Family-based Homology Detection via Pairwise Sequence Comparison

William Noble Grundy
Department of Computer Science and Engineering
University of California, San Diego (619) 453-4364
La Jolla, California 92093-0114 FAX 534-7029
bgrundy@cs.ucsd.edu

Abstract

The function of an unknown biological sequence can often be accurately inferred by identifying sequences homologous to the original sequence. Given a query set of known homologs, there exist at least three general classes of techniques for finding additional homologs: pairwise sequence comparisons, motif analysis, and hidden Markov modeling. Pairwise sequence comparisons are typically employed when only a single query sequence is known. Hidden Markov models (HMMs), on the other hand, are usually trained with sets of more than 100 sequences. Motif-based methods fall in between these two extremes.

The current work compares the performance of representative examples of these three homology detection techniques—using the BLAST, MEME and HMMER software—across a wide range of protein families, using query sets of varying sizes. A linear combination of multiple pairwise sequence comparisons outperforms motif-based and HMM methods for all query set sizes. Furthermore, heuristic pairwise comparison algorithms are much more efficient than the training algorithms for statistical models.

1 Introduction

The Human Genome Project and similar work on other species are producing biological sequence data at an accelerating rate. However, this data represents only a first step toward the goal of understanding the functions of these genetic sequences. Computational methods, although they will probably never replace the wet lab techniques of molecular biology, provide an important set of tools for inferring function.

One of the most effective means of inferring the function of an unidentified protein is to ask what functions are performed by homologous proteins. Two proteins are homologous if they share a common ancestor. Since the actual sequence of the common ancestor is unavailable, sequence homology can only be inferred by statistical means.

The most widely used means of inferring homology involves performing a pairwise comparisons between a single query sequence and a sequence in a protein database. Dynamic programming algorithms, such as the Smith-Waterman algorithm, or related heuristic algorithms, such as BLAST [2, 3] and FASTA [30], can be used to assign to each sequence in a database a score indicating the likelihood that this sequence is homologous to the query sequence.

Because homology inferences are based upon statistical measures, they become increasingly uncertain when the evidence for homology is weak. The *twilight zone* of sequence similarity sets the boundary of confidence levels for detecting evolutionary relatedness of proteins [15]. For most pairwise alignment programs, the twilight zone falls between 20-25% sequence identity [14].

In order to push back the twilight zone and thereby discover more remote homologs, additional information is needed. Family-based methods of homology detection leverage the information contained in a set of proteins that are known to be homologous. In a diverse family of proteins, individual members may have very low pairwise sequence similarity and hence might be missed by a pairwise analysis. Using a representative set of sequences from the family, however, can uncover these missed relationships because homology is transitive [31, 3].

The simplest means of detecting homologs using a set of related query proteins is to perform multiple pairwise comparisons. Each sequence in the database is compared with each sequence in the query set and the resulting scores are combined into an overall score for that sequence. This approach may be augmented by adding the newly discovered homologs to the query set and iterating until a transitive closure of the homology relationship is computed [27].

More sophisticated homology detection methods involve two steps: first building a statistical model of the family and then comparing that model to each sequence in the database. For example, hidden Markov models (HMMs) have been used extensively to model protein families [24, 11, 16]. These statistical models have a strong theoretical basis in probability and are supported by efficient algorithms for training, database searching,

Proceedings of the Second Annual International Conference on Computational Molecular Biology. March 22-25, 1998. pp. 94–100.

and multiple sequence alignment. The model parameters are learned via expectation-maximization, and the homology detection algorithm is a form of dynamic programming. One drawback to modeling proteins using HMMs is that they contain many free parameters and therefore require a large amount of training data. A typical, 200-state HMM may contain on the order of 5000 trainable parameters. The use of empirically derived Dirichlet mixture priors [13] can partially offset the need for larger training sets.

The size of the model may be greatly reduced by focusing only upon regions that are highly conserved across family members. Usually these regions, called motifs, have been conserved by evolution for important structural or functional reasons. As such, the motifs constitute a summary of the essential details of the family of proteins. Motifs may be represented as regular expressions [9, 28], or more generally as profiles [17] or position-specific scoring matrices, in which each column in the matrix represents a distribution across the amino acids at that position in the motif. These matrix models are formally equivalent to a class of HMMs and hence may be learned via expectation-maximization [5] or Gibbs sampling [25] from a set of unaligned protein sequences. Because motif-based methods ignore the poorly conserved spacer regions between motifs, they can be trained using smaller sets of related sequences.

