimation, the parameters $\hat{\mathcal{T}}(r|q)$ and $\hat{\mathcal{P}}(x|q)$ are further modified by considering the prior probability of the model before they are used to replace the old parameters.

4. Step 2 and 3 are repeated until the parameters of the current model change only insignificantly.

Since the quality of the current model (as measured by (3) or (5)) increases in each iteration, and no model is arbitrarily good, the process eventually terminates and produces a model that is, at least locally, the best model for the training sequences to within some specified precision of the parameters (Dempster *et al.*, 1977). Typically, this occurs very rapidly (*e.g.* in less than ten iterations) even for large models and large sets of training sequences.

The main computational bottleneck in the algorithm is step 2, since individually examining each possible path for every training sequence would generally take time exponential in the length of the longest training sequence. However, it is possible to use a dynamic programming technique known as the *forward-backward* procedure to speed up this step. Using this method, the new parameter estimates can be calculated in time proportional to the number of states in the model multiplied by the total length of all the training sequences. Details are given in the excellent tutorial article on HMMs by Rabiner (1989).

The forward part of the forward-backward procedure can also be used to efficiently compute $-\log \mathrm{Prob}(\mathrm{sequence}|\mathrm{model})$, the negative logarithm of the probability of a sequence given the model (as defined in (2)), without summing over all possible paths for the sequence (Rabiner, 1989). We call this the *negative log likelihood (NLL)-score* of the sequence. The average NLL-score of a training sequence is inversely related to the likelihood of the model, given by (3), and hence serves as a numerical measure of progress for each iteration of the EM procedure. The NLL-score can also be used to evaluate how well the model fits a novel "test" sequence not present in the training set, as described in Section 2.6 below.

2.3 The Viterbi algorithm and multiple alignment from an HMM

The forward-backward procedure is related to the dynamic programming technique used to align one sequence to another, or more generally to align a sequence to a profile. A variant of the forward-backward procedure known as the *Viterbi algorithm* is similar to the standard profile alignment algorithm (Waterman & Perlwitz, 1986; Barton & Sternberg, 1990; Gribskov *et al.*, 1990). Instead of calculating the NLL-score for a sequence, which implicitly involves all possible paths for that sequence through the model, the Viterbi algorithm computes the negative logarithm of the probability of the single most likely path for the sequence. We can write this as

$$-\log \max_{paths} \mathrm{Prob}(s, \mathrm{path}|\mathrm{model}), \tag{6}$$

where $\mathrm{Prob}(s, \mathrm{path}|\mathrm{model})$ is given in (1), with $s = x_1 ... x_L$ and path $= q_0 ... q_{N+1}$. Instead of first maximizing the probability of the path and then taking the negative logarithm, it is convenient (and equivalent) to simply minimize the negative logarithm of the probability over all paths. This minimum we will call the *distance* from the sequence to the model,

$$\begin{aligned}
\mathrm{dist}(s, \mathrm{model}) &= \min_{paths}\{-\log \mathrm{Prob}(s, \mathrm{path}|\mathrm{model})\} \\
&= \min_{paths} \sum_{i=1}^{N+1} [-\log \mathcal{T}(q_i|q_{i-1}) - \log \mathcal{P}(x_{l(i)}|q_i)].
\end{aligned}$$

This distance from a sequence to a model is analogous to the standard "edit distance" from one sequence to another (with gap penalties), see *e.g.* (Waterman, 1989), but is perhaps more related to the distance from a sequence to a profile. The term $-\log \mathcal{P}(x_{l(i)}|q_i)$ represents a penalty for aligning the amino acid $x_{l(i)}$ to the position represented by state q_i in the model. The term $-\log \mathcal{T}(q_i|q_{i-1})$ corresponds to a penalty for using the transition from q_{i-1} to q_i in the model. If this is a transition from a match state to a delete state, then this represents a gap-initiation penalty; if it is from a delete state to a delete state it represents a gap-extension penalty; if it is from a match state to an insert state, it represents an insertion-initiation penalty; and if it is a transition from an insertion state to itself (a "self-loop"), then it represents an insertion extension penalty. One of the main features of this distance measure is that all these penalties depend on the position in the model, whereas they would be fixed in most standard pairwise alignment methods. Often the most likely path has a significantly higher probability than all other paths, and in that case the distance defined here will be approximately equal to the NLL-score defined earlier.

The computation time for the Viterbi algorithm is proportional to the number of states in the model multiplied by the length of the sequence being aligned, *i.e.* the same as the time for the forward-backward algorithm. In addition, with a simple extension to the algorithm, the most probable path itself can be found using the usual backtracking technique (Rabiner, 1989). This is the method we use to obtain our multiple alignments: each sequence is aligned to the model by the Viterbi algorithm, after which the mutual alignment of the sequences among themselves is then determined. [3]

2.4 Using the HMM to cluster sequences and discover subfamilies

When a relatively large number of sequences are available, it is sometimes possible to obtain improved results by dividing these sequences into clusters of similar sequences and training a different HMM for each cluster/subfamily. The results of this are illustrated in more detail in Section 3.1. Given a large set of unlabeled and unaligned sequences, a simple extension of the hidden Markov model enables us to use the EM training algorithm to automatically partition the sequences into clusters of similar sequences. By iteratively splitting clusters, this method might be useful for building phylogenetic trees in a "top-down" manner. However, when the clusters become too small there will be an insufficient number of sequences in each cluster to construct an accurate model, so some "bottom-up" processing may still be necessary.

In order to discover w clusters in the data, we make w copies of the HMM, one for each cluster. We call these *components* of the (composite) HMM. Presently, the number w of clusters and the initial lengths of the models for these clusters are determined empirically. We then add a new begin state with w outgoing transitions, one to each of the begin states of the component HMMs (see Figure 2).

This new begin state is analogous to the other begin states in that it generates no amino acid. We then train this composite model with the EM algorithm as described in Section 2.2. The EM reestimation of a component model is the same as the reestimation of a single model, except that the weight that a sequence has in the reestimation of a component is proportional to the probability of the sequence given that component model. Thus, sequences that have better NLL-scores for a particular HMM component have greater influence in reestimating the parameters of that component, and this causes the parameters of that component to change in such a way that the component further "specializes" in modeling those sequences. The "surgery" procedure described below in Section 2.7 is used to adapt the length of that

[3] We make no attempt to align portions of the sequence that use the insert states of the model.

component to further specialize it. In this manner, the individual components evolve during training to represent clusters in the training sequences. This way of using EM is called mixture modeling in the statistics literature (Duda & Hart, 1973; Everitt & Hand, 1981), and is known as '(soft) competitive learning' in the neural network literature (Nowlan, 1990).

When the model is trained, the probability of a sequence given any of the submodels can be calculated, *i.e.*, the probability that the sequence belongs to the corresponding cluster/subclass. The negative logarithm of this probability corresponds to the NLL-score calculated for a simple HMM. As with the standard HMM we use, this yields a quantitative measure of how well the model fits the data. The clusters found can also be compared to known subfamilies of the sequences. Experiments with the clustering of globin sequences are described in Section 3.1.

2.5 Modeling protein domains with an HMM

There are many cases when one does not want to build a statistical model of a family of *whole* proteins like globins, but instead to build a model of a *structural motif* or *domain* that occurs as a subsequence in many different kinds of proteins, such as the EF-hand motif (Nakayama *et al.*, 1992) or the kinase catalytic domain (Hanks & Quinn, 1991). Here we expect our model to only match a relatively small subsequence of any given protein, with many other unmatched amino acids appearing before and after this subsequence. One approach to this problem is to alter the dynamic programming method used to align a sequence to a model so that it tries all possible ways of aligning each subsequence of the sequence to a model (Waterman, 1989). We use a simpler (but almost equivalent) method in which only the HMM model is altered, so that the same standard procedures (forward-backward and Viterbi) which we use for models of whole proteins can be used without modification for models of domains.

Consider a training set of many unaligned sequences consisting not of complete proteins, but of a specific domain. Our first step is to train an HMM for these sequences exactly as described earlier. As shown in Figure 1, this HMM will have initial and final "dummy" match states $\mathtt{m}_0$ and $\mathtt{m}_{N+1}$ (where $N+1=5$ in Figure 1) that do not match any amino acid. To alter the HMM to represent a protein domain, we create two new insert states $\mathtt{i}_B$ and $\mathtt{i}_E$, adding $\mathtt{i}_B$ to the model before the state $\mathtt{m}_0$ and $\mathtt{i}_E$ at the end of the model after $\mathtt{m}_{N+1}$ (see Figure 3).

We then add a new dummy "BEGIN" state before $\mathtt{i}_B$ and a new dummy "END" state after $\mathtt{i}_E$. Eight new transitions are also added to the model. The first four are from "BEGIN" to $\mathtt{i}_B$, from $\mathtt{m}_{N+1}$ to $\mathtt{i}_E$, and the self-loops from $\mathtt{i}_B$ to itself and from $\mathtt{i}_E$ to itself. These all have the same probability p, for some p between 0 and 1. The second four transitions are from $\mathtt{i}_B$ to $\mathtt{m}_0$, from "BEGIN" to $\mathtt{m}_0$, from $\mathtt{i}_E$ to "END", and from $\mathtt{m}_{N+1}$ to "END. These all have probability $1-p$. The new states added before and after the model, along with these transitions, form two new modules, one for matching the extra amino acids that occur in the sequence before the domain, and the other for matching the amino acids after the domain.

The choice of the parameter p does affect the way that the overall model aligns with a given sequence. To see how, it is convenient to think of the negative logarithm of the probability of a transition as a penalty for using that transition, as described in Section 2.3.

