Scoring Hidden Markov Models

Christian Barrett Richard Hughey Kevin Karplus

Department of Computer Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064, USA
Tel: (408) 459-2939, Fax: (408) 459-4829, email: rph@cse.ucsc.edu

1 Abstract

1.1 Motivation

Statistical sequence comparison techniques, such as hidden Markov models and generalized profiles, calculate the probability that a sequence was generated by a given model. Log-odds scoring is a means of evaluating this probability by comparing it to a null hypothesis, usually a simpler statistical model intended to represent the universe of sequences as a whole, rather than the group of interest. Such scoring leads to two immediate questions: what should the null model be, and what threshold of log-odds score should be deemed a match to the model.

1.2 Results

This paper experimentally analyses these two issues. Within the context of the Sequence Alignment and Modeling software suite (SAM), we consider a variety of null models and suitable thresholds. Additionally, we consider HMMer's log-odds scoring and SAM's original Z-scoring method. Among the null model choices, a simple looping null model that emits characters according to the geometric mean of the character probabilities in the columns modeled by the HMM performs well or best across all four discrimination experiments.

1.3 Availability

Information on obtaining the SAM program suite (free for academic use), as well as a server interface, is available from http://www.cse.ucsc.edu/research/compbio/sam.html. HMMer is freely available from http://genome.wustl.edu/eddy/hmm.html.

1.4 Contact

Richard Hughey. Department of Computer Engineering; University of California, Santa Cruz; Santa Cruz, CA 95064, USA. Email: rph@cse.ucsc.edu

2 Introduction

Linear hidden Markov models (HMMs) (Haussler *et al.*, 1993; Krogh *et al.*, 1994) and the related generalized profiles (Bucher & Bairoch, 1994) have been quite successful in detecting conserved patterns in multiple sequences (Hughey & Krogh, 1996; Baldi *et al.*, 1994; Eddy *et al.*, 1995; Eddy, 1995; Bucher *et al.*, 1996; McClure *et al.*, 1996). By providing unaligned sequences that contain instances of a family, the resulting HMM identifies a set of positions that describes, more or less, conserved first order structure in the family.

The resulting model is useful for three purposes. Using a dynamic programming method one can generate a multiple alignment of the unaligned sequences from which the model was built. Thus one can inspect the regions in these training sequences that the process found to be "homologous". By studying the model itself, one can glean further insight by noting what it reveals about the common structure underlying the sequences in the family. Finally, the model can be used to discriminate between family and non-family sequences when employed for database searching.

It is this last issue that we address here using the Sequence Analysis and Modeling software suite (SAM) (Hughey & Krogh, 1996). While previous work with SAM has shown notable success in sequence family discrimination, the methods used to determine sequence homology were admittedly crude (Krogh *et al.*, 1994). The basic scores reported by SAM are highly length dependent, so a heuristic was used to Z-score each sequence based on the distribution of scores for sequences of similar length. This paper studies techniques to overcome

problems with the Z-score heuristic. The problem of length dependence in the score is addressed using log-odds scoring, discussed by Altschul and used by HMMer (Altschul, 1991; Eddy *et al.*, 1995). This method evaluates how much better a sequence fits the model than some underlying background distribution or null model. The significance of these scores is evaluated with the algorithmic significance method (Milosavljević & Jurka, 1993). The primary result of this paper is an experimental evaluation of null model alternatives for log-odds scoring.

3 System and Methods

The work discussed herein was performed using the SAM HMM program suite on UltraSparc 140 workstations. SAM is programmed in ANSI C for Unix workstations. Information on obtaining SAM, as well as several World-Wide Web interfaces to SAM, can be found at http://www.cse.ucsc.edu/research/compbio/sam.html, or by sending email to sam-info@cse.ucsc.edu. The HMMer experiments were performed with HMMer version 1.8.1, while the SAM experiments used a pre-release of SAM version 2.0.

4 Algorithm

4.1 Hidden Markov Models

Since full treatment of SAM's HMMs is available elsewhere (Krogh *et al.*, 1994; Hughey & Krogh, 1996), discussion will be limited to a brief overview of the method.

Each position, or module, in a linear HMM (Figure 1) has three states. A state shown as a rectangular box is a match state that models the distribution of letters in the corresponding column of an alignment. Diamond-shaped states model insertions of letters between two alignment positions, and circular states model a deletion, corresponding to a gap in an alignment. States of neighboring positions are connected, as shown by lines. For each of these lines there is an associated *transition probability*, which is the probability of going from one state to the other. This model topology features insertions and deletions similar to biological sequence alignment techniques based on affine gap penalties (Gotoh, 1982).