This work compares the performance of each of these three homology detection methods and examines the extent to which each is dependent upon the size of the query set. For all query set sizes investigated here, the average score of multiple pairwise sequence comparisons performs better than both competing methods. All three methods have difficulty recognizing larger protein families. Also, larger query sets uniformly lead to improved homology detection. Pairwise comparisons and hidden Markov models have difficulty recognizing families containing repeated elements. Overall, heuristic pairwise comparison algorithms provide excellent performance and are much more computationally efficient than competing, model-based methods of homology detection.

2 Methods

Protein families

A collection of 75 protein families [7] was used in the homology detection experiments. These families were selected from the Prosite database [9] for their difficulty, based upon the number of false positives reported in the Prosite annotations. The Prosite IDs of these families are listed in Appendix A. The families range in size from 5 to 109 sequences, and from 949 to 58 015 amino acids. The associated release of SwissProt [10] contains 36 000 sequences and nearly 12.5 million amino acids.

Attempts to model families of related proteins are necessarily biased because the set of known protein sequences does not uniformly sample from the space of existing family members. Sequence weighting schemes attempt to compensate for this bias by assigning weights to individual sequences, usually based upon the level of sequence similarity. Such schemes may significantly improve the performance of homology detection algorithms [1, 33, 34]; however, Henikoff and Henikoff [21] have shown that many weighting schemes perform almost as well as one another. Accordingly, all the experiments reported here employ a simple, binary weighting scheme based upon BLAST similarity scores [26]. This approach is simple, since the highly similar sequences can be removed at once before any analysis is performed, and leads to faster training, since the sizes of the weighted training sets are reduced. For these experiments, a BLAST similarity threshold of 200 was used. The sizes of the weighted Prosite families ranged from 1 to 73 sequences with an average of 10.7 sequences, and from 394 to 18 702 amino acids with an average of 4202.

For each family, nested query sets of sizes 2, 4, 8, 16 and 32 sequences were randomly selected from the set of weighted sequences. This resulted in 73 query sets of size 2, 57 sets of size 4, 35 of size 8, 16 of size 16 and 3 query sets of size 32.

```
procedure average_score (query_set, database)
  for i ← 1 to size(database)
    target ← database[i]
    sum_of_scores = 0.0
    for j ← 1 to size(query_set)
      query ← query_set[j]
      (evalue, score) ← BLAST(query, target)
      if (evalue < 1000) then
        sum_of_scores = sum_of_scores + score
      end
    end
    scores[i] = sum_of_scores / size(query_set)
  end
  return scores
```

Figure 1: The algorithm used to compute average BLAST scores.

Pairwise sequence comparison

For homology detection using pairwise comparisons, gapped BLAST version 2.0 was used [2]. BLAST is a heuristic approximation of a dynamic programming optimization of maximal segment pair scores. The program was run with its default parameters, including the filtering of low-complexity regions and the use of the BLOSUM62 scoring matrix. Although BLAST was designed for single-sentence queries, it was extended here to multiple-sequence queries by searching the database separately with each member of the query set. For each search, an expectation cutoff of 1000 was used, and database sequences which would have received E-values larger than 1000 were assigned bit scores of 0.0. The result, from a set of n training sequences, was a set of n normalized BLAST bit scores for each sequence in the database. The n scores were then averaged to yield a final score for that database sequence. This algorithm is summarized in Figure 1.

Motif analysis

Ungapped motifs were discovered using MEME version 2.1 [5] with the default parameter settings from the web interface [19]. These defaults include empirical Dirichlet mixture priors weighted according to the megaprior heuristic [6], a minimum motif width of 12 and a maximum of 55, and a motif model biased toward zero or one motif occurrence per sequence. A total of ten motifs was discovered from each query set, and motif significance was judged using the majority occurrence heuristic [20]: motifs that did not appear in more than half of the query sequences were discarded. This heuristic excludes motifs that are specific to subfamilies of the given query set. For eight-sequence queries, the heuristic selected an average of 5.2 motifs.

