Chapter 15

Predicting protein structure using SAM, UCSC's hidden Markov model tools

Kevin Karplus

This is a pre-print of a chapter from the book Protein Structure Prediction: Bioinformatic Approach *Published by International University Line, La Jolla, CA, 2001. www.iul-press.com*

15.1 A naive view of protein structure prediction

A complete solution of the "protein folding problem" would take the amino acid sequence of a protein and produce its native fold(s) from physical first principles, getting not only the correct structure, but the entire energy landscape for the protein, and consequently all other low-energy conformations and the dynamic behavior of the protein.

There are only five atom types in the standard amino acids (hydrogen, carbon, oxygen, nitrogen, and sulfur), so if these atoms and the interactions between them could be adequately modeled, it should not require too many physical parameters to get a complete solution of the folding problem.

This problem is intellectually important, but is much too difficult for our current knowledge and computational capacity. In practice, no one has made this approach work for anything more than toy problems. (More precisely, there have been occasional successes on small proteins, but not any consistent success.)

There are several problems with this naive view:

1. We can't currently model the various atomic interactions with sufficient accuracy. Because real proteins have evolved to be only barely stable at their operating temperatures, the energy computations involve subtracting terms that nearly cancel. Relatively small errors in any part of the computation completely change where in the conformation space the min-

ima are located, thus changing which conformations of the protein appear to have the minimum free energy.

2. The energy function must model phenomena such as Van der Waals attraction and repulsion of atoms. The very narrow range of distances over which interactions like this are strongly favorable means that small changes in conformation result in large changes in free energy. That is to say, the energy function is very rough, with many local minima.

3. A typical protein domain has about 200 residues and 600 degrees of freedom (two degrees of freedom for each residue for the ϕ and ψ torsion angles and about one per sidechain). The solvent around the protein also needs accurate modeling, raising the number of degrees of freedom by six for each of hundreds of water molecules. Searching such a high-dimensional space with a fractal energy function takes enormous amounts of computation—far beyond current computer capability.

Because this pure *ab initio* approach to protein folding seems out of reach for a long time to come, but protein structure prediction is needed now, we must adopt a less purist approach to make progress.

Although one approach that has been tried is to approximate the energy function with smoother, more optimizable functions, the most pragmatic approach is to use all available information about a protein—not just its amino-acid sequence—in order to predict its structure. The most voluminous information is sequence information—the high-throughput DNA sequencing projects continue to produce an exponentially expanding supply of sequence information from a variety of organisms. Also increasing rapidly (though not as rapidly as the sequence information) is the database of solved protein structures. Other, more specific, information (such as disulfide bridges, active-site residues, proximity from cross-linking experiments, chemical protections, ...) may be known for specific target proteins.

Our goal is to make the best use we can of this wealth of information.

15.2 Fold recognition

The most successful structure-prediction techniques rely on fold recognition—finding a known structure (the *template*) that is similar to the protein whose structure is to be predicted (the *target*) and aligning the target sequence to the template sequence. The expectation is that aligned residues will play similar roles in the protein, producing similar structures. If finding the template is trivially done with simple sequence-based search (such as BLAST [1] or FASTA [2]), then the problem is referred to as *homology modeling*, and the emphasis is mainly on the quality of the alignment. Here we'll concentrate on those cases where the template is not trivially found.

Fold recognition reduces the dimensionality of our search enormously. Instead of 100s or 1000s of degrees of freedom, we have only a few thousand

templates to examine. There may be many possible alignments of a target to a template, but if we use efficient alignment procedures, we need only examine a few alignments for each template.

This chapter will focus on one fairly successful approach to fold recognition that relies on hidden Markov models (HMMs) for both selecting the template and for aligning the target to the template. One of the advantages of HMMs is that they allow the same efficient algorithms to be used for both tasks, while providing a rigorous probabilistic framework. The technique has been used successfully in three of the Critical Assessment of Structure Prediction (CASP) experiments. [3, 4, 5]

In its most basic form, a hidden Markov model, M, is just a *stochastic model* for a family of protein sequences, that is, it is an efficient way to compute a probability density function over all possible amino-acid sequences, $P_M(\text{seq})$.

Stochastic modeling of proteins is useful, because natural proteins are not designed, but are the results of a stochastic process of mutation and selection. If we can combine the information from several different proteins that result from the same ancestral sequence and essentially the same selection pressures, then we can have a much clearer picture of what is being selected for and what other proteins might result from the same (or sufficiently similar) process.

The stochastic modeling technique can be applied in two ways: we can model the family of the target sequence and search for a template belonging to that family, or we can build a library of template models and see if the target fits any of them. Both methods can be combined, to reinforce the weak signal from remotely related proteins.

An HMM is not a fully general stochastic model of a protein. Many important effects, such as long-range interactions and disulfide bridges cannot be captured in an HMM, which treats each position of the protein sequence as essentially independent of the other positions. We accept this weakness of the modeling power in return for very efficient algorithms for doing computation with the HMMs. Many attempts have been made to create models that capture the tertiary interactions, most notably "threading" models that use pairwise terms [6, 7, 8, 9, 10], but these have met with limited success [11] and relatively high computational cost [12]. Other than the disulfide bridges, the pairwise terms appear not to contain as much additional information as is commonly believed [13].

To build a good HMM of the family of a protein, we need several representatives of the family. Ideally, we would like to have sequences that represent the full diversity possible for the family. Note that we have a bit of a circular problem here—our goal is to find a template sequence similar to the target sequence, but we need to know what sequences are similar to the target to build the HMM that we will search with.

As with other such circular problems, we can avoid the problem by using iteration. We first build an HMM based on a single sequence, and use that model to find similar sequences in a large database of proteins. These sequences can be used to build a better stochastic model of the protein family, which is used in turn for a new search and so on. This iterated search technique for building

multiple alignments is the basis for the popular PSI-BLAST tool at NCBI [14] and for the SAM-T98 [15], SAM-T99, SAM-T2K [5], ... techniques developed at the University of California, Santa Cruz.

15.3 Hidden Markov models

Let's look now at how a hidden Markov model (HMM) represents a protein family, how it is used for searching a database, how an alignment is made with an HMM, and how an HMM can be created from a family of protein sequences. The method described here is used with only minor variations in SAM-T98, SAM-T99, and SAM-T2K. The differences between the methods are mainly in the parameter settings and in minor improvements in the readability and maintainability of the scripts.

There are two approaches for explaining HMMs: for computer scientists and computer engineers, we start with finite-state machines and add probabilities, and for biologists and chemists, we start with multiple alignments. Although the first approach is more general, we are primarily interested in profile HMMs for this chapter, and so will take the second approach. A *profile* HMM is an HMM with a particularly simple structure explained in the next few paragraphs.