Building an HMM is a process that iteratively estimates the parameters from a set of unaligned training sequences. SAM uses a maximum *a posteriori* (MAP) approach, where a prior probability distribution, or *regularizer*, over all possible parameters is assumed. These priors encapsulate our beliefs on what a model should be like. The MAP estimation process tries to maximize

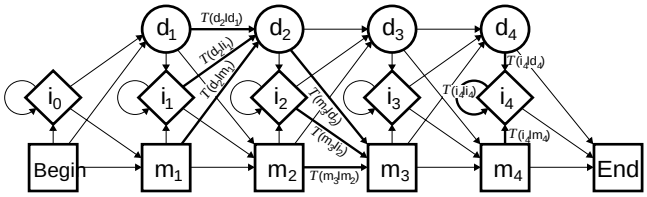


Figure 1: A linear hidden Markov model.

the posterior probability of the model given the training sequences. While this process is not guaranteed to find the best model in a global sense, various heuristics, including noise injection and model surgery, are used to (hopefully) approach a global, rather than merely local, maximum.

Once an HMM has been trained (or during training, if the sequences are not trimmed to the region of interest), Free Insertion Modules (FIMs) are added to the ends of the model. The purpose of the FIMs is to allow the HMM to model a region within a larger sequence, rather than the larger sequence in its entirety. Each FIM is a table of residue insertion probabilities with a no-cost, or unity probability, loop. In essence, the FIMs allow the model core to match the most likely part of the sequence and treat the rest as randomly generated residues with no length distribution. The FIMs enable the HMM to position itself within a sequence and to find the best match.

Once an HMM has been built for a family of proteins, negative log-likelihood (NLL) scores, $-\log(P(s|m))$, based on the probability that the sequence s was generated by the model m are returned by SAM. This raw NLL-score is exceedingly dependent on the lengths of s and m, and cannot be used directly to classify the sequence.

Using the original Z-score method, SAM overcame this problem by appropriately normalizing the NLL-score. First, all proteins in a standard database, such as SwissProt (Bairoch & Boeckmann, 1994), are scored against the model, and smoothed average curves of NLL-score versus length and standard deviation versus sequence length are calculated. Using this data, the difference between the NLL-score of a sequence and the mean NLL-score of sequences of similar length in standard deviations is calculated. This value is the Z-score for the sequence. A Z-score cutoff is then chosen by inspecting the histogram of Z-scores for sequences in the database and choosing an appropriate value, ideally one that cleanly separates the distributions of family members and non-members. Sequences with Z-scores above this cutoff are deemed to be in the family modeled by the HMM.

The major shortcomings of this procedure include the difficulty of scoring a single sequence without scoring an entire database, the subjective choice of Z-score cutoff, and the presumption of a normal distribution of scores.

We present an analysis of methods that solve these problems and in the process eliminate the need for Z-scoring altogether.

4.2 Scoring

Sequence alignments under probabilistic models can be scored using the log-odds ratio, an information-theoretic means of scoring an alignment (Altschul, 1991) that has been used by HMMer since its inception (Eddy *et al.*, 1995). The log-odds score asserts whether or not the probability that the sequence s was generated by the model m is larger than the probability that the sequence was generated by the null model ϕ:

$$\text{score}(s) = \log_z \frac{P_m(s)}{P_\phi(s)},$$

where the logarithm can be to any base z, most often to base 2, in which case the score is reported in bits. For historical reasons, SAM uses natural logarithms.

Once the model m has been trained, a key question is, of course, what should be used for the null model ϕ in the calculation of these log-odds, or NLL-NULL, scores.

One obvious choice is to have the null model match the FIMs appended to the model. The FIMs model parts of a sequence not modeled by the core of the HMM, and this is just the effect desired in a null model. Making this choice, the probability that the entire sequence was randomly generated is simply the product of the sequence residue probabilities in the FIM.

One possible problem with the simple looping null model is that, unlike the trained model, it has no distribution over sequence lengths. We have evaluated an alternative complex null model that is an HMM with transition probabilities identical to those of the trained model, but without position-specific character emission probabilities. Because the character emission probabilities are the same in all states (and identical to the FIM probabilities), with the complex null model, $P_{\phi_c}(s)$ can be decomposed into two parts for efficient calculation: the length-independent loop of the simple null model and the length distribution of the complex null model: $P_{\phi_c}(s) = P_{\phi_c}(|s|) \cdot P_\phi(s)$.

Having decided on the relation between FIMs and the null model — looked at in the other direction, FIMs are just null models attached the sides (or middle, in the case of a 2-part domain) of the model — the distribution to place in the null model must be decided.