Homology detection was performed using MAST [8]. For each sequence in the database, MAST computes a p-value for each given motif and combines these values assuming that motif occurrences are statistically independent. The resulting sequence-level E-value scores were used to rank the sequences in the database.

Hidden Markov model analysis

Hidden Markov models of each query set were built using the HMMER software package version 1.8 [16]. Models were trained using expectation-maximization coupled with simulated annealing. The default geometric annealing schedule was used, and Dirichlet mixture priors were used in order to allow the models to be trained with smaller training sets.

Preliminary experiments showed that, of the four search programs provided in HMMER, `hmmsw` consistently provides the best results when searching protein databases. That program uses a modified form of the Smith-Waterman algorithm to search for sequence-to-model matches, allowing partial matches to either the sequence or the model. Thus, `hmmsw` performs a semi-local search. For each database sequence, the program returns a log-odds scores in bits.

Evaluating search results

A modified version of the Receiver Operating Characteristic, called ROC_{50} [18], was used to compare the three search techniques. The ROC score is the area under a curve which plots true positives versus true negatives for varying score thresholds. ROC analysis combines measures of a search's sensitivity and selectivity. The ROC_{50} score is the area under the ROC curve, up to the first 50 false positives. This value has the advantages of yielding a wider spread of values, of requiring less storage space, and of corresponding to the typical biologist's willingness to sift through only approximately fifty false positives. ROC_{50} scores are normalized to range from 0.0 to 1.0, with 1.0 corresponding to the most sensitive and selective search. For BLAST, sequences were ranked according to the average bit score, as described above. For MAST, sequences were ranked by E-value, and for HMMER, sequences were ranked by log-odds score.

3 Results

Figure 2(a) shows the average ROC_{50} scores for all 75 protein families in the study. In comparing distributions of ROC_{50} scores, a two-sample t test was used, and differences were deemed significant at a 1% confidence level. In Figure 2(a), for query sets of size 2, 4, 8 and 16, all differences between search techniques at a given query set size are significant. Thus, BLAST uniformly outperforms the other two methods, and MEME outperforms HMMER. Only three families contained more than 32 weighted members, so the differences between techniques at that query set size are not significant.

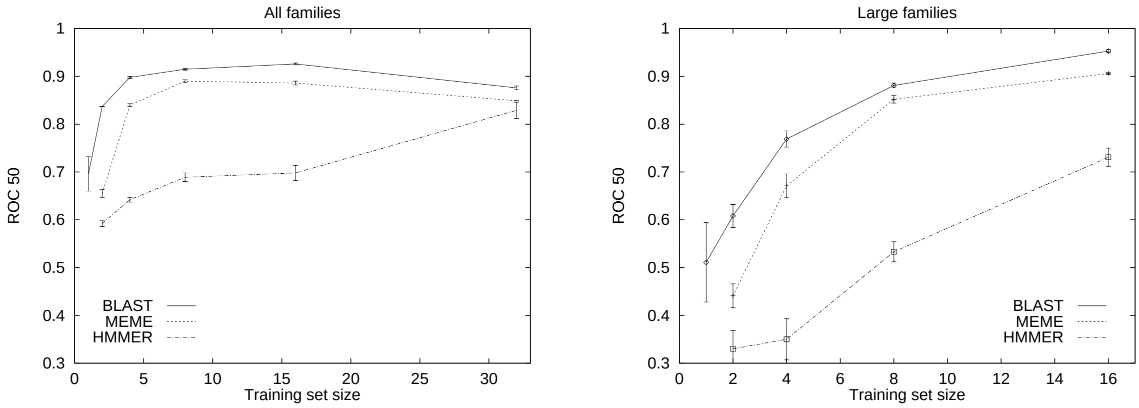
One unexpected characteristic of Figure 2(a) is the downward trend of the BLAST and MEME scores as the query set size increases. However, this trend is an artifact of the presentation of the data: the 73 2-sequence queries contain many sequences from very small families. For these small families, the task of homology detection is relatively easy. The sixteen 16-sequence query sets, however, each correspond to a relatively large and hence difficult-to-recognize protein family. The effect of family size upon recognition difficulty is illustrated in Figure 3, in which ROC_{50} scores are plotted as a function of family size. The scores show a significant downward trend as the family size increases.