We can view a profile HMM as a condensed summary of the information in a multiple alignment of a family of proteins. It does not contain all the information of the multiple alignment—any information about individual sequences is lost, so procedures that rely on individual sequences (such as phylogenetic analysis) can't be done with just the HMM. Fortunately, for the tasks we want to do (identifying whether a protein is a member of the family and aligning the protein to a member of the family), the information in the HMM is usually sufficient.

In summarizing a multiple alignment we distinguish between two classes of residues—those in *alignment columns*, which are assumed to have corresponding structure or function in the different proteins of the family, and *insertions*, which occur between the alignment columns and about which the multiple alignment makes no claims of similarity. Some proteins in the multiple alignment may not have any residue in a particular alignment column—the protein is said to have a *deletion* in that position. Note that insertions and deletions are defined here relative to the multiple alignment, not to a known or hypothesized ancestral sequence—there is no evolutionary relationship implied by these terms here.

Our profile HMMs will have two *states* for each alignment column: the *match state* and the *delete state*. Associated with each match state we have a table giving the probability of seeing each amino acid in that position of the multiple alignment. These probabilities are referred to as the *emission* probabilities for the state.

Simple dynamic programming algorithms can be used to compute the probability of a specific sequence given an HMM and to provide the most probable alignment of the sequence to the states of the HMM, hence to the alignment columns of the multiple alignment. These algorithms are beyond the scope of this chapter, but there is an excellent text available [16].

One detail that is not well presented in that book is how to compute the probability vector for each state from the multiple alignment. We have found it best to weight the sequences, using any of several strategies [17, 18, 19, 20, 15], then compute the mean posterior probability of each amino acid given the (weighted) observed counts and a prior that is a mixture of Dirichlet distributions. [21]. A detailed discussion of Dirichlet mixtures is beyond the scope of this chapter, and many of the papers (including the recommended text [16]) get the formula for Dirichlet mixtures wrong—the tutorial paper by Sjölander et al. [21] is probably the most accessible presentation, and has no major errors. [1]

A suite of tools for building and using profile HMMs, the Sequence Alignment and Modeling (SAM) suite, is available from the University of California, Santa Cruz (`http://www.soe.ucsc.edu/research/compbio/sam.html`). Another tool suite for profile HMMs, HMMer [22, 23], is available as open source, but provides somewhat less functionality, especially for building HMMs.

An unusual feature of the SAM methods is the use of *reverse-sequence null models*, explained in the next few paragraphs.

In any stochastic modeling approach, one wants to compare the probability of a model, M, with the probability of some null model, N, given some sequence, s. This is usually expressed as the logarithm of the posterior odds ratio (that is, the ratio of the probabilities we assign to the model and the null model after we have seen the sequence):

$$\begin{aligned} \log \frac{P(M|s)}{P(N|s)} &= \log \frac{P(s|M)P(M)}{P(s|N)P(N)} \\ &= \log \frac{P(s|M)}{P(s|N)} + \log \frac{P(M)}{P(N)} \ . \end{aligned}$$

Note that this formulation reduces to calculating the probability of the sequence given the model and given the null model and choosing a constant, the log prior odds ratio. A sequence is well fit by a model if the posterior probability of the model is much higher than the posterior probability of the null model.

When simple null models are chosen, there are some anomalous effects. For example, a cysteine-rich protein (such as metallothionein) will have a high probability with any HMM for a protein family that has several conserved cysteines, even if the protein is unrelated.[2] As another example, amphipathic helices have a fairly standard 7-residue repeating pattern that extends for many residues, giving high probabilities for any sequence with a long amphipathic helix if the HMM is for a family with a long amphipathic helix. These anomalies cause some false positives with very strong scores in almost any homology-detection

[1] The only "error" I'm aware of in the tutorial is that Sjölander chose to give the likelihood formulas for multisets rather than for strings, introducing a multinomial correction that is rarely needed. This choice does not affect the formulas for the mean posterior estimate, nor for the optimization of Dirichlet mixtures, so is rarely of any importance.

[2] Not all sequences with biased composition cause such problems. For example, hisactophilin, with histidine for over a quarter of its residues, rarely turns up as a false positive, since few protein families are characterized primarily by patterns of conserved histidines.

method, because the sequences are indeed much more similar than chance, even though they are not homologous.

Sequence-based search techniques, such as BLAST [24], often rely on low-complexity filters and compositional correction to try to remove these anomalies, but neither the cysteine-richness nor the amphipathic-helix anomalies are effectively removed by these standard techniques.

We can greatly reduce the number of such false positives by using a more sophisticated null model that uses the reversal of the sequence:

$$P(s|N) \quad = \quad P(s^r|M) \ .$$

Since the reversed sequence s^r has the same composition and length as s, and since the reversal of the periodic pattern of an amphipathic helix also looks like an amphipathic helix, most of the misleading signals are canceled by this null model, and the log-odds ratio is centered at 0 for any reasonable definition of random sequences of similar size (that is, a definition that makes the probability of sequences and their reversals the same).

This reverse-sequence null model can be used with almost any scoring system, including BLAST [1] and Smith-Waterman [25], but we have applied it mainly to HMM scoring.

15.3.1 Multitrack hidden Markov models

We can get improved fold-recognition results by incorporating more information about our templates than just the amino-acid sequence. For example, we could use the known secondary structure of the template and match it against the predicted secondary structure of our target. One mechanism for including such information in an HMM-based fold-recognition scheme is the *multitrack* HMM.

A multitrack HMM has the same basic structure as an ordinary HMM, but instead of one table of emission probabilities for each match or insertion state, there are multiple tables, one for each *track* of the HMM. A two-track HMM might have one table for amino-acid probabilities and another for probabilities of secondary-structure codes (see Figure 15.1). When the HMM is used for scoring, we need to have two input strings—the amino-acid sequence and the string of secondary-structure codes for the protein.

To build a two-track HMM for a target sequence, we can create a multiple alignment using SAM-T2K, then build an amino-acid-only HMM in the usual way. The multiple alignment can also be used to predict the secondary structure of the target (say using a neural net that outputs a probability vector for each alignment column), and the probabilities can be put into the tables for the second track.

If we have a library of template sequences with associated known secondary structure codes, we can use the two-track HMM to score each of the templates. In Section 15.5, Figure 15.5, we will see how much improvement one such two-track scheme provides.