4.3 Null model selection

First, the null model can be fixed for all scoring sequences and all models. For example, the FIM and null model tables can be set to the flat distribution or the background distribution of amino acids over all proteins. This second one is commonly used in sequence analysis. HMMer uses these background frequencies as its default null model. Both of these distributions are invariant over all protein families, and thus have an advantage in their simplicity.

Second, the null model can be fixed for all scoring sequences, but different for each model. The goal here is to correct for compositional bias within the model. For example, if a certain family is alanine-rich, the core of its model may produce a high score for *any* alanine-rich protein.

Third, the null model can be different for every scoring sequence, but the same for each model. For example, the tables can contain the amino acid distribution of the sequence currently being scored. This is a pessimistic null model, as it demands not that the HMM model the sequence better than fixed background frequencies, but that it model the sequence significantly better than frequencies exactly matching the sequence's composition. This will both correct for compositional bias and ensure that a sequence scores well only if the extra effort of creating an HMM for the family outperforms simpler representations.

Fourth, the null model can depend on both the scored sequence and the model. We have not investigated this possibility.

The experimental work has shown that the second category, null models that depend on the trained model, are usually best. This work examines three null models in this category.

- The distribution of amino acids in the set of training sequences. This distribution is simple to calculate, and has the advantage of being available at the start of model training. By default, SAM will use these letter frequencies for the first iteration of training. For scoring, this choice is exceptionally poor in the pathological, but occasionally occurring (as our group found out!) case of a specific amino acid not existing in the training set, resulting in a zero probability for that letter in the null model.

- The average amino acid distribution over all match states (columns) modeled by the HMM for the training sequences. In modeling a motif, the HMM being trained does not model all positions in each sequence. Especially in the case of a short motif within long sequences, the modeled region may have a different character distribution than the training

sequences as a whole. This match count average option will correct for this possible bias. Because the counts are taken after regularization, there will never be zero probabilities in the null model.

- The geometric mean of the amino acid distributions in the match states normalized to sum to unity. When calculating the probability $P_m(s)$, the appropriate letter emission probabilities are multiplied, along with the transition costs, to form the final $P_m(s)$. Considering log probabilities, the letter emission log probabilities are summed for the length of the sequence. A particular letter's average contribution to the sum of log probabilities, assuming it is equally likely to occur in any state, is the average over all match states of the log probabilities for generating that character. Moving back into the probability world, this corresponds exactly to the geometric mean of amino acid distributions.

  The geometric mean allows non family members with exactly the same composition as family members to benefit equally from using the model or the null model. That is, a randomly shuffled family member will score about zero because the null model and the average gain from the trained model closely match. On the other hand, the true family member, which has a linear sequence of amino acids representative of the family, will be able to make effective use of the node-to-node variations in the model.

This third method, because it has a strong foundation, can be calculated directly from a model (rather than from information saved during model training), and performs similarly or better than the other choices, is our favorite. As a result of this work, SAM has been modified to use the geometric mean by default and, during the training procedure, to update the insert and FIM tables to be the geometric mean of the match states after each training iteration.

Once a null model has been selected, the task remains to settle how to distinguish sequences that contain the model from sequences that do not.

4.4 Significance

When the log-odds score is greater than zero, the HMM m fits sequence s better than the null model ϕ fits sequence s. The question remains, however, whether or not this difference is significant. 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 logarithm base), assuming that the null model is a reasonably accurate description of the space the sequences are drawn from (Milosavljević & Jurka, 1993). 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 the same as BLAST's E parameter, the expected number of false positives (Altschul *et al.*, 1994).

Consider searching a database of length L residues in S sequences with an HMM. What value should be assigned to N? If the HMM is considered to be matching each sequence in its entirety, then N should just be the number of sequences S. On the other hand, if a motif is being searched for (with, for example, free insertion modules added to the ends of the HMM), then an experiment is effectively performed at each starting point within the database, so N is the total length L of the database. Because HMMs can have insertions and deletions, however, perhaps L^2, representing the number of starting points and ending points tested, should be considered. In practice, HMMs tend to be rigid enough that once the starting point is picked, the ending point is not a free variable.

Taking the middle ground, letting N be the number of starting points within the database, one further adjustment can be made, also based on the rigidity of HMMs. Because the HMM has a certain preferential length (adjusted by SAM's surgery heuristics), sequences shorter than the HMM really only contribute one placement to N: does the model fit the sequence starting at the beginning and ending at the end. Similarly, a longer sequence will contribute a number of placements to N approximately equal to the difference between its length and the HMM's. So, for this work, we set

$$N = \sum_{i=1}^{S} \max(\text{length}_{s_i} - \text{length}_m, 1).$$

With semi-local scoring, which allows a complete sequence to match only part of the model, the number of placements is the adjusted sequence length plus the model length. With full local scoring, which allows a subregion of the sequence to match a subregion of the model

(Smith & Waterman, 1981), the number of placements is the sum of the length of the model and the length of the sequence. These corrections are critical when the scores of a single sequence against different models are compared.