Figure 2(b) corrects for differences in family size by including only families containing between 16 and 31 members. Here, the trend toward better performance with more query sequences is clearer. All three methods improve significantly (again with a 1% confidence level) at each increase in the query set size except for HMMER between queries of size 2 and 4.

Some protein families are difficult to recognize regardless of the homology detection method employed. Table 1 shows the fifteen families that received the lowest ROC_{50} scores from all three methods. The data show a strong correlation between the families for which BLAST and MEME had difficulty: the seven most difficult families for each method are the same. This agreement indicates that, for these families, a low ROC_{50} score indicates a family that is difficult to recognize, rather than a problem with the homology detection method.

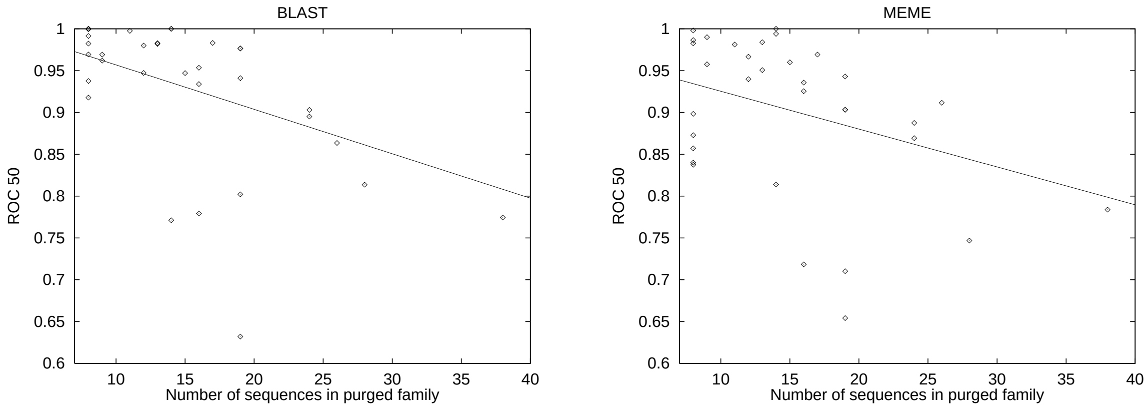
Any evaluation of homology detection methods can only be as accurate as the curated list of family members upon which the evaluations are based. Unannotated family members will cause all three methods to apparently perform poorly on that family. Thus, for example, the first 50 false positives that BLAST uncovered for the cytochrome c family contain six sequences for which the annotation includes the words `CYTOCHROME C`. Three of these false positive sequences are cytochrome C precursors; three more are listed as cytochrome c family members in a later version of SwissProt (one as a potential member).

An important difference between BLAST and MEME on the one hand and HMMER on the other is that the former two algorithms employ local search techniques. The `hmmsw` program, in contrast, performs a semi-local search, in which a single subsequence of the HMM can match a subsequence of the database protein sequence. The three-row topology of the standard linear HMM implies a simple model of evolution, involving point mutations, insertions and deletions. Semi-local searching adds to this model the possibility of large-scale deletions and insertions at either end of the protein. Still, however, the linear topology cannot accurately model protein families in which motifs or domains are repeated or shuffled. Accordingly, one would expect HMMER to perform poorly on families known to contain repeated elements. Figure 4 illustrates this effect. Prosite annotations were used to separate out those



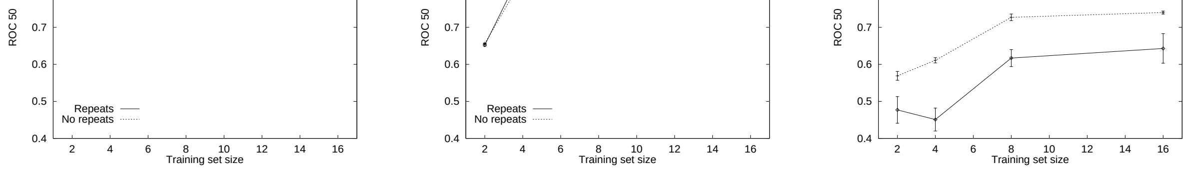
(a) (b)

Figure 2: **ROC$_{50}$ score as a function of query set size.** For each of the 75 Prosite families, nested query sets were randomly selected after binary weighting was carried out. Figure (a) includes data for all families in the study; Figure (b) only includes data from families containing more than fifteen and less than 32 members after binary sequence weighting. Error bars represent standard error. Figure (a) includes 73 query sets of size 2, 57 sets of size 4, 35 of size 8, 16 of size 16 and 3 sets of size 32; Figure (b) includes 13 query sets of each size.



