UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Bayesian Approach to Motif-based Protein Modeling

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy

in Computer Science and Cognitive Science

by

William Noble Grundy

Committee in charge:

Professor Charles Elkan, Chair
Professor Richard Belew
Professor Garrison Cottrell
Professor Clark Glymour
Professor Terrence Sejnowski

1998

The dissertation of William Noble Grundy is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

1998

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Charles Elkan, for not only providing research ideas and direction, but for teaching me to be a scientist and for encouraging me throughout the past several years. Thanks also go to my collaborators, Timothy Bailey and Michael Baker, who contributed enthusiasm, ideas and expertise—Timothy in AI and Michael in biology. I thank Michael Gribskov for numerous helpful discussions and for several curated data sets used in Chapter IV.

Finally, I would like to thank my roommate, Christian Gurtner, for keeping me from working too hard during the past four years, as well as my family—Mom, Dad, John and Christopher—who have supported me for the past 29.

The text of Chapter III, in part, is a reprint of the material as it appears in *Proceedings of the Second International Conference on Computational Molecular Biology* [59]. The dissertation author was the sole author listed in this publication.

The text of Section IV.B, in part, is a reprint of the material as it appears in *Computer Applications in the Biosciences* [62]. The dissertation author was the primary author, and the co-authors listed in this publication directed and supervised the research which forms the basis for this chapter.

The text of Section IV.C, in part, is a reprint of the material as it appears in *Biochemical and Biophysical Research Communications* [61]. The dissertation author was the primary author, and the co-authors listed in this publication directed and supervised the research which forms the basis for this chapter.

VITA

1991	B.S., Stanford University
1992-1993	United States Peace Corps Volunteer, Lesotho
1996	M.S., University of California, San Diego
1998	Doctor of Philosophy University of California, San Diego

PUBLICATIONS

W. N. Grundy. "Homology Detection via Family Pairwise Search." *Journal of Computational Biology.* In press, 1998.

W. N. Grundy. "Family-based Homology Detection via Pairwise Sequence Comparison." *Proceedings of the Second Annual International Conference on Computational Molecular Biology.* S. Istrail, P. Pevzner and M. Waterman, editors. March 22-25, 1998. pp. 94–100.

T. L. Bailey, M. E. Baker, C. P. Elkan and W. N. Grundy. "MEME, MAST, and Meta-MEME: New Tools for Motif Discovery in Protein Sequences" in *Pattern Discovery in Molecular Biology.* J. Wang, B. Shapiro and D. Shasha, editors. Oxford UP, 1998.

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.

W. N. Grundy, T. L. Bailey, C. P. Elkan and M. E. Baker. "Hidden Markov Model Analysis of Motifs in Steroid Dehydrogenases and their Homologs." *Biochemical and Biophysical Research Communications.* 231(3):760–766, 1997.

J. Batali and W. N. Grundy. "Modeling the Evolution of Motivation." *Evolutionary Computation.* 4(3):235–270, 1996.

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.

W. N. Grundy. "Solar Cookers and Social Classes in Southern Africa." *Techné: Journal of Technology Studies.* G. Norton, editor. Vol. V, Winter 1995. pp. 3-7.

W. Grundy and R. Grundy. "Diffusion of Innovation: Solar Oven Use in Lesotho (Africa)." *Advances in Solar Cooking: Proceedings of the 2nd International Conference on Solar Cooker Use and Technology.* S. S. Nandwani, editor. July 12-15, 1994. pp. 240-247.

B. Grundy. "The Coinage of Sicyon." *The Numismatist.* November, 1986. pp. 2307-8.

ABSTRACT OF THE DISSERTATION

A Bayesian Approach to
Motif-based Protein Modeling

by

William Noble Grundy
Doctor of Philosophy in Computer Science and Cognitive Science
University of California, San Diego, 1998
Professor Charles Elkan, Chair

The increasing stream of data produced by the Human Genome Project and similar work on other species requires sophisticated computational analysis. This dissertation describes Meta-MEME, a software toolkit for modeling families of related proteins. Meta-MEME produces probabilistic models that provide insight into the structural and functional operation of proteins, and may be used to discover functional and evolutionary relationships among proteins. In addition, the dissertation introduces Family Pairwise Search, a heuristic homology detection algorithm based upon the linear combination of multiple pairwise sequence comparison scores.

Meta-MEME combines two existing technologies—motif discovery via expectation-maximization and hidden Markov modeling (HMMs)—to build motif-based models of protein families. A *motif* is a subsequence that is conserved across all or most members of a protein family. Biologically, a motif corresponds to a region of the protein that is essential for the proper functioning or structural conformation of the protein. MEME is an unsupervised motif discovery tool that, given an unaligned set of related protein or DNA sequences, builds statistical models of one or more motifs. Meta-MEME combines these motif models within a hidden Markov model framework. A Meta-MEME model improves upon the collection of individual motif models by including information about the typical order and spacing of motifs within the family.

Meta-MEME provides two important improvements over existing protein HMMs. First, because Meta-MEME's models focus on motif regions, they are much smaller than traditional protein HMMs. This decreased size makes the models more computationally efficient and allows the models to be trained from smaller data sets. Second, the generalized topology of Meta-MEME models implies a complex model of molecular evolution, allowing for the repetition or shuffling of motif-sized elements within a single protein sequence.

The models produced by Meta-MEME provide biologists with insight into the characteristics of the given family of related proteins. Furthermore, the models may be used to search protein databases for previously unidentified homologs and to generate multiple alignments of the motif regions of the proteins. Family Pairwise Search, although lacking an explicit model and accurate statistics, is much more efficient than Meta-MEME and provides better homology detection performance.

Chapter I

Introduction

The Human Genome Project is a fifteen-year, three billion dollar research effort coordinated by the U.S. Department of Energy and National Institute of Health, the goal of which is to discover the entire sequence of the human chromosome by the year 2005 [75]. This is a monumental task, involving the sequencing of approximately 100 000 genes in 3 billion base pairs of DNA. At the same time, the genetic sequences of other organisms, including bacteria, yeast and mice, are being decoded. The GenBank database of publicly available DNA sequences currently contains over 1.6 million entries [55], and the number of known sequences is increasing rapidly as various genome projects come on line [52, 54, 81].

This wealth of data represents a challenge for molecular biologists. Determining the sequence of As, Cs, Gs and Ts that make up the human genetic code is only the first step in the effort to understand the functions of the genes encoded therein. Ultimately, biologists hope to understand how the genes encoded on the chromosome work individually and in concert to affect and direct the development of the human organism. This knowledge, in addition to being of intrinsic scientific interest, will allow researchers to predict, prevent and treat a wide range of diseases that involve a genetic component. However, the task of discovering the function of each newly-sequenced gene is expensive and time-consuming. Given the rapid pace at which sequencing centers are now producing sequence data, functional analysis in the wet lab can only be applied to a small percentage of the new data.

On the other hand, since biological sequence data can be accurately represented in a computer database, the wealth of new sequence data provides an opportunity for computer scientists. Relatively fast analyses of these data can provide insight into the functions of new genes, both by analyzing the genes themselves and by teasing out similarities with genes for which functional information is already available. Computational analyses of biological sequence data may never replace the wet lab techniques of the molecular biologist. However, by mining statistically significant trends from genetic databases, the computer scientist can direct the attention of molecular biology, uncovering biologically significant functional information that might otherwise have remained undiscovered.

This dissertation presents a set of computational techniques for the analysis of genetic sequence data. Specifically, the methods developed here can be used to characterize families of evolutionarily related proteins. Drawing upon techniques from artificial intelligence and speech recognition, we produce probabilistic models of protein families. These models provide biologists with insight into the general characteristics of the given family, and may be used to identify evolutionarily related positions within the protein sequences and to search protein databases for previously unidentified family members. The accompanying software toolkit, Meta-MEME, implements the algorithms described herein. Meta-MEME is freely available as ANSI C source code and is accessible over the web for use by biologists [63].

The general approach adopted here is Bayesian[1] in the sense that every modeled event is represented by a random variable. Bayesian modeling involves selecting a set of relevant random variables and a corpus of relevant prior knowledge, then using the given evidence, the background knowledge, and the laws of probability to draw conclusions. Several useful introductions to Bayesian statistics are available [110, 132].

In keeping with this approach, we begin by stating explicitly some of the

[1]Thomas Bayes (1702–1761) was an English mathematician and cleric who developed the foundation of what would later become probability theory.

background knowledge that informs the models produced by Meta-MEME. Section I.A describes the biological background knowledge that Meta-MEME employs. In addition to basic knowledge about what proteins consist of and how they work, Meta-MEME assumes that proteins can be best described as consisting of a series of motifs. In this context, a *motif* is a structural or functional unit that appears in a similar form in many evolutionarily related proteins. Meta-MEME's models are motif-based in the sense that they assume that most of the relevant information about a protein is concentrated in these motif regions. Meta-MEME assumes that, besides the contents of the motif regions, the only important information about a protein resides in the order and spacing of the motifs within the protein. Thus, the prior information that proteins are motif-based allows Meta-MEME to focus its modeling efforts on these relatively small regions, ignoring the noisier, spacer regions between motifs.

The motif-based nature of proteins fits well with a second piece of background knowledge implicit in Meta-MEME, namely, that molecular evolution is largely domain-based [119, 36, 45, 25, 44, 103, 95]. A *domain* is a separately evolved, independent structural unit of a protein. Generally speaking, domains are larger than motifs, and a single domain may contain multiple motifs. In a domain-based model of evolution, proteins evolve via point mutation, but also by less well-understood, large-scale mechanisms such as gene duplication, exon shuffling and transposable elements. The topology of the Meta-MEME's models reflects this domain-based view of molecular evolution.

In addition to providing an overview of the relevant molecular biology, Chapter I describes three important tasks in this domain that can be addressed by computational means: multiple alignment, phylogenetic inference, and homology detection. Chapter I also introduces the class of statistical models, called hidden Markov models, that will be employed in the modeling of protein families.

Chapter II describes in detail the architecture of the protein models used by Meta-MEME, as well as the algorithms employed in building, training, and using these models.