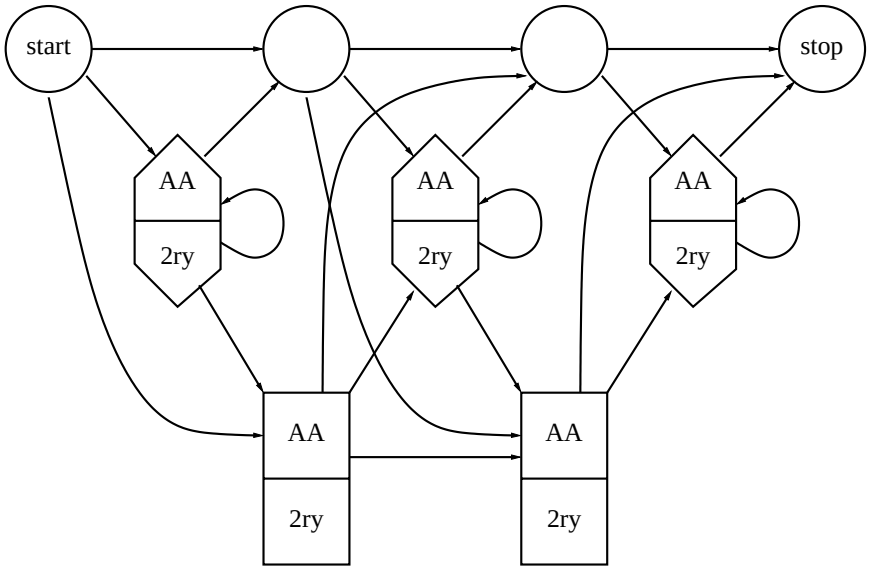

Figure 15.1: A profile HMM with two tracks (AA for amino acids and 2ry for secondary structure). Each match state (rectangle) corresponds to one column of a multiple alignment, each insert state (hexagon) corresponds to an insertion position, and each delete state (circle) provides a way to skip a particular alignment column.
For the match and insert states, both an amino acid and a secondary-structure code is matched each time the state is entered. This picture has only two match states, corresponding to a two-column multiple alignment, but typical profile HMMs have hundreds of states (one for each column of a multiple alignment).

15.3.2 Statistical significance for hidden Markov models

When searching for remote homologs, one has to examine the statistical significance of the results carefully, as it is difficult to distinguish a weak signal from noise. Although many methods use some arbitrary internal scoring scheme, most search methods provide assistance in interpreting the results by converting the internal scoring system into *E-values*, the expected number of sequences that would score this well by chance when scoring the database. Of course, "by chance" requires a definition of a null model, and a poorly chosen null model could make the E-values uninformative.

In SAM version 3, the E-value is computed using the reverse-sequence null model, using the following formula:

$$\begin{aligned} E(a) &= NP(\text{cost} < a) \\ &= N(1 + e^{-\lambda a})^{-1} \end{aligned}$$

where N is the size of the database being searched, and λ is a scaling parameter.

The derivation of the formula (still unpublished) implies that the scaling parameter λ should be 1, and this is the default. Forthcoming versions of the SAM package will allow calibration of λ (and fitting more complicated distribution functions) to improve the E-value computations. This calibration is essential for multitrack HMMs, as the default parameter values are not close to being correct for the multitrack HMMs.

One other important warning: the E-value reported is an estimate of how many sequences would score that well with the final model in a random database. A very low E-value means that the sequence or a very similar one was included in the training set from which the model was built—it does not necessarily mean that the sequence is very similar to the seed sequence. With any iterated search method (SAM-T98, SAM-T99, SAM-T2K, PSIBLAST [14], ISS [26], ...) including a false positive on any iteration results in much too strong a score for that sequence and similar sequences on subsequent iterations.

Since the default thresholds for SAM-T2K include sequences with E-values as large as 0.005 (measured in the non-redundant database) for the last iteration, all E-values less than

$$0.005 \frac{\text{size(searched database)}}{\text{size(nonredundant protein database)}}$$

are roughly equivalent to each other.

15.4 Using SAM-T2K for superfamily modeling

The SAM-T2K method has been encapsulated in a single perl script: `target2k`. This script accepts a single protein sequence as input (the *target* sequence), does a search of a non-redundant protein database, and returns a multiple alignment of sequences similar to the target. The default parameters have been adjusted to give good performance at recognizing sequences related at the superfamily level of the SCOP database [27].

For example, let's start with a single hemoglobin (say Swissprot sequence HBA_HUMAN), and try to find as many other globins as we can.

The command

```
target2k -seed hba_human.seq -out hba_human-t2k
```

will search the non-redundant protein database [28] and return a multiple alignment of protein sequences similar to hba_human. The seed sequence is provided first, followed by the similar sequences sorted in order of how well they fit the HMM constructed from the multiple alignment of the penultimate iteration. The best-scoring sequence is quite frequently not the sequence that was used as the seed. Note that the multiple alignment hba_human-t2k.a2m, with almost 1300 sequences, contains not only hemoglobins, but also myoglobins and leghemoglobins.

Sometimes this multiple alignment of putative homologs is the primary goal of the method, and it is analyzed with various tools. But for tertiary structure prediction, you generally want to create another HMM from the final multiple

alignment, in order to search another database (for example, the PDB database), or to score sequences that were selected by some other method. Although you can use `modelfromalign` directly to do the HMM construction, the SAM package includes some simple scripts that set the parameters appropriately.

These scripts do three things: first they remove redundant sequences from the multiple alignment, then they calculate a weight for each sequence, and finally build an HMM from the weighted counts at each position in the multiple alignment. The sequence weighting is intended to reduce further the effect of biased sampling of protein sequences, increasing the weight of under-represented sequences in the multiple alignment. One popular weighting scheme, developed by the Henikoffs [17], works about as well as any other for relative sequence weighting. When we use Dirichlet mixtures to estimate the probabilities of the amino acids, we also need to set the total weight appropriately. For the SAM programs, one specifies how general one wants the model to be (in terms of how many bits more information each match state has on average, compared to the background frequencies). The programs then adjust the total weight to get the desired level of generality. For comparison, the popular BLOSUM50 and BLOSUM62 matrices [29] save about 0.5 and 0.7 bits per position [30].

We used the `w0.5` script as follows

```
w0.5 hba_human-t2k.a2m hba_human-t2k-w0.5.mod
```

to build an HMM that averages about 0.5 bits of information per column—the sequence logo in Figure 15.2 shows what the match columns of the HMM are looking for.

This model can then be used for scoring a set of sequences:

```
hmmscore hba_human-scop -i hba_human-t2k-w0.5.mod -db pdb90d -sw 2
```

producing hba_human-scop.dist (the option -sw 2 specifies that we want local alignment, rather than global alignment). If we take all the SCOP domains from version 1.53, reduced to sequences with 90% or less residue identity [32], then all the globins have E-values $\leq$ 4.55e-11, and the best-scoring non-globin has E-value 1.13e-04 providing excellent separation. Note that this HMM does much better than prior reports of HMMs constructed automatically for globin recognition, which did not achieve perfect separation even when given 400 known globins [33, 20], rather than just one (except for SAM-T99 [34] which works as well as SAM-T2K on globins). Warning: globins are a fairly easy protein domain to recognize, and this clean a separation cannot be expected for more difficult domains.