In the modified model, all sequences must suffer a penalty of $-\log(1-p)$ to enter and again to exit the domain part of the model, no matter which path they take. Hence this penalty is a fixed cost, which can be ignored when comparing the distances or NLL-scores of two sequences with respect to the model. In addition to this penalty, all sequences will suffer a penalty of $K(-\log p + \log 20)$, where $K \geq 0$ is the number of amino acids that are not matched to the original domain model, but are instead matched in the states $\mathtt{i}_B$ and $\mathtt{i}_E$. The $-\log 20$ term arises because we set the probabilities of each amino acid to $(1/20)$ in the insertion states $\mathtt{i}_B$ and $\mathtt{i}_E$ (see Section 2.8). Thus p will determine the 'pressure' on the sequence to align something to the domain model, *i.e.*, if p is low it advantageous to squeeze many amino acids into the domain model, using the insert states in this part of the model. If it is high, it is possible that most sequences would prefer to pass through the delete states in the domain model, aligning everything instead to the new modules before and after it. It is straightforward to estimate p the same way as all the other parameters, the only additional problem is that the same value must be used in all the transitions that use this value, 'tying' these parameters to each other. Otherwise the model might become biased towards aligning the domain either near the beginning of the sequence or near the end of the sequence. We have not attempted to estimate p. Rather, we have used a fixed $p = 1$ with good results. (This should be thought of as a limit of p approaching 1, otherwise $-\log(1-p)$ is infinite.)

Using this construction, it may also be possible to discover interesting domains by training on whole protein sequences, and letting EM determine which part of the proteins to model. Furthermore, if more than one occurrence of the same domain is expected in some sequences, then this model can be further modified to find all occurrences. This is accomplished by simply adding a transition from the "END" state back to the "BEGIN" state.

2.6 Searching a database with an HMM

Once an HMM is built for a family of proteins, it can be used to search a database such as PIR or SWISS-PROT for other proteins in this family. Similarly, if an HMM is built for a protein domain or motif, then it can be used to search for occurrences of this domain or motif in the database, much like a PROSITE expression (Bairoch, 1992), a commonly used method for searching for patterns found in protein sequences. Like a profile (Waterman & Perlwitz, 1986; Barton & Sternberg, 1990; Gribskov *et al.*, 1990; Bowie *et al.*, 1991; Lüthy *et al.*, 1991), an HMM has an advantage over a PROSITE expression for database searches, because it takes into account a large amount of statistical information in matching a sequence, and weighs this information appropriately, rather than relying on relatively rigid matching rules.

As described in Section 2.2, the forward part of the forward-backward dynamic programming method calculates a NLL-score for any test sequence that measures how well it fits the model. This NLL-score is the negative logarithm of the probability of the sequence given the model. It turns out that this raw NLL-score is too dependent on the length of the test sequence to be used directly to decide if the sequence is in the family modeled by the HMM or not. However, we can overcome this problem by normalizing this NLL-score appropriately.

Whenever we build an HMM for a family of proteins or for a protein domain, we run all the proteins in a standard database (for instance, SWISS-PROT) through this HMM and compute the NLL-score for each sequence. A scatter plot of sequence length versus

NLL-score for our kinase catalytic domain model is given in Figure 9.

Most proteins tend to lie on a fairly straight line (towards the top of the plot) indicating that the NLL-score for these proteins is proportional to their lengths. These proteins are the ones that do not contain the kinase catalytic domain and thus look like "random proteins" to the kinase catalytic domain model. In contrast, the proteins that *do* contain the kinase catalytic domain tend to have NLL-scores that are much lower than expected for proteins of their length, and hence appear below the linear band of non-kinase proteins.

We can quantify the difference between NLL-scores for proteins containing the kinase catalytic domain and NLL-scores for proteins *not* containing the domain by a simple statistical method, as follows. Using a local windowing technique,[4] we first calculate a smooth average curve for the roughly linear band of the NLL-score versus length plot. The standard deviation around this average curve is also calculated. Using this, we calculate the difference between the NLL-score of a sequence and the average NLL-score of typical sequences of that same length, measured in standard deviations. This number is called the *Z-score* for the sequence. We then choose a Z-score cut-off, either *a priori* or by looking at the histogram of Z-scores for sequences in the database (see Figure 10), and use it to decide if a given sequence fits the model or not. We have found that a Z-score of approximately 5 appears a good choice in most cases we have examined, but we suggest carefully checking the histogram by eye before deciding on a cutoff. For example, for our HMM of the kinase catalytic domain, sequences with Z-scores below 5 are classified as not containing the kinase catalytic domain, and sequences with Z-scores above 5 are classified as containing the catalytic domain. If the Z-score of a sequence indicates that it contains the catalytic domain, we can align the sequence to the catalytic domain HMM to find out where this domain occurs in the sequence. The time it takes to do a database search is proportional to the numbers of residues in the database times the length of the model. For our globin model (length 147) we can search the SWISS-PROT database (about 8,375,000 residues) in approximately 2 CPU hours on a Sun Sparcstation 1. Using the shorter EF-hand model (length 29) it takes only 18 CPU seconds (11 user minutes) on a Sun Sparcstation 2. A parallel implemetation of the search procedure (not yet implemented) will speed up these searches substantially, as it has the EM training procedure.

While the statistical techniques we have used to determine Z-scores are still quite crude, we have found that the HMMs are sufficiently good models that these techniques work well enough in practice. However, it may be that more sophisticated techniques are needed in certain cases.

[4] The average curve is calculated as follows. For each length i starting at $i = 1$, the length l_i is computed such that there are at least 500 proteins of lengths i to l_i and less than 500 proteins of lengths i to $l_i - 1$. The length interval i to l_i is called a *window*. The average curve is piecewise linear through the points corresponding to the average length and average NLL-score for each window. The first and last parts of the curve are calculated by linear regression in the first and last window respectively. The standard deviation of the points from the smooth curve is also calculated for each window. The estimate of the average curve can be improved by eliminating outliers, *i.e.* NLL-scores that lie many standard deviations from the average. We iterate the process of removing outliers and reestimating the average curve until no more outliers remain.

2.7 Initial model, local minima, and choice of model length

As mentioned in Section 2.2, when estimating the model from the training sequences, the EM algorithm does not guarantee convergence to the best model. It is basically a steepest-descent-type algorithm that climbs the nearest peak (local maximum) of the likelihood function (or the posterior probability in MAP estimation). Since finding the globally optimal model seems to be a difficult optimization problem in general (Abe & Warmuth, 1990), we have experimented with various heuristic methods to improve the performance of the method.

Probably the best method is to give the model a hint if something is already known about the sequences, which is often the case. A good starting point makes it much more likely that the nearest peak is at least close to optimal. This is done by setting the probabilities in the initial model to values reflecting that knowledge. If, for instance, an alignment of some of the sequences is available, it is straightforward to translate that into a model by simply calculating the relative frequency of the amino acids and the transition frequencies in each position, as in the profile method (Gribskov *et al.*, 1990).

It is of course even more interesting if the model can be found from a *tabula rasa*, *i.e.*, using no knowledge about the sequences. For that we have used an initial model where all equivalent probabilities are the same, *i.e.*, $\mathcal{T}(\mathtt{m}_k|\mathtt{m}_{k+1})$ is independent of the position k in the model, and similarly for all other transition probabilities, and $\mathcal{P}(x|\mathtt{m}_k)$ is also independent of k. To avoid the smaller local maxima, noise is added to the model during the iteration before each reestimation. Initially quite a lot of noise is added, but over ten iterations the noise is decreased linearly to zero. Since noise is added directly to the model, it is not like the usual implementation of simulated annealing, but the principle is the same. The "annealing schedule" is presently rather arbitrary, but it does seem to give reasonable results[5] if it is applied several times, and the best of the models found is used as the final model.

It is important that the best model be selected, since suboptimal models do produce inferior alignments in general. However, when studying alignments from suboptimal globin models, we noted that they tend to align some regions well, occasionally getting better alignments in those regions than the best overall model found, while in other regions they are completely incorrect. This leaves open the intriguing possibility of combining the best solutions found for different regions into a new overall best model. We have not yet explored this possibility.

The length of the model is also a crucial parameter that needs to be chosen *a priori*. However, we have developed a simple heuristic that selects a good model length, and even helps in the problem of local maxima. The heuristic is this: After learning, if more than a fraction[6] γ_{del} of the paths of the sequences choose $\mathtt{d}_k$, the delete state at position k, that position is removed from the model. Similarly, if more than a fraction γ_{ins} make insertions at position k (in state $\mathtt{i}_k$), a number of new positions equal to the average number of insertions made at that position are inserted into the model after position k. After these changes in

[5]An alternate method that also appears to give good results has been developed by Baldi *et al.* (Baldi *et al.*, 1993; Baldi & Chauvin, 1993). This method uses stochastic gradient descent in place of the EM method, which may help in avoiding local minima.

[6]Currently we choose γ_{del} and γ_{ins} each to be 1/2.

the model, it is retrained, and this cycle is repeated until no more changes are needed. We call this "model surgery."

2.8 Over-fitting and MAP estimation

A model with too many free parameters cannot be estimated well from a relatively small data set of training sequences. If we try to estimate such a model, we run into the problem of *overfitting*, in which the model fits the training sequences very well, but gives a poor fit to related (test) sequences that were not included in the training set. We say that the model does not "generalize" well to test sequences. This phenomenon has been well documented in statistics and machine learning (see *e.g.* (Geman *et al.*, 1992; Berger, 1985)). One way to deal with this problem is to control the effective number of free parameters in the model by using prior information. This can be accomplished with MAP estimation. Parameters that we assume (via our prior distribution on models) can be well-estimated *a priori* in effect become less adaptive, because it takes a lot of data to override our prior beliefs about them, whereas those about which we have only weak prior knowledge are estimated in almost the same manner as in maximum likelihood estimation. In this way, the model can have a very large number of parameters, but a much smaller number of "effectively free" parameters.