In Chapter III, an alternative method for solving the homology detection task is presented. This method, called Family Pairwise Search (FPS), is a relatively simple algorithm that involves combining the scores of multiple pairwise comparisons. FPS was developed as a baseline for comparison with other homology detection methods, and its excellent performance is surprising [59]. Accordingly, in Chapter III we develop a hybrid, motif-based extension to FPS and show that incorporating motif models into the algorithm leads to still better performance.

The final chapter of the dissertation presents experimental results that examine Meta-MEME's effectiveness on the multiple alignment and homology detection tasks. The primary conclusion is that Meta-MEME, in its current instantiation, fails to improve upon the homology detection performance of its non-HMM counterpart, MAST. A multi-motif model of the type constructed by Meta-MEME should be able to exploit information about the order and spacing of motifs within the family. This meta-motif information is not available to a method that treats motif models separately. The assumption that this information is valuable in detecting homologs is not violated by Meta-MEME's poor performance. Rather, the experiments reported in Chapter IV suggest that the usual expectation-maximization training algorithm for HMMs is inappropriate for Meta-MEME models. Furthermore, the relatively poor performance of purely motif-based methods, compared with that of the motif-based extension of Family Pairwise Search, suggests that a method which focuses on motifs but retains information from non-motif regions provides the best possible homology detection performance.

I.A Biology background

I.A.1 Genes and proteins

Genes are commonly understood as discrete genetic elements, each of which determines a particular phenotypic trait. The gene, for example, for eye color de-

```
TACACAGAGAGCCACGGCCAGGGCTGAAACAGTCTGTTGAGTGCAGCCATGGGGGACGTCCTGGAACAGTTCTTCATCCTCACAGGGCTGCTGGT
GTGCCTGGCCTGCCTGGCGAAGTGCGTGAGATTCTCCAGATGTGTTTTACTGAACTACTGGAAAGTTTTGCCAAAGTCTTTCTTGCGGTCAATGG
GACAGTGGGCAGTGATCACTGGAGCAGGCGATGGAATTGGGAAAGCGTACTCGTTCGAGCTAGCAAAACGTGGACTCAATGTTGTCCTTATTAGC
CGGACGCTGGAAAAACTAGAGGCCATTGCCACAGAGATCGAGCGGACTACAGGGAGGAGTGTGAAGATTATACAAGCAGATTTTACAAAAGATGA
CATCTACGAGCATATTAAAGAAAAACTTGCAGGCTTAGAAATTGGAATTTTAGTCAACAATGTCGGAATGCTTCCAAACCTTCTCCCAAGCCATT
TCCTGAACGCACCGGATGAAATCCAGAGCCTCATCCATTGTAACATCACCTCCGTAGTCAAGATGACACAGCTAATTCTGAAACATATGGAATCA
AGGCAGAAAGGTCTCATCCTGAACATTTCTTCTGGGATAGCCCTGTTTCCTTGGCCTCTCTACTCCATGTACTCAGCTTCCAAGGCGTTTGTGTG
CGCATTTTCCAAGGCCCTGCAAGAGGAATATAAAGCAAAAGAAGTCATCATCCAGGTGCTGACCCCATATGCTGTCTCGACTGCAATGACAAAGT
ATCTAAATACAAATGTGATAACCAAGACTGCTGATGAGTTTGTCAAAGAGTCATTGAATTATGTCACAATTGGAGGTGAAACCTGTGGCTGCCTT
GCCCATGAAATCTTGGCGGGCTTTCTGAGCCTGATCCCGGCCTGGGCCTTCTACAGCGGTGCCTTCCAAAGGCTGCTCCTGACACACTATGTGGC
ATACCTGAAGCTCAACACCAAGGTCAGGTAGCCAGGCGGTGAGGAGTCCAGCACAACCTTTTCCTCACCAGTCCCATGCTGGCTGAAGAGGACCA
GAGGAGCAGACCAGCACTTCAACCTAGTCCGCTGAAGATGGAGGGGGCTGGGGTCACAGAGGCATAGAATACACATTTTTTGCCACTTT
```

(a)

```
MGDVLEQFFILTGLLVCLACLAKCVRFSRCVLLNYWKVLPKSFLRSMGQWAVITGAGDGIGKAYSFELAKRGLNVVLISRTLEKLEAIATEIERT
TGRSVKIIQADFTKDDIYEHIKEKLAGLEIGILVNNVGMLPNLLPSHFLNAPDEIQSLIHCNITSVVKMTQLILKHMESRQKGLILNISSGIALF
PWPLYSMYSASKAFVCAFSKALQEEYKAKEVIIQVLTPYAVSTAMTKYLNTNVITKTADEFVKESLNYVTIGGETCGCLAHEILAGFLSLIPAWA
FYSGAFQRLLLTHYVAYLKLNTKVR
```

(b)

Figure I.1: **The DNA (a) and corresponding amino acid (b) sequences of human 17β hydroxysteroid dehydrogenase.**

termines whether your eyes will be blue, green, brown or some other shade. The truth, of course, is much more complicated. Figure I.1(a) shows a single human gene. Rather than determining a particular trait, this gene, like the vast majority of all genes, encodes a sequence of DNA bases (adenine, guanine, thymine and cytosine) that serves as the blueprint for a particular protein. Occasionally, as in the case of eye color, the protein thus encoded has a function that is closely linked to a particular phenotypic trait. Usually, however, the function of the protein, and hence of the gene that codes for it, does not have a directly observable phenotype.

Figure I.1(b) shows the amino acid sequence for the protein coded by the gene in Figure I.1(a). Rather than being composed of four bases, proteins are constructed from an alphabet of twenty amino acids. The universal genetic code translates from triples of DNA bases into single amino acids. The mechanism by which a gene is transcribed and translated from DNA bases on the chromosome into a separate protein is complex; however, for our purposes, it is enough to know that this process of transcription and translation occurs. We are primarily interested in the resulting proteins.

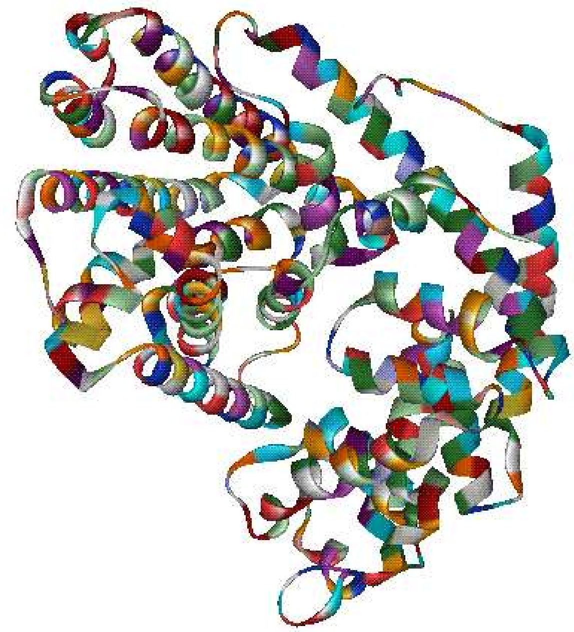
I.A.2 Protein structure and function

Proteins are of scientific interest because they perform essentially every task that an organism needs to accomplish at the molecular level. Thus, proteins perform tasks as varied as opening and closing ion channels in neurons, transporting molecules to various parts of the cell or across permeable membranes, and latching onto and thereby disarming invading parasites.

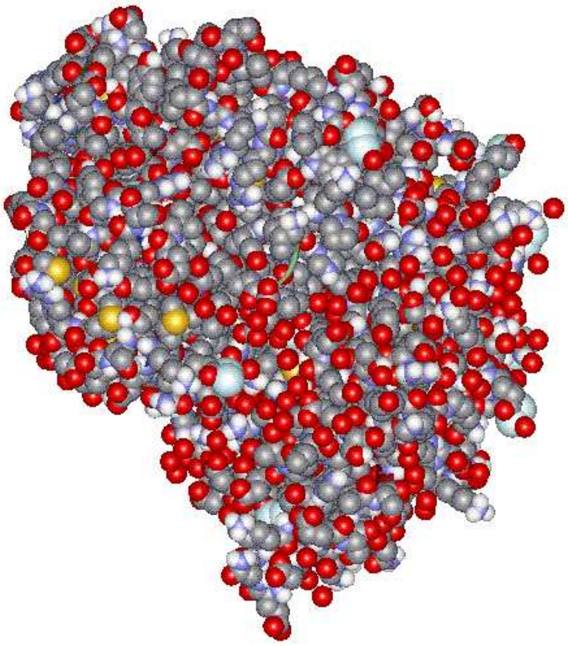
Any protein can be represented as a sequence of amino acids, varying in length from around 50 to over 5000. However, this amino acid representation of the protein does not tell the whole story. Figure I.2 shows two views of the three-dimensional structure of tobacco 5-epi-aristolochene synthase [120]. In Figure I.2(a), only the backbone of the molecule is portrayed. This backbone corresponds to the amino acid sequence shown in Figure I.2. The three-dimensional configuration of this backbone is well defined: every time a cell synthesizes this particular sequence of amino acids, the sequence always folds into this exact three-dimensional structure. Each amino acid consists of between 10 and 22 atoms. Figure I.2(b) shows the same molecule with all of the atoms included. This three-dimensional structure is the complete picture of tobacco 5-epi-aristolochene synthase.

The three-dimensional structure of a protein is important because it determines the protein's function. For example, enzymes are proteins that specifically bind to a wide range of molecules. The target molecule, called the substrate, attaches to a particular site on the enzyme, called the active site. The active site consists of a small set of amino acids that are adjacent to one another on the surface of the enzyme. The function of the enzyme critically depends upon the chemical properties of the amino acids in the active site. If these properties change—if, for example, a mutation in the DNA causes one of the amino acids in the active site to be different from usual—then the enzyme may fail to perform its function because it does not bind properly to the substrate.

The three-dimensional structure of many proteins can be determined directly using crystallographic techniques. However, these techniques are even more expen-



(a)



(b)

Figure I.2: **Two views of the three-dimensional structure of tobacco 5-epi-aristolochene synthase [120].**

sive and time-consuming than the functional analyses of proteins. Finding the crystal structure of a single protein can take up to eighteen months and is not guaranteed to be successful. Indeed, for large classes of proteins, including most transmembrane proteins, the proteins' lack of solubility precludes the use of crystallography for determining structure.