The phycocyanins, which are in the same SCOP superfamily as globins, but a different SCOP family, had E-values ranging from 4.71e-02 to 3060, and were emphatically not recognized by the HMM. Thus the model generated from hba_human should be regarded as a family model, not as a superfamily model. If we score the paramecium hemoglobins, which are quite remote from the others and aren't in the SCOP test set, we get E-values ranging from 6.16e-03 to 9.72e-02 (using a database size of 4861, to match the SCOP test set). These are not quite recognized by the HMM—there are 5 false positives before the first paramecium hemoglobin and 24 before all the paramecium hemoglobins

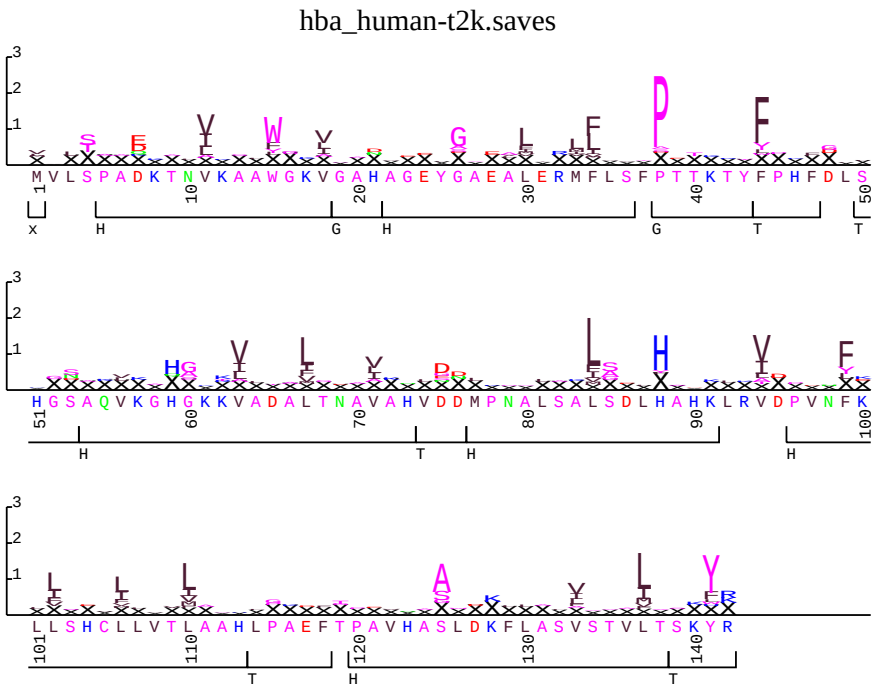

Figure 15.2: A sequence logo for the match columns of an HMM built using script `w0.5` from the multiple alignment of almost 1300 sequences found in the non-redundant protein database (NR) by SAM-T2K starting with hemoglobin hba_human as a seed. The height of each column indicates the conservation in each position (expressed as the number of bits of information). The letters in each column are sized according to the probability of each amino acid, and sorted with the most probable amino acid on top. The sequence logos are normally in color, but are here reproduced in black-and-white for economy.

Note the high conservation of the iron-coordinating histidine in position 88. There are only 3 sequences with residues other than histidine at H88 (probably from sequencing or alignment errors), but histidine is assigned a probability of only 60%, because of prior probability of substituting other aromatics is so high that the conservation here is not fully recognized.

Other highly conserved residues in the heme-binding site include F44, V63, L67, L84, V94, and L102. The lines below the amino-acid sequence show the true secondary structure (as determined by STRIDE [31] on PDB structure 1abwA: H=helix, T=turn, G=3_{10} helix).

are found.

For this SCOP test, E-values less than about 3e-5 should be considered roughly equivalent, because the threshold E-value of 0.005 on the last iteration of search in NR (which is about 130 times bigger than the test set) will include sequences scoring better than that into the multiple alignment used to build the HMM. Since all the globins score much better than 3e-5, about all we can say is that they fit the model—there are no sequences being found that are very different from the ones used in creating the HMM.

On a test of 1997 sequences with sequence identity less than 25% (FSSP representative sequences [35]), with homology defined as containing domains classified as similar by SCOP [36], the E-values can be seen to be somewhat conservative for values larger than 1, but the false-positive rate remains above 0.01 even for very small E-values (see Figure 15.3). This "fat-tail" phenomenon occurs not only for SAM-T2K, but for almost all homology-detection methods—there are some statistically significant sequence similarities that do not cause proteins to fold the same way. Most of the strong false positives result from these sequences being admitted into the training set and contaminating the multiple alignment, causing it to model two superfamilies, rather than one. If one has the knowledge and the time to remove these false training sequences by hand, the HMMs can be made much more selective [37].

At the University of California, Santa Cruz, we have relied heavily on HMM techniques for protein structure prediction in three of the Critical Assessment of Structure Prediction (CASP) experiments. We did well in the fold-recognition category in each of them [3, 4, 5]. In CASP4, we had both an automatic server (SAM-T99) and a hand-assisted method (SAM-T2K) evaluated. The automatic server is available for free use on the World-Wide Web[3]. The methods of the automatic server are primarily what is described in this chapter, though one of the main improvements of SAM-T2K, multitrack HMMs, was discussed in Section 15.3.1.

15.5 Improved verification of homology

If we have a conjectured homology between two sequences, perhaps as a result of a search using a SAM-T2K model, we can improve our confidence in the result by combining the results from two different searches, starting with each of the two sequences as a seed. For each sequence, we build a multiple alignment using `target2k`, then create an HMM from the alignment and use it to score the other sequence. The two E-values can be combined by taking their geometric mean, providing a stronger signal if both scores are better than expected by chance. Figure 15.4 shows the improvement one gets from averaging E-values in this way. There are many other possible ways to combine scores, particularly if we can estimate the relative quality of the target and template HMMs, but this simple averaging technique is surprisingly effective.