This value of N continues to be pessimistic because, as mentioned, it assumes independence among the N placements. This is a problem at two levels. First, the sequences in most databases are not independent, though various non-redundant databases strive to correct this problem. Second, within a sequence, if the HMM does not produce a significant score for a sequence s starting at the first residue, it almost certainly will not starting at the second either, though it may very well match the model somewhere further along the sequence. Furthermore, SAM's default scoring method will only report one motif in a sequence even when more than one exist. The independence of placements can be experimentally evaluated for any given model, or estimated as some number of residues or fraction of the overall model length. Rather than delve into these subjective choices, we choose instead to use the pessimistic N. Log-odds scoring has always been used by HMMer, which suggests use of $\log(L)$ for discrimination ($\sigma = 1$) (Eddy, 1995).

5 Results

To evaluate scoring methods, we performed discrimination experiments on three families of sequences, globins, EF-hands, and ferredoxins, and searched for remote homologues of 1hurA.

To build models, we created a model based on the unaligned sequences in the training set using SAM's default settings and a Dirichlet mixture prior (Sjölander *et al.*, 1996). From this model, we generated a multiple alignment of the training sequences and calculated position-based sequence weights (Henikoff & Henikoff, 1994), using an unpublished method to determine total sequence weight. With these weights, the unaligned sequences, and the mixture prior, we built the final HMM. After SAM's surgery heuristic, the globin model had 148 positions, the EF-hand model 27, the ferredoxin model 95, and the 1hurA model 180. For the HMMer models, we used the default settings, which include an internal weighting scheme. Because we did not adjust any parameters, our models may not be as good as those of an experienced HMMer user.

In scoring the models, we chose to use semi-local scoring, which allows a fragment to match only part of the model. We decided on this after discovering that many of our globin false negatives were short fragments. The switch to semi-local scoring enables us to move several, but not all, of these fragments to the true and possible positive scores, rather than being buried in the database.

For evaluation of false positives and false negatives, we use the Prosite classification in the first three experiments and the FSSP tree in the last. Because there may be unidentified EF-hands, globins, or ferredoxins in the databases, some of the sequences identified as false positives in the experiments may actually be true positives.

We investigated the following simple null models:

flat Each amino acid is equally likely

background The background distribution over all proteins

train counts The amino acid frequencies of the training set

match counts The average amino acid frequencies found in the model's match states for the training set

geometric mean The normalized geometric mean of the match state probabilities

scored counts The amino acid frequencies of the scored sequence.

We also analyzed the complex null model based on the trained model's transitions and the geometric mean of its match states, SAM's Z-scoring heuristic, and HMMer's hmmsw.

5.1 Globins

For the globins, we used Prosite (Bairoch & Bucher, 1994) to extract 685 globin sequences from SwissProt 32 (Bairoch & Boeckmann, 1994) of lengths 19 to 403. From these 685 sequences, we used the same 333 training sequences of lengths 40 to 403, as in a previous study (Bucher *et al.*, 1996). Because the globin domain is quite conserved, it is relatively easy to model.

Figure 2a plots false negatives versus false positives, each of which are a function of the setting of threshold score or σ. At each integral point on, for example, the false positive axis, the value of the false negative coordinate is a linear interpolation between the two nearest integral false negative values based on the scores of the two false negative points and the false positive point in question. This means, for example, that if two negative examples score 10 and 20, and two positive examples score 19 and 21, the point (1,1.1) will be plotted for the score of 19, indicating that that if the threshold is set at 19, there will be one false negative (the sequence that