In theory, however, since nearly every protein sequence folds into a unique three-dimensional structure, it should be possible to predict that structure from the amino acid sequence alone. This prediction task is called the protein folding problem. Since a protein's function is almost solely dependent upon the protein's three-dimensional structure, and since biologists are very interested in discovering the functions of newly sequenced proteins, there is strong motivation to solve the protein folding problem. Indeed, scientists have been attempting to do so since the late 1940s, but with very limited success. In the absence of a solution to the protein folding problem, we must search for other means of inferring protein function.

Currently, the most successful means of inferring protein function exploits information about evolutionary relationships among proteins. Proteins that share a common evolutionary ancestor are said to be *homologous*. A set of homologous proteins, all of which are descended from a common ancestor, is called a protein family. Because the overall three-dimensional structure, or fold, of a protein remains fairly constant over evolutionary time, the various members of a protein family typically share a common fold. This similarity of fold implies a similarity of function, with the degree of functional similarity depending upon the degree of evolutionary divergence that has occurred within the family.

I.A.3 Three tasks

One of a biologist's most useful tools for displaying the features of a set of evolutionarily related sequences is the multiple alignment. Figure I.4 shows an example of such an alignment. The alignment specifies the position-by-position correspondence between homologous proteins. This correspondence has evolutionary, and often

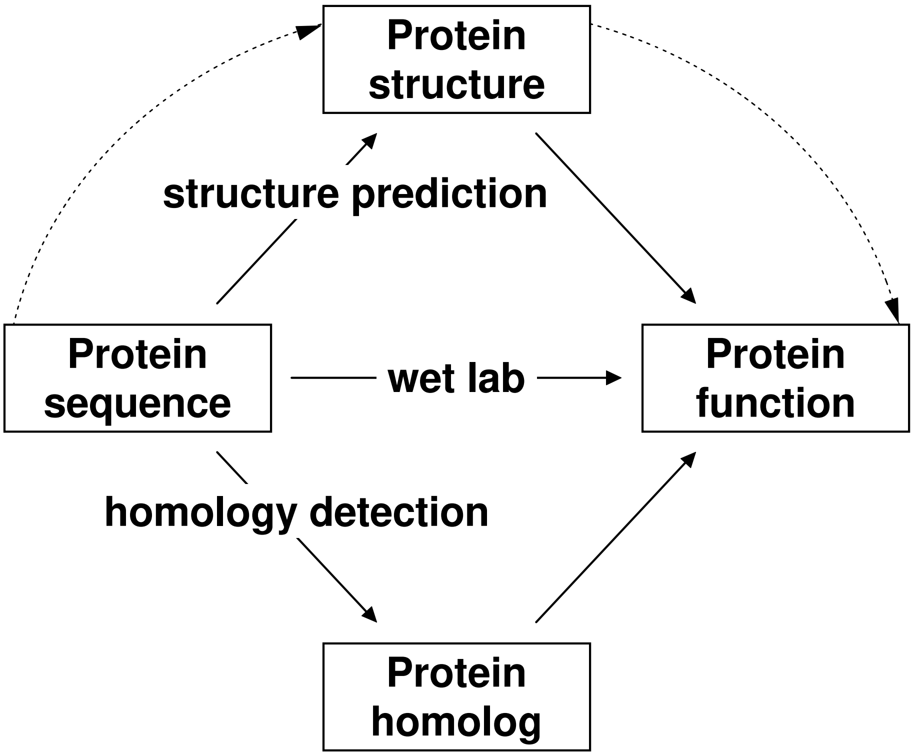


Figure I.3: **Three different means of inferring protein function.** The function of a protein can be inferred directly via wet lab experiments, or indirectly either by discovering the three-dimensional structure of the protein or by finding one or more homologous proteins that provide evidence for the original protein's function. The dotted line in the figure represents the biological determination of protein function. The protein consists of a sequence of amino acids that fold into a three-dimensional structure which determines the protein's function.

```
ASG2_ECOLI ALALPNITILATGGTIAGGGDSATKSN.YTVGKVGVENLV
ASG2_HAEIN AADLPNITILATGGTIAGSGQSSVNSA.YKAGQLAIDTLI
ASPG_WOLSU .MAKPQVTILATGGTIAGSGESSVKSS.YSAGAVTVDKLL
ASPG_ERWCH ADKLPNIVILATGGTIAGSAATGTQTTGYKAGALGVDTLI
 consensus AXALPNITILATGGTIAGSGXSSXKSXGYKAGA GVDTLI
```

Figure I.4: **An example of four aligned sequences.** See text for description of the underlined region.

structural or functional, significance. The alignment can show which regions of the proteins are highly conserved and hence likely to be evolutionarily significant. Furthermore, for example, if we know the three-dimensional structure of ASG2_ECOLI and from that structure we find that the active site consists of the underlined amino acids (GGTIAGGGD), then from the given multiple alignment we could safely infer the locations of the active sites of the other sequences in the alignment, even though we may not have structural information for these sequences. This information could then be used, for example, to decide which sites to mutate when performing a functional analysis of ASG2_HAEIN.

In addition to providing functional insights, multiple alignments may serve as input to phylogenetic inference algorithms. The usual means of inferring the phylogenetic tree that relates a set of species is to extract one or more proteins from each species in the set, build an alignment of the proteins, and then infer a phylogenetic tree from the alignment using a criterion such as maximum parsimony [51] or maximum likelihood [49]. Thus, in this method, the multiple alignment is a necessary first step in finding the phylogenetic tree.

In order to build a multiple alignment, we must start with a set of sequences that are known to be homologous. Furthermore, one of the most effective means of inferring the function of an unidentified protein is to ask what functions are performed by homologous proteins. Thus, discovering homologies between sequences is very useful. Unfortunately, the only way to conclusively prove that two sequences are homologous would be to prove their descent from a common ancestral sequence. In practice, this common ancestor is not available, so the only means of detecting homology is to infer it by statistical means.

The most widely used means of inferring homology involves performing a pairwise comparison between a single query sequence and a sequence in a protein database. Dynamic programming algorithms, such as the Smith-Waterman algorithm [118] and its heuristic approximations, BLAST [2] and FASTA [104], can be used to assign to each sequence in a database a score indicating the probability 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 [43]. For most pairwise alignment programs, the twilight zone falls between 20-25% sequence identity [38].

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, homology inference algorithms can uncover these missed relationships because homology is transitive [106, 3].

The simplest means of detecting homologs using a set of related query proteins is to perform multiple pairwise comparisons [59]. 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 [98].

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. Examples of such models include profiles [57] and hidden Markov models (HMMs) [83, 22, 46]. One drawback to model-based homology detection is that the models usually contain many free parameters and therefore require a large amount of training data. For example, a typical, 200-state HMM may contain on the order of 5000 trainable parameters.

GUNA_BUTFI	VIYEICNEP
GUNB_RUMAL	LIFEGLNEP
GUN_BACS6	IIWELANEP
GUN3_FIBSU	LFFELLNEP
GUNC_PSEFL	IGIDVFNEP
GUN1_BUTFI	LVFETMNEP
GUNB_BACLA	LMFESVNEP
GUN5_THEFU	VLYEIANEP
GUND_CLOCL	LIFETMNEP
GUNH_CLOTM	LLFEIMNEP
GUNA_XANCP	LGLDLKNEP
GUNC_CLOTM	IAFELLNEV
GUNB_CLOTM	IGFDLKNEP
EXG1_YEAST	IGIELINEP

Figure I.5: **A typical motif.** Column 1 consists of the names of the sequences belonging to the family; column 2 contains the corresponding motif instances. This particular motif is the glycosyl hydrolases family 5 signature. [12, 74]

I.A.4 Motifs

The size of the model may be greatly reduced by focusing only upon motif regions. A *motif* is a short subsequence that is highly conserved across family members. Figure I.5 shows an example of one motif. The motif is represented by a subsequence of length nine excised from each of fourteen members of the family. The level of conservation within the motif is high: all members of the family have an asparagine (N) in the seventh position and a glutamic acid (E) in the eighth position. Even the less conserved columns contain amino acids that are biochemically similar, such as the valines (V), leucines (L) and isoleucines (I) in the first column.

Usually, motifs are conserved by evolution for important structural or functional reasons. For example, the motif may participate in the binding site of the protein, or it may play a critical role in stabilizing the overall structure of the molecule. As such, the motifs constitute a summary, or fingerprint [4], of the biologically essential details of the family of proteins [91].

Motifs may be modeled as regular expressions [12, 99], or more generally as profiles [57] 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 may be learned via expectation-maximization [6] or Gibbs sampling [87] 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 than are required for complete sequence modeling.

Unfortunately, using multiple models in concert to find homologous sequences can be difficult. One means of combining multiple motif models is to search the target database with each motif model separately and then to combine the scores, assuming that motif occurrences are statistically independent. This is the approach taken by MAST [11]. Although the independence assumption is clearly invalid in theory, the technique works well in practice.

One important improvement upon MAST's approach would be to exploit information about the order and spacing of motifs within the protein family. This information is a critical part of the motif signature for the family [92] and can be critical in determining the statistical significance of a weak match to the motif model. For example, consider a training set of sixteen sequences, each containing three motifs in the order 1-2-3 and separated by spacers of length 10. If a candidate sequence contains motifs 1 and 2 in the right positions, and a weak match to motif 3 ten amino acids after motif 2, then that match is very likely to be genuine. On the other hand, if the same weak match appears in a sequence with no other motif matches, then the match is likely to be false. MAST, by treating motif occurrences independently, sacrifices the opportunity to exploit information about the order and spacing of motifs within the family.

In contrast, the BLOCKS method for protein family classification [71, 67] does take the order and spacing of motifs into account when searching for homologous sequences. The BLOCKS database [28] contains, for each known protein family, an ordered set of motif models (called blocks) along with the minimum and maximum observed spacings between the blocks in the training set. The BLIMPS program [73] searches this database of blocks using a single sequence as a query, thus taking into account the order and spacing of motifs. The primary drawback to the BLOCKS method is the rigidity of the spacer length specifications and the lack of a statistical model from which accurate scores can be derived.