The parameters in our HMM models describe probability distributions over sets of 20 values (the amino acids), or over 3 values (the three possible transitions out of a given state to the match, insert and delete states following). Let us focus on the parameters describing distributions on the amino acids; the other parameters are treated analogously. We have denoted the parameter defining the probability that amino acid x occurs in the match state at position k by $\mathcal{P}(x|\mathtt{m}_k)$. For simplicity, let us now fix the position k, and let $p_1, \ldots, p_{20}$ denote the probabilities of the amino acids in this match state at position k. When we reestimate these probabilities in each iteration of the EM method, the data we use is a set of (estimated) counts $n_1, \ldots, n_{20}$, where n_i is the estimated number of times amino acid i occurs in this state, based on alignments of the training data to the current model, for each i between 1 and 20. Let $n = \sum_{i=1}^{20} n_i$. In maximum likelihood estimation, the old value p_i is simply replaced by the new value $\hat{p}_i = n_i/n$, *i.e.* the probability is estimated by the fraction of times the amino acid occurs. In the Bayesian MAP approach that we use, the new value is defined by

$$\hat{p}_i = \frac{n_i + \alpha_i}{n + \alpha}, \tag{7}$$

where $\alpha_1, \ldots, \alpha_{20} > 0$ and $\alpha = \sum_{i=1}^{20} \alpha_i$. The numbers $\alpha_1, \ldots, \alpha_{20}$ define an *a priori* probability density over the set of all possible vectors $(p_1, \ldots, p_{20})$ such that $p_i > 0$ and $\sum_{i=1}^{20} p_i = 1$. This density is known as the *Dirichlet distribution* with parameters $\alpha_1, \ldots, \alpha_{20}$ (Berger, 1985; Santner & Duffy, 1989). The $\hat{p}_i$ defined in (7) are the posterior estimates of the parameters, given the count data and this prior.[7] We use the same $\alpha_1, \ldots, \alpha_{20}$ for reestimating the probabilities of the amino acids in each match state.

[7]Assuming that $\alpha_i > 1$, the MAP estimate is actually obtained by subtracting one from each α_i before applying Equation 7. The formula given in Equation 7 corresponds to a Bayesian least-squares estimate of the p_is.

In order to apply the Bayesian method, one must decide on the parameters $\alpha_1, \ldots, \alpha_{20}$. We have done extensive experiments in this area which are reported in detail elsewhere (Brown *et al.*, 1993). Here we only give a brief description of our approach. The main idea is to analyze previously built HMMs for other protein families to see what kind of probability distributions on amino acids appear in their match states, and use this information to determine the parameters $\alpha_1, \ldots, \alpha_{20}$ of the Dirichlet distribution that defines our prior beliefs. Unfortunately, since we have not yet built many HMMs, we cannot use this approach effectively at the present time. Therefore we have elected to use data from existing multiple alignments of homologous proteins in place of previously built HMMs. In particular, we have used the 129 multiple alignments described in the recent paper of Sander and Schneider (1991). Multiple alignments with less than 30 proteins are discarded, leaving 41 alignments with a total of 5670 positions. For each position in each alignment we extract the vector of counts $n_1, \ldots, n_{20}$, where n_i is the number of times amino acid i occurs in that position among all proteins in that alignment. Then we take these 5670 count vectors and find the parameters $\alpha_1, \ldots, \alpha_{20}$ that fit them best. Here we use the maximum likelihood method, assuming that each vector of counts was generated by selecting a probability distribution $p_1, \ldots, p_{20}$ at random according to the Dirichlet density defined by the (unknown) $\alpha_1, \ldots, \alpha_{20}$, and then randomly generating amino acids according to this probability distribution. The mathematical details are described in (Brown *et al.*, 1993). The final vector of α_i's we obtain is { 0.162339 0.037220 0.107508 0.123557 0.074544 0.122092 0.072662 0.112151 0.128548 0.138534 0.063912 0.113368 0.074824 0.103722 0.110612 0.170739 0.154307 0.143584 0.028017 0.069302 }, where the order of the amino acids is alphabetical with respect to the one letter amino acid code.

Having determined the prior density for the probability distribution of amino acids in a match state, we still need to find a prior for the distributions of amino acids in the insert states, and for the probability distributions governing the transitions between states. We deal with the transitions as follows. For each type of state (match, insert, and delete), we determine a vector of three αs which we denote $\alpha_{match}, \alpha_{insert}, \alpha_{delete}$. This gives nine αs in all. For a match state, α_{match} is used to regularize the reestimation of the probability of transition from this match state into the next match state, α_{insert} the transition to the next insert state, and α_{delete} the transition to the next delete state. Similar conventions are used for insert and delete states. Each of these three vectors of αs is determined by a maximum likelihood method like that described above for $\alpha_1, \ldots, \alpha_{20}$, except here we really do use previously built HMMs. Namely, we use an HMM for globins with 147 positions and an HMM for the kinase catalytic domain with 255 positions to provide 402 examples of transition distributions out of match states to estimate the parameters $\alpha_{match}, \alpha_{insert}, \alpha_{delete}$ of a Dirichlet prior on these transition distributions, and similarly for the insert and delete states.

	Kinase			Globin		
	to match	to delete	to insert	to match	to delete	to insert
From match	15.521340	0.254944	0.265967	48.0	1.0	1.0
From delete	1.819972	1.886984	0.225758	44.0	5.0	1.0
From insert	3.764209	0.37648	4.006562	29.0	1.0	20.0

The first half of the table shows the alpha values used in the kinase experiments. Because these regularization values were obtained from only two HMMs, and both HMM's were built using previous ad hoc regularization methods, these α values are highly preliminary, and will probably need to be refined considerably before they are generally useful. In the globin experiments we used (for historical reasons) a different regularizer.

Finally, we have found that it does not help to try to model the probability distribution of the 20 amino acids in an insert state. Therefore, to avoid having 20 additional free parameters for each insertion state, we simply set the amino acid distributions in the insert states to be uniform distributions, *i.e.*, each amino acid has probability 1/20. In our previous work we have used the global frequencies of the amino acids (Haussler *et al.*, 1993). We have found that it usually makes little difference which one is used, but the uniform distributions tend to improve the performance of the models when searching databases.

The previous discussion implicitly assumed that the model in principle is almost perfect, *i.e.*, given enough training data a very good model can be found even without a regularizer. That might not be the case. After all, the model is based on some very simplified assumptions about proteins, and in some cases the model structure might have an inherent bias towards models that are wrong biologically. In our globin experiments, for instance, we have seen cases where the EM algorithm chose to model a lot of proteins as only insertions and deletions (*i.e.*, skipping a large percentage of the match states). By regularization it is possible to bias the model toward using the match states for their intended purpose. In the globin experiment, this was done by increasing the strength of the regularization on the transitions going from delete to match. Although related, this kind of regularization serves a different purpose than correcting for the bias introduced by a limited training set.

3 Results

3.1 Globin experiments

The modeling was first tested on the globins, a large family of heme-containing proteins involved in the storage and transport of oxygen that have different oligomeric states and overall architecture (for a review see (Dickerson & Geis, 1983)). Hemoglobins are tetramers composed of two α chains and two other subunits (usually β, γ, δ or θ). Myoglobin is a single chain, some insect globins are present as dimers and some intracellular invertebrate globins occur in large complexes of many subunits.

Globin sequences were extracted from the SWISS-PROT database (release 19) by searching for the keyword "globin". Eliminating the false positives, resulted in 625 genuine globin sequences of average length 145 amino acids. We left three non-globins in the sample for illustrational purposes giving a total of 628 sequences. The sample of globins in the database is *not* the random sample a statistician would prefer, but is perhaps one of the best and largest collection of protein sequences from a homologous family. Searching for the words “alpha”, “beta”, “gamma”, “delta”, “theta”, and “myoglobin” in the data file yielded 224 alpha, 199 beta, 16 gamma, 8 delta and 5 theta chains and 79 myoglobins, which adds up to 531 sequences. These should naturally be considered minimum numbers, but they give a good picture of how skewed the sample is.

To test our method, we trained an HMM using the method described in Sections 2.2 and 2.7. We used a homogeneous initial model that contained no knowledge about the globin family. Its probability parameters were derived from the prior, and were the same for all equivalent transitions (*i.e.*, 9 different transition probabilities). All amino acid probabilities (the $\mathcal{P}$ distributions) were set equal to the distribution of the amino acids given in Section 2.8. In the insert states we used a probability of 1/20 for all amino acids. The only model parameters set by hand are the initial transition probabilities and corresponding regularization parameters (see Section 2.8). From our experience, the method does not seem to be very sensitive to the choice of these parameters, but it would require considerable further experimentation to verify this quantitatively.

For our training set, we picked 400 sequences at random from the 628 sequences. We witheld the remaining 228 sequences in order to test the model on data not used in the training process. The model was trained using noise and model surgery ($\gamma_{del} = \gamma_{ins} = 0.5$), as described in Section 2.7. This procedure was repeated about 20 times with model lengths chosen randomly between 145 and 170. The average run-time was around 60 cpu minutes on a Sun Sparcstation I. For each run we computed a NLL-score for the model, which was the average of the NLL-scores for the training sequences, as defined in Section 2.2. The final NLL-scores varied considerably for these runs but the best was 210.7.

We then took this model, produced 10 new models by adding noise, and optimized these. These models all generated approximately the same NLL-score and we picked the model with the best NLL-score, 210.3, having a length of 147. We validated this model[8] in two ways: from the alignments it produced, and by its ability to discriminate between globins and non-globins. The results are described below.