(a) (b)

Figure 3: **ROC$_{50}$ score as a function of family size.** Each figure includes ROC$_{50}$ scores from 35 8-sequence queries. The slope of the regression line in Figure (a) is -0.0053 and in Figure (b) is -0.0045. Both slopes are significantly different from 0.0 at a 1% level of confidence. In each figure, two outlying families (with 53 and 73 sequences) are left out for the sake of scale.



(a) (b) (c)

Figure 4: **ROC$_{50}$ score as a function of query set size for families with and without repeated elements.** Each figure contains data for 21 families containing repeats and 52 families without repeats. Error bars represent standard error.

Family	BLAST		HMMER		MEME		Total
	ROC_{50}	Rank	ROC_{50}	Rank	ROC_{50}	Rank	rank
Cytochrome c	0.562	1	0.043	2	0.548	1	4
ABC transporters	0.774	4	0.292	4	0.784	6	14
N-6 Adenine-specific DNA methylases	0.814	7	0.257	3	0.747	5	15
Aminoacyl-transfer RNA synthetases class-II	0.632	2	0.455	11	0.654	2	15
Binding-protein-dependent transport systems inner membrane component	0.802	6	0.431	9	0.710	3	18
Lipases	0.771	3	0.528	13	0.814	7	23
Gram-positive cocci surface proteins	0.779	5	0.764	16	0.718	4	25
Eukaryotic putative RNA-binding region RNP-1	0.903	11	0.389	8	0.887	13	32
Myc-type, helix-loop-helix dimerization domain	0.864	8	0.375	7	0.912	17	32
Short-chain alcohol dehydrogenases	0.895	9	0.512	12	0.869	11	32
GTP-binding elongation factors	0.938	14	0.752	15	0.873	12	41
Glycosyl hydrolases	0.941	15	0.360	6	0.943	22	43
4Fe-4S ferredoxins	0.897	10	0.837	18	0.918	18	46
Growth factor and cytokines receptors	0.954	18	0.444	10	0.925	19	47
C-5 cytosine-specific DNA methylases	0.934	13	0.844	19	0.936	20	52

Table 1: **Difficult families.** Listed are the fifteen families that contain eight or more weighted sequences and that received the lowest ROC_{50} scores for 8-sequence queries. For each method, the families were ranked by increasing ROC_{50} score. They are listed in order of increasing total rank.

Program	Query set size		
	2	4	8
BLAST	18.2	39.4	82.3
MEME	67.2	170.4	548.0
MAST	65.5	39.7	33.9
hmmt	41.6	62.6	1716.7
hmmsw	8965.2	8772.3	8692.0

Table 2: **Typical execution times for the three homology detection methods.** Times reported are total CPU time in seconds on a 167 MHz Sparc Ultra for one protein family.

families containing repeated elements. For all query set sizes, HMMER performs significantly worse (with 1% confidence) on families containing repeated domains. For MEME, although some differences between ROC_{50} scores for families with and without repeats are significant, those differences are smaller, and no consistent trend appears. Surprisingly, however, BLAST performs better on families without repeated domains. This difference most likely arises because, in the gapped BLAST algorithm, a given sequence segment cannot participate in more than one MSP score.

One important reason for using homology detection to infer protein function is speed. Most wet lab experiments are slow relative to a protein database search. However, not all computational methods are equally fast. In this respect, BLAST clearly outperforms both MEME and HMMER. For example, Table 2 shows typical timing data for one protein family. For an 8-member query set, BLAST requires only 82 seconds; combined, MEME and MAST require 9.7 minutes, and HMMER training and searching require 2.9 hours. BLAST implements a linear algorithm, whereas the training algorithms for both MEME and HMMER are roughly $O(n^2)$ in the size of the training set. On the other hand, the MAST search algorithm is considerably faster than the corresponding HMMER search algorithm, `hmmsw`. A MAST query requires less than a minute, but with `hmmsw`, searching even a relatively small database like SwissProt takes nearly 2.5 hours on a fast workstation.

4 Discussion

For family-based homology detection using query sets of the sizes investigated here, computing the average score from a pairwise sequence comparison algorithm such as BLAST is clearly preferable to the more computationally expensive statistical modeling methods tested here. The reasons for BLAST's excellent performance are three-fold.

First, the average BLAST score incorporates information from multiple sequence comparisons into a single score. The method thereby allows for the detection of remote homologs that lack significant similarity with one or more of the training set sequences. The average BLAST score is therefore similar to the intermediate sequence approach suggested by Pearson [32] and Park *et al.* [29].

Also, BLAST may perform well relative to motif-based methods because BLAST allows for query-to-target matches along the entire length of the sequences, rather than only within the motif regions. These non-motif regions often contain important evidence of homology [32].

Finally, by avoiding a position-specific scoring matrix representation of the training sequences, the average BLAST score method does not assume that the occurrences of amino acids at a particular site in the protein are independent of amino acid occurrences at other sites in the same protein. If, in fact, covariation between sites imposes a significant evolutionary constraint, then searching separately with each training set sequence will respect that constraint.

For fairness of comparison, the experiments reported here employed the default settings of each technique. It may be the case, however, that selecting different parameter settings for the various homology detection methods may result in slightly different results. For example, although both MEME and HMMER employ Dirichlet mixture priors, MEME weights the prior more heavily by default. This heuristic may have given MEME an advantage for the smaller training sets.

The large difference in performance between single-sequence BLAST queries on the one hand and family-based homology detection methods on the other suggests a bootstrap approach when only a single query sequence is available. In such an approach, BLAST would be used initially to search for close homologs, which would then be given to a family-based homology detection algorithm.

Iterating this bootstrap procedure should provide even better homology information than the single pass reported here. Iterative applications of BLAST have been suggested by Koonin and Tatusov [23] and implemented in Probe [27] and PSI-BLAST [3]. However, Probe and PSI-BLAST search the database using position-specific scoring matrix representations. In order to test the usefulness of these matrix representations, it would be interesting to compare the performance of PSI-BLAST with that of an iterative search that employs the average BLAST score. Such a comparison will be the subject of future research.

Acknowledgments

The author would like to thank Timothy Bailey for helpful discussion and for providing the 75 protein families and the accompanying database, and Alfonso Dufour for help with the statistical analyses of results. W. N. G. is funded by the National Defense Science and Engineering Graduate Fellowship Program.

References

[1] S. F. Altschul, R. J. Carroll, and D. J. Lipman. Weights for data related by a tree. *Journal of Molecular Biology*, 207(4):647–53, 1989.

[2] 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.

[3] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[4] T. L. Bailey. MEME – Multiple EM for Motif Elicitation. `http://www.sdsc.edu/MEME`, 1998.

[5] T. L. Bailey and C. P. Elkan. Fitting a mixture model by expectation-maximization to discover motifs in biopolymers. In R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, editors, *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 1994.

[6] T. L. Bailey and M. Gribskov. The megaprior heuristic for discovering protein sequence patterns. In D. J. States, P. Agarwal, T. Gaasterland, L. Hunter, and R. Smith, editors, *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, pages 15–24. AAAI Press, 1996.

[7] T. L. Bailey and M. Gribskov. Score distributions for simultaneous matching to multiple motifs. *Journal of Computational Biology*, 4(1):45–59, 1997.

[8] T. L. Bailey and M. Gribskov. Combining evidence using p-values: Application to sequence homology searches. *Bioinformatics*, 14(1):48–54, 1998.

[9] A. Bairoch. PROSITE: A dictionary of sites and patterns in proteins. *Nucleic Acids Research*, 20:2013–2018, 1992.

[10] A. Bairoch. The SWISS-PROT protein sequence data bank: Current status. *Nucleic Acids Research*, 22(17):3578–3580, September 1994.

[11] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure. Hidden Markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences of the United States of America*, 91(3):1059–1063, 1994.

[12] NCBI BLAST search. `http://www.ncbi.nlm.nih.gov/-BLAST`, 1997.

[13] M. Brown, R. Hughey, A. Krogh, I. Mian, K. Sjolander, and D. Haussler. Using Dirichlet mixture priors to derive hidden Markov models for protein families. In C. Rawlings et al., editor, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 47–55. AAAI Press, 1995.

[14] S. Y. Chung and S. Subbiah. A structural explanation for the twilight zone of protein sequence homology. *Structure*, 4(10):1123–1127, 1996.

[15] R. F. Doolittle. *Of Urfs and Orfs: Primer on how to analyze derived amino acid sequences.* University Science Books, 1986.

[16] S. R. Eddy. Multiple alignment using hidden Markov models. In C. Rawlings et al., editor, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 114–120. AAAI Press, 1995.

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

[18] M. Gribskov and N. L. Robinson. Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Computers and Chemistry*, 20(1):25–33, 1996.

[19] W. N. Grundy, T. L. Bailey, and C. P. Elkan. ParaMEME: A parallel implementation and a web interface for a DNA and protein motif discovery tool. *Computer Applications in the Biosciences*, 12(4):303–310, 1996.

[20] W. N. Grundy, T. L. Bailey, C. P. Elkan, and M. E. Baker. Meta-MEME: Motif-based hidden Markov models of protein families. *Computer Applications in the Biosciences*, 13(4):397–406, 1997.

[21] S. Henikoff and J. G. Henikoff. Position-based sequence weights. *Journal of Molecular Biology*, 243:574–578, 1994.

[22] S. R. Eddy group, Dept. of Genetics, Washington University. `http://genome.wustl.edu/eddy/hmm.html`, 1997.

[23] E. V. Koonin and R. L. Tatusov. Computer analysis of bacterial haloacid dehalogenases defines a large superfamily of hydrolases with diverse specificity: application of an iterative approach to database search. *Journal of Molecular Biology*, 244(1):125–132, 1994.

[24] A. Krogh, M. Brown, I. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 1994.

[25] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.

[26] A. F. Neuwald and P. Green. Detecting patterns in protein sequences. *Journal of Molecular Biology*, 239(5):698–712, 1994.

[27] A. F. Neuwald, J. Liu, D. Lipman, and C. Lawrence. Extracting protein alignment models from the sequence data database. *Nucleic Acids Research*, 25(9):1665–1677, 1997.

[28] C. G. Nevill-Manning, K. S. Sethi, T. D. Wu, and D. L. Brutlag. Enumerating and ranking discrete motifs. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, 1997.

[29] J. Park, S. A. Teichmann, T. Hubbard, and C. Chothia. Intermediate sequences increase the detection of homology between sequences. *Journal of Molecular Biology*, 273:1–6, 1997.

[30] W. R. Pearson. Rapid and sensitive sequence comparisions with FASTP and FASTA. *Methods in Enzymology*, 183:63–98, 1985.

[31] W. R. Pearson. Effective protein sequence comparison. *Methods in Enzymology*, 266:227–258, 1996.

[32] W. R. Pearson. Identifying distantly related protein sequences. *Computer Applications in the Biosciences*, 13(4):325–332, 1997.

[33] P. R. Sibbald and P. Argos. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *Journal of Molecular Biology*, 216(4):813–8, 1990.

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

A Prosite families

PS00030	PS00037	PS00038	PS00043	PS00060
PS00061	PS00070	PS00075	PS00077	PS00079
PS00092	PS00095	PS00099	PS00118	PS00120
PS00133	PS00141	PS00144	PS00158	PS00180
PS00185	PS00188	PS00190	PS00194	PS00198
PS00209	PS00211	PS00215	PS00217	PS00225
PS00281	PS00283	PS00287	PS00301	PS00338
PS00339	PS00340	PS00343	PS00372	PS00399
PS00401	PS00402	PS00422	PS00435	PS00436
PS00490	PS00548	PS00589	PS00599	PS00606
PS00624	PS00626	PS00637	PS00639	PS00640
PS00643	PS00656	PS00659	PS00675	PS00676
PS00678	PS00687	PS00697	PS00700	PS00716
PS00741	PS00760	PS00761	PS00831	PS00850
PS00867	PS00869	PS00881	PS00904	PS00933

Prosite IDs of the 75 families included in this study.