A preferable approach, rather than using motif models independently or combining them in an *ad hoc* fashion, would be to build a completely probabilistic, motif-based model of the protein family. This model would contain all of the information in the individual motif models, as well as information about the order and spacing of motifs in the family. An appropriate framework for such a model is provided by the theory of hidden Markov models (HMMs). HMMs have a well-founded probabilistic interpretation and are supported by efficient algorithms for training and use. Accordingly, the Meta-MEME toolkit builds, trains and uses motif-based HMMs of protein families. Before describing the details of the Meta-MEME algorithms, however, we provide the reader with background on hidden Markov models in general and on their use in computational molecular biology.

I.B Hidden Markov models

I.B.1 Definition

A hidden Markov model (see Figure I.6) is a mathematical framework that models a series of observations based upon a hypothesized underlying but hidden process. The model consists of a set of states and transitions between these states. Each state emits a signal, based upon a state-specific emission probability distribution, and then stochastically transitions to some other state, based upon a transition probability distribution. If we denote the state at time t as q_t, then a hidden Markov model is completely characterized by the following parameters [112]:

- the number N of states in the model, with individual states denoted as S_i for

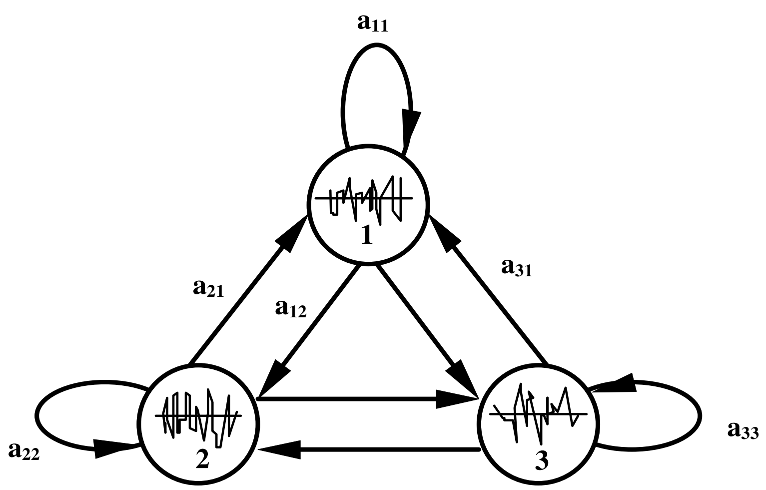


Figure I.6: **A simple, three-state hidden Markov model.** Each state has an associated emission probability distribution that determines what observation is emitted by the state and a transition probability distribution that determines which state will be visited next.

$1 \le i \le N$,

- the number M of distinct observation symbols per state, with individual symbols denoted as v_i for $1 \le i \le M$,
- initial state probabilities $\pi_i = Pr(q_1 = S_i)$ for $1 \le i \le N$,
- state transition probabilities $A = [a_{ij}]$, where $a_{ij} = Pr(q_{t+1} = S_j | q_t = S_i)$ for $1 \le i, j \le N$, and
- observation probabilities $B = \{b_i(k)\}$, where $b_i(k) = Pr(v_k \ at \ t | q_t = S_i)$ for $i \le i \le N$ and $1 \le k \le M$.

Although introduced relatively recently to computational molecular biology, HMMs have been in use for speech recognition for many years [14]. In speech recognition, the series of observations being modeled is a spoken utterance; in computational biology, the series of observations is a biological sequence of DNA bases or amino acids.

Hidden Markov models have a strong theoretical basis in probability and are supported by efficient algorithms for training, database searching, and multiple

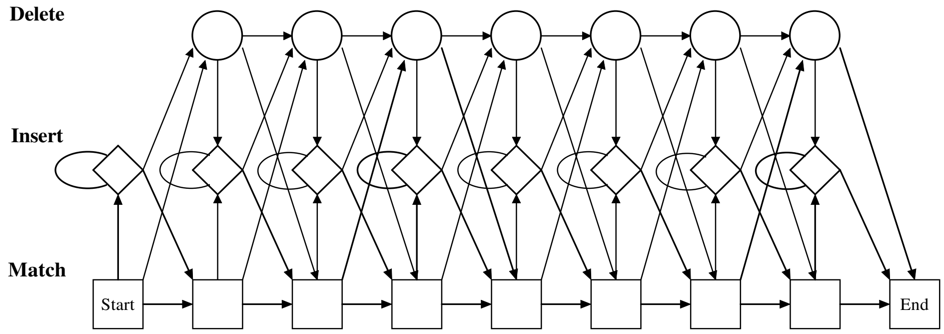


Figure I.7: **The topology of a standard linear HMM.** Emission probability distributions for match and insert states are not shown.

sequence alignment. A useful HMM tutorial was written by Rabiner [112], and more detailed information is available in [113]. The tutorial describes three basic problems for HMMs: given an observation sequence and a model, how do we (1) efficiently compute the probability of the observation sequence, given the model, (2) choose a corresponding state sequence which is optimal in some meaningful sense (i.e., best "explains" the observations), and (3) adjust the parameters of the model to maximize the probability of the sequence, given the model? In computational biology, Rabiner's three problems correspond to (1) determining whether a given sequence belongs to the modeled family, (2) finding an alignment of the given sequence to the rest of the family, and (3) training the model based upon known members of the family. The model parameters are learned via expectation-maximization [42], and the sequences are aligned and homolog detected via dynamic programming.

I.B.2 The standard HMM topology

Hidden Markov models were first applied to problems in molecular biology by Churchill [39]. Krogh *et al.* [83] applied HMMs to protein modeling and brought widespread recognition to the approach. We refer to the linear HMMs described in that paper as *standard HMMs*. This standard topology has subsequently been used by many researchers [46, 22]. Two standard HMM packages are freely available, SAM [78, 115] and HMMER [46, 76].

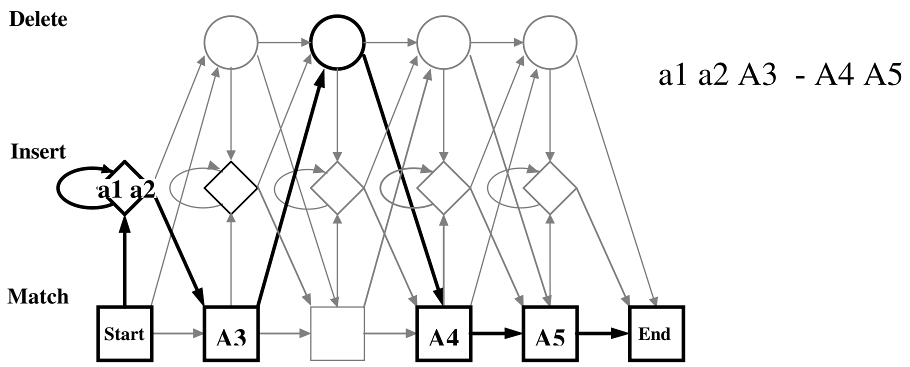


Figure I.8: **The HMM as a generative model.** The HMM generates the sequence a1 a2 A3 A4 - A5 by alternately emitting a symbol according to the current state's emission distribution and then transitioning to a new state based upon the transition distribution. The "-" symbol corresponds to a non-emitting delete state in the model and would be a gap in a multiple alignment.

The topology of the standard HMM (see Figure I.7) attempts to reflect the process of molecular evolution. The core of the standard model is a sequence of states, called *match states*, that represent the canonical sequence for this family. Each match state corresponds to one position in the canonical sequence. This series of states is similar to a profile [57] or to a MEME motif model, since the emission probabilities at each state are distributed across the alphabet of amino acids.

To model the process of evolution, two additional types of states—*insert* and *delete states*—are included in the HMM. One delete state lies in parallel with each match state and allows the match state to be skipped. Since delete state do not emit characters, aligning a sequence to a delete state corresponds to the sequence having a deletion at that position. Insert states with self-loops are juxtaposed between match states, allowing one or more bases to be inserted between two match states. These three series of states are connected as shown in Figure I.7. The topology of the model is linear: once a state has been traversed, it cannot be entered a second time. The three types of states in the standard model imply a simple model of molecular evolution that involves only point mutations, insertions and deletions.

Although rarely used in this fashion, HMMs may be understood as gener-

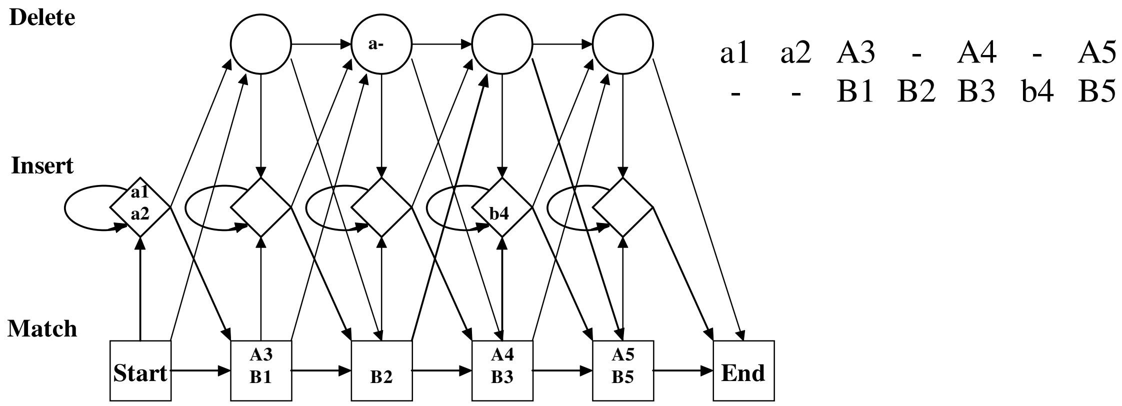


Figure I.9: **Aligning two sequences according to the Viterbi paths.** The Viterbi path is the most likely path through the model, given the sequence. A sequence alignment is generated by aligning symbols emitted from corresponding states. Multiple symbols emitted from a single insert state are not aligned.

ative models (see Figure I.8). Starting at some pre-determined, non-emitting initial state, the succeeding state is selected randomly according to the transition probability distribution at the start state. At the new state, an amino acid is randomly emitted according to the emission probability distribution at that state. From there the process repeats, emitting a symbol and transitioning to a new state. The process terminates when the stop state is reached.