Multiple sequence alignments

A multiple alignment of many globin sequences has been produced by Bashford *et al.* (1987) by including into the alignment procedure tertiary-structure information of seven globins (Figure 4). This was achieved by aligning these seven sequences and then aligning the rest of the 226 studied to the closest of these seven. In contrast, generating multiple alignments with HMMs requires no prior knowledge of underlying structure. Using the globin HMM, we produced a multiple alignment of all the 625 globin sequences by the Viterbi algorithm as described in Section 2.3. Figure 5 shows this alignment for the seven sequences from (Bashford *et al.*, 1987).

[8]We stress that the final model was chosen according to an objective measure, namely the NLL-score on the training set, and not retroactively on the basis of how well it did in multiple alignment or database search tasks.

The alignment found in this experiment agrees extremely well with the structurally derived alignment of Bashford *et al.* . Our alignment differs in the region between the C and E helices. However, this is a highly variable area since only some globins possess a D helix. The difference in the F/G-helices is more pronounced, with the remaining discrepancies possibly representing an alternative alignment. Four of the insertions the model chose are in variable regions between or at the end of helices, *i.e.*, between secondary structure elements. The last two insertions appear in the F/G region.

Database search: Discriminating globins from non-globins

The globin HMM model we found was also tested on all the 25,044 proteins in the SWISS-PROT database release 22.0 of length less than 5000 amino acids (which is all but 2). A NLL-score and a Z-score were computed for each of these sequences as described in Section 2.6. These are plotted in Figures 6 and 7 as a scatter plot and a histogram, respectively.

For the histogram (but not the scatter plot), the data were filtered as follows:

- All sequences with a Z-score > 3.5 and either more than a total of 25 or more than 15% unknown residues were removed (a total of 23). Currently, we treat an unknown amino acid, X, as being the most probable amino acid at the position it is matched to, so sequences with many Xs spuriously match the model very well.
- Since we searched a newer release of SWISS-PROT (rel. 22) than the one from which the globin training set was extracted (rel. 19), 8 new globins were found and incorporated into the test set.
- Five globin fragments of length 19–45 were removed from the data.
- Three non-globin sequences in the globin file that were identified as outliers in Figure 6 were removed. One of these non-globins was left as part of the training set to illustrate the robustness of the method.

The model distinguishes extremely well between globins and non-globins. Choosing a Z-score cutoff of 5 we would miss 2 out of 628 globins[9] and get essentially no false positive globins. There is one "non-globin", a bacterial haemoglobin-like protein (SWISS-PROT id HMP_ECOLI), that may or may not be counted as a false positive depending on your point of view. Only one sequence, the heme containing catalase of *Penicillium Vitale* (CATA_PENVI, Z-score 4.7), has a Z-score between 4.2 and 5.1, so any cutoff in this range will essentially give the same separation. The two sequences falling between a Z-score of 1 and 4 (GLB_PARCA and GLB_TETPY) are protozoan, whereas the other globins are metazoan. The primary sequences of these globins are similar and have little similarity with other eukaryotic globins. Note also that both of these sequences are in the test set.

[9] 628 in the original data set, plus 8 new, minus 3 spurious, minus 5 fragments. 397 were left from the training, and the remaining 231 made up the test set.

Discovering subfamilies of globins

We also performed an experiment to automatically discover subfamilies of globins using the method described in Section 2.4. An HMM with 10 component HMMs was used. The initial lengths of the components were chosen randomly between 120 and 170, but were adjusted by model surgery during training. We trained this HMM on *all* 628 globins and then calculated the NLL-score for each sequence for each of the 10 component HMMs. A sequence was classified as belonging to the cluster represented by the component HMM that gave the lowest NLL-score, *i.e.*, the one giving the highest probability to that sequence.[10] Three of these clusters were empty and the remaining 7 nonempty ones represented chains from known globin subfamilies:

Class 1 233 sequences: principally all α, a few ζ (an α-type chain of mammalian embryonic hemoglobin), π/π' (the counterpart of the α chain in major early embryonic hemoglobin P), and θ-1 chains (early erythrocyte α-like).

Class 2 232 sequences, almost all β, a few δ (β-like), ϵ (β-type found in early embryos), γ (comprise fetal hemoglobin F in combination with two α chains), ρ (major early embryonic β-type chain) and θ chains (embryonic β-type chain).

Class 3 71 myoglobins.

Class 4 58 sequences. The 13 highest scoring in this cluster are leghemoglobins. This class contain a variety of sequences including the three non-globins in original data set.

Class 5 19 sequences. Midge globins.

Class 6 8 sequences. Globins from agnatha (jawless fish).

Class 7 7 sequences. Varied.

We have not repeated this experiment using different randomization to ascertain if better results can be obtained. However, we are encouraged by the results of this first experiment since it is able to classify correctly the major globin subfamilies (alpha, beta and myoglobin).

The final globin model

Examination of the model itself yields information on the structure of globins. Figure 8 shows the normalized frequency counts (the numbers used to re-estimate the parameters of the model) from some parts of the final model. The thickness of a line indicates what fraction of the 400 training sequences made that transition or used that particular amino acid. A dashed line indicates that less than 5% of the sequences used that transition. (The continued delete is mostly due to fragments that have to make many deletions.) The histogram in a match state shows the distribution of amino acids that were matched to that

[10] We can also calculate the posterior probability of each cluster by looking at the transition probabilities out of the global start state, and thereby obtaining a posterior distribution over the 10 clusters for each sequence. However, these posteriors are very sharply peaked, so this adds little to the analysis.

state. The number in an insert state shows the average length of an insertion beginning at that position.

For the amino acids the ordering proposed by Taylor (1986) is used. Starting from the top, the amino acids are medium-sized and non-polar, small and medium polar (around G and P), medium sized and polar (around K), large medium-polar (around F and Y), and finally below they are medium-large and non-polar. There does seem to be some tendency for the distributions to peak around neighboring amino acids when using this ordering, as one would expect. When one looks at the whole model, regions that are highly conserved are also readily distinguished from the more variable regions, both as a function of the probability that a position is skipped, and the entropy of the distribution of amino acids at that position.

3.2 Kinase experiments

Protein kinases are defined as enzymes that transfer a phosphate group from a phosphate donor onto an acceptor amino acid in a substrate protein (Hunter, 1991; Hanks *et al.*, 1988). Based upon the acceptor amino acid specificity, they have been classified into serine/threonine, tyrosine, histidine, cysteine, aspartyl and glutamyl kinases. Only enzymes in the first two categories have been well characterized and recent developments indicate that some can phosphorylate both alcohol (serine/threonine) and phenol (tyrosine) groups, the so called dual-specificity protein kinases (Lindberg *et al.*, 1992). It is the region comprising the catalytic domain of these hydroxyamino acid phosphorylating enzymes that we model by an HMM and which we subsequently refer to as protein kinases or simply kinases. Despite the differences in size, substrate specificity, mechanism of activation, subunit composition and subcellular localization, all these kinases share a homologous catalytic core containing 12 conserved subdomains or regions (Hanks & Quinn, 1991; Hanks *et al.*, 1988).

Because the kinase catalytic domain is only a subsequence embedded in a larger protein, the kinase experiments differed from the globin experiments. The HMM used in the globin experiments modeled the *entire* protein rather than simply a segment of a protein as is the case for the kinase family. Modeling domains requires several modifications to our standard HMM training which are described in Section 2.5.

The training set for these experiments is a group of 193 sequences from the March 1992 release of the protein kinase catalytic domain database maintained by S. K. Hanks and A. M. Quinn (91). This set is composed of serine/threonine, tyrosine and dual-specificity kinases principally from vertebrates and higher eukaryotes but also includes some from lower eukaryotes, retroviruses and herpes virus.

We trained 10 HMMs on all 193 (unaligned) sequences in this data set using the prior distributions described in Section 2.8. No parameters of the modeling process were set manually and the initial model lengths ranged from 242 to 282 positions (this encompasses the average length of the sequences in our kinase catalytic domain training set). At the end of the ten training runs, the best kinase model had a NLL-score (the average $-\log P(sequence|model)$ over the training set) of 588.39 and a length of 254. Modules were added at the beginning and end of this model as described in Section 2.5. We tested this model in the same manner as described earlier for the globin model.

Our main tests were discrimination tests, in which we utilized the model to search the

SWISS-PROT version 22 database (25,044 sequences) for proteins containing the kinase catalytic domain.

As described in Section 2.6, a NLL-score was computed for each of the sequences in the database and this information was used to compute a sequence's deviation from the average curve as measured by a Z-score. The data was then filtered to remove all sequences with any unknown residues (353) and all sequences having length less than 200 (4230), since complete protein kinase catalytic domains range from 250 to 300 residues (Hanks *et al.*, 1988). This filtering removed a total of 4386 sequences. A scatter plot of NLL-score versus length for the SWISS-PROT sequences in given in Figure 9.

A cutoff of 6.0 was chosen because there are no sequences with Z-scores between 4.935 and 6.773. See figure 10 for a histogram of the resulting Z-scores. Any sequence having a Z-score > 6.0 was therefore classified as containing the kinase catalytic domain while those with Z-score < 6.0 were classified as not possessing the domain. With this cutoff, 296 sequences were classified as containing the kinase catalytic domain. The remaining 20357 sequences were rejected.

The general issue of estimating the number of false negatives and false positives when distinguishing sequences belonging to a given family from non-members is a complex one. In the case of the globins, it is "relatively" straightforward since it is possible to identify all the globins in the database by performing a keyword or title string search. The situation for the kinase domain or the EF-hand motif (see Section 3.3) is less obvious and thus more problematic. For instance, while a given protein may possess the sequence characteristics for this motif or domain, functionally, the region may not bind calcium or possess kinase activity. We have attempted to address this complicated matter as best we can as described below. However, we stress that we do not feel able to give a definitive answer as to the number of true false negatives and true false positives in our kinase or EF-hand database discrimination tests.