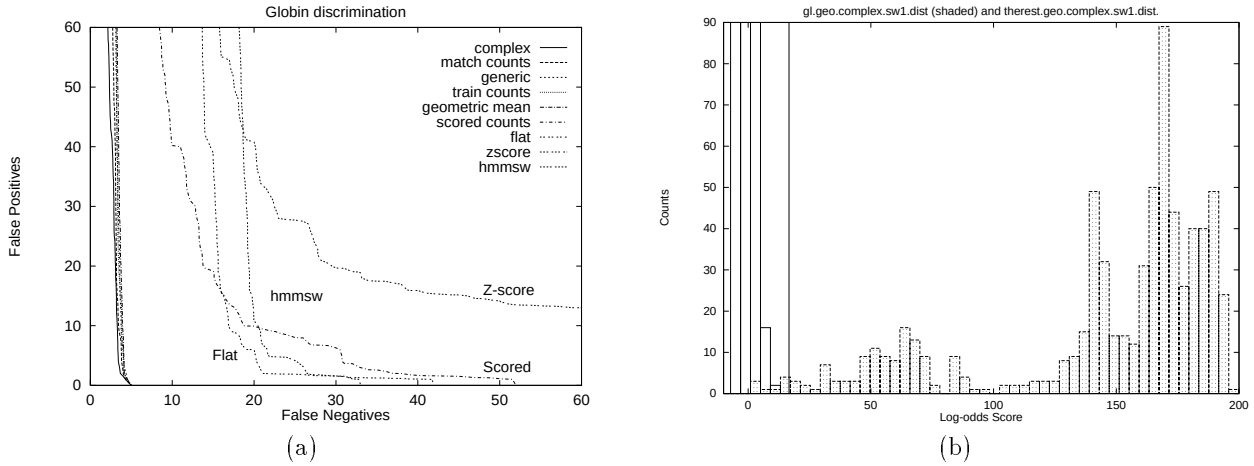


Figure 2: Globin false positives versus false negatives for different thresholds using the different null models (a). For the cluster of roughly equivalent methods, the theoretical significance level occurs at 6 false negatives and no false positives, although on typical histogram (b, for the complex null model, with a vertical line at $\sigma = 1$) slightly lower settings provide 5 and 0 or 4 and 1 false negatives and false positives. Scoring methods are listed in order of intersection with the top of the plot.

scores 21) and a little over one false positive (the sequence scoring 19 is 10% of the way between the two negative examples).

In the case of the globins, the model achieved excellent, and comparable, discrimination for all null models except the flat distribution and the scored letter counts distribution. The Z-scoring method does poorly, in part because of problems with false positives with more than 20% wildcard characters. If these are trimmed from the database, Z-scoring performance is similar to hmmsw. The hmmsw performance is more indicative of the model building (and our unfamiliarity with it) than the scoring: hmmsw uses log-odds scoring with either a background amino acid distribution (default) or a user-specified null model.

Because of the ease of discrimination, the good null models show little variation between themselves. An ideal graph would show complete separation between the false positives and false negatives, an 'L' shape that includes the origin.

Looking at the histogram for a typical example (Figure 2b), the theoretical significance setting does not exactly provide the number of false positives, as the threshold must be lowered (to approximately $\sigma = 10$) before a false positive is found.

For the well-performing null models, the low-scoring positives include, in order of lowest to highest score, three fragments with 29–41 residues (GLB4_LAMSP, HMPA_SALTY, and GLB1_LAMSP), and GLBH_CAEEL (159 residues). Under the best null model, the complex, the -7.9 score of GLBH_CAEEL is higher than any false positve, and would certainly be found by inspection. The other three scores are 2.6, 3.2, and 4.4, corresponding to 140, 75, and 25 false positives. The problem is not strictly in detecting fragments, as the smallest globin fragment, HBB2_UROHA at 19 residues, always scores as an identifiable globin. Of these five sequences, only GLB1_LAMSP was in the training set. The two highest-scoring non-globins are YF06_HAEIN (9.3) and WN8B_XENLA (6.8).

The primary result of this examination of globin scoring is that when the model is exceptionally good, the choice of null model makes little difference, apart from avoiding the flat and the scored letter counts null models.

5.2 EF-Hands

The EF-hand motif is a 29-residue structure present in cytosolic calcium-modulated proteins (Nakayama *et al.*, 1992; Persechini *et al.*, 1989; Moncrief *et al.*, 1990). These proteins bind the second messenger calcium and in their active form function as enzymes or regulate other enzymes and structural proteins. Although a number of proteins possess the EF-hand motif, some of these regions have lost their calcium binding property. For a training set, we used the same set of EF-hand sequences as in earlier work (Krogh *et al.*, 1994). This set is based on the June 1992 set of EF-hand sequences maintained

by Kretsinger and co-workers (Nakayama *et al.*, 1992). The EF-hand structures were extracted from each of the 242 sequences in the EF-hand database to obtain 885 EF-hand motifs having an average length of 29. We used Prosite to classify 324 full and partial sequences of lengths 25 to 2477 containing the EF-hand motif as true positives. We placed Prosite's unknown sequences in the group of non-EF-hands. Because of the motif's shortness and its diversity, it is a difficult family to model.