I.B.3 Using HMMs for multiple alignment

The evolutionary model implicit in the topology of an HMM enables these models to be used for multiple sequence alignment. The Viterbi algorithm [41] is a dynamic programming algorithm that calculates the series of model states $q_0 \dots q_T$ most likely to have generated a given sequence $O_0 \dots O_T$; i.e., the algorithm calculates

$$\max_q \pi_0 \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(O_t). \tag{I.1}$$

The resulting series of states is called the Viterbi path. A multiple alignment for a set of sequences may be generated by finding the Viterbi path for each sequence and then aligning each path to the original model, as shown in Figure I.9.

I.B.4 Using HMMs for homology detection

HMMs also provide an accurate means of detecting sequence homologies. The forward algorithm is similar to the Viterbi algorithm, except that it computes the total probability of a sequence, given a model. Thus, during the updating of cells in the dynamic programming matrix, the maximum operation performed by the Viterbi algorithm is replaced by a summation to find the total probability of the sequence. An appropriately normalized version of this probability score may be used to determine whether a candidate sequence from a database belongs to the modeled family. The ability of an HMM to discriminate, with a high degree of precision and recall, family members from non-family members in a large sequence database indicates that the model incorporates necessary and sufficient conditions for family membership, with respect to the proteins in the given database.

Because the similarity between protein family members typically reflects a similarity in their three-dimensional structures, HMMs are implicitly attacking a version of the protein folding problem. However, instead of determining how a given protein will fold, the HMM only determines (roughly) whether the protein will fold in a particular way, i.e., the way that other members of the family fold. In this respect, hidden Markov modeling resembles a threading approach to the protein folding problem [29].

I.B.5 Drawbacks of the standard topology

Hidden Markov models have been successfully applied in the domains of speech recognition and biological sequence modeling. One immediately apparent difference between these two domains is the amount of available training data. Training sets for state-of-the-art speech recognition systems frequently contain many gigabytes of recorded speech; in contrast, families of related biological sequences usually consist of kilobytes or even hundreds of bytes of characters. Even for speech recognition systems, for which the training set size is relatively large, researchers attempt to simplify their models in order to reduce the number of trainable parameters [134]. When modeling biological sequences, the need for smaller models is even more pronounced.

The large number of model parameters is a major weakness of the standard protein HMM. A standard HMM of length n using an alphabet of size 20 contains 6 trainable transition probabilities and 19 trainable match state emission probabilities for each of n positions, as well as 19 insert state emission probabilities, yielding a total of $25n + 19$ trainable parameters. For a protein family whose members have an average length of 200, such a model contains 5019 parameters. Many small families of biological sequences contain less than this number of characters in all known family members combined. In order to estimate the values of the model parameters reliably, a large training set of proteins that are already known to be related is required. For example, over 200 randomly selected sequences are required to adequately model the globin family [32], and Krogh *et al.* [83] mention a lower limit of approximately 70 carefully selected training sequences in order to model the same family. Smaller families cannot effectively train a standard HMM because reliable training requires that the number of samples greatly exceeds the number of free parameters. A model based upon a smaller data set may overfit the data, modeling details specific to the training set but not to the larger protein family.

In order to avoid overfitting, standard HMMs often rely upon a set of Bayesian prior probabilities, such as Dirichlet mixture priors empirically derived from known multiple alignments [32, 116]. However, even with accurate prior probabilities, when the training set is small and the model is large, the trained model will depend upon the prior probabilities more than it reflects the training sequences. The only effective means of ensuring that the trained model reflects the characteristics of a particular protein family is to keep the number of model parameters small.

Another important disadvantage of the standard topology is that it implies an over-simplified model of molecular evolution that involves only point mutations, insertions and deletions. The actual mechanisms of molecular evolution are quite complex, involving point mutations, crossover, exon shuffling, gene duplication and various types of transposable elements [89]. Many of these mechanisms operate on a larger scale than single amino acids. The separately evolved, independent structural units that comprise most proteins are called protein *domains* [25, 44, 103, 95] The diversity of protein functions results primarily from the combinatorial arrangement of a finite number of these domains [36, 45]. Thus, most proteins can be understood and accurately modeled as an arrangement of autonomously structured domains [119]. A protein family such as the kinases may consist of individual proteins that contain a common set of domains in different orders. More frequently, small sections of an ancestral protein may be cut out and re-inserted multiple times, with the result that a single protein may contain up to 50 copies of a given subsequence. These phenomena cannot be accounted for by a linear topology.

The Meta-MEME software toolkit directly addresses both of these drawbacks of standard HMMs. By constructing motif-based models, Meta-MEME greatly reduces the number of parameters in a typical model. And by allowing for the creation of models with non-linear topologies, Meta-MEME allows for a more realistic implicit model of molecular evolution.

Chapter II

Meta-MEME

The Meta-MEME software toolkit [61, 62], along with the MEME motif discovery tool [6], provides biologists with a complete set of Bayesian, motif-based sequence analysis tools. Given a training set of known family members, the biologist can discover a representative set of motifs, combine those motifs into a single, motif-based hidden Markov model, re-train the motifs and the transitions between the motifs in the context of the HMM, and then use the model to build multiple alignments and detect previously unknown homologies. This process is summarized in Figure II.1 and will be described in detail in this chapter.

II.A Motif discovery using MEME

MEME (Multiple Elicitation of Motifs by Expectation-maximization) is an unsupervised motif discovery tool [6, 8, 7, 60]. Given an unaligned set of related protein or DNA sequences, MEME discovers therein one or more conserved motif regions and builds statistical models of those regions. The parameters of the models are trained via expectation-maximization so as to maximize the posterior probability of the data, given the model. Since MEME does not allow insertions or deletions within motifs, the models that it builds are matrices in which each column of the matrix contains an amino acid distribution. Thus, such a model is formally equivalent

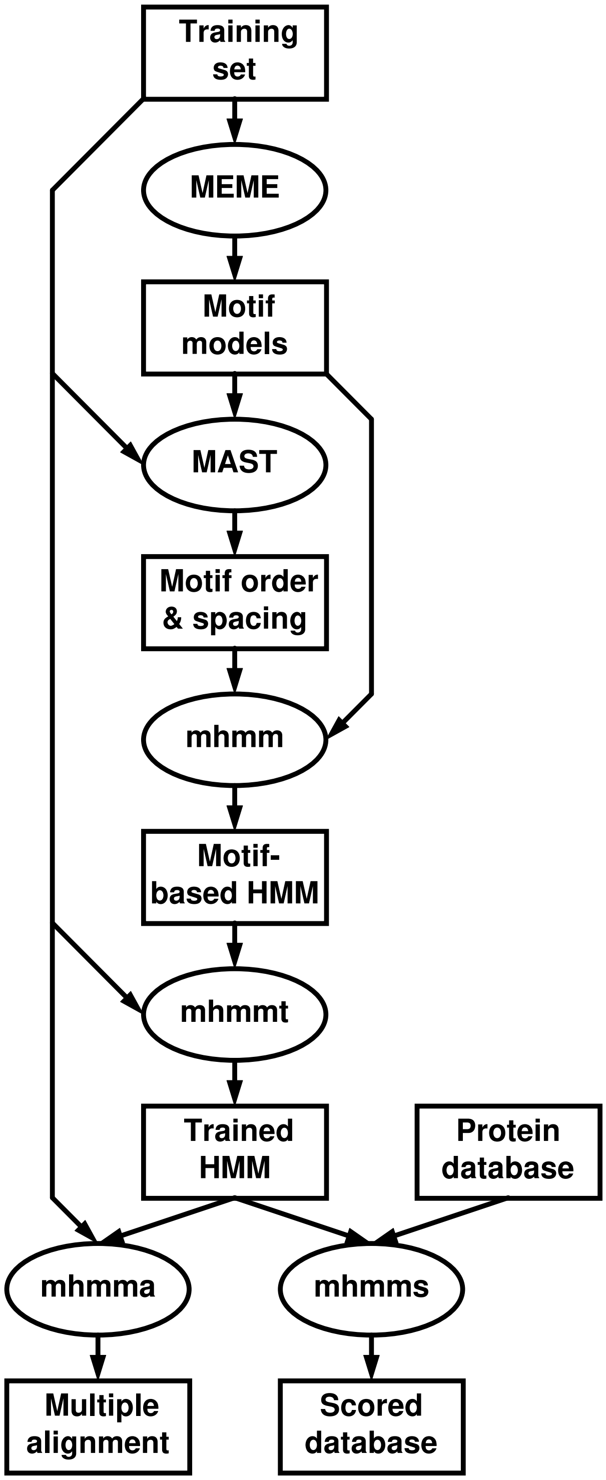


Figure II.1: **Schematic diagram of the Meta-MEME toolkit.**

to a standard linear HMM with the insert and delete states removed. A parallelized version of MEME running on a supercomputer is available on the web [60].

MAST (Motif Annotation and Search Tool) [11, 10] is a companion program for MEME that uses MEME motif models to search a sequence database for homologs of the initial training set. For each sequence in the database, MAST computes a p-value for each motif in the query and combines these values assuming that motif occurrences are statistically independent. The resulting sequence-level p-value scores are used to rank the sequences in the database.

As mentioned in section I.A.4, MAST's discriminative ability would likely improve if it searched concurrently with multiple motifs, rather than assuming that motif occurrences are statistically independent. By treating motifs independently, rather than in concert, MAST ignores important information about the order and spacing of motifs within a family. This information is an essential aspect of the family signature [92].

Meta-MEME addresses the problem of searching using multiple motif models by combining motif models within a hidden Markov model framework. Although Meta-MEME has only been tested using motifs discovered by MEME, extending the approach to motifs discovered by other means, such as the Gibbs sampler [87] or BLOCKMAKER [71, 67], would be straightforward. Meta-MEME models fall into two categories: linear models, which are a constrained version of the standard topology described previously, and completely connected models, which allow for a more complex implicit model of molecular evolution. The following sections describe these two types of topologies.