A list of potential protein kinases was created from the union of sequences designated as being kinases from four independent sources: our HMM, PROSITE (a dictionary of sites and patterns in proteins (Bairoch, 1992)), PROFILESEARCH (a technique used to search for relationships between a protein sequence and multiply aligned sequences (Gribskov *et al.*, 1990)) and a keyword search.

Two regions of the catalytic domain of eucaryotic protein kinases have been used to build PROSITE signature patterns. The first pattern corresponds to an area believed to be involved in ATP binding (PROSITE entry PROTEIN_KINASE_ATP, sequence motif [LIV]G.G.[FYM][SG].V). There are two signature patterns for the second region important for catalytic activity: one specific for serine/threonine kinases (PROTEIN_KINASE_ST, [LIVMFYC].[HY].D[LIVMFY]K.2N[LIVMFYC]3) and the other for tyrosine kinases (PROTEIN_KINASE_TYR, [LIVMFYC].[HY].D[LIVMFY][RSTA].2N[LIVMFYC]3). Since PROSITE expressions do not allow for flexible gapping or insertions, a profile of kinases was constructed from an alignment of seven kinases and employed for database discrimination tests (M. Gribskov, personal communication) using the program PROFILESEARCH (Gribskov *et al.*, 1990). The seven kinases used to generate the profile are, bovine cAMP dependent protein kinase (PIR code OKBOG), bovine cGMP dependent protein kinase (OKBO2C), bovine protein kinase C (KIBOC), human *mos* kinase related transforming protein (TVHUF6), human *ref-a* kinase related transforming protein (TVHUMS), mouse *pim-1* kinase related transforming protein (TVMSP1), and human *fes/fps* kinase related transforming protein (TVHUFF). The keyword search consisted of searching the descriptions of the sequences in SWISS-PROT for the following strings: "SERINE/THREONINE-PROTEIN KINASE, SER/THR-PROTEIN KINASE, PROTEIN-SERINE/THREONINE KINASE, PROTEIN-SER/THR KINASE, TYROSINE-PROTEIN KINASE, TYR-PROTEIN KINASE, PROTEIN-TYROSINE KINASE, PROTEIN-TYR KINASE, V-ABL, C-ABL, V-FGR, C-FGR, V-FMS, C-FMS, V-FPS/FES, V-FES/FPS, C-FPS/FES, C-FES/FPS, V-FYN, C-FYN, V-KIT, C-KIT, V-ROS, C-ROS, V-SEA, C-SEA, V-SRC, C-SRC, V-YES, C-YES, V-ERBB".

Of the 296 SWISS-PROT 22 sequences that were above the Z-score cutoff of 6.0 and were thus classified as containing a kinase domain by our HMM, 278 were similarly classified by PROSITE, PROFILESEARCH and the keyword search. These 278 sequences may be considered to constitute "certain kinases". Figure 11 shows the multiple sequence alignment generated by our HMM of some representative kinases from this set (sequences 1-22). Sequences 23-40 are the 18 sequences (296 minus 278) that were designated as kinases by the HMM and one or two of the three other methods. For PROSITE, we consider a sequence to be a kinase if it satisfies one or more of the three patterns PROTEIN_KINASE_ATP, PROTEIN_KINASE_ST or PROTEIN_KINASE_TYR as a true positive ("T" in Figure 11B). PROSITE false negatives ("N"), potential hits ("P") and false positives ("F", sequences which do not belong to the set under consideration) are ignored.

Among the 18 sequences classified as kinases by our HMM, eight (23-26, 35, 38-40) were also deemed to be kinases by the keyword search and PROSITE, and one (27) by PROFILESEARCH and PROSITE. The remainder (28-34, 36-37, those indicated by "%" in Figure 11B) are particulate guanylyl cyclases and except for 36-37, PROFILESEARCH also defines them as possessing a kinase domain. These guanylyl cyclases contain a single transmembrane domain, a cyclase catalytic domain and an intracellular protein kinase-like domain in which protein kinase activity has not been seen to date (reviewed in (Garbers, 1992)). Although these sequences are not kinases in terms of function, they possess all the conserved subdomains except for subdomain I (the nucleotide binding loop) and the majority of conserved residues present in "certain kinases" (see "Subdomain" of Figure 11A and positions indicated by "*").

Sequences 41-50 are the top ten sequences in SWISS-PROT immediately below our cutoff of 6.0. Of these, the first three (41-43) were classified as kinases by two out of PROSITE, PROFILESEARCH and the keyword search. Our cutoff was chosen from a visual inspection of a histogram of Z-scores which indicated that 6.0 lay in a large gap (see Figure 10). If the Z-score cutoff is lowered to the next largest gap (from Z-score 3.9 to 4.8) between sequences 43-44, then these three viral sequences (41-43) would also be categorized as kinases by the HMM.

Of the eight sequences (41, 51-53, 56-57, 59-60) that were not classified as kinases by our HMM but were classified only by the keyword search and PROSITE, one (41) is the first sequence below our cutoff discussed above. Four (56-57, 59-60) are partial sequences where the kinase domain is absent. Three (51-53) possess divergent forms of many of the conserved regions and like 41-43, although they are below our cutoff, the HMM is able to generate an alignment that correctly identifies divergent forms of conserved regions. Finally, there are three aminoglycoside 3'-phosphotransferase sequences (54-55, 58) which are only designated as kinases because they satisfy the PROSITE expression for the catalytic loop.

Inspection of Figure 11B permits an estimation of the accuracy of the various methods in distinguishing kinases from non-kinases in database discrimination tests. The HMM generates 6 false negatives (41-43, 51-53) of which the first three fall immediately below our kinase cutoff. For PROFILESEARCH, there are 12 false negatives (23-26, 35, 38-41, 51-53) but it should be recalled that nine of these (those indicated by "$" in Figure 11B) do not appear in the results obtained from searching SWISS-PROT 25 provided to us by M. Gribskov (personal communication). We suspect that at least four (23-26) would be correctly classified as kinases by PROFILESEARCH leaving an estimate of 3-8 false negatives. In the case of PROSITE, using our assumption of a kinase to be a true positive ("T") sequence for any one of the three patterns, there are 3 false negatives (39, 42-43). However, the actual performance of the PROSITE patterns themselves is much worse; scans of SWISS-PROT 22 with each of the patterns PROTEIN_KINASE_ATP, PROTEIN_KINASE_ST and PROTEIN_KINASE_TYR individually yield 40, 2 and 3 false negatives respectively.

The difficulty in quantifying the precise number of false positives and false negatives produced by the database discrimination tests may be illustrated by employing an alternative mechanism for assessing the number of false negatives. If simply the number of sequences denoted as kinases only by all three other methods is evalutated, the number of false negatives for each of the techniques differ from the more detailed analysis: 3 for the HMM (39, 42-43), 7 for PROFILESEARCH (23-26, 35, 38, 40) and 0 for PROSITE (ignoring known false negatives as above). This general problem is further highlighted by the guanylyl cyclases (indicated by "%" in Figure 11B). If the definition of a kinase is based upon function and not possession of particular sequence patterns, then the guanylyl cyclases are the only false positives for both the HMM and PROFILESEARCH. The PROSITE patterns PROTEIN_KINASE_ATP, PROTEIN_KINASE_ST and PROTEIN_KINASE_TYR produce 8, 0 and 2 false positives respectively, giving some indication of the actual PROSITE performance.

Overall, both the HMM and PROFILESEARCH appear to perform generally better than PROSITE in the discrimination tests, with the HMM possibly having a slight advantage over PROFILESEARCH.

The HMM database search did not suggest any new putative kinases in SWISS-PROT 22. However, a comparative examination of the HMM produced multiple sequence alignment and the crystal structure of the catalytic subunit of cAMP-dependent protein kinase (Knighton *et al.*, 1991) (sequence 1), a template for the protein kinase family, yields insights into the conserved regions and their functions in kinases of unknown structure. Figure 11A displays the location of secondary structure elements obtained from this crystal structure. An invariant Asp in subdomain VIb (Asp 166 in (Knighton *et al.*, 1991)) that is proposed to be the catalytic base is known to diverge in guanylyl cyclases (28-34, 36-37) even though the immediate region is highly conserved (Garbers, 1992). Our results indicate that other invariant residues appear to be replaced as well. In the sea urchin spermatozoan cell-surface receptor for the chemotactic peptide "resact" (sequences 32 and 38), a Lys in subdomain II (Lys 72) that forms part of the ATP α- and β-phosphate binding site is changed to His. The heat-stable entertoxin receptor of rat (40) replaces an Asp in subdomain IX (Asp 200) that contributes directly to stabilisation of the catalytic loop by Glu. Like these guanylyl cyclases, yeast VPS15 (sequence 39), a probable serine/threonine kinase that is auto-phosphorylated, also lacks subdomain I. In addition, a conserved ion-pair that stabilises ATP (Glu 91 - Lys 72) would be disrupted in VPS15 because the Glu in subdomain III is altered to Arg resulting in the apposition of two positively charged residues. In the putative B12 kinases of two strains of vaccinia virus (42-43), the proposed Asp catalytic base is replaced by Lys (*cf* guanylyl cyclases). This is accompanied by a further change in the "general" sequence of the catalytic loop: the normally positively charged residue at $n+2$ has been altered to Glu. In general, all the sequences below our cutoff and the last one above it (40-60) appear to lack α-helix F (see "X-ray" in Figure 11A). The functional and or structural consequences of these modifications on any kinase activity are not clear.