In the case of the EF-hand experiments (Figure 3), discrimination was not nearly so good, but there was differentiation between null models. Here, the minimal number of false negatives is approximately 6 for the complex, background distribution, and geometric average null models. The match counts, letter counts, flat, scored sequence, and hmmsw curves become vertical around 10–15 false negatives, indicating more sequences that could not be found regardless of the number of false positives. Scored letter counts achieves zero false positives at 330 false negatives, match counts and letter counts at 240 false negatives, hmmsw at 220 false negatives, and the remaining null models at 200–210 false negatives.

During the experiments, we found that many of the false positives were sequences containing variant EF-motifs that no longer bind calcium, and thus are not labeled as EF-hands by Prosite. To account for this, we changed the classifications of 43 sequences to being an EF-hand based on their SwissProt annotations. Specifically, we moved 36 sequences that are similar to other EF-hand calcium binding proteins (with scores between 25.4 and 16.6), but do not bind calcium, 4 sequences that contain potential calcium binding sites (AAC2_HUMAN (20.8), AAC3_HUMAN (20.0), AACT_CHICK (19.3), and YNZ1_CAEEL (16.6)), and three sequences (SPCA_HUMAN (18.2), CC23_ORCLI (17.9), CAGC_PIG (17.5)) that bind calcium but are not noted as such in Prosite. Two sequences that scored quite well (SM21_SCHMA (22.2) and YKT5_CAEEL (18.2)) do not have any calcium-related annotations and were kept in the negative category, as were 6 additional proteins that are involved in calcium channel function. The results of this switch, which changes the curve offsets but not the relative ordering, are shown in Figure 4. Additionally, these diagrams show that the theoretical $\sigma = 1.0$ falls close to where it should, though hand inspection of sequences and score histograms is still required for a neighborhood around the significance setting. This observation was common, to varying degrees, in all four experiments.

The EF-hand experiments favor the use of complex, geometric, or background distribution null models.

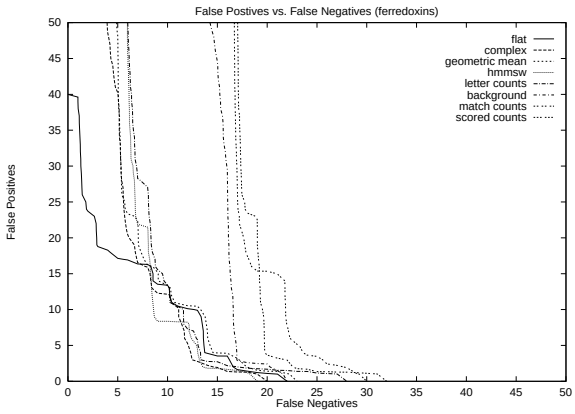


Figure 5: Ferredoxin false negatives versus false positives. Scoring methods are listed in order of intersection with the top of the plot.

5.3 Ferredoxins

The 2Fe-2S ferredoxins are a subgroup of the ferredoxin family, which is composed of iron-sulfur proteins that mediate electron transfer in a diverse range of metabolic reactions in a broad spectrum of organisms, including plants, algae and archaebacteria. As such, the family has witnessed numerous sequence and structural evolutionary changes. This, plus the sequence and structural conservation needed to retain functional properties, make the ferredoxin family as a whole difficult to model. Thus we chose the 2Fe-2S subgroup.

The 2Fe-2S ferredoxins are proteins or domains of approximately 100 amino acid residues that bind a single 2Fe-2S iron-sulfur cluster (Sander & Schneider, 1991). Using the Prosite classification of true 2Fe-2S ferredoxins, we extracted 100 sequences from SwissProt 32 of lengths 32 to 1353. Two of these were fragments. Model training involved 49 sequences, with fragments excluded.

Discrimination ability for the 2Fe-2S ferredoxins fell between that of the globins and EF-hands (Figure 5). The best tradeoff for false negatives and false positives is for the complex and the flat null models, closely followed by the training set letter counts and geometric mean null models. For a low number of false positives (3–7), there are 12–15 false negatives. The flat null model performs particularly well, though we are not entirely sure why.

Scored counts, background, and match counts all perform poorly. In the globin and EF-hand experiments, the train counts and match counts performed similarly. The difference here is that the ferredoxin model was trained on unclipped sequences that contained the ferredoxin domain plus extraneous residues. Because the amino acid distributions of the motif itself and the complete se-