II.B Linear models

The topology of the linear models produced by Meta-MEME can be represented as a constrained version of the standard HMM topology (see Figure II.2). As in MEME, the motifs themselves allow neither gaps nor insertions; thus, each motif

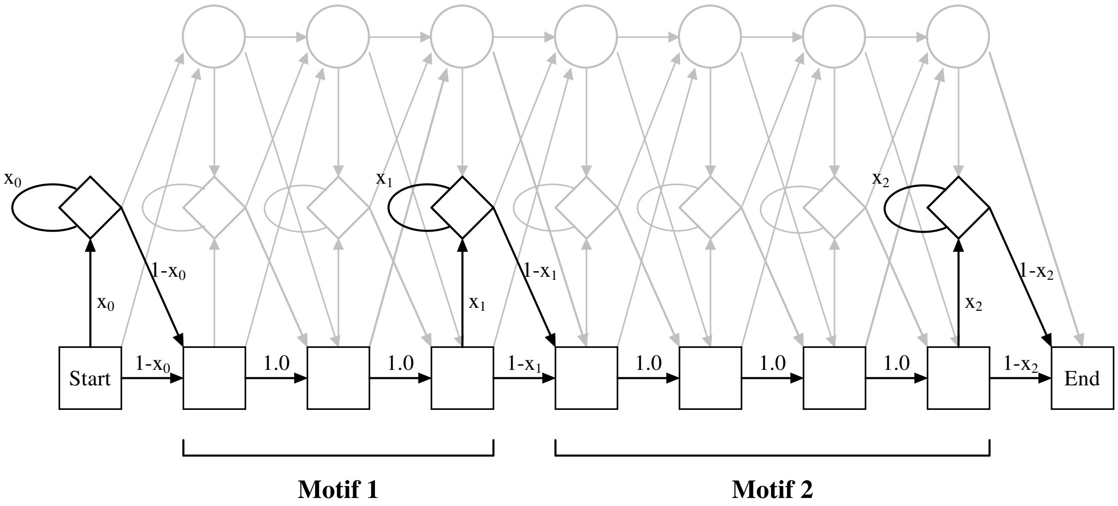


Figure II.2: **A small, linear motif-based HMM.** Only the darker nodes and transitions are used in the model; the gray background nodes would appear in a standard HMM but are unreachable in this HMM. Note that this is a simplified example; real motifs generated by MEME are longer.

is modeled by a sequence of match states, with transition probabilities of 1.0 between adjacent states. Spacer regions between motifs are represented by insert states, as described in more detail in section II.D.

In order for Meta-MEME to build multi-motif models from MEME output in an unsupervised way, the program must decide automatically how many motifs to include in the model. For models with a linear topology, Meta-MEME uses a simple heuristic. As MEME generates successive motifs for a data set, it first finds the highly significant motifs and then begins to model motifs which are conserved in only a subset of the given sequences. In effect, MEME finds motifs representing subfamilies of the given family. Since such subfamily motifs are not useful for characterizing the entire family, they should not be included in the Meta-MEME model. Models generated by Meta-MEME, therefore, only incorporate those motifs for which the motif occurs in the majority of the training sequences, up to some user-selected maximum number of motifs. Motif occurrences are defined by a threshold on the p-values calculated by MAST.

Once the motif models have been generated by MEME and selected according to the majority occurrence heuristic, they must be combined into a single, 25-[4]-21-[1]-12-[2]-2-[5]-67-[3]-40

Figure II.3: **An example of a motif occurrence diagram as generated by MAST.** The diagram shows the lengths of non-motif regions, alternating with the indices of five motifs (in brackets). Motifs are indexed according to the order in which MEME discovers them.

linear model. Ideally, the order and spacing of motifs should reflect the canonical order and spacing of motifs in the family. To determine this canonical motif schema, MAST searches the given training set of sequences. For each such sequence, MAST produces a motif occurrence diagram (see Figure II.3) that shows the motif occurrences with p-values less than 0.0001, as well as the lengths of the spacers between occurrences. Meta-MEME selects from this output the highest-scoring sequence containing significant matches to each of the motifs selected for use in the HMM. The motif occurrence diagram associated with this sequence is then used as a template for building the linear HMM.

A Meta-MEME model with a linear topology solves one of the two major problems faced by standard HMMs. By focusing on motif regions, such a model reduces the number of parameters relative to a standard HMM and hence is more easily trained. Admittedly, a Meta-MEME model also effectively reduces the size of the training set by throwing out training data from the noisy regions. However, because proteins are motif-based, Meta-MEME models can be accurately trained from smaller data sets than can standard HMMs. The simplest way to see this is to think of the extreme case, in which all of the information in the protein family is concentrated in the motif regions. In this case, the non-motif regions would contain pure noise. For a standard HMM of such a family, the number of observations per state in the motif and non-motif regions would be the same, but the non-motif regions would still be completely untrained: training them would amount to randomly perturbing the parameters. This is the sense in which the parameters of such a model are underdetermined: the non-motif regions may be trained by the same number of observations per state as the motif regions, but the non-motif counts are effectively fewer because of the noise. By throwing out the non-motif regions, the average "trained-ness" of the parameters increases.

II.C Completely connected models

Although the Meta-MEME models described above are more easily trained than standard HMMs, both types of models are linear; hence, both topologies fail adequately to account for complex, domain-based mechanisms of molecular evolution. Modeling phenomena such as repeated or shuffled protein domains requires the introduction of cycles into the HMM topology. For the standard HMM, introducing such non-linearity is too expensive, with respect both to the size of the model and the cost of computation. Meta-MEME's motif-based HMMs, however, can be straightforwardly generalized, with relatively small cost, to allow a complete set of inter-motif connections.

Figure II.4 shows an example of the generalized topology of the standard Meta-MEME model. Rather than connecting motifs in a particular order, this model contains a complete set of transitions among motifs. This topology allows for the accurate modeling of families such as the ice-nucleation proteins, which contain up to 57 consecutive copies of a single subsequence [133, 64]. The completely connected topology also allows for the modeling of families, such as the kinases, in which a set of domains appears in different orders in different family members. In the model, connections between motifs represent the less-conserved spacer regions, the contents of which are modeled by insert states, as in the linear Meta-MEME models. Although the motifs in this model are completely connected, the total number of motifs is a low constant value, and within each motif the topology is strictly linear. Furthermore, by discarding most information from the noisy, inter-motif spacers, the total number of parameters is small relative to a standard linear HMM.

In addition to improving the range of protein families that can be accurately

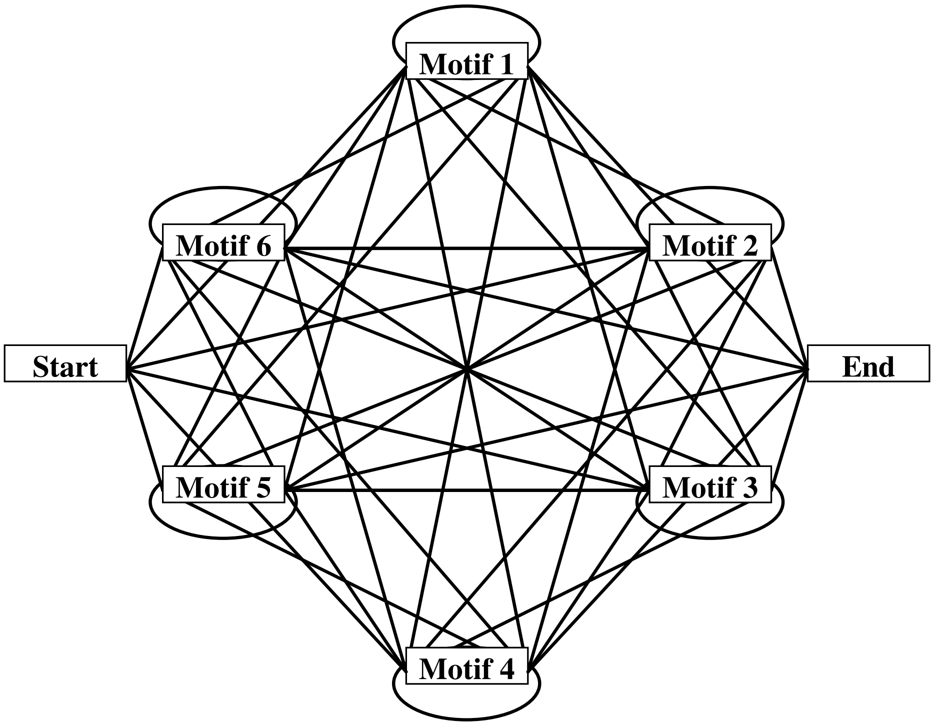


Figure II.4: **Topology of a completely connected Meta-MEME model.** Each edge in the graph contains a model of an inter-motif spacer region.

modeled, the completely connected topology brings with it an additional advantage: the majority occurrence heuristic can be discarded. Because the completely connected topology allows for an arbitrary ordering of motifs within the family, motifs which represent subfamilies can be included in the model without loss of accuracy for non-subfamily members. Furthermore, the training by expectation-maximization of inter-motif transition probabilities will minimize the chance of transitioning to very weak motifs that do not accurately represent the family. Thus, training will effectively eliminate non-representative motifs from the model.

Just as for the linear topology, information from a MAST analysis of the training set can be used to set informed initial values for the model's transition probabilities. However, rather than relying upon a single, canonical diagram, the completely connected models are initialized using all of the motif occurrence information in the training set. Information from all of the training set diagrams is combined into two matrices. The first, the average length matrix, contains in position (x, y) the average observed length of the spacers between motif x and motif y. Spacers that are not observed in the training set are assigned an arbitrary length of 10 amino acids. The second, motif-to-motif frequency matrix, contains at (x, y) the frequency with which motif x is followed by motif y. This matrix is initialized with pseudocounts of one motif-to-motif transition in each position, so as to avoid building a model containing probabilities of 0.0. The transition probabilities of the completely connected model are calculated so as to reflect the observed data in these two matrices. Thus, the transition probabilities from the last state of a motif are copied directly from the transition frequencies observed in the training set. And the transition probabilities on the self-loops of the spacer states are calculated so as to reflect the observed average spacer lengths. The details of this calculation are described in the following section.

II.D Modeling spacer regions

In Meta-MEME, the regions between motifs are not modeled very precisely, since the contents of these spacer regions are not highly conserved. In the simplest form of model, each spacer region is modeled using a single insert state. The transition probabilities into this state and on the state's self-loop are calculated such that the expected length of the emission from this state equals the length of the corresponding spacer region in the canonical motif occurrence diagram (for linear models) or the corresponding average observed spacer length (for completely connected models). In effect, then, the length of each spacer region is modeled by a single parameter.