3.3 EF-hand experiments

For these experiments we used the June 1992 database of EF-hand sequences maintained by Kretsinger and co-workers (Nakayama *et al.*, 1992). Sequences in this database are proteins containing two or more copies of the EF-hand motif, a 29-residue structure present in cytosolic calcium-modulated proteins (Nakayama *et al.*, 1992; Persechini *et al.*, 1989; Moncrief *et al.*, 1990). These proteins bind the second messenger calcium and in their active form function as enzymes or regulate other enzymes and structural proteins. The motif consists of an α-helix, a loop binding a Ca^{2+} ion followed by a second helix. Although a number of proteins possess the EF-hand motif, some of these regions have lost their calcium-binding property.

For our training set, we extracted the EF-hand structures from each of the 242 sequences in the database, obtaining 885 EF-hand motifs having an average length of 29. For our first experiment we trained five HMMs on all 885 EF-hand motifs, using the standard techniques described earlier. (In subsequent experiments, described below, we trained on smaller subsets of these 885 sequences.) The best model had a final length of 29, and a NLL-score (the average $-\log P(\text{sequence}|\text{model})$) of 61.41.

As described in Section 2.5, we modified the final model to enable it to search the SWISS-PROT database for sequences containing the EF-hand motif. We computed Z-scores for all sequences as described in Section 2.6 and Figure 12 shows the resulting histogram. In contrast to the kinases, a visual inspection of the histogram of Z-scores did not indicate the presence of a distinct gap thus making the selection of a cutoff more difficult. After choosing by eye a cutoff of 4.75 and excluding all sequences with unknown residues (Xs), the model classified 232 sequences as containing the EF-hand sequence motif.

As with the kinase experiments in the previous section, false positives and false negatives were identified in the following manner. A list of "certain EF-hands" was created from the union of sequences determined to be contain the EF-hand motif by three independent sources: PROSITE, a keyword search, and the results of Michael Gribskov's PROFILESEARCH.

We obtained a list of sequences which satisfied the PROSITE pattern "EF-HAND" i.e. those possessing the sequence "D.[DNS] [$\wedge$ ILVFYW] [DENSTG] [DNQGHRK] [$\wedge$ GP] [LIVMC] [DENQSTAGC].2 [DE] [LIVMFYW]" which includes the complete EF-hand loop as well as the first residue following the loop. We consider a sequence to be an EF-hand as defined by PROSITE if it is listed as a true positive for the pattern ("T" in Figure 13B). False negatives ("N") and potential hits ("P") are ignored. The keyword search consisted of searching the annotations of sequences in SWISS-PROT for one or more of the following keywords: "ALPHA-ACTININ, ALPHA ACTININ, AEQUORIN, CALBINDIN, CALCINEURIN, CALCIUM VECTOR PROTEIN, CA VECTOR PROTEIN, CALCYPHOSIN, P24 THYROID PROTEIN, CALCIUM-DEPENDENT PROTEIN KINASE, CALCIUM DEPENDENT PROTEIN KINASE, CA DEPENDENT PROTEIN KINASE, CALCIUM-BINDING PROTEIN, CALCIUM BINDING PROTEIN, ICABP, CALCYCLIN, CALGIZZARIN, CALGRANULIN, CALMODULIN, CALPAIN, CALRETININ, CALTRACTIN, CDC31, DIACYLGLYCEROL KINASE, EIGHT-DOMAIN PROTEIN, F-ANTIGEN, FIMBRIN, LPS1, LUCIFERIN-BINDING PROTEIN, LUCIFERIN BINDING PROTEIN, MYOSIN REGULATORY LIGHT CHAIN, MYOSIN ESSENTIAL LIGHT CHAIN, ONCOMODULIN, OSTEONECTIN, PARVALBUMIN, PLACENTAL-CABP, PLACENTAL CALCIUM BINDING PR0TEIN, PLACENTAL CALCIUM-BINDING PR0TEIN, 18A2, S100, S-100, SARCOPLASMIC CALCIUM-BINDING PROTEIN, SARC1, SARC2, SORCIN, SPEC1, SPEC2, SPECTRIN, SQUIDULIN, TCBP-10, VISININ, RECOVERIN, TROPONIN C, TWO-DOMAIN CALCIUM-BINDING PROTEIN" and "TCPB10". This set of keywords yielded some extraneous sequences, most of which could be eliminated by removing those sequences containing "CALMODULIN-BINDING, HYPOTHETICAL PROTEIN, CALMODULIN-DEPENDENT, CALMODULIN-SENSITIVE" and "CALPAIN INHIBITOR". Two different PROFILESEARCH experiments were conducted for us by M. Gribskov (personal communication). The first employed a profile generated using the multiple sequence alignment of sequences classified as EF-hands by our HMM and the second was constructed using an alignment of the following four sequences: *E. coli* galactose binding protein (JGECG, 1 EF-hand motif), rabbit parvalbumin (PVRB, 2), human troponin (TPHUCS, 4) and human calmodulin (MCHU, 4).

Although a sequence may possess multiple copies of the EF-hand (or any other) motif, only the one which most closely resembles that described by the HMM is identified. Of the 232 SWISS-PROT 22 sequences that were above the cutoff (Z-score > 4.75) and were thus classified as containing an EF-hand motif by our HMM, 163 were similarly classified by PROSITE, both PROFILESEARCH experiments and the keyword search (if only one of the PROFILESEARCH experiments is considered, then there are an additional 14 sequences making a total of 177). These may be considered to constitute "certain EF-hands" and Figure 13 shows the multiple sequence alignment generated by our HMM of some representative EF-hands from this set (sequences 1-27). Of the 69 (232 minus 163) or 55 (232 minus 177) sequences above the cutoff and not categorized as EF-hands by all three other methods, 33 possess the motif but do not bind calcium (indicated by "%" in Figure 13B) and six (64, 72, 88, 89, 91, 94) were classed as EF-hands by only one other method.

The identification of "certain" EF-hands as compared to "certain kinases" is not as straightforward, making it difficult to ascertain the precise number of classification errors made by each technique. This problem arises partly because of the absence of a pronounced gap in the histogram of Z-scores and the resultant uncertainty in assigning an exact cutoff (*cf* Figures 10 and 12). The mnemonic developed to identify EF-hand homologues and distinguish them from analogues (Nakayama *et al.*, 1992) (see *Ca-binding* in Figure 13A) is known to generate errors and is unable to detect 8 of the 27 sequences known to be EF-hands (sequences 1-27 in Figure 13). Therefore, the sensitivity and specificity of the EF-hand database discrimination tests is unlikely to be comparable to the kinases. Using Figure 13B, an estimate of the false negative rate for each method was determined by using the simple notion of evaluating the number of sequences classified as EF-hands by all methods other than the one being considered. (Those which possess the motif but do not bind calcium, denoted by "%" in Figure 13B, are not considered.) Using this criterion, the number of false negatives are 1 for the HMM (101), 20 for PROFILESEARCH using four sequences (28, 47, 56-57, 59, 67, 74-82, 84-85, 92-93, 96), 7 for PROFILESEARCH using our EF-hand alignment (28, 57, 74, 79-80, 92-93), 1 for the keyword search (58) and 2 for PROSITE (60, 70). A similar analysis of false positives produces 6 for the HMM (52, 71, 83, 86, 90, 94) 9 for PROFILESEARCH using four sequences (97, 99, 111-112, 121-122, 129, 132-133), 8 for the keyword (123, 126, 130-131, 134-137) and 1 for PROSITE (120). It should be noted however, that a search of SWISS-PROT 22 using the PROSITE pattern "EF-HAND" produces different results: 3 false negatives and 24 false positives (compare with 2 and 1 using the simple criterion). A total of 26 sequences were not designated as EF-hands by the HMM but were classified so by PROFILESEARCH, PROSITE or the keyword search. Of these, 19 were classified as such by only one of these methods. This includes five fragments where the EF-hand motif is missing: human and murine spectrin *alpha*- and *beta*-chains (123, 126, 131, 134) and rabbit calgizzarin (125).

Inspection of the HMM produced alignment and examination of the putative calcium binding ligands (Figure 13) for the twenty sequences immediately below the cutoff (97-116) and the false negatives and positives suggests that many possess potential EF-hand motifs. This includes 6 sequences whose Z-scores lie above our cutoff but are not classed as EF-hands by any other method: chicken myosin light chain alkali, smooth muscle (52); bovine calpactain I light chain (71); *Arabidopsis thaliana* inorganic pyrophosphatase (83); rat placental calcium-binding protein (90) (note however that the mouse protein, sequence 88, *is* designated as an EF-hand by the keyword search); and rat and bovine 1-phosphatidylinositol-4,5-bisphosphate phosphodiesterase III (86 and 94). A notable example among the false negatives is the α-1 subunit of L-type calcium channels from carp and rabbit skeletal muscle (97, 99) and rat and rabbit cardiac muscle (111, 112). These proteins play an important role in excitation-contraction coupling and carry the calcium antagonist binding domains (reviewed in (Grabner *et al.*, 1991)). They possess a highly conserved and evolutionarily preserved putative intracellular region of 155 residues near the carboxyl terminus immediately following the fourth internal repeat. This region has been suggested to contain functional domains that are typical or essential for all L-type calcium channels regardless of whether they couple to ryanodine receptors, conduct ions or both (Grabner *et al.*, 1991). The inferred EF-hands for these proteins occur within this conserved 155 residue segment.

The above results were for an HMM trained on all 885 EF-hand motifs from the Kretsinger database. There is considerable overlap between this training set and the EF-hand motifs found in SWISSPROT 22, so in order to provide some clearer cross validation of our results we also did another series of experiments. In these experiments, models were estimated using training sets consisting different numbers of randomly chosen EF-hand sequences from the database of 885 EF-hand sequences. For training sets consisting of 5, 10, and 20 random EF-hand sequences, 15 models were estimated, each using a different randomly chosen training set. For training sets consisting of 40, 80, 100, 200, and 400 random EF-hand sequences, 5 models were estimated. In all, seventy models were estimated. A model's performance after training was gauged on how well it performed on a test set which consisted of motifs from database of 885 sequences that were not used in the training set. Thus for each model, two NLL-scores were computed (see Section 2.6), one for the training set and one for the test set. These NLL-scores serve as a quantitative measure of how well the model is representing the sequence data. Figure 14 shows that for small training set sizes, the model overfits the training data. This is shown by low training NLL-scores but very high testing NLL-scores. This effect largely disappears when the training set size reaches about 100 sequences.

A model's performance was also gauged on how well it searches a database for sequences containing the EF-hand motif. For each training set size, one model was randomly chosen to search SWISSPROT 22. A histogram of the resultant Z-scores was plotted and a cutoff was chosen by eye. The number of false positives was computed, as described earlier in this section, by taking a list of "certain EF-hands" (i.e. determined to contain the EF-hand motif by the three independent sources) and counting the number of sequences above the Z-score cutoff in the HMM database search that were not in the "certain EF-hand" list. Figure 15 shows that models built from small training sets have large numbers of false positives. Again, this effect disappears substantially when the training set size reaches about 100 sequences.

4 Discussion

A new method to model protein families using hidden Markov models has been introduced. The method is capable of tapping into the tremendous amount of statistical information contained in many unaligned sequences from the same family. For the cases of globins, kinases and EF-hands, the results have shown that by using this method, it is possible to obtain multiple alignments that mirror structural alignments, having only the unaligned primary sequences as input. The results have also shown that the model can be used successfully in database searches for putative analogs of sequences in a given protein family or domain. Finally, we believe that the model itself is a valuable tool for representing the family or domain.

The HMM method we have proposed requires that many sequences be available from the family or domain one wants to model. Since the number of sequences in the protein databases is growing rapidly, this may be less of a problem in the future, but it will always be a serious issue. Currently, only a relatively small number of sequences are available for most protein families and domains. For the globin family, we found that 400 sequences is certainly sufficient. Preliminary results indicate that 200 is enough, and even as few as 70 may suffice if they are chosen carefully from our database of 628 (70 chosen at random will be nearly all α- and β-chains). Our experiments using smaller numbers of EF-hand sequences for training, as described in Section 3.3, show a similar trend. Using careful regularization, these numbers might even be lowered further. However, there will be a limit on how small the number of available sequences can be if one hopes to obtain a reasonable model starting from a *tabula rasa*.

We believe that the answer to the problem of small training sets is to add more prior knowledge into the training process. One way to do this is by starting with a better initial model. We have performed several experiments in which we have started with a model obtained from a small set of aligned sequences, and then trained the model further using a larger set of unaligned sequences. These will be reported in a future paper. We find that this technique can often give better results. This also suggests that one application of HMMs may be in maintaining multiple alignments as the number of sequences in the alignment grows. Each time new sequences are added to a dataset of homologous sequences, we can begin with the HMM based on the alignment of the previous set of sequences, train it with the larger dataset that includes the new sequences, and then create a new multiple alignment for the larger dataset from this HMM. Not only will the new sequences be included in the new alignment, but the alignment of the old sequences may be improved by utilizing the statistical information present in the larger dataset.[11]

Another way to add more prior knowledge into the training process is to use a more sophisticated Bayesian prior. We are currently exploring the use of a prior on the probability distribution over the amino acids in a match state of the model consisting of a mixture of Dirichlet priors (Brown *et al.*, 1993). This prior is derived in a manner analogous to that described in Section 2.8. Using such a prior is like "soft-tying" the distributions in the states of the HMM. By "soft-tying" we mean a combination of the idea of tying states, see *e.g.* (Rabiner, 1989), in which the number of free parameters is reduced by having groups of states all share the same distribution on the output alphabet (the 20 amino acids in this case), and the idea of *soft weight sharing* from (Nowlan & Hinton, 1992), in which the regularizer (in this case the prior for the distribution of amino acids) is also adaptively modified during learning. We have shown that this method can be used to estimate good EF-hand models using substantially fewer training sequences. Other types of more sophisticated priors can be obtained by switching from the alphabet of the primary sequences to a different representation based more on the structural or chemical properties of the amino acids in the sequence. We plan to explore these as well.

It is interesting to note that we have obtained quite good results in multiple alignment and database searching without using any special weighting schemes to make up for the statistical bias in our training sets (see *e.g.* (Sibbald & Argos, 1990)), or employing Dayhoff's matrix or any of its analogs (see *e.g.* (Waterman, 1989)) to take explicit mutation probabilities between amino acids into account. It also remains to be seen whether or not incorporating any of these extensions into the HMM approach will yield even better results.

We also believe that some of the errors made by our HMM models are due to the fact that these models are suboptimal, in the sense that their NLL-scores are not as low as they could be. This is because the EM procedure is not guaranteed to find the globally optimal model for a given training set. In other experiments, reported in (Haussler *et al.*, 1993), we trained an HMM for globins beginning with a model derived from the Bashford *et al.* alignment, and obtained a slightly lower NLL-score than any model from our experiments using EM on unaligned training sequences (208 compared to 210.3). Hence, we know that EM is not locating the globally optimal model in this case. An important open problem is

[11]This point was suggested to us by an anonymous referee of one of our previous reports.

to find a reliable way to prevent EM from getting stuck and returning a suboptimal solution.

Another issue is the adequacy of the hidden Markov model itself as a statistical model of the sequence variation within a protein family. Clearly an HMM provides at best a "first order" model of sequence variation. There are many kinds of interactions in proteins that are not easily modeled by HMMs, for example, pairwise correlations between amino acid distributions in positions that are widely separated in the primary sequence, but close in the three-dimensional structure (see e.g. (Klinger & Brutlag, 1993)). It would be very valuable to have more general models that incorporate such interactions while still remaining computationally tractable. We are currently exploring the potential of one model class of this type to capture the base-pairing in RNA families (Sakakibara *et al.*, 1993), and hope eventually to incorporate some of the features of these models into our protein models.

Finally, we are encouraged by the quality of the multiple sequence alignments generated by the HMMs and the accuracy of the database searches. For example, the kinase HMM is able to align correctly class III receptor tyrosine kinases which possess a domain that differs from other receptor tyrosine kinases by the insertion of a stretch of seventy to one hundred residues (see the insertion between the "D" and "E" helices in sequence 8, the β-chain of the platelet-derived growth factor receptor, in Figure 11A). With respect to the database discrimination tests, we would eventually like to see HMMs built for all the domains and families currently indexed by PROSITE expressions. In many cases, HMMs for subfamilies could be constructed automatically using the method described in Section 2.4. Once this is done, this might then lead to the construction of a simple "protein language parser" using HMMs. This parser could be constructed by connecting all these individual HMMs in parallel into a single large HMM with a global "BEGIN" and "END" state, and a transition from the "END" state back to the "BEGIN" state. In principle, this parser should be capable of finding all occurrences of each of the PROSITE-indexed domains in a single long protein, using the Viterbi algorithm. The remaining portions of the sequence could be marked as "unknown". While this would not constitute a complete "parse" of the sequence, it would be very useful in providing some automatic annotation of new sequences, which is of critical importance as the rate of growth of the protein databases continues to accelerate. A related approach to protein annotation is given in (Stultz *et al.*, 1993), and a related HMM-based DNA parser for *E. coli* is described in (Krogh *et al.*, 1993).

A comparative examination of the HMM produced kinase multiple sequence alignment and the crystal structure of the catalytic subunit of cAMP-dependent protein kinase (Knighton *et al.*, 1991) indicates a number of conserved residues in kinases of unknown structure that may be suitable for further experimental study (see Section 3.2). Results from our database discrimination tests suggest the presence of an EF-hand calcium binding motif in a highly conserved and evolutionarily preserved putative intracellular region of 155 residues in the α-1 subunit of L-type calcium channels which play an important role in excitation-contraction coupling (see Section 3.3). This region has been suggested to contain the functional domains that are typical or essential for all L-type calcium channels regardless of whether they couple to ryanodine receptors, conduct ions or both. Our EF-hand HMM indicates the following proteins may also possess this motif: chicken myosin light chain alkali (smooth muscle), bovine calpactain I light chain, *Arabidopsis thaliana* inorganic pyrophosphatase, rat placental calcium-binding protein and rat and bovine 1-phosphatidylinositol-4,5-bisphosphate phosphodiesterase III.

Although there are many experiments left to be done, based on our experience, we believe that HMMs and the EM algorithm have tremendous potential in the area of statistical modeling of biological macromolecules. Currently, most of this potential remains to be realized.

Acknowledgements

We would like to thank Peter Brown, Søren Brunak, Richard Durbin, Harry Noller, Martin Vingron, Don Morris, and Michael Zuker for valuable comments on this work. Very special thanks to Richard Hughey for implementing our software on a MASPAR parallel machine and to MASPAR for providing compute time on their machine for some of these experiments, and very special thanks to Michael Gribskov for running his PROFILESEARCH program for the kinases and EF-hands so that we could compare the results to those found with the HMM. This work was supported by NSF grants CDA-9115268 and IRI-9123692, ONR grant N00014-91-J-1162, NIH grant number GM17129, and a grant from the Danish Natural Science Research Council. The full alignments and Z-score tables described in this paper are available in electronic form, and can be obtained by anonymous ftp from `ftp.cse.ucsc.edu`. Our HMM building program and other tools (written in C) will also be made available from the same ftp site.