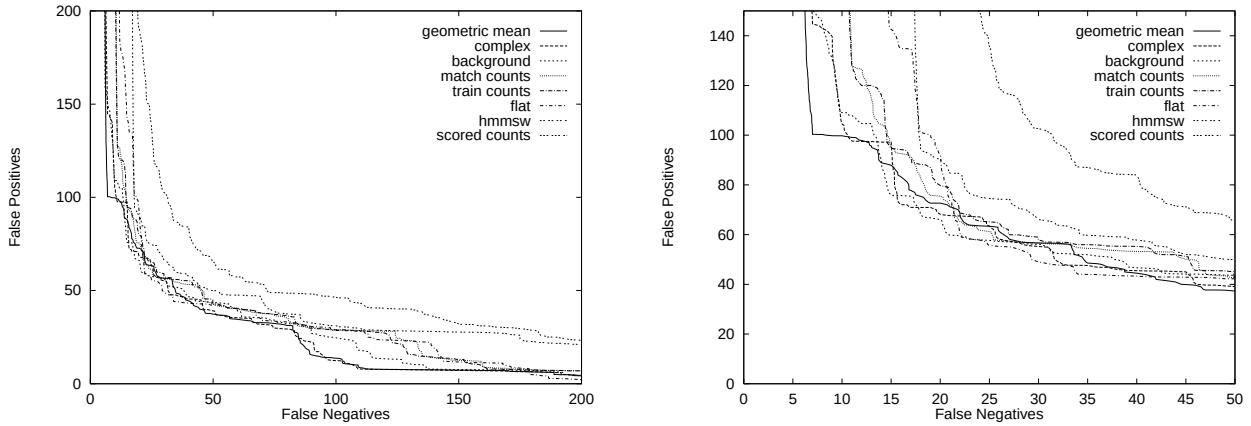
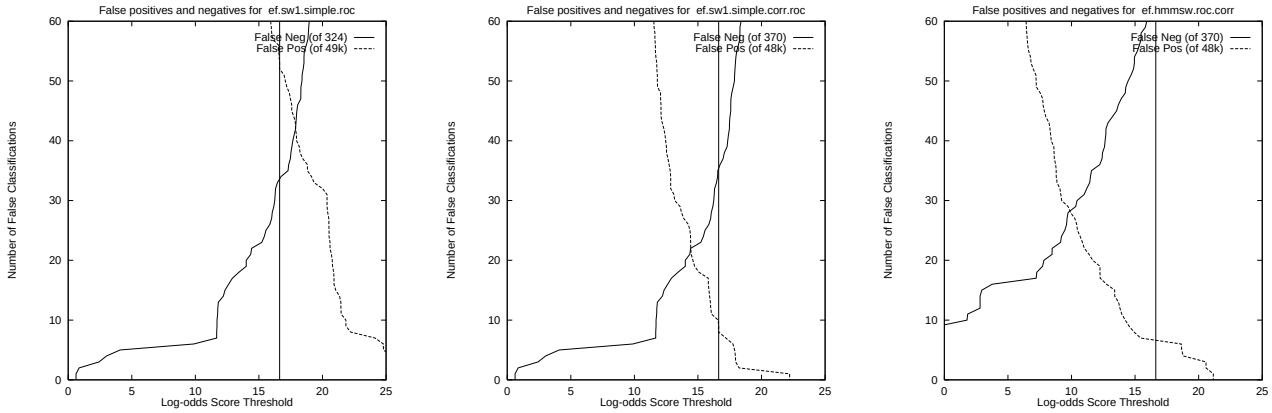


Figure 3: EF-hand false negatives versus false positives the different null models and HMMer. Scoring methods are listed in order of intersection with the top of the plot. Plot (b) is a subregion of (a) at higher magnification.



(a) Unadjusted Geometric Average (b) Adjusted Geometric Average (c) Adjusted HMMer hmmsw

Figure 4: EF-hand false positives and false negatives as a function of threshold setting before (a) and after (b, c) moving 43 sequences from the negative to the positive category. The vertical bar corresponds to $\sigma = 1.0$.

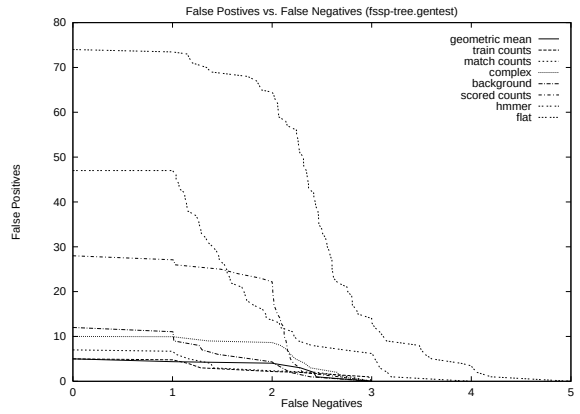


Figure 6: False negatives versus false positives when searching for remote homologues of 1hurA. Scoring methods are listed in order of intersection with the vertical axis.

quences that contain the motif are different, train counts and match counts produce different results.