[3] `http://www.soe.ucsc.edu/research/compbio/HMM-apps/target99-query.html`

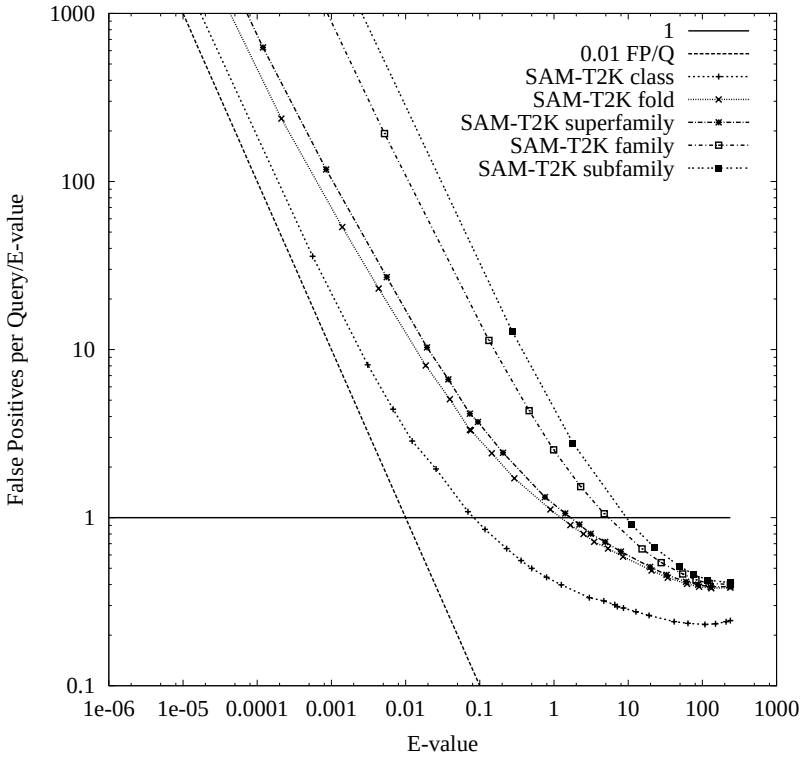

Figure 15.3: This graph compares the theoretically expected number of false positives per query (the E-value), with the observed number of false positives per query in an all-vs-all test of 1997 proteins (FSSP representative sequences [35]). The y-axis shows the ratio of the observed false positives to those predicted by the E-value computation, which should ideally be near 1. The line for 0.01 false positives/query is included as it appears to be an asymptote for the observed false positives. The different curves reflect the different levels of similarity one can define using the SCOP hierarchy. The superfamily level is most often used as a working definition of homology. The actual number of false positives deviates from the expected number by a factor of 2–10 for E-values larger than 0.01, but is much larger than expected for smaller E-values, staying above 0.01 false positives/query even for quite small E-values.. The multiple alignments were built with the default values of the `target2k` script, and the HMMs were built using `w0.5`, and scored with `hmmscore` using local alignment. No calibration was used on the HMMs, so these results are for the default e-value computation with $\lambda = 1$.

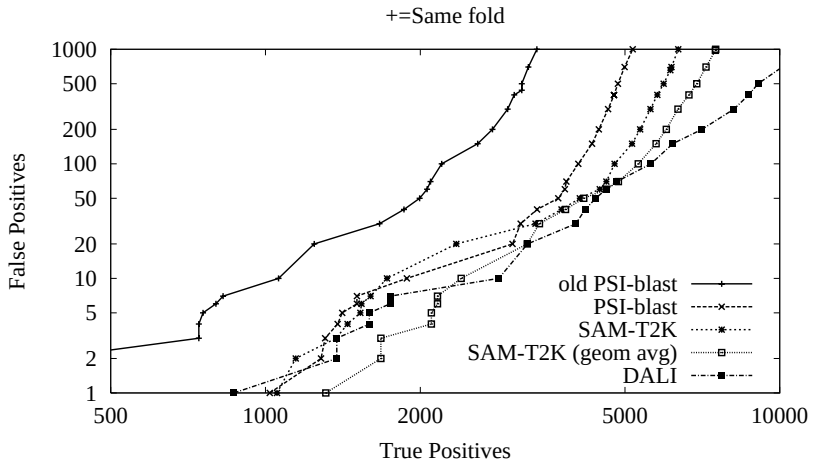

Figure 15.4: This graph the number of false positives versus the number of true positives (where "true positives" means containing a SCOP domain from the same fold) in an all-vs-all test of 1997 FSSP representative sequences. The multiple alignments were built with the default values of the `target2k` script, and the HMMs were built using `w0.5` and scored after being calibrated. The "geom avg" line is for the method of combining the E-values for HMM A scoring sequence B with HMM B scoring sequence A. The averaging method finds considerably more fold relationships at almost every level of false positives accepted. For comparison, results from using PSI-BLAST [14] and from using the structure-comparison program DALI [38] are also plotted. Note that for fewer than 0.05 false positives/query (100 false positives), the SAM-T2K methods do as well at fold recognition as a method that uses the structural information.

As described in Section 15.3.1, we can improve fold recognition using a target model by predicting the secondary structure from the multiple alignment for the target sequence, and using the predicted probabilities of each of the secondary-structure codes to build a second track for the target HMM. This two-track HMM can be used to score template sequences together with their known secondary structure sequences.

Note that this construction technique does not allow us to construct and score a two-track template HMM, because we do not have a reliable secondary structure for the target sequence. Because of this asymmetry in our knowledge, we can't use the trick of combining scores of two-track target and two-track template models. We can, however, combine the two-track HMM scores with the scores from the standard amino-acid-only template HMMs to get much more sensitivity (see Figure 15.5).

Somewhat surprisingly, the two-track HMMs provide increases in selectivity, rather than increases in sensitivity. That is, they increase the number of true

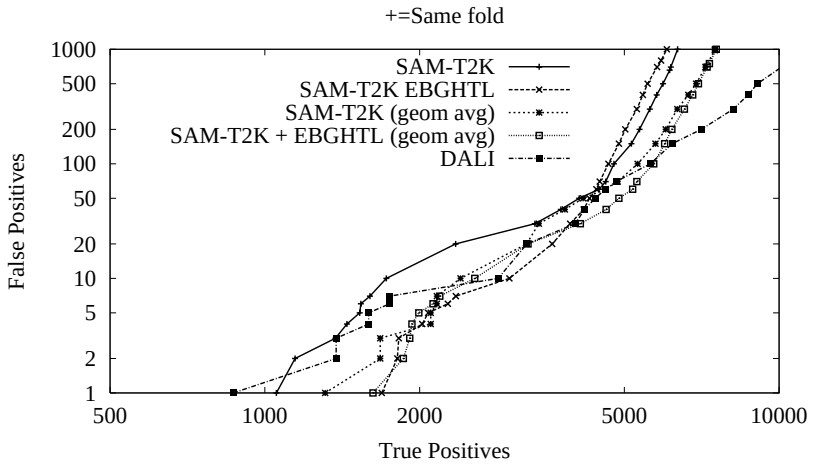

Figure 15.5: This graph shows the number of false positives versus the number of true positives (where "true positives" means containing a SCOP domain from the same fold) in an all-vs-all test of 1997 FSSP representative sequences. The multiple alignments were built with the default values of the `target2k` script, and the HMMs were built using `w0.5` and scored after being calibrated. The two-track HMMs had a second track predicting either a six-state alphabet based on STRIDE's definitions (E=beta strand, B=beta bridge, G=3_{10} helix, H=alpha helix, T=turn, L=other). Although the two-track HMMs are more selective than the amino-acid-only HMMs, combining scores from either style of target HMM with the amino-acid-only HMMs provides similar fold-recognition results. For comparison, results from using the structure-comparison program DALI [38] are also plotted. Note that though DALI has much more information (the complete structures), it does not do better at fold recognition for low false-positive rates.

positives for low false-positive rates, but not for higher false-positive rates.