References

Abe, N. & Warmuth, M. (1990). On the computational complexity of approximating distributions by probabilistic automata. In: *Proceedings of the 3nd Workshop on Computational Learning Theory* pp. 52–66, Rochester, NY: Morgan Kaufmann.

Allison, L., Wallace, C. S., & Yee, C. N. (1992). Finite-state models in the alignment of macromolecules. *Journal of Molecular Evolution,* **35**, 77–89.

Asai, K. and Hayamizu, S. and Onizuka, K. (1993). HMM with protein structure grammar. In: *Proceedings of the Hawaii International Conference on System Sciences* pp. 783–791, Los Alamitos, CA: IEEE Computer Society Press.

Bairoch, A. (1992). Prosite: a dictionary of sites and patterns in proteins. *Nucleic Acids Research,* **20**, 2013–2018.

Baldi, P. & Chauvin, Y. (1993). A smooth learning algorithm for hidden Markov models. To appear in Neural Computation.

Baldi, P., Chauvin, Y., Hunkapiller, T., & McClure, M. A. (1993). Hidden Markov models in molecular biology: new algorithms and applications. In: *Advances in Neural Information Processing Systems 5*, (Hanson, Cowan, & Giles, eds) pp. 747–754, San Mateo, CA: Morgan Kauffmann Publishers.

Barton, G. J. (1990). Protein multiple sequence alignment and flexible pattern matching. *Methods in Enzymology,* **183**, 403–428.

Barton, G. J. & Sternberg, M. J. (1990). Flexible protein sequence patterns: A sensitive method to detect weak structural similarities. *Journal of Molecular Biology,* **212** (2), 389–402.

Bashford, D., Chothia, C., & Lesk, A. M. (1987). Determinants of a protein fold: Unique features of the globin amino acid sequence. *Journal of Molecular Biology,* **196**, 199–216.

Berger, J. (1985). *Statistical Decision Theory and Bayesian Analysis.* New York: Springer-Verlag.

Bowie, J. U., Lüthy, R., & Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science,* **253**, 164–170.

Brown, M. P., Hughey, R., Krogh, A., Mian, I. S., Sjölander, K., & Haussler, D. (1993). Using Dirichlet mixture priors to derive hidden Markov models for protein families. Proceedings of Workshop on AI in Molecular Biology, Wash. D.C. in press.

Cardon, L. R. & Stormo, G. D. (1992). Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *Journal of Molecular Biology,* **223**, 159–170.

Churchill, G. A. (1989). Stochastic models for heterogeneous DNA sequences. *Bull Math Biol,* **51**, 79–94.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the *EM* algorithm. *J. Roy. Statist. Soc. B,* **39**, 1–38.

Dickerson, R. E. & Geis, I. (1983). *Hemoglobin : structure, function, evolution, and pathology.* Menlo Park, California: Benjamin/Cummings Pub. Co.

Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* New York: Wiley.

Everitt, B. S. & Hand, D. J. (1981). *Finite mixture distributions.* London: Chapman and Hall.

Feng, D. F. & Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution,* **25**, 351–360.

Garbers, D. L. (1992). Guanylyl cyclase receptors and their endocrine, paracrine and autocrine ligands. *Cell,* **71**, 1–4.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation,* **4**, 1–58.

Grabner, M., Friedrich, K., Knaus, H.-G., Striessnig, J., Scheffauer, F., Staudinger, R., Koch, W. J., Schwartz, A., & Glossmann, H. (1991). Calcium channels from *cyprinus carpio* skeletal muscle. *Proceedings of the National Academy of Sciences of the United States of America,* **88**, 727–731.

Gribskov, M., Lüthy, R., & Eisenberg, D. (1990). Profile analysis. *Methods in Enzymology,* **183**, 146–159.

Hanks, S. K. & Quinn, A. M. (1991). Protein kinase catalytic domain sequence database: identification of conserved features of primary structure and classification of family members. *Methods in Enzymology,* **200**, 38–62.

Hanks, S. K., Quinn, A. M., & Hunter, T. (1988). The protein kinase family: conserved features and deduced phylogeny of the catalytic domain. *Science,* **241**, 42–52.

Haussler, D. & Krogh, A. (1992). Protein alignment and clustering. Presented at the conference Neural Networks for Computing.

Haussler, D., Krogh, A., Mian, I. S., & Sjölander, K. (1992). Protein modeling using hidden Markov models: Analysis of globins. Technical Report UCSC-CRL-92-23 University of California at Santa Cruz Computer Science, UC Santa Cruz, CA 95064.

Haussler, D., Krogh, A., Mian, I. S., & Sjölander, K. (1993). Protein modeling using hidden Markov models: Analysis of globins. In: *Proceedings of the Hawaii International Conference on System Sciences* , Los Alamitos, CA: IEEE Computer Society Press.

Hunter, T. (1991). Protein kinase classification. *Methods in Enzymology,* **200**, 3–37.

Jurka, J. & Milosavljevic, A. (1991). Reconstruction and analysis of human alu genes. *Journal of Molecular Evolution,* **32**, 105–121.

Klinger, T. & Brutlag, D. (1993). Detection of correlations in trna sequences with structural implications. In: *First International Conference on Intelligent Systems for Molecular Biology,* (Hunter, L., Searls, D., & Shavlik, J., eds) , Menlo Park: AAAI Press.

Knighton, D. R., Zheng, J., Eyck, L. F. T., Ashford, V. A., Xuong, N.-H., Taylor, S. S., & Sowadski, J. M. (1991). Crystal structure of the catalytic subunit of cyclic adenosine monophosphate-dependent protein kinase. *Science,* **253**, 407–414.

Krogh, A., Mian, I. S., & Haussler, D. (1993). A hidden Markov model that finds genes in *e. coli* DNA. Technical Report UCSC-CRL-93-33 University of California at Santa Cruz Computer Science, UC Santa Cruz, CA 95064. in preparation.

Lander, E. S. & Green, P. (1987). Construction of multilocus genetic linkage maps in humans. *Proceedings of the National Academy of Sciences of the United States of America,* **84**, 2363–2367.

Lawrence, C. E. & Reilly, A. A. (1990). An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins,* **7**, 41–51.

Lindberg, R. A., Quinn, A. M., & Hunter, T. (1992). Dual-specificity protein kinases: will any hydroxyl do? *Trends in Biochemical Sciences,* **17**, 114–119.

Lüthy, R., McLachlan, A. D., & Eisenberg, D. (1991). Secondary structure-based profiles: Use of structure-conserving scoring table in searching protein sequence databases for structural similarities. *PROTEINS: Structure, Function, and Genetics,* **10**, 229–239.

Moncrief, N. D., Kretsinger, R. H., & Goodman, M. (1990). Evolution of EF-hand calcium-modulated proteins. I. relationships based on amino acid sequences. *Journal of Molecular Evolution,* **30**, 522–562.

Nakayama, S., Moncrief, N. D., & Kretsinger, R. H. (1992). Evolution of EF-hand calcium-modulated proteins. ii. domains of several subfamilies have diverse evolutionary histories. *Journal of Molecular Evolution,* **34**, 416–448.

Nowlan, S. (1990). Maximum likelihood competitive learning. In: *Advances in Neural Information Processing Systems*, (Touretsky, D., ed) volume 2 pp. 574–582. Morgan Kaufmann.

Nowlan, S. J. & Hinton, G. E. (1992). Soft weight-sharing. In: *Advances in Neural Information Processing Systems 4*, (Moody, Hanson, & Lippmann, eds) , San Mateo, CA: Morgan Kauffmann Publishers.

Persechini, A., Moncrief, N. D., & Kretsinger, R. H. (1989). The EF-hand family of calcium-modulated proteins. *Trends in Neurosciences,* **12** (11), 462–467.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE,* **77** (2), 257–286.

Sakakibara, Y., Brown, M., Underwood, R., Mian, I. S., & Haussler, D. (1993). Stochastic context-free grammars for modeling rna. Technical Report UCSC-CRL-93-16 University of California at Santa Cruz Computer Science, UC Santa Cruz, CA 95064.

Sander, C. & Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins,* **9** (1), 56–68.

Santner, T. J. & Duffy, D. E. (1989). *The Statistical Analysis of Discrete Data.* New York: Springer Verlag.

Sibbald, P. & Argos, P. (1990). Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *Journal of Molecular Biology,* **216**, 813–818.

Stultz, C. M., White, J. V., & Smith, T. F. (1993). Structural analysis based on state-space modeling. *Protein Science,* **2**, 305–315.

Subbiah, S. & Harrison, S. C. (1989). A method for multiple sequence alignment with gaps. *Journal of Molecular Biology,* **209**, 539–548.

Tanaka, H., Ishikawa, M., Asai, K., & Konagaya, A. (1993). Hidden markov models and iterative aligners. In: *First International Conference on Intelligent Systems for Molecular Biology* , Menlo Park: AAAI Press.

Taylor, W. R. (1986). The classification of amino acid conservation. *Journal of Theoretical Biology,* **119**, 205–218.

Vingron, M. & Argos, P. (1991). Motif recognition and alignment for many sequences by comparison of dot-matrices. *Journal of Molecular Biology,* **218**, 33–43.

Waterman, M. S. (1989). Sequence alignments. In: *Mathematical Methods for DNA Sequences*, (Waterman, M. S., ed). CRC Press.

Waterman, M. S. & Perlwitz, M. D. (1986). Line geometries for sequence comparisons. *Bull. Math. Biol.* **46**, 567–577.

White, J. V., Stultz, C. M., & Smith, T. F. (1991). Protein classification by nonlinear optimal filtering of amino-acid sequences. Unpublished manuscript.