For the 2Fe-2S ferredoxins, the flat, geometric average, complex, and train counts null models provided the best discrimination.

5.4 Remote Homologue Detection

In order to test the effect of null model choice on searches for remote homologs, we selected an HMM trained on 1hurA (human ADP-ribosylation factor 1) and its close homologs, as given in the HSSP database (Sander & Schneider, 1991). We used this HMM to search the FSSP database of representative proteins (Holm & Sander, 1996). Because no two sequences in the FSSP database have greater than 25% residue identity, we are looking for remote homologs, not close ones.

Deciding how distant a homolog should be counted as a correct one is a little arbitrary — we defined the following 6 proteins as desired hits: 5p21, 1tag, 1hurA, 1efm, 1efgA, and 1eft. These proteins form a subtree of the FSSP classification tree and all have a Z-score higher than 9 in the 1hurA.fssp file.

All of the methods score 1hurA and 5p21 better than any of the incorrect ones, and all but the flat null model score 1tag better than any of the incorrect ones. Only hmmsw and the flat null model score 1efgA worse than an incorrect protein. The two most difficult hits are 1eft and 1efm, both of which score worse than the best of the incorrect proteins for all models.

Because the behavior on false negatives is so similar across all null models (Figure 6), we get the most sensitive test of the different null model choices by considering how many false positives are found when there are no false negatives. In increasing order these are training sequence count and geometric mean (4 false positives), match state count (6 false positives), complex null model (10 false positives), generic background (11 false positives), hmmsw (46 false positives), and flat background (73 false positives). Thus, the remote homologue experiments favor the training sequence counts or geometric mean null models.

6 Discussion

The paper has studied seven different log-odds scoring methods, all of which compare the probability of a sequence being generated by an HMM to that of being generated by a null model. Among the three categories of methods considered, model-specific null models seem better than sequence specific and global models. Among the model specific null models, the geometric average of the match state probabilities performs well, and is appealing in that it requires no information external to the model (such as the background distribution of amino acids or characteristics of the training set), and has an intuitive justification. The complex null model, which corrects for the length distribution of the HMM, can improve discrimination.

The study has also shown that SAM's sensitivity to fragments has room for improvement, and that scoring methods require further refinement before the theoretical significance setting can be effectively used.

The experiment with finding remote homologs of 1hurA can be repeated in other parts of the fssp-tree. We plan a comprehensive series of evaluations of the ability of HMMs to find remote homologs.

7 Acknowledgements

We would like to thank Philipp Bucher and David Haussler for their valuable conversations, and Rachel Karchin for developing some of the data analysis software. This research was supported in part by DOE grant DE-FG03-95ER62112 and NSF grant BIR-9408579.

References

Altschul, S. F. (1991). Amino acid substitution matrices from an information theoretic perspective. *JMB,* **219**, 555–565.

Altschul, S. F. *et al.* (1994). Issues in searching molecular sequence databases. *Nature Genetics,* **6**, 119–129.

Bairoch, A. & Boeckmann, B. (1994). The SWISS-PROT protein sequence data bank: current status. *Nucleic Acids Research,* **22**, 3578–3580.

Bairoch, A. & Bucher, P. (1994). PROSITE: recent developments. *NAR,* **22**, 3583–3589.

Baldi, P., Chauvin, Y., Hunkapillar, T., & McClure, M. (1994). Hidden Markov models of biological primary sequence information. *PNAS,* **91**, 1059–1063.

Bucher, P. & Bairoch, A. (1994). A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation. In: *ISMB-94* pp. 53–61, Menlo Park, CA: AAAI/MIT Press.

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

Eddy, S. (1995). Multiple alignment using hidden Markov models. In: *ISMB-95*, (Rallings, C. *et al.*, eds) pp. 114–120, Menlo Park, CA: AAAI/MIT Press.

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

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *JMB,* **162** (3), 705–708.

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

Henikoff, S. & Henikoff, J. G. (1994). Position-based sequence weights. *JMB,* **243** (4), 574–578.

Holm, L. & Sander, C. (1996). The FSSP database: Fold classification based on structure-structure alignment of proteins. *NAR,* **24** (1), 206–209.

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

Krogh, A., Brown, M., Mian, I. S., Sjölander, K., & Haussler, D. (1994). Hidden Markov models in computational biology: Applications to protein modeling. *JMB,* **235**, 1501–1531.

McClure, M., Smith, C., & Elton, P. (1996). Parameterization studies for the SAM and HMMER methods of hidden Markov model generation. In: *ISMB-96* pp. 155–164, St. Louis: AAAI Press.

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

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

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

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

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

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

Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. *JMB,* **147**, 195–197.