15.6 Family-level multiple alignments

The default parameters for the `target2k` script have been set for fairly good performance on recognizing SCOP superfamilies, producing HMMs that generalize broadly, though for some seeds (such as hba_human) only SCOP-family-level recognition is achieved. If you want to separate one SCOP family from another (as Pfam does [39]), the level of generalization in SAM-T2K is usually too much. By specifying the option `-family`, you can request a different set of parameters intended for building family-level multiple alignments from a seed sequence. These parameters have not been tested yet, and are offered only as a starting

point for further tweaking.

The Pfam database creates family-level HMMs from hand-curated seed alignments [39]. The SAM-T2K method can also be started with a seed alignment, rather than a single sequence—simply provide the alignment with `target2k`'s `-seed` option. Using the Pfam seed alignments directly with SAM could cause some problems, as the seed alignments often have large regions which would be better modeled as insertions than as alignment columns. The multiple alignment format used for the Pfam seeds cannot express the notion of unaligned residues, so the HMMer package has heuristics to guess which alignment columns should be converted to insertions for better generalization of the HMM. SAM's `buildmodel` program has similar heuristics in the *model surgery* options of the `buildmodel` program, but these options are not used by default in the `target2k` script.

The `-seed` option is also useful for superfamily recognition—especially if a structural alignment is available to use as a seed, though a recent study by Julian Gough shows that using multiple HMMs to represent a superfamily works better than trying to build a single HMM from a structural alignment [37].

15.7 Modeling non-contiguous domains

Some protein domains are formed from non-contiguous pieces of the backbone. For example, the SCOP database [27] has a GroES-like fold for alcohol dehydrogenase (d2ohxa1) consisting of residues 1–174 and 325–374, with a Rossmann-fold domain in the middle.

To model a domain like this, the HMM needs to represent the long insertion in the middle. Although SAM has a special mechanism (the free insertion module, or FIM) for handling arbitrary length insertions, we have found it works just as well to include the insertion as lower-case letters in the seed alignment, and let the insertion probabilities be computed automatically.

If we create a file `d2ohxa1.seq` with the appropriate seed alignment for the multi-helical domain:

```
>d2ohxa1 2.22.1.2.1 (1-174,325-374) Alcohol dehydrogenase [horse]
STAGKVIKCKAAVLWEEKKPFSIEEVEVAPPKAHEVRIKMVATGICRSDDHVVSGTLVTP
LPVIAGHEAAGIVESIGEGVTTVRPGDKVIPLFTPQCGKCRVCKHPEGNFCLKNDLSMPR
GTMQDGTSRFTCRGKPIHHFLGTSTFSQYTVVDEISVAKIDAASPLEKVCLIGC
gfstgygsavkvakvtqgstcavfglggvglsvimgckaagaariigvdinkdkfakake
vgatecvnpqdykkpiqevltemsnggvdfsfevigrldtmvtalsccqeaygvsvivgv
ppdsqnlsmnpmlllsgrtwkgaifggfks
KDSVPKLVADFMAKKFALDPLITHVLPFEKINEGFDLLRSGESIRTILTF
```

then the command

```
target2k -seed d2ohxa1.seq -out d2ohxa1-t2k
```

will build the multiple alignment, and we can create an HMM with the `w0.5` script as before.

The HMM `d2ohxa1-t2k-w0.5.mod` scores the alcohol dehydrogenase family members (in SCOP version 1.53) with E-values $\leq$ 7.6e-40. The best-scoring

sequence outside the superfamily (the first false positive) is 1c17M (subunit A of F1F0 ATP synthase) with E-value 7.17e-04.

We can also look for false negatives—sequences in the GroES-like superfamily that do not score well. The chaperonin-10 sequences, which are in the superfamily but not the alcohol dehydrogenase family, have E-values 133, 211, and 3820 after several hundred unrelated sequences, so this HMM is best viewed as a family-level model, not a superfamily model.

15.8 Building an HMM from a structural alignment

The central domain of 2ohxA is an NAD(P)-binding Rossmann-fold domain, a large superfamily of alpha/beta/alpha units with a parallel beta-sheet of 6 strands (structure 321456). This superfamily is fairly diverse, and difficult to capture in a single HMM. Let's start with `d2ohxa2.seq`:

```
>d2ohxa2 3.19.1.1.1 (175-324) Alcohol dehydrogenase [horse]
GFSTGYGSAVKVAKVTQGSTCAVFGLGGVGLSVIMGCKAAGAARIIGVDINKDKFAKAKE
VGATECVNPQDYKKPIQEVLTEMSNGGVDFSFEVIGRLDTMVTALSCCQEAYGVSVIVGV
PPDSQNLSMNPMLLLSGRTWKGAIFGGFKS
```

and build the HMM as before

```
target2k -seed d2ohxa2.seq -out d2ohxa2-t2k
w0.5 d2ohxa2-t2k.a2m d2ohxa2-t2k-w0.5.mod
```

Scoring the SCOP database gives us the first false positive at E-value 1.64e-06 after 10 true positives, at the minimum error point (1.11e-02) there are 34 true positives, 54 false negatives, and 15 false positives.

All eight members of the same family as d2ohxa2 score extremely well (E-values <8e-27), but several of the other families in the same superfamily also score well, with only occasional false positives, so this can genuinely be viewed as a superfamily model, and not just a family model.

We can try to make a better model for the Rossmann-fold superfamily containing d2ohxa2, by starting from a structural alignment of two of its members. The model built using just d2ohxa2 had particular difficulty recognizing the lactate dehydrogenases and the malate dehydrogenases, so perhaps we could improve the model by including one of them in the alignment. In the FSSP [35] alignment for 2ohxA, the highest Z-score of these is for 2cmd (*E. coli* malate dehydrogenase).

From the FSSP structural alignment, we can extract the aligned section that seems to match the domain we are interested in:

```
             177
               |
2ohxA-domain STCAVFG.LGGVGLSVIMGCKAAG...AARIIGVDINKDKFAKAKEV........GA
2cmd-domain  MKVAVLGaAGGIGQALALLLKTQLpsg-SELSLYDIAPVTPGVAVDLshiptavk--

2ohxA-domain TECVNPQDYKKPIQEVLTEMSNGGVDFSFEVIG...............RLDTMVTAL
2cmd-domain  IKGFSGE-----DATPAL----EGADVVLISAGvrrkpgmdrsdlfnvNAGIVKNLV

2ohxA-domain SCC..QEAYGVSVIVGVP.........PDSQNLSMNP..MLLL..SGRTWKGAIFGG
2cmd-domain  QQVakTCPKACIGIITNPvnttvaiaa------EVLKkaGVYDkn---KLFGVT-TL
                 327
                  |
2ohxA-domain FK.SKDS
2cmd-domain  DIiRSNT
```

If we build the HMM as before

```
target2k -seed 2ohxA-2cmd.a2m -out 2ohxA-2cmd-t2k
w0.5 2ohxA-2cmd-t2k.a2m 2ohxA-2cmd-t2k-w0.5.mod
```

and score the SCOP 90% database, we get the first false positive at E-value at 1.18e-05 (DNA polymerase beta) after 31 true positives, and at the minimum-error point (E-value 1.85e-02), there are 51 true positives, 12 false positives, and 37 false negatives.