To calculate this parameter, consider a spacer state for which the incoming transition probability is x, the outgoing transition probability is $1-x$, and the probability of a self-loop is x. Let n be the number of times the node is visited. Then the expected number of visits, μ, to such a node is, by definition,

$$\mu = \sum_{n=0}^{\infty} n(1-x)x^n \tag{II.1}$$

At first there are two possibilities: visit the node with probability x, or skip it with probability $1-x$. Skipping the node gives a spacer of length 0, while visiting it gives a spacer length 1 plus the expected remaining path length, ν. So we have

$$\mu = (1-x)0 + x(1+\nu) \tag{II.2}$$

Because of the Markov property, regardless of the path length so far, if we reach this node again then the expected path length from it is simply μ. So we have

$$\mu = x(1+\mu) \tag{II.3}$$

Solving for x yields

$$x = \mu/(1+\mu) \tag{II.4}$$

This equation is used to calculate transition probabilities for spacer states.

One important drawback to the use of a single insert state as a model of inter-motif spacer regions is that this model generates sequences whose lengths follow

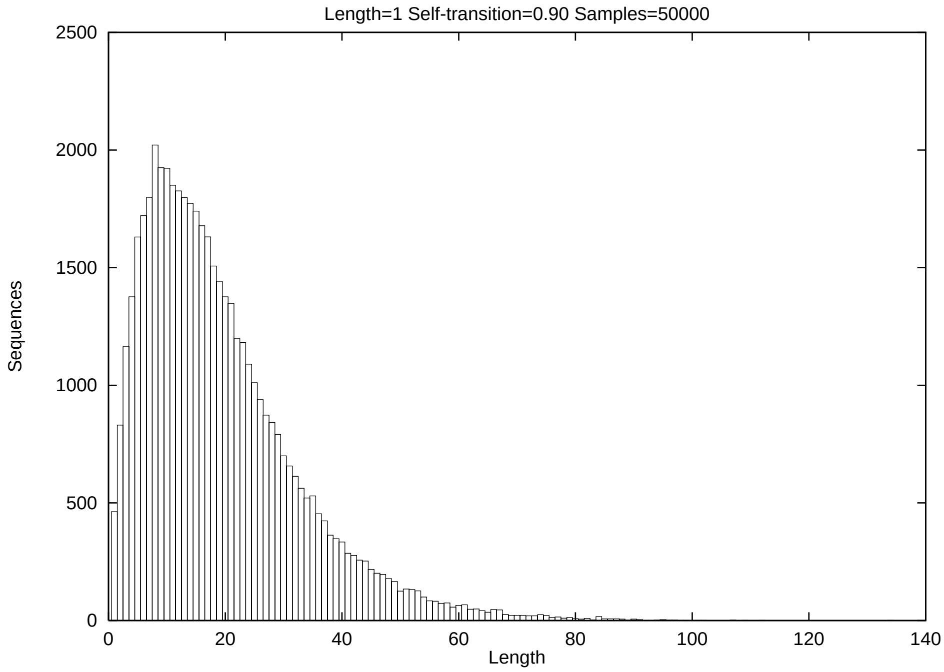


Figure II.5: **An exponential distribution generated by a single-state spacer model.** The figure shows a histogram of the lengths of sequences randomly generated by a standard HMM with length 1 and a self-transition probability of 0.9 on the insert state.

an exponential distribution. Figure II.5 is a histogram of spacer lengths randomly generated by a single insert state with a self-transition probability of 0.9. The exponential nature of the distribution is evident. Thus, the single-state model only makes sense if we have reason to believe that the lengths of spacers in actual proteins also follow such a distribution.

An informal examination of the spacer lengths within a large set of protein families leads to little evidence that the distributions tend to a particular form. Figure II.6 shows spacer length distributions for a typical family. These distributions take many forms. Some are sharply peaked. Some appear to have long tails to either the right or left. None of the observed histograms appeared to be strongly exponential.

Given that little can be said about the general form of the distributions of spacer lengths, the usual choice would be to model these distributions using a bell-shaped curve such as a Gaussian. Explicitly representing this length distribution within the HMM is straightforward [88]. However, including in the model single states that generate multiple amino acids violates the Markov property of the model and therefore results in a large increase in computational complexity. For each of the algorithms discussed previously (the Baum-Welch, Viterbi and forward algorithms), the running time for a model with explicit length distributions increases by a factor of n, where n is the maximum length of a sequence emitted by a single state in the model. This increase occurs because during the update of the cells in the dynamic programming matrix, rather than only considering cells in the previous column, each cell in the previous n columns must be considered.

In order to avoid the computational overhead of modeling length distributions explicitly, Meta-MEME employs multi-state spacer models, as shown in Figure II.7. These multi-state spacers exploit the Central Limit Theorem [132], which states that, in the limit, the sum of multiple, independent distributions with finite means and variances approaches a normal distribution. The example shown in Figure II.7 contains eight states. A distribution generated by a similar eight-state model

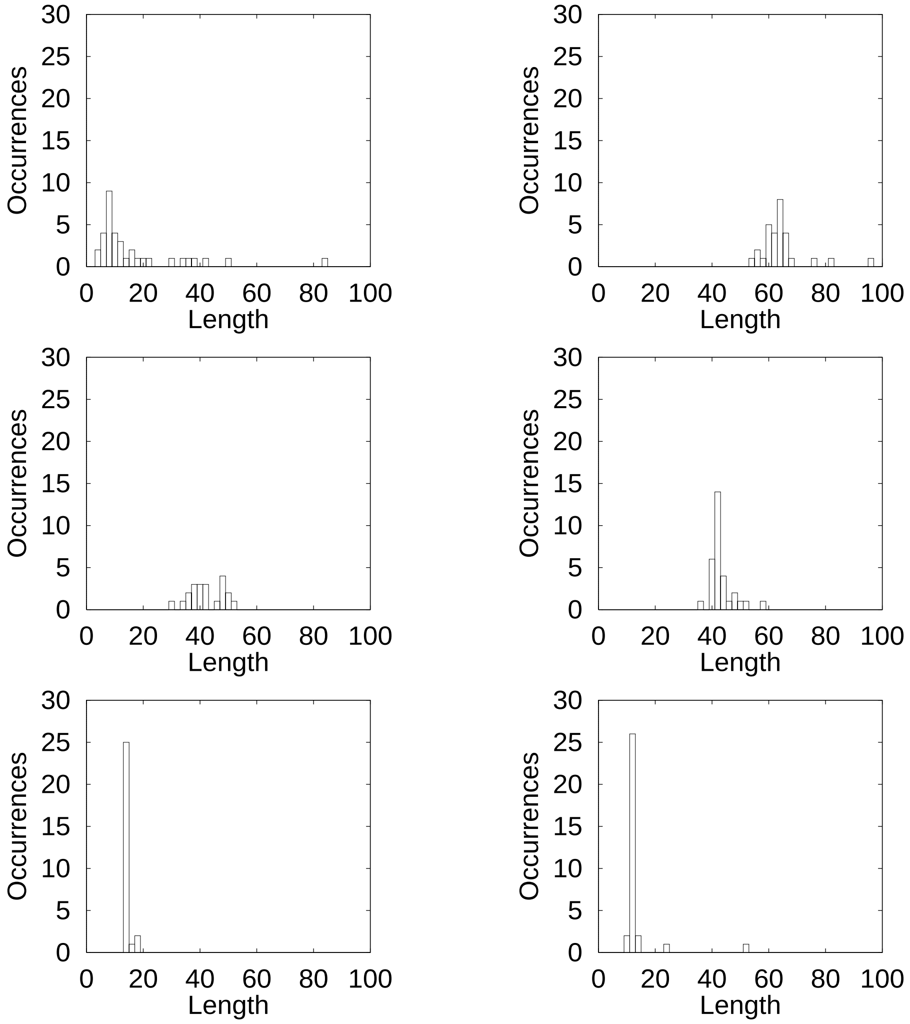


Figure II.6: **Observed spacer length distributions of a typical protein family.** A family of short chain alcohol dehydrogenases from PROSITE [12] was used. Highly similar sequences were removed, and six motifs were discovered in the resulting set of divergent sequences. The figures are histograms of sequence lengths, as determined from MAST motif occurrence diagrams of the data set, using a p-value threshold of 0.0001.

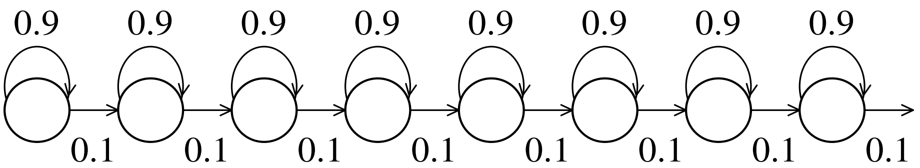


Figure II.7: **An 8-state spacer model.**

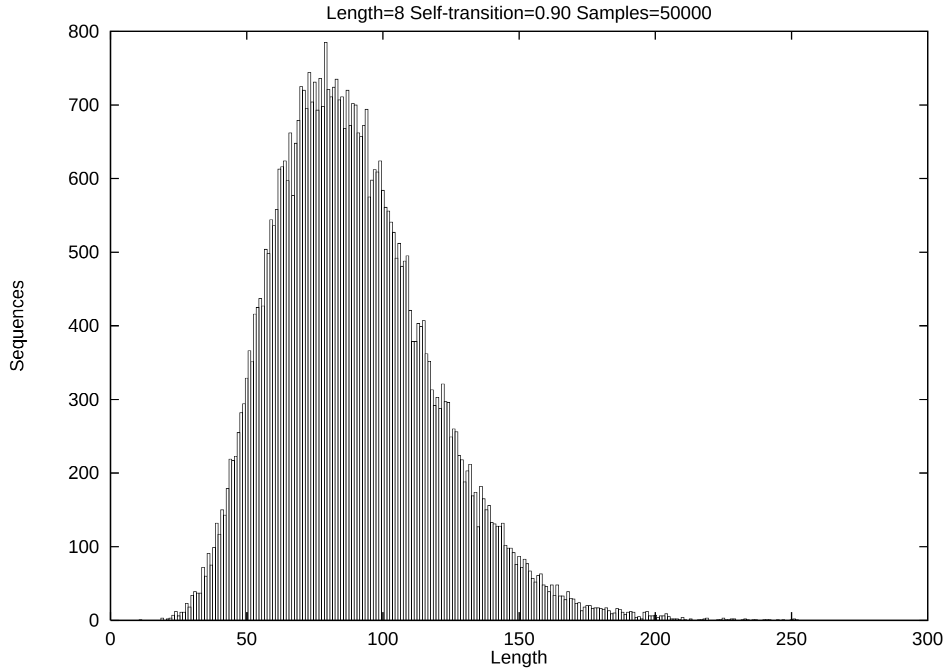


Figure II.8: **Approximation of a normal distribution generated by a multiple insert states.** This histogram of sequence lengths was generated by a standard HMM of length 8 with probabilities of 0.9 on each insert state self-transition.

is shown in Figure II.8.[1] As can be seen, although the distribution still has a longer tail to the right, it is much more bell-shaped than the distribution created by a single spacer state. The expected length of the spacer generated by an n-state spacer is just the sum of the expected lengths of n single-state spacers. Hence, equation II.4 can be straightforwardly generalized in order to compute initial self-transition probabilities.

An obvious apparent drawback to the multi-state spacer model is the increased number of parameters involved. In order to avoid overfitting the training data, Meta-MEME ties these parameters. Thus, during training, all of the self-loop transition probabilities in one spacer are treated as a single parameter and are updated in lock-step.

Because multi-state spacers approximate a normal distribution by summing

[1]The architecture of the model used to generate Figure II.8 differs from that shown in Figure II.7 only because the software for emitting samples from an HMM assumes the standard topology. This topology does not allow adjacent insert states, so the mean of the distribution in Figure II.8 is 8 greater than the predicted value because the model contains a match state juxtaposed between every pair of insert states.

exponential distributions, these spacer models are only effective in conjunction with total probability scoring. The Viterbi algorithm does not create the desired distribution because the algorithm only considers the probability of the single most likely path through the model. To see that this is so, consider the case of an n-length sequence being scored against an m-state spacer model. In order for the distribution of scores generated by this m-state model to be bell-shaped, it must be possible for the score of an n-length sequence to be less than that of a sequence of length $n+1$. We will show that, with respect to transition probabilities, this inequality can hold for total probability scoring but not for Viterbi scoring.

If the self-transition loop on each spacer state has probability x, any n-length path through the model will have a probability of $x^{n-m}(1-x)^m$. In this case, since all paths are equi-probable, the Viterbi score of this sequence is the probability of any path through the model. We can show that the Viterbi score of an n-length sequence is necessarily greater than or equal to the score of $n+1$-length sequence as follows:

$$x^{n-m}(1-x)^m >= x^{n+1-m}(1-x)^m \tag{II.5}$$

$$x^{n-m} >= x * x^{n-m} \tag{II.6}$$

$$1 >= x \tag{II.7}$$

Since x is the probability of a self-transition on a spacer state, it is necessarily less than or equal to 1. This implies that the original inequality is true, and that therefore the Viterbi score of an n-length sequence is always greater than the score of an $n+1$-length sequence. Thus, since the Viterbi scores generated by a multi-state spacer are non-increasing in the length of the sequence, the distribution of such scores cannot be a bell-shaped curve.

A similar argument shows that it is possible for total probability scores to either increase or decrease as the length of the sequence changes. The total probability score for the sequence-to-model match is the sum of the probabilities of all paths through the model. We have shown that the probability of any single path is $x^{n-m}(1-x)^m$. To count the total number of such paths, we consider choosing from the $n-1$

inter-character positions in the sequence a set of $m-1$ places to transition between states in the spacer model. This results in $\binom{n-1}{m-1}$ possible paths and a total probability score of

$$\binom{n-1}{m-1}x^{n-m}(1-x)^m$$

We can characterize the situations in which the sequence score changes from increasing to decreasing with sequence length as follows:

$$\binom{n-1}{m-1}x^{n-m}(1-x)^m = \binom{n}{m-1}x^{(n+1)-m}(1-x)^m \tag{II.8}$$

This equation can be simplified algebraically as follows:

$$\begin{aligned}
\binom{n-1}{m-1} &= \binom{(n+1)-1}{m-1}x && \text{(II.9)}\\
\frac{(n-1)!}{(m-1)!(n-m)!} &= \frac{n!}{(m-1)!(n-(m-1))!}x && \text{(II.10)}\\
1 &= \frac{n}{n-(m-1)}x && \text{(II.11)}\\
n-(m-1) &= xn && \text{(II.12)}\\
n &= \frac{m-1}{1-x} && \text{(II.13)}\\
x &= \frac{n-m+1}{n} && \text{(II.14)}
\end{aligned}$$

Equation II.13 shows that the total probability scores generated by a spacer model of length m with self-transitions x will increase with sequence length until n reaches $(m-1)/(1-x)$, after which the scores will decrease. Thus, the point $(m-1)/(1-x)$ corresponds to the peak of the bell-shaped curve. Equation II.14 suggests that, for example, modeling a spacer of average length 30 using a 3-state model requires spacer state self-transitions of $30-3+1/30 = 0.933$. Note that, for single-state spacers, Equation II.14 collapses to $x=1$. This makes sense, since a single-state spacer can only generate a single path, corresponding to the Viterbi path, and we have already shown that the Viterbi score is non-increasing with increasing sequence length.

Model topology	Trainable parameters	ADH	Fer4
standard	$25n$	5.95	5.12
linear Meta-MEME	$19l + m$	1.00	1.00
completely connected Meta-MEME	$19l + m + m^2$	1.03	1.05

Table II.1: **Comparison of model sizes for different HMM topologies.** The second and third columns contain the model sizes for two families, the short chain alcohol dehydrogenases (ADHs), and the 4Fe-4S ferredoxins (Fer4). n is the average length of sequences in the modeled family; l is the total length of the motifs in the family, and m is the total number of motifs. The values of l, m, and s are averages taken from models built in Chapter IV. For the ADHs, $l = 58$, $m = 6$ and $n = 264$; for the ferredoxins, $l = 35.2$, $m = 6$ and $n = 138.3$.

II.E Model size and computational complexity

An important advantage of Meta-MEME over standard HMMs is the small size of motif-based HMMs. Linear Meta-MEME models, in particular, have fewer parameters relative to a non-motif-based HMM. For a family of average length n, a standard HMM contains a distribution across twenty amino acids at each of n match states, as well as nine transition probabilities from each set of match-insert-delete states to the next. Emission distributions at insert states are not trained. Since the emission and transition distributions from each state must sum to 1.0, the standard HMM contains 25 trainable parameters for each match state. In contrast, if the family can be characterized by a set of m motifs of total length l, then a linear Meta-MEME model of the family contains $19l$ trainable emission probabilities within the motifs and $m + 1$ trainable transition probabilities in the spacers. Table II.1 shows model sizes for two typical families, the short chain alcohol dehydrogenases (ADHs) and the 4Fe-4S ferredoxins. For these two families, the respective standard HMMs are 6.0 and 5.1 times larger than the corresponding linear Meta-MEME models. Even when the topology is generalized to allow complete inter-motif connectivity, Meta-MEME's models contain fewer parameters than a standard HMM. Generalizing the topology introduces m^2 additional transition parameters. As shown in Table II.1, this results in a 3% and 5% increase in model size respectively for ADHs and ferredoxins. Thus, both Meta-MEME topologies offer a significant reduction in parameters, thereby allowing

Model topology	Complexity	ADHs	Ferredoxins
standard	$9n^2$	36.6	29.5
linear Meta-MEME, s=1	$(l+m+1)n$	1.0	1.0
linear Meta-MEME, s=3	$(l+3(m+1))n$	1.2	1.3
complete Meta-MEME, s=1	$(l+m+m^2+1)n$	1.6	1.9
complete Meta-MEME, s=3	$(l+3(m+m^2+1))n$	2.9	3.9

Table II.2: **Computational complexity of HMM dynamic programming algorithms using different topologies.** The final two columns contain the ratios of running times relative to linear Meta-MEME for two families, the short chain alcohol dehydrogenases (ADHs), and the 4Fe-4S ferredoxins. n is the length of the protein sequence; m is the number of motifs in the motif-based HMM; l is the total length of those motifs, and s is the number of HMM states used to model a single spacer. For the ADHs, $l = 58$, $m = 6$ and $n = 264$; for the ferredoxins, $l = 35.2$, $m = 6$ and $n = 138.3$. Each parameters is an average over five randomly selected training sets.

the models to be trained from smaller training sets.

In addition to improving the trainability of the models, reducing their size improves their efficiency. The computational complexity of each of the HMM algorithms, including the dynamic programming algorithms that underlie Baum-Welch training, is $O(tn)$, where t is the number of transitions in the model and n is the number of positions in the sequence under consideration. For standard HMMs, t is approximately $9n$, since the model has one match state for each amino acid in a typical sequence, and since each set of match-insert-delete states has nine corresponding transitions. A linear Meta-MEME model, by contrast, contains $l + s(m + 1)$ transitions, where s is the number of states representing a single spacer. Generalizing the topology to completely connect the motifs adds an additional sm^2 transitions. Table II.2 compares the running times of the HMM algorithms for standard, linear Meta-MEME and completely connected Meta-MEME models. A linear Meta-MEME model of the ADHs improves on the standard HMM's running time by a factor of 36.6. Even with complete connections among motifs, Meta-MEME still performs 23.5 times faster than a standard HMM.

```
ICYA_MANSE  gdifyp.........GYCPDVKPVnd....FDLSAFAGAWHeiaklplen
LACB_BOVIN  mkclllalal.....TCGAQALIVtqtmkgLDIQKVAGTWYslamaasdi
BBP_PIEBR   nvyhd..........GACPEVKPVdn....FDWSNYHGKWWevakypnsv
RETB_BOVIN  erdcr..........VSSFRVKEN......FDKARFAGTWYamakkdpeg
MUP2_MOUSE  mkmllllclgltlvcVHAEEASSTgrn...FNVEKINