Predicting protein structure using hidden Markov models

Kevin Karplus[†]
Kimmen Sjölander[‡]
Christian Barrett[‡]
Melissa Cline[‡]
David Haussler[‡]
Richard Hughey[†]
Liisa Holm[§]

Chris Sander[§]

UCSC-CRL-97-13
3 Nov 1997

[†] Computer Engineering
U.C. Santa Cruz

[‡] Computer Science
U.C. Santa Cruz

[§] The Sanger Centre
England

Baskin Center for
Computer Engineering & Information Sciences
University of California, Santa Cruz
Santa Cruz, CA 95064 USA

ABSTRACT

We discuss how methods based on hidden Markov models performed in the fold recognition section of the CASP2 experiment. Hidden Markov models were built for a set of about a thousand structures from the PDB database, and each CASP2 target sequence was scored against this library of hidden Markov models. In addition, a hidden Markov model was built for each of the target sequences, and all of the sequences in PDB were scored against that target model. Having high scores from both methods was found to be highly indicative of the target and a structure being homologous.

Predictions were made based on several criteria: the scores with the structure models, the scores with the target models, consistency between the secondary structure in the known structure and predictions for the target (using the program PhD), human examination of predicted alignments between target and structure (using RASMOL), and solvation preferences in the alignment of the target and structure. The method worked well in comparison to other methods used at CASP2 for targets of moderate difficulty, where the closest structure in PDB could be aligned to the target with at least 15% residue identity. There was no evidence for the method's effectiveness for harder cases, where the residue identity was much lower than 15%.

Keywords: CASP-2, fold-recognition, HMM, structure library, remote homology

1 Introduction

One method of protein sequence analysis is the identification of *homologous* proteins—proteins which share a common evolutionary history and have similar overall structure and function [21]. Homology is straightforward to infer when two sequences share 25% residue identity over a stretch of 80 or more residues [52]. Recognizing *remote homologs*, with lower primary sequence identity, is more difficult. Here we report how new extensions of the hidden Markov model (HMM) methods for remote homolog recognition reported in [39, 34] fared in the fold recognition and alignment section of the CASP2 experiment [25].

Finding these remote homologs is one of the primary motivating forces behind the development of statistical models for protein families and domains, such as *profiles* and their many offshoots [28, 27, 18, 8, 3, 64, 62, 61, 8, 15, 46, 18, 26], *Position-Specific Scoring Matrices* [32], and *hidden Markov models (*HMM*s)* [19, 66, 59, 39, 34, 7, 6, 4, 22, 24, 23].

HMMs have been used in a variety of fields to do discrete time series analysis, most notably in speech recognition. A general introduction to them can be found in [50]. In biosequence analysis they are used in protein analysis [19, 66, 59, 39, 34, 7, 6, 4, 22, 24, 23], genefinding [40, 41, 35] and other areas [42, 19]. When used as statistical models of protein motifs or domains, they combine the best aspects of weight matrices and Smith-Waterman methods. They are currently used in the analysis of nematode and human DNA sequencing efforts at the genome center at Washington University and at the Sanger Centre [23], have led to the discovery of fibronectin type III domains in yeast [9] and members of the immunoglobulin superfamily in bacteria [11], were used in the analysis of lectins [29], and they have led to other discoveries as well [10, 49, 20].

The structure of the HMMs that are most commonly used in protein modeling is similar to that of a weight matrix or *profile* [28, 27, 15, 30], except that it has specific states and transitions at each position to model insertions and deletions, and the probability parameters for these are different in each position. One advantage of an HMM is that it defines a formal statistical model for sequences in the given protein family, so one can calculate the likelihood of a sequence and find the most probable locations for the insertions and deletions, i.e. the most probable alignment of the sequence to the "consensus model" for the family. The likelihood is calculated by the *forward* algorithm and the most probable alignment by the *Viterbi algorithm*. Each is a dynamic programming method similar to the Smith-Waterman method used to align two sequences. The forward algorithm can be used to search a database for homologs of the protein family represented by the HMM, and the Viterbi algorithm can be used to create a multiple alignment of all family members. The parameters of the HMM can be estimated from a set of unaligned family members using an expectation-maximization method known as the *forward-backward* algorithm [50]. The two most extensively used systems for applying HMMs to protein sequence analysis are SAM[1] [34], which was used in the experiments reported here, and HMMer[2] [23].

The HMM method of fold recognition differs from protein threading methods [17, 36, 57, 47, 37, 67, 44, 56, 43, 45, 65] in that pairwise interactions between amino acids are not modeled or used. This results in more efficient computation, but may involve some loss of information. The HMM method is similar to the profile approach of Eisenberg and his colleagues [26], but the method we used relies much less on structural information for parameter estimation, using instead a Bayesian method to incorporate prior information about amino acid substitution probabilities and insertion and deletion probabilities (see Section 2.2).

An important component of this Bayesian method is the incorporation of prior information about amino acid distributions that typically occur in columns of multiple alignments. We reported in [58] a method to condense the information in databases of multiple alignments into a mixture of *Dirichlet* densities [13, 12, 53] over amino acid distributions, and to combine this prior information with the observed amino acids from easily recognized homologs to form estimates of the parameters of profiles and HMMs.

[1] http://www.cse.ucsc.edu/research/compbio/sam.html

[2] http://genome.wustl.edu/eddy/hmm.html

With accurate prior information about which kinds of amino acid distributions are reasonable in columns of alignments, it is possible with only a few sequences to identify what other amino acids may occur in a particular environment. The models produced with Dirichlet mixture priors are more effective at generalizing to previously unseen data, and are often superior at database search and discrimination experiments [63, 34, 38, 5, 60, 16]. In the CASP2 experiments, we combined this method with a new sequence-weighting scheme, described in Section 2.4, to tune the models for remote homolog recognition, and to compensate for the overrepresentation of very similar homologs in the HMM training set.

In addition to extending our previous HMM methods, as described in the following section, we developed a number of posthoc analysis tools used to decide among closely scoring predictions. These tools were used to select predictions from among the top candidates. This additional intervention generally helped in identifying the most promising hits, but was occasionally misleading—as were a number of attempts to hand-edit the automatically produced alignments. A summary of the results is given in Section 3.1, with discussion in Section 3.4.

2 Methods

Our general method for predicting the structure of a target sequence involved a two-pronged approach: (1) constructing a target HMM from the target and identified homologs, and searching a sequence version of the Protein Data Base (PDB) with this model, and (2) scoring each target sequence against a library of HMMs constructed on a representative subset of PDB.

Those PDB sequences that were scored well with the target HMM and whose HMM scored the target sequence well were examined more closely using several analysis tools (see Section 2.6).

2.1 Scoring

Two of the main uses for an HMM are sequence alignment and discrimination. Since an HMM is a stochastic process, one can talk about the probability of a sequence *seq* of amino acids being generated by a given HMM, which we will denote $P(seq|\text{HMM})$. Sequences that resemble those for which the model was trained will be given a high probability, and others a low probability. By itself, this probability is not very meaningful. A more informative value is one that compares the likelihood of the sequence under the HMM to the likelihood of the sequence for a simple "null model" [1]. For our work, in the null model each residue is generated independently according to a background distribution over the amino acids. Using the HMM, we compute the log likelihood ratio

$$\ln \frac{P(seq|\text{HMM})}{P(seq|\text{NULL})},$$

which forms a better method of scoring the sequence.

For reasons that are not yet theoretically justifiable, we found that more comparable scores were produced if the length of the database sequence being scored was taken into account. This leads to the adjusted log-likelihood score

$$\ln \frac{P(seq|\text{HMM})}{P(seq|\text{NULL})} - \ln(\text{number of residues in seq}),$$

which is a suitable score to use for database searching with HMMs. For historical reasons, we actually use the negation of this score, which we call the adjusted NLL-NULL score. Thus our score is defined by

$$\text{NLL} - \text{NULL} \quad = \quad -\ln \frac{P(seq|\text{HMM})}{P(seq|\text{NULL})} + \ln(\text{number of residues in database sequence})$$

However, this score does not take into account the size of the database. When searching a very large database, there are many possible places in the database where one has a possibility of finding a match, and thus there is a increased probability that one will find a good log-likelihood ratio for one of these sequences simply "by chance". To address this issue, we make use of Milosavljević's algorithmic significance test, which asserts that the probability of getting a score larger than d is less than or equal to z^{-d} (where z is the base of the logarithm), assuming that the null model is a reasonably accurate description of the space the sequences are drawn from [48]. When a database is searched, and the search includes N individual placements of the model, for a given d, the equation becomes:

$$P(\exists_i, \text{score}_i \geq d) \leq \sum_{i=1}^{N} P(\text{score}_i \geq d) \leq N z^{-d}.$$

Thus, to assure a certain level of significance σ (typically 0.01 to 10.0), a score such that $\sigma \leq N z^{-d}$, or $d \geq -\log_z(\sigma) + \log_z(N)$ will certainly indicate significance, though this significance level is pessimistic as it assumes an independence of placements that does not exist in HMMs. The meaning of σ is roughly the same as BLAST's E parameter, the expected number of false positives [2].

Because we work with natural logarithms, the NLL−NULL is reported in terms of nats. The SAM module `hmmscore` recommends some thresholds to use for significance. In practice, we find that these values are conservative, but do provide a reasonable guideline.

There are two different scoring methods for determining $P(seq|\text{HMM})$, the probability of a sequence given an HMM. The Viterbi score is determined from the most probable path through the model that could generate the sequence. In contrast, the all-paths score is a sum of the probabilities for all possible paths that could have generated a sequence. This is computed by the forward algorithm. The all-paths score gives the true probability of the sequence given the model, but the Viterbi score is faster to compute and provides a useful approximation to the true probability.

For the CASP2 contest, we used all-paths scoring (i.e., the forward algorithm) over local alignments, that is, the paths in the HMM and in the sequence were not constrained to include the ends of the model or the ends of the sequence.

2.2 The HMM library of proteins of solved structure

We originally selected a representative subset of proteins of solved structure from the Protein Data Bank (PDB) [14]. This set was increased to include PDB sequences which scored well against target models during the course of the experiment, and has since been increased to include a more complete set of representative PDB sequences. The library grew from about 800 models in the beginning to about 1000 at the end of the CASP2 contest, and now has 1312 models.

For each of these representative structures, we constructed a hidden Markov model (a *structure model*), using for an initial alignment and training set the alignment of the structure and its close homologs as given in the HSSP database [54]. To refine the HMMs produced from the HSSP alignments (since they omit inserted residues), we re-estimated the initial models in two stages, using the complete SWISSPROT sequences of the homologs in the alignment. In the first stage, model length and match state amino acid distributions were kept fixed, and only the transition probabilities were allowed to change. In the second stage, all model parameters (except for model length) were re-estimated. We found it helpful to turn off the introduction of noise into the parameter estimation process for these procedures. This process generally produced alignments that revealed greater pairwise sequence identity and produced models that gave higher probability to the training sequences. To allow the HMMs to generalize to remotely related proteins, we applied sequence weighting (see Section 2.4) and priors over transition probabilities in various structural environments into the parameter estimation.

The transition priors were estimated from the observed transitions (with sequence weights) in the re-estimated alignments. The alignment columns were grouped into 48 different structural environments (based on secondary structure of the residue and its two neighbors and the solvent accessibility

of the residue), and a single-component Dirichlet prior (pseudocount) transition regularizer was optimized for each environment. The appropriate pseudocount regularizers were employed to estimate the parameters of the final model, allowing us to incorporate general structural information, such as the low probability of an insert in the middle of a helix, into the HMM estimation process.

Finally, we added *FIMs* (free-insertion modules) to the ends of each HMM. FIMs are special HMM architectural elements which can generate (or model) an arbitrary number of residues in a sequence. This allows the most appropriate region of a sequence to align to the HMM. The FIMs are also effective in building HMMs for subregions of training sequences.

While this describes our construction of a library of *whole chain models*, our original intent was to build an HMM library of protein domains. These domains were from a comprehensive list that had been identified in the FSSP database by Liisa Holm. For each of these domains she identified a representative structure and the domain's corresponding position in the structure's amino acid sequence. The construction of a domain HMM began by first training an HMM for the entire representative structure sequence (or chain) which contained the domain. The domain HMM would simply be the result of excising relevant parts of this "whole chain HMM."

Because of time constraints and experimentation with building reliable whole chain HMMs, a satisfactory library of domain HMMs was never constructed.

2.3 Building the target model

Target models were built using the target sequence and purely sequence-based methods. The basic approach was to find a set of fairly close homologs to the target, align the target and the homologs, build a model corresponding to the alignment, then generalize the model to find more distant homologs by using sequence weighting (as explained in Section 2.4).

The SAM suite of tools was used for all alignment and model building, but not with their default parameters. The most important difference is that noise was turned off completely in `buildmodel`, SAM's expectation-maximization parameter estimation module, and all model-building was done by modifying a previously existing model. The models always used a Dirichlet mixture as a Bayesian prior to get estimates of amino acid probabilities, rather than the default pseudocount regularizer.

To find putative homologs, a BLAST search was done at NCBI to find the database entry for the target sequence and the "protein neighbors" in the ENTREZ database were retrieved. It turned out that these neighbors were not always homologs of the target sequence. For example, on target t0012, the proregion of procaricain, the structure of the mature enzyme was already known and many of the "neighbors" were similar to the mature enzyme but did not include the proregion that was the target. In other cases, some of the proteins were "neighbors" only because they were on the same clone as a homologous protein.

The first step of the model building was to select a subset of the putative homologs for use in building the models. First, `modelfromalign` was used to create a model (t_0) from the target sequence having exactly one match state for each position of the sequence. This model was used to score the putative homologs, and only those that scored better than a rather arbitrary *cost threshold* of -10 nats were selected for building a model. Note that the more negative the score, the better the fit to the model.

A new model (t_1) was built by retraining the single-sequence model on the selected sequences. Retraining the model preserved the one-to-one relationship between the residues of the target sequence and the match states of the model. The transitions in the model were set to give low costs to insertions and deletions, so that the sequences could be easily aligned on the conserved regions.

The retrained model was used to select again from the set of putative homologs. Fairly often, a few more sequences were selected to be included for further training. The alignments from the t_1 models generally had too many insertions and deletions, and so they were retrained with a different transition regularizer that made continuing an insertion or deletion cheap, but starting one expensive. The retrained (t_2) models generally produced much cleaner-looking alignments. The conserved blocks were essentially the same, but the variable regions had fewer locations where

insertions and deletions could occur and fewer instances of matching just one or two amino acids in the middle of a variable region.

Because some of the targets could have been domains that can occur repeatedly in a protein, the `multdomain` module of SAM was used with the t_2 models to select subsequences from the putative homolog set that were particularly high-scoring. On target t0004 (the nucleotidyltransferase S1 motif that turned out to have the structure of a cold-shock protein) some of the homologs had 3 or 4 regions that matched the model well.

The t_2 models were retrained on the results of the domain search to create t_3 models. Where there had been only one domain found in each sequence, this retraining made very little difference to the models, but where there were several domains, the t_3 models scored remote homologs somewhat better. The t_3 models were used with `multdomain` to select and align subsequences from the putative homolog set. This alignment was the final alignment used to build the generalized models.

Each sequence in the final alignment was assigned a sequence weight (as described in Section 2.4), so that the average entropy of the match states was 0.3 bits less than the entropy of the background frequency distribution. One iteration of `buildmodel` was done to build the generalized model from the t_3 model and alignment. The transition probabilities were also set so that insertions or deletions were unlikely except in places where they had already occurred in the alignment.

The generalized models were used to score all the sequences for which structures were recorded in the PDB database, and several of the top-scoring sequences were selected for more careful examination. Table 3.1 shows how well the sequences we predicted scored and how well the correct structure scored (for those sequences that had a correct structure in the database).

For some targets (t0011, t0019, t0026, t0030), the initial set of training sequences was too small, and so a search was done of a non-redundant protein database using the generalized model, and the sequences with cost less than -8.0 (a fairly loose threshold) were considered possible homologs. The model-building procedure was repeated for this larger training set, and the results are reported in Table 3.1.

The diversity increased significantly by including this larger set of homologs, except for target t0030 (see Table 2.1—the models after searching the larger database have "-nrp" after their names). The number of sequences sometimes dropped, because the initial training set had identical sequences, and the non-redundant protein database eliminated this duplication. The models for other targets could probably also be improved by doing such a search, but the difference would be small, since they already have a fair amount of diversity in the training set.

When we predicted a high-scoring sequence, it was usually correct (t0002, t0004, t0031, but not t0020). The reasons for failure are examined in Section 3.3.

2.4 Weighting schemes

Almost any set of homologous protein sequences will contain some very similar sequences and some less similar ones. If we construct a model from a set of sequences, the model will tend to favor the most highly represented sequences. Sequences very similar to the most common ones in the training set are easily recognized by the model, but more distant homologs may not be recognized.

Several methods have been proposed for reducing the inherent bias in training data [55, 31, 62]. The two most common methods involve removing from the training set any sequences that are too similar to others in the set (as is done for creating the BLAST substitution matrices, for example), and using sequence weighting to give less weight to sequences which are very similar to others in the set and more weight to unusual ones. The former method is really just a special case of the latter, restricting sequence weights to 0 and 1. We have chosen to use sequence weighting to reduce the training set bias.

When using a Dirichlet mixture to convert amino acid counts to probabilities, the total weight assigned to the set of sequences is an important control parameter. When the total weight is very low, the probabilities are very close to the background probabilities of amino acids. When the total weight is high, the probabilities are very close to the observed frequencies. By adjusting the total sequence weight, one can smoothly interpolate between these two extremes.

There are two opposing goals that determine the range of values that works well for the total weight. First, we need sufficient specificity in the model that it recognizes members of the protein family and rejects proteins that are not members. Second, we need sufficient generality that we recognize members of the family that we have not seen before, even if they are from a subfamily that has not been previously observed.

The optimum balance between these two goals is dependent on how distant a relationship we which to consider "being in the same family". If we wish to distinguish hemoglobins from myoglobins, for example, we need very specific models, but if we want to recognize all TIM barrels, a more general model is called for. That is, the correct sequence weighting does not depend just on the data, but on the use to which the model will be put.

In remote homolog recognition using HMMs or profile methods, we need to estimate models that generalize as much as possible without losing the ability to recognize the training set. This means that the total weight assigned to the sequences in the training set should be fairly small.

We want to assign a larger weight when we have a very diverse set of homologs in the training set (since we have already seen most of the acceptable variation), and a smaller weight when we have a rather homogeneous training set. Rather than have the user assign a total weight to the data, the user creates the desired effect in the final model by specifying the average relative entropy of the match states relative to the background frequency. The sequence weighting computation is done just from the frozen alignment, eliminating any need for knowledge of the model.

The entropy of the background distribution P_0 is

$$H_0 = - \sum_{\text{amino acid } a} P_0(a) \log_2 P_0(a) \ ,$$

while the entropy of a match state or alignment column c is

$$H_c = - \sum_{\text{amino acid } a} \hat{P}_c(a) \log_2 \hat{P}_c(a) \ ,$$

where $\hat{P}_c(a)$ is the mean posterior estimate of the probability of amino acid a under the Dirichlet mixture given the weighted counts of amino acids in column c. The control parameter is the value of $H_0 - H_c$ averaged over all match states (or alignment columns).

Another way of viewing the control parameter is that the user specifies the average number of bits of information each column of the alignment should contain. We sometimes refer to this as the "bits saved" by the model. Related approaches for selecting the right PAM distance in a substitution matrix used for BLAST searches are discussed in [1].

The total weight is set by an iterative algorithm, which guesses a total weight, computes the average entropy, then adjusts the weight. The relative sequence weights can be set in several different ways. The method used for the CASP2 contest is an unpublished method that takes the entropy of each sequence raised to the 10th power as the relative weight, but very similar results would have been obtained by using the Henikoff's position-based weighting scheme [31] for the relative weights.

Table 2.1 shows the number of sequences and total weight assigned to the training sets (the domains found by the t_3 models) for each of the targets we submitted predictions for. The total sequence weight needed to save 1.4 bits per position is a good approximation to the number of "different" sequences in the training set (it is approximately 1 if only one sequence is used).

Some of the targets had very few distinct homologs in the training set (for example, t0011, t0019, and t0026), resulting in poor generalization to remote homologs. For these targets, a search of a non-redundant protein database was performed with the generalized model and a fairly loose threshold to find other potential homologs, and the full model-building procedure was repeated with this larger set of potential homologs. This procedure was not used for targets 27 and 28, because there was a close homolog of the target found in the PDB database, and remote homolog searching was not needed.

target name	number of sequences	weight for 0.3 bits	weight for 1.4 bits
t0002	89	1.1772	16.2566
t0004	83	0.4602	7.8122
t0011	48	0.2285	1.6410
t0011-nrp	96	0.3240	3.5491
t0012	147	1.1880	19.1128
t0019	8	0.2882	1.7553
t0019-nrp	8	0.5275	3.6353
t0020	11	0.4458	3.9479
t0023	14	0.3794	3.1689
t0026	10	0.2329	1.3902
t0026-nrp	5	0.3223	2.7309
t0027	9	0.2815	1.7427
t0028	10	0.3389	2.3813
t0030	18	0.2844	1.5711
t0030-nrp	11	0.2826	1.5689
t0031	13	0.3434	2.4183
t0038	14	0.6331	5.4846

Table 2.1: This table presents the number of sequences in the final alignment for each target model and total weight assigned to the sequences to get 0.3 or 1.4 bits saved per column. The latter number is a good approximation of how many "different" sequences there are in the training set.

For targets with fewer than two different sequences by this measure (except target t0027, which had a close homolog of known structure), a search of a non-redundant protein database was done to try to find more potential homologs, and new models were built with this larger training set (labeled with "-nrp"). For target t0030, the iteration did not increase the diversity of the training set.

2.5 Estimating joint models for two protein families

The structure models (Section 2.2) and target models (Section 2.3) do a good job of identifying somewhat remote homologs, but as the evolutionary distance increases, the alignments of remote homologs to the model get worse, and the discrimination ability of the models is reduced. The highly conserved positions that provide most of the recognition signal are usually aligned well, but the regions of low sequence similarity are often very poorly aligned.

When we align the homologs of a structure to a target model, the proteins may not maintain their mutual alignment (similarly for homologs of a target sequence and a library model). This obviously reduces our ability to predict with confidence the correct pairwise alignment between a target and a structure.

If two sets of proteins share a common structure and evolutionary history, then we ought to be able to construct a statistical model that gives high probability to both sets. We developed two methods for estimating *joint models*. In both methods the joint HMMs were trained on the target sequence, its homologs, the sequence of known structure, and its homologs. By training on both sets of homologs we were able to estimate models that successfully produced multiple alignments of both sets of homologs, retained the mutual alignment within each group, and provided better alignments between the two groups' regions of lower primary sequence identity.

One method for constructing a joint model employed the same method used for building the target model, except that the initial possible homolog set was increased to include the homologs of the desired PDB sequence, and the threshold for acceptance into the training set was lowered so that at least one homolog of the PDB sequence would be included. The t_3 models (not generalized by using sequence weighting) were used to provide the final joint alignment.

The other method retrained an existing model using both sets of homologs. The model length was kept fixed, and sequence weights were assigned to allow roughly equal weight to both groups of sequences. Two joint models were produced with this method—one starting with the library model and the other starting with the target model.

The different joint models usually produced somewhat different alignments. We are in the process of doing a quantitative study to determine which method is best, but preliminary results show no clear winner. All joint models, though, seem to produce superior alignments than the search models trained on just the target homologs or just the structure homologs. For the CASP2 contest, we examined each alignment, looking for agreement between predicted secondary structure for the target (predicted by PhD [51]) and the real secondary structure, reasonable matching of hydrophobicity patterns, residue identity, and solvation scores. This is described further in the following section.

2.6 Posthoc analysis tools

Most of our prediction work involved automated methods to identify potential matches between a target and each of the solved structures in our library and the production of alignments between the target and structure. When the pool of potential matches between a target and solved structures had been narrowed down to a reasonable number by automated methods, it was narrowed down further using the analysis tools described below.

Prospective alignments of targets and structures were checked using a solvation analysis tool written by Liisa Holm. This score analyzes how stable the prospective alignment would be given the exposed positions and given the hydropathy of each residue. Ultimately, we learned that this type of analysis is very sensitive to small errors in alignments. Due to this sensitivity, it was not always clear if an alignment was somewhat incorrect, if the match between target and structure was correct, or whether the match between target and structure was simply wrong.

Our methods did not incorporate secondary structure prediction into the initial screening for matches between targets and structures. However, when the number of potential matches had been narrowed down sufficiently, we checked the secondary structure at each position in the pairwise alignments and observed how well this prediction correlated with the PHD secondary structure.

Finally, we checked the plausibility of each of the remaining alignments using SAE, a graphical tool combining RASMOL with an alignment viewer, written by Leslie Grate. Under SAE, we would check that insertions and deletions occurred in reasonable places and that the resulting protein structures were compact and contiguous.

3 Results

3.1 Fold-recognition results

Table 3.1 shows how our HMM models scored on the thirteen targets for which we submitted predictions. Targets t0027 and t0028 had a close homolog of known structure in the training sets for the target models, which accounts for the extremely high scores. Target t0030 was the only one for which we were sure enough of our methods to predict that the fold was a new one, and not that we had just missed finding the fold.

Because the methods for building the target models and scoring the library evolved over the course of the CASP2 contest, these results are from models built using the most recent methods.

From this table, we can see that when there was a similar fold in the existing PDB databse, the target model scored one of the similar folds within the top 25 (out of 7991 sequences) and one of the very similar folds within the top 100, except for target t0012, which had only weakly similar folds and did not score them highly.

For the library models, we see a similar phenomenon—there is a similar fold in the top 15 (out of 1312 models) and a very similar one in the top 60 (except for t0012 again).

target	structure	target		library		DALI
		cost	rank/7991	cost	rank/1312	rescaled
t0002	1ubsB	-36.008	4	-17.037	2	1.00
t0002	1ttpB/1ttqB	-38.372	2,3	-17.188	1	1.00
t0002	1wsyB†	-38.594	1	-17.022	3	1.00
t0004	1csp†	-7.114	1	-0.948	223	1.00
t0004	1mjc†	-5.382	7	-2.628	24	1.00
t0011	1grl†	-1.999	1656	-2.660	55	0.00
t0011	3gapB†	-3.532	197	-5.997	3	0.00
t0011	1frpA†	-3.115	505	-5.887	4	0.00
t0012	1mdyA†	-0.649	3173	-6.047	2	0.00
t0012	1pht†	-2.052	302	-2.090	85	0.00
t0012	1atr	-0.697	2829	-0.726	432	0.22
t0012	1gerA	-1.921	372			0.28
t0019	1klnA†	-0.937	2919	-1.894	77	
t0019	1ribA†	-2.040	685	-2.473	35	
t0019	1pdnC†	-3.589	881	-2.674	27	
t0020	1arv†	-7.525	2	-1.359	319	0.00
t0020	1tahA	-2.490	1527	-1.441	288	0.35
t0020	7aatA	-1.786	2903	-4.977	4	0.43
t0020	1xad	-3.264	536	-1.316	334	0.45
t0020	1scuB	-2.741	1130	-3.068	42	0.55
t0020	1ecl	-1.471	3821	-2.341	93	0.60
t0020	2dln	-5.595	24	-0.614	705	0.82
t0020	1minA	-4.854	67	-2.667	64	1.00
t0023	1ubsA†	-2.659	335	-2.109	105	
t0026	1hstA†	-0.010	7097	-1.012	106	
t0026	1scmB†	-0.471	3492	-5.495	2	
t0026	1htmB†	-0.815	1417	-1.845	33	
t0026	1top†	-3.201	39	-3.497	9	
t0026	1cmg†	-2.834	53	-4.775	5	
t0027	1pcl	-20.108	2	-63.398	1	
t0027	2pec†	-286.850	1	-12.331	2	
t0028	1celA†	-321.974	1	-282.178	1	
t0030	2hwf1†	-1.218	805	-0.389	446	0.00
t0030	1hsbA†	-0.321	4254	-2.923	7	0.00
t0030	NONE†					1.00
t0031	1fon(A,B)	-10.798	1,2			1.00
t0031	1hcgA	-1.885	990	-12.285	6	1.00
t0031	4ptp	-1.828	1085	-14.918	3	1.00
t0031	1elt†	-8.568	4	-11.341	7	1.00
t0031	1mctA†	-2.143	816	-14.976	1	1.00
t0031	1try†	-10.487	3	-13.118	5	1.00
t0038	1exg†	-2.985	200	-0.857	217	0.57
t0038	1lpaB,1lpbB	-5.242	1,2	-0.725	268	0.85
t0038	1celA	-0.809	2897	-2.390	22	1.00
t0038	1bglA	-4.309	75	-0.969	181	1.00
t0038	2ayh	-0.760	3082	-0.266	705	1.00

Table 3.1: This table shows how the generalized target models and library models scored the sequences that we predicted (marked with †) and some of the lowest-cost sequences which DALI [33] considered to have a similar structure. Where DALI scores are available, we have listed the structures in increasing order of similarity to the known structure. The DALI scores are rescaled so that $Z \leq 2$ is 0 and $Z \geq 6$ is 1. The rank numbers are from the October 1996 version of the PDB database and the April 1997 version of our HMM library. Ranks are somewhat inflated by redundancy in the database and the library (e.g., there are 5 sequences identical to 1csp in the PDB database, so the rank of 7 for 1mjc would be 3 in a non-redundant database).

Our manual attempts to combine the results from the two sets of searches and to use PhD-predicted secondary structure to confirm or reject the matches were fairly successful in reducing the number of predicted matches.

For targets t0002, t0004, and t0031, all the predicted structures were excellent structural matches, and the top-scoring match was correct in the target model. The top-scoring match was correct in the library for t0002 and t0031, but for t0004, the first correct match was rank 4.

For target t0002, like most groups, we misunderstood the rather cryptic comment about partial homology and only predicted the domain homologous to 1wsyB for which simple sequence methods already provided an adequate prediction of homology. We had made some attempts to predict the other domain, but we did not come up with a prediction sufficiently believable to be submitted. We hope that future contests label targets more clearly when partial prediction is desired.

For t0038, the prediction 1exg was an adequate match, though there were much better matches in the database. The top-scoring match to the target model (1lpaB) would have been a better prediction, and there was an even better match in the top 22 for the library (1celA). We'll discuss what went wrong for this prediction in Section 3.3.

For targets t0011 and t0030, there were no good structural matches, and we had only weak predictions. Indeed, for t0030 we decided to put 80% of our "bet" on this being a new fold. We might have done so for t0011 as well, but that early in the summer we were not sure enough of our methods to predict something as a new fold.

For target t0012, only rather poor structural matches existed in the database. Our predictions for this target (1mdyA and 1pht) scored well in the library, but not in the target model. 1mdyA is a helix-turn helix, which, in terms of secondary structure, is aligned very well by our 1mdyA structure model to a helix-turn-helix in t0012. Unfortunately, the angle between the helices does not match closely enough to superimpose the structures well.

For target t0020, our incorrect prediction (1arv) scored well with the target model, but not in the library. There were high-scoring correct structures, which we should have reported. We explain what went wrong for target t0020 in Section 3.3.

For targets t0019, t0023, and t0026, we still have not gotten any feedback on what the correct folds are, and so we do not know how well we did.

For t0019, we predicted three possibilities (1ribA, 1pdnC, and 1klnA). All three score well in the library but poorly with the target model, and so are not expected to be very good structural matches.

For t0023, we predicted just 1ubsA (a TIM barrel), but this scored poorly for both the target and the library models. There were several high-scoring TIM barrels with both the target model and the library, but we did not have time to refine the alignments for them. We're still fairly confident that t0023 is a TIM barrel, though there are almost certainly better matches that 1ubsA in the database.

All five predictions for t0026 score well in the library, but only 1top and 1cmg score well in the target model.

Targets t0027 and t0028 were not fold-recognition targets—the fold was known and we were just attempting to find better alignments.

3.2 Quality of alignments

We submitted alignments for ten fold-recognition targets and three comparative modeling targets (counting t0002 as a comparative modeling target). Results are available for eight of those targets, and Table 3.2 summarizes our alignments for the three comparative modeling targets (t0002, t0027, and t0028) and the two fold-recognition targets for which we identified a correct fold (t0004 and t0031).

Our method searched for global, rather than local, alignments between a target and a structure. While the average shift in the alignments is generally quite low, this resulted in alignments with a higher RMS distance compared to other groups. Loop regions have a high degree of divergence, and so identifying these regions and removing them from the alignment would improve the evaluation of our alignments—this improvement is one we hope to have implemented by the next CASP contest.

target	structure		Alignment Length	Residues Aligned Correctly	Avg. Shift	Avg. RMSD	SC%ID	%ID	Alignment Specitivity	Alignment Sensitivity
t0002	1wsyB	VAST	245	117	1.316	5.15	20.07	24.08	47.76	51.09
t0004	1csp	VAST	63	34.80	0.338	3.52	24.53	29.37	55.24	65.66
t0004	1csp	DALI	63	34.80	0.450	3.52	21.31	29.37	55.24	57.05
t0004	1csp	SSAP	63	28.80	0.826	3.52	25.00	29.37	45.71	49.66
t0004	1mjc	VAST	62	36.74	0.300	3.64	26.92	23.20	58.79	70.77
t0004	1mjc	DALI	62	39.14	0.471	3.64	22.58	23.20	62.62	63.23
t0004	1mjc	SSAP	62	36.74	0.593	3.64	18.00	23.20	58.79	57.50
t0027	2pec	VAST	319	85	3.596	14.40	18.26	24.14	26.65	38.81
t0027	2pec	DALI	319	99	3.938	14.40	21.93	24.14	31.03	36.80
t0028	1celA	VAST	359	319	0.188	2.37	46.97	49.30	88.86	91.93
t0028	1celA	DALI	359	342	0.205	2.37	48.74	49.30	95.26	95.80
t0031	1elt	VAST	200	95	2.567	8.73	14.44	15.50	47.50	52.78
t0031	1elt	DALI	200	111	2.427	8.73	13.90	15.50	55.50	59.36
t0031	1elt	SSAP	200	59	0.600	8.73	3.00	15.50	29.50	69.41
t0031	1mctA	VAST	195	99	2.263	8.77	14.44	20.51	50.77	55.00
t0031	1mctA	DALI	195	105	2.437	8.77	15.85	20.51	53.85	57.38
t0031	1mctA	SSAP	195	58	0.481	8.77	14.00	20.51	29.74	70.73
t0031	1try	VAST	198	99	1.270	7.45	16.95	18.69	50.00	55.93
t0031	1try	DALI	198	101	1.624	7.45	17.74	18.69	51.01	54.30
t0031	1try	SSAP	198	46	1.301	7.45	20.00	18.69	23.23	57.50

Table 3.2: This table compares our alignments of the targets to the structural alignments produced by VAST, DALI, and SSAP respectively. *Alignment length* refers to the total number of residues aligned, including loop regions. *Residues Aligned Correctly* describes the number of positions in which the alignment was correct, as compared to the structural alignments. *Avg. RMSD* and *Avg. Shift* refer to the average RMS deviation and shift, as computed by the assessors. *SC%ID* describes the percent residue identity for each structural alignments, and *%ID* describes the percent residue identity of our alignment. *Aligment Specificity* and *Alignment Sensitivity* refer to the number of correctly aligned residues as a fraction of the number aligned in the prediction and the number aligned in the structural alignment, respectively.

Perhaps the most striking difference between the RMS measure and the other measures of correctness for the alignment is for our t0031 predictions. The number of exactly correct residues is high and the average shift is quite low, but the RMS deviation is suprisingly high. This results from a single segment (residues 163-183) which is badly misaligned (see Figure 3.1). The segment should be aligned to residues 149-168 of 1mctA which includes the edge strand of a conserved beta sheet. Instead the edge strand was skipped and the segment was aligned to a loop and helix on the surface. This misalignment should have been detected before we submitted the prediction, since it results in a large distance between the predicted positions of residues 162 and 163, but we failed to notice the problem.

Our alignment for t0027 and 2pec was reasonable in the beta sheets of the core, but we included alignments for the rather variable surface helices which turned out to be different in the two structures. Trimming our global alignment to remove the surface elements would have considerably improved the statistics for the prediction.

It is interesting that we got better alignments for t0004 and t0031, which were classified as a fold-recognition target, than for t0027, which was classified as a comparative modeling target. Perhaps in future CASP contests, the targets should not be pre-classified, but all targets should be made available for all prediction types. The assessment for each type of prediction can then focus on the targets that show a difference between the predictors. Existing servers for sequence-based alignment can be used as baseline comparisons to see whether the more sophisticated methods provide better results on the easy targets.

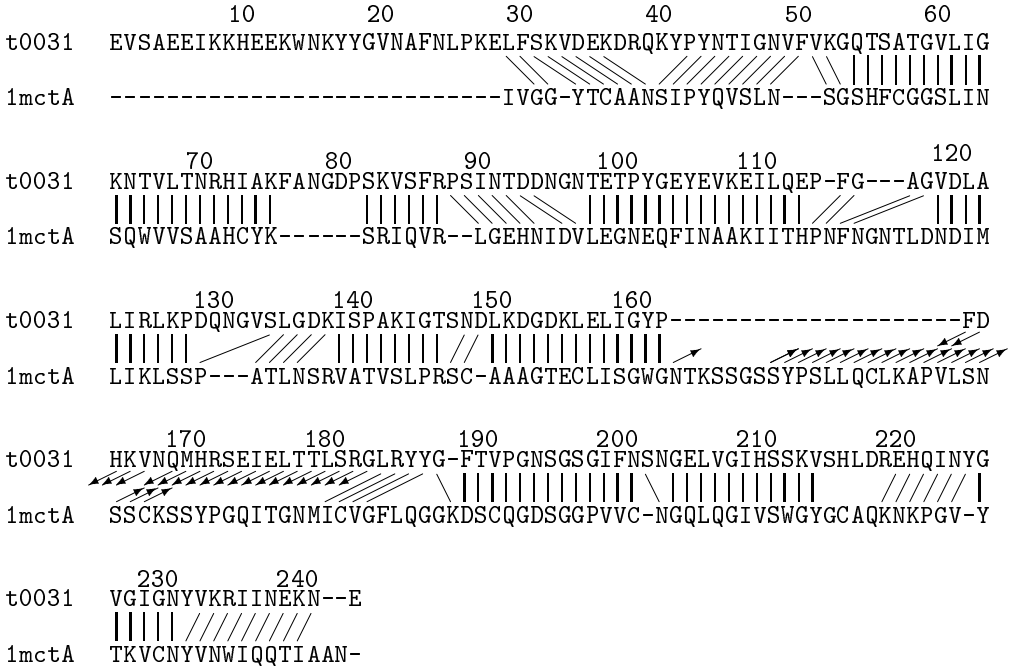

Figure 3.1: The alignment we predicted for t0031 and 1mctA, with bars indicating positions aligned by the structure-structure aligner DALI. The numbers are the residue numbers in t0031. Most of the segments are shifted by only one or two, but the segment from 163 to 183 is shifted by 16 residues, as indicated by the arrows.

3.3 Mistakes

Since we started the summer of 1996 with no experience in predicting protein structure, we had to develop our methods on the fly, learning as we went. Because of the tight time constraints, we did not have time to carefully evaluate each of our methods before applying it to the targets. We are now going back and doing quantitative studies of our search and alignment techniques for remote homology detection and structure prediction, but expect these studies to take several more months to complete.

This section contains some of the things we learned over the summer, and some things we did not learn until the true structures were known.

One thing we learned fairly early was that the "protein neighbors" in the Entrez database is a somewhat noisy source for putative homologs. We found three sources for error here:

- Sequences that had high similarity to the full protein, but not to the domain in the target sequence. This was a particular problem for target t0012, the proregion of procaricain, since many of the "neighbors" were similar to the mature enzyme, but did not contain the proregion of interest.
- Sequences that had no discernable similarity to target. For the most part these seemed to be database errors in which proteins coded for on the same clone as a homologous protein also got reported as neighbors.
- Sequences with weak similarity to a portion of the target, but not easily alignable for the rest. These sequences probably represented different structures that shared a short motif. Luckily there were not many of them.

The model-building techniques for the target models were modified to reject from the initial set of possible homologs any that did not fit the model being built. This may have cost us a few genuine remote homologs, but did prevent the trash from destroying the specificity of the models.

The protein neighbors found by Entrez were often not a very complete set. We could find a larger set of homologs by using the generalized target model to search a non-redundant protein database,

but because this was not completely automated, we only did this for a few cases, where the initial set of homologs lacked diversity.

Another problem with our methods is that using sequence weight to generalize our models results in very low total sequence weight, and so very little information gain in any residue position. The sequence-weighting method already gives more importance to positions that are conserved than ones that are variable, but does not distinguish between conservation due to accidents of evolution and conservation due to functional or structural requirements.

We would have liked to have had available a weighting scheme that could increase or decrease the weights of the residues that align to specific positions in the model, so that we could require better matches for active-site residues or other positions for which we had biological or chemical evidence that high conservation was required. Software to do this was not available to us during the CASP2 contest, but has since been developed and seems to be very useful in remote homology detection—we hope to incorporate this sort of information in future structure predictions.

We had originally hoped to use the negative log-likelihood costs reported by SAM directly in choosing between library models, but we found that some models scored all sequences substantially better than other models. For example, a model for a coiled-coil structure scored all sequences containing helices very highly. We computed an average score for each model and subtracted it off from the raw score, in order to get more comparable numbers. Unfortunately, time constraints prevented us from using a large database of scambled sequences to compute this average score, and so we just used the target sequences of the CASP contest to compute it, since we had to score them anyway. Clearly, we should compute a less biased normalization for each model.

Our alignments, because they were constructed from joint models, generally aligned all the residues of either the target protein or of the structure template (whichever was the basis for the joint model). We knew that the loop regions were highly variable and unlikely to be alignable, but did not have the time to identify the regions that were most likely to be misaligned and remove them from the alignment. Doing so would undoubtedly improve the rms distance measure of alignment quality, and is probably an important step to take before using homology modeling to construct 3D structures. The HMMs can provide indirect information about whether the residues in a given alignment column are all from a single type of structural environment or from different environments, and this information could be used to trim the alignment down to the core elements.

Two targets that we feel we should have been able to make better predictions for are targets t0020 (ferrochelatase) and t0038 (CBDN1).

For target t0020, we did consider 1minA (an excellent structural match), 2dln and 1ecl (structurally somewhat similar), and more distantly related structures such as 1xad and 1tahA. Joint models were built for 1minA, 1ecl, 1xad, and 1tahA, and both 1ecl and 1tahA were considered excellent candidates at some point in our analysis. We ended up concentrating on 1arv and related proteins, because of the perceived need for an iron-binding site and because of functional relationships.

For unknown reasons, we never examined 1lpaB for target t0038, even though it was our highest scoring sequence in the target model, and turned out to be a somewhat similar structure. We also did not examine 1bglA (which turned out to be a very similar structure), though we looked at several sequences that scored worse. Note: the sequence we predicted, 1exg, is structurally somewhat similar to t0038, but 1lpaB and 1bglA are much better and scored much higher with the target model. We also did not consider 1celA, though it scored very well with the library models.

We did consider 2ayh (which is an excellent structural match), though it scored very poorly in the target model. We created an alignment for 2ayh with high residue identity and good agreement with PhD using the methods of Section 2.5. Unfortunately, we rejected this good alignment based on a too strict interpretation of solvation scores.

For target t0038, we clearly did a very poor job of our post hoc analysis, as we did not consider several candidates that turned out to be correct, despite their high scores. Even when we considered one of the correct structures, we rejected it due to a misinterpretation of the meaning of the solvation scores, which are only suitable for choosing between alignments to the same structure, not for choosing between structures.

3.4 Conclusions

Overall, the results of this CASP2 experiment show that fold recognition and alignment by HMMs show some promise, but there were too few targets with clear structural homologs in PDB to yield enough experimental tests of the method to draw a definitive conclusion. Some evidence is given that the method may be effective in cases where the residue identity between the target and the sequence of known structure is in the 15–25% range, which brings us some distance into the "twilight zone." However, no evidence is given that the method will be effective in harder cases, where the residue identity is less than 15%.

Even if the current method is not effective in these harder cases, which we suspect may be true, and more sophisticated methods are required, the HMM method still has the advantage that it is computationally efficient in comparison to threading methods, and makes minimal use of structure information, so it can also be used with little modification to search for remote homologs of protein families that contain no sequence with a known structure.

Acknowledgments

This work reflects the contributions of many, including Leslie Grate, Chris Tarnas, Ole Winther, Rachel Karchin, Marc Hansen, Mark Diekhans, Rey Rivera, Tony Fink, and Lydia Gregoret.

This work was supported in part by NSF grants CDA-9115268, IRI-9123692, and BIR-9408579; DOE grant 94-12-048216; ONR grant N00014-91-J-1162; NIH grant GM17129; NSF and GANN Fellowships; and the UCSC Division of Natural Sciences.

References

[1] S. F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *JMB*, 219:555–565, 1991.

[2] S. F. Altschul et al. Issues in searching molecular sequence databases. *Nature Genetics*, 6:119–129, 1994.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lippman. Basic local alignment search tool. *JMB*, 215:403–410, 1990.

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

[5] T. L. Bailey and C. Elkan. The value of prior knowledge in discovering motifs with MEME. In *ISMB-95*, pages 21–29, Menlo Park, CA, July 1995. AAAI/MIT Press.

[6] P. Baldi and Y. Chauvin. Smooth on-line learning algorithms for hidden Markov models. *Neural Computation*, 6(2):305–316, 1994.

[7] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure. Adaptive algorithms for modeling and analysis of biological primary sequence information. Technical report, Net-ID, Inc., 8 Cathy Place, Menlo Park, CA 94305, 1992.

[8] G. J. Barton and M. J. Sternberg. Flexible protein sequence patterns: A sensitive method to detect weak structural similarities. *JMB*, 212(2):389–402, 1990.

[9] A. Bateman. Fibronectin type III domains in yeast detected by a hidden Markov model. *Current Biology*, 6(12):1544–1547, December 1996.

[10] A. Bateman. The structure of a domain common to archaebacteria and the homocystinuria disease protein. *Trends in Biochemical Sciences*, 22(1):12–13, January 1997.

[11] A. Bateman, S. Eddy, and C. Chothia. Members of the immunoglobulin superfamily in bacteria. *Protein Sci.*, 5:1939–1941, 1996.

[12] J. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, New York, 1985.

[13] J. Bernardo and A. Smith. *Bayesian Theory.* John Wiley and Sons, first edition, 1994.

[14] F. Bernstein, T. F. Koetzle, G. J. Williams, E. E. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: a computer-based archival file for macromolecular structures. *JMB*, 112:535–542, 1977.

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

[16] M. P. Brown, R. Hughey, A. Krogh, I. S. Mian, K. Sjölander, and D. Haussler. Using Dirichlet mixture priors to derive hidden Markov models for protein families. In L. Hunter, D. Searls, and J. Shavlik, editors, *ISMB-93*, pages 47–55, Menlo Park, CA, July 1993. AAAI/MIT Press.

[17] S. H. Bryant and C. E. Lawrence. An empirical energy function for threading protein sequence through the folding motif. *Proteins: Structure, Function, and Genetics*, 16(1):92–112, May 1993.

[18] P. Bucher, K. Karplus, N. Moeri, and K. Hoffman. A flexible motif search technique based on generalized profiles. *Computers and Chemistry*, 20(1):3–24, Jan. 1996.

[19] G. A. Churchill. Stochastic models for heterogeneous DNA sequences. *Bull Math Biol*, 51:79–94, 1989.

[20] J. Dalgaard, M. Moser, R. Hughey, and I. Mian. Statistical modeling, phylogenetic analysis and structure prediction of a protein splicing domain common to inteins and hedgehog proteins. *Journal of Computational Biology*, 4:193–214, 1997.

[21] R. F. Doolittle. *Of URFs and ORFs: A primer on how to analyze derived amino acid sequences.* University Science Books, Mill Valley, California, 1986.

[22] S. Eddy. Multiple alignment using hidden Markov models. In C. Rallings et al., editors, *ISMB-95*, pages 114–120, Menlo Park, CA, July 1995. AAAI/MIT Press.

[23] S. Eddy. Hidden Markov models. *Curr. Opin. Struct. Biol.*, 6(3):361–365, 1996.

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

[25] D. Eisenberg. Into the black of night. *Nature Structural Biology*, 4:95–97, Feb 1997.

[26] D. Fischer and D. Eisenberg. Protein fold recognition using sequence-derived predictions. *Protein Sci.*, 5(5):947–955, May 1996.

[27] M. Gribskov, R. Lüthy, and D. Eisenberg. Profile analysis. *Methods in Enzymology*, 183:146–159, 1990.

[28] M. Gribskov, A. D. McLachlan, and D. Eisenberg. Profile analysis: Detection of distantly related proteins. *PNAS*, 84:4355–4358, July 1987.

[29] B. Hazes. The (QxW)3 domain: A flexible lectin scaffold. *Protein Sci.*, 5(8):1490–1501, August 1996.

[30] S. Henikoff and J. G. Henikoff. Automated assembly of protein blocks for database searching. *NAR*, 19(23):6565–6572, 1991.

[31] S. Henikoff and J. G. Henikoff. Position-based sequence weights. *JMB*, 243(4):574–578, Nov. 1994.

[32] S. Henikoff, J. C. Wallace, and J. P. Brown. Finding protein similarities with nucleotide sequence databases. *Methods Enzymol.*, 183:111–132, 1990.

[33] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *JMB*, 233(1):123–138, 5 Sept 1993.

[34] R. Hughey and A. Krogh. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS*, 12(2):95–107, 1996.

[35] K. F. J. Henderson, S. Salzberg. Finding genes in human DNA with a hidden Markov model. In *ISMB-96*, St. Louis, 1996. AAAI.

[36] R. L. Jernigan and D. G. Covell. Conformations of folded proteins in restricted spaces. *Biochemistry*, 29(13):3287–94, Apr. 1990.

[37] D. Jones and J. Thornton. Protein fold recognition. *J. Comput. Aided Mol. Des.*, 7:439–456, 1993.

[38] K. Karplus. Regularizers for estimating distributions of amino acids from small samples. In *ISMB-95*, Menlo Park, CA, July 1995. AAAI/MIT Press.

[39] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *JMB*, 235:1501–1531, Feb. 1994.

[40] A. Krogh, I. S. Mian, and D. Haussler. A Hidden Markov Model that finds genes in *E. coli* DNA. *NAR*, 22:4768–4778, 1994.

[41] D. Kulp, D. Haussler, M. Reese, and F. Eeckman. A generalized hidden Markov model for the recognition of human genes in DNA. In *ISMB-96*, St. Louis, June 1996. AAAI Press.

[42] E. S. Lander and P. Green. Construction of multilocus genetic linkage maps in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 84:2363–2367, 1987.

[43] R. Lathrop, L. Ljubomir, R. Nambudripad, J. White, L. L. Conte, B. Bryant, and T. Smith. Threading through the levinthal paradox. *Nature*, To appear.

[44] R. H. Lathrop and T. F. Smith. A branch-and-bound algorithm for optimal protein threading with pairwise (contact potential) amino acid interactions. In *Proceedings of the 27th Hawaii International Conference on System Sciences*, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[45] C. Lemer, M. Rooman, and S. Wodak. Protein structure prediction by threading methods: evaluation of current techniques. *Proteins: Structure, Function, and Genetics*, 23:337–355, 1995.

[46] R. Lüthy, A. D. McLachlan, and D. Eisenberg. 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, 1991.

[47] V. N. Maiorov and G. M. Crippen. Learning about protein folding via potential functions. *Proteins*, 20(2):167–73, Oct. 1994.

[48] A. Milosavljević and J. Jurka. Discovering simple DNA sequences by the algorithmic similarity method. *CABIOS*, 9(4):407–411, 1993.

[49] G. N., T. J.L., and J. D.T. Using evolutionary trees in protein secondary structure prediction and other comparative sequence analyses. *JMB*, 263(2):196–208, October 25 1996.

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

[51] B. Rost. Phd: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol.*, 2666:525–39, 1996.

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

[53] T. J. Santner and D. E. Duffy. *The Statistical Analysis of Discrete Data*. Springer Verlag, New York, 1989.

[54] R. Schneider and C. Sander. The HSSP database of protein structure-sequence alignments. *NAR*, 24(1):201–205, 1 Jan 1996.

[55] P. Sibbald and P. Argos. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *JMB*, 216:813–818, 1990.

[56] M. Sippl. Knowledge-based potentials for proteins. *Curr. Opin. Struct. Biol.*, 5:229–235, 1995.

[57] M. J. Sippl. Boltzmann's principle, knowledge-based mean fields and protein folding: an approach to the computational determination of protein structures. *Journal of Computer-Aided Molecular Design*, 7(4):473–501, Aug. 1993.

[58] K. Sjölander, K. Karplus, M. P. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *CABIOS*, 12(4):327–345, 1996.

[59] C. M. Stultz, J. V. White, and T. F. Smith. Structural analysis based on state-space modeling. *Protein Sci.*, 2:305–315, 1993.

[60] R. L. Tatusov, S. F. Altschul, and E. V. Koonin. Detection of conserved segments in proteins: Iterative scanning of sequence databases with alignment blocks. *PNAS*, 91:12091–12095, Dec. 1994.

[61] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties, and weight matrix choice. *NAR*, 22(22):4673–4680, 1994.

[62] J. D. Thompson, D. G. Higgins, and T. J. Gibson. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *CABIOS*, 10(1):19–29, 1994.

[63] J. T. L. Wang, T. G. Marr, D. Shasha, B. Shapiro, G.-W. Chirn, and T. Y. Lee. Complementary classification approaches for protein sequences. *Protein Engr.*, to appear 1996.

[64] M. S. Waterman and M. D. Perlwitz. Line geometries for sequence comparisons. *Bull. Math. Biol.*, 46:567–577, 1986.

[65] D. R. Westhead, V. P. Collura, M. D. Eldridge, M. A. Firth, J. Li, and C. W. Murray. Protein fold recognition by threading: comparison of algorithms and analysis of results. *Protein Engineering*, 8(12):1197–1204, 1995.

[66] J. V. White, C. M. Stultz, and T. F. Smith. Protein classification by stochastic modeling and optimal filtering of amino-acid sequences. *Mathematical Biosciences*, 119:35–75, 1994.

[67] S. Wodak and M. Rooman. Generating and testing protein folds. *Curr. Opin. Struct. Biol.*, 3:247–259, 1993.