Although this model scores the superfamily well (58 of the 88 members have E-values less than 1.0), it also scores 97 incorrect domains as well. The HMM built with a single sequence as a seed does not do as well—to get 58 true positives it would need to include 212 false positives.

If we add to the structural alignment one more sequence that was not found with the 2ohxaA-2cmd-t2k HMM (say 2pgd), we can improve the model further. Starting from the alignment

```
2pgd  DIALIG.LAVMGQNLILNMNDHG...F.VVCAFNRTVSKVDDFLANEAKGT....KVLGAH..S
2cmd  KVAVLGaAGGIGQALALLLKTQLpsgS.ELSLYDIA-PVTPGVAVD-LSHIptavKIKGFSgeD
2ohxA TCAVFG.LGGVGLSVIMGCKAAG...AaRIIGVDINKDKFAKAKEV-----ga..-TECVN..P

2pgd  .....LEEMVSKLKKPR.RIILLVK.....AGQAVDNFIEKL....VPLLDIGDIIIDGGNSEY
2cmd  .....ATPALEGA---D.VVLISAGv15nvNAGIVKNLVQQV....AKTCP-KACIGIITNPVN
2ohxA qd7piQEVLTEMSNGGVdFSFEVIG.....----RLDTMVTAlscc---QEAYGVSVIVGVPPD

2pgd  RD.....TMRRCRDLK..DKGI..LFVGSGVS
2cmd  TT.....VAIAAEVLKkaGVYDknKLFGVTTL
2ohxA SQnlsmn-----PMLL..LSGR..TWKGAIFG
```

and building a SAM-T2K alignment as before, the HMM finds 31 true positives before the first false positive (sarcosine oxidase) at E-value 3.93e-05, 47 true positives and 9 false positives at the minimum error point (E-value=1.01e-2), and 60 true positives with 76 false positives at E-value < 1. To get 58 true positives, we have to accept 30 false positives, so we can see that this model is considerably more selective than either of the first two. We could continue generalizing the model by adding poorly scored members of the superfamily, perhaps 1ofgA next, since it scores very poorly but has a reasonably high Z-score in FSSP.

If we have a structural alignment of dissimilar homologs, it may not seem

necessary to run the `target2k` method. If we build a model from just the structural alignment with the command

```
w0.5 2pgd-2cmd-2ohxA.a2m 2pgd-2cmd-2ohxA-w0.5.mod
```

we get only 14 true positives before a false positive at E-value 1.48e-04, 24 true positives with 9 false positives, and 58 true positives with 566 false positives.

For very low false-positive rates, our best results on this domain were with the SAM-T2K method applied to a structural alignment of multiple sequences, with 31 true positives before a false positive. Also, if we want to find the remote homologs, accepting a few false positives, we again get the best results by using a structural alignment as the seed for SAM-T2K (58 true positives with only 30 false positives). The SAM-T2K method does get better results on this example than using just a structural alignment without searching for more homologs.

In tests, we have found using many HMMs, each built with SAM-T2K from a single seed, simpler and as effective at finding remote homologs as building HMMs from structural alignments [37]. We believe that combining information from diverse seeds would be useful, but that building an HMM from a structural alignment is not as valuable as one might expect.

15.9 Improving existing multiple alignments

You can clean up an existing multiple alignment by using it as a seed and specifying the `-tuneup` option. This option turns off the search for similar sequences and turns off the `-force_seed 1` option that normally copies the seed alignment without modification.

The seed alignment will be used to create an HMM, then the sequences in the alignment will be used as the set of potential homologs to search and align. The output alignment may contain only a subset of the original sequences, if some of the sequences score too poorly to meet the thresholds. If you want to include all the sequences, set the `-thresholds` variable to have a very large final threshold.

The `-tuneup` option can also be used to add unaligned sequences to an existing multiple alignment with the `-homologs` option. For example,

```
target2k -seed hba_human.seq -homologs globins50.seq -tuneup \
         -db_size 600000 -out hba_human-50
```

adds 50 globin sequences to the single hba_human globin alignment, creating a multiple alignment of 51 sequences. The `-db_size` option says what size database should be assumed for computing E-values. If it is omitted, the size of the non-redundant database that would normally be searched is used.

15.10 Creating a multiple alignment from unaligned sequences

It isn't really necessary to specify a seed for the SAM-T2K method—if the `-close` or `-homologs` option is used, then an initial multiple alignment can be

created by `buildmodel` from the specified unaligned sequences. If both `-close` and `-homologs` are given, then only the close set are used for the initial alignment, but the full set of homologs are used for subsequent iterations.

For example,

```
target2k -homologs globins50.seq -tuneup -db_size 600000 \
         -out globins50-tuned
```

will align the 50 globin sequences without using a seed sequence, leaving the alignment in globins50-tuned.a2m.

We have tested this multiple alignment method using the BAliBASE benchmark [40] and the SAM-T99 tuneup option, and found that SAM-T99 produces multiple alignments of about the same quality as ClustalW [41, 42]. These tests showed us that there was little to gain from realigning SAM alignments using CLUSTAL, but also that SAM-T99 was a slow way to align small numbers of sequences. Although the SAM multiple alignment algorithm is linear in the number of the sequences, while CLUSTAL is quadratic, the constants for SAM are large enough that CLUSTAL is faster if there are fewer than 500 sequences to align.

15.11 Conclusions

Hidden Markov models have been an effective method for predicting protein structure through fold recognition and alignment, but there is still a lot to do.

We expect the next few years to bring several improvements in fold recognition, not only from the constantly increasing sequence and template databases, but also from improvements in the methods. For example, we have only just started examining multitrack HMMs, and have not yet determined what local properties of protein structures are best to use as auxiliary tracks. Improved computation of statistical significance will let us set our thresholds more precisely, giving better sensitivity and selectivity. We could try more sophisticated ways of combining results from multiple HMMs for the same fold, such as the product-of-p-values method of Bailey and Grundy [43].

We could try adding "sanity checks" to select fold-recognition predictions that produce compact, protein-like structures—the current alignment methods do not examine the predicted 3d structure at all, and often result in obviously bad predictions. Although almost all hand-assisted predictions include some of these sanity checks, automating the procedure has been surprisingly difficult to do well.

We can also combine our fold-recognition methods with other methods for predicting protein structure, such as the fragment-packing methods of Rosetta [44] or the keyword matching of SAWTED [45].

Reference list

[1] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] W. Pearson and D. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences, USA*, 85:2444–2448, 1988.

[3] Kevin Karplus, Kimmen Sjölander, Christian Barrett, Melissa Cline, David Haussler, Richard Hughey, Liisa Holm, and Chris Sander. Predicting protein structure using hidden Markov models. *Proteins: Structure, Function, and Genetics*, Suppl. 1:134–139, 1997.

[4] Kevin Karplus, Christian Barrett, Melissa Cline, Mark Diekhans, Leslie Grate, and Richard Hughey. Predicting protein structure using only sequence information. *Proteins: Structure, Function, and Genetics*, Supplement 3(1):121–125, 1999.

[5] Kevin Karplus, Rachel Karchin, Christian Barrett, Spencer Tu, Melissa Cline, Mark Diekhans, Leslie Grate, Jonathan Casper, and Richard Hughey. What is the value added by human intervention in protein structure prediction? *Proteins: Structure, Function, and Genetics*, 2001. accepted.

[6] R. Lathrop and T. Smith. Global optimum protein threading with gapped alignment and empirical pair score functions. *Journal of Molecular Biology*, 255(4):641–65, 1996.

[7] A. R. Panchenko, A. Marcher-Bauer, and S. H. Bryant. Combination of threading potentials and sequence profiles improves fold recognition. *Proteins: Structure, Function, and Genetics*, Supplement 3(1):133–40, 1999.

[8] S. Miyazawa and R. Jernigan. Residue-residue potentials with a favorable contact pair term and unfavorable high packing density term, for simulation and threading. *Journal of Molecular Biology*, 256(3):623–44, 1996.

[9] C. M. Lemer, M. J. Rooman, and S. J. Wodak. Protein structure prediction by threading methods: evaluation of current techniques. *Proteins: Structure, Function, and Genetics*, 23(3):227,55, 1995.

[10] M. Sippl. Knowledge-based potentials for proteins. *Current Opinion in Structural Biology*, 5:229–235, 1995.

[11] L. A. Mirny and E. I. Shakhnovich. Protein structure prediction by threading. why it works and why it does not. *Journal of Molecular Biology*, 283(2):507–26, 1998.

[12] 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.

[13] Melissa Cline. *Protein sequence alignment reliablity: prediction and measurement*. PhD thesis, University of California, Computer Science, UC Santa Cruz, CA 95064, 2000.

[14] S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3899–3402, 1997.

[15] Kevin Karplus, Christian Barrett, and Richard Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.

[16] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[17] Steven Henikoff and Jorja G. Henikoff. Position-based sequence weights. *Journal of Molecular Biology*, 243(4):574–578, November 1994.

[18] Stephen F. Altschul, Raymond J. Carroll, and David J. Lipman. Weights for data related by a tree. *Journal of Molecular Biology*, 207:647–653, 1989.

[19] Martin Vingron and Peter R. Sibbald. Weighting in sequence space: a comparison of methods in terms of generalized sequences. *Proceedings of the National Academy of Sciences, USA*, 90:8777–8781, October 1993.

[20] R. Karchin and R. Hughey. Weighting hidden Markov models for maximum discrimination. *Bioinformatics*, 14(9):772–782, 1998.

[21] 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. *Computer Applications in the Biosciences*, 12(4):327–345, August 1996.

[22] Sean Eddy. HMMER WWW site. `http://hmmer.wustl.edu/`.

[23] S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–63, 1998.

[24] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[25] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[26] J. Park, S. Teichmann, T. Hubbard, and C. Chothia. Intermediate sequences increase the detection of homology between sequences. *Journal of Molecular Biology*, 273:349–354, 1997.

[27] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995. Information on SCOP is available at `http://scop.mrc-lmb.cam.ac.uk/scop`.

[28] NR (All non-redundant GenBank CDS translations+PDB+SwissProt+PIR+PRF Database) Distributed on the Internet via anonymous FTP from `ftp://ftp.ncbi.nlm.nih.gov/blast/db`. Information on NR is available at `http://www.ncbi.nlm.nih.gov/BLAST/blast_databases.html`.

[29] Steven Henikoff and Jorja G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences, USA*, 89:10915–10919, November 1992.

[30] Stephen F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology*, 219:555–565, 1991.

[31] Dimitrij Frishman and Patrick Argos. Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Genetics*, 23:566–579, 1995.

[32] S.E. Brenner, P. Koehl, and M. Levitt. The ASTRAL compendium for sequence and structure analysis. *Nucleic Acids Research*, 28:254–256, 2000. `http://astral.stanford.edu/`.

[33] D. Haussler, A. Krogh, I. S. Mian, and K. Sjölander. Protein modeling using hidden Markov models: Analysis of globins. In *Proceedings of the Hawaii International Conference on System Sciences*, volume 1, pages 792–802, Los Alamitos, CA, 1993. IEEE Computer Society Press.

[34] Richard Hughey, Kevin Karplus, and Anders Krogh. SAM: Sequence alignment and modeling software system, version 3. Technical Report UCSC-CRL-99-11, University of California, Santa Cruz, Computer Engineering, UC Santa Cruz, CA 95064, October 1999. Available from `http://www.cse.ucsc.edu/research/compbio/sam.html`.

[35] Liisa Holm and Chris Sander. Mapping the protein universe. *Science*, 273(5275):595–603, Aug 2 1996.

[36] T. Hubbard, A. Murzin, S. Brenner, and C. Chothia. SCOP: a structural classification of proteins database. *Nucleic Acids Research*, 25(1):236–9, January 1997.

[37] Julian Gough, Kevin Karplus, Richard Hughey, and Cyrus Chothia. Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *Journal of Molecular Biology*, 2001. submitted.

[38] Liisa Holm and Chris Sander. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233(1):123–138, 5 Sept 1993.

[39] E.L.L Sonnhammer, S.R. Eddy, and R. Durbin. Pfam: A comprehensive database of protein families based on seed alignments. *Proteins: Structure, Function, and Genetics*, 28:405–420, 1997.

[40] Julie D. Thompson, Frederick Plewniak, and Oliver Poch. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1):87–8, 1999.

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

[42] Kevin Karplus and Birong Hu. Evaluation of protein multiple alignments by SAM-T99 using the BaliBASE multiple alignment test set. *Bioinformatics*, 17:713–720, August 2001.

[43] Timothy L. Bailey and William N. Grundy. Classifying proteins by family using the product of correlated p-values. In *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB99)*, pages 10–14. ACM Press, April 11-14 1999.

[44] Kim T. Simons, Rich Bonneau, Ingo Ruczinski, and David Baker. Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins: Structure, Function, and Genetics*, Supplement 3(1):171–176, 1999.

[45] Robert M. MacCallum, Lawrence A. Kelley, and Michael J. E. Sternberg. SAWTED: Structure assignment with text description-enhanced detection of remote homologues with automated SWISS-PROT annotation comparisons. *Bioinformatics*, 16(2):125–129, February 2000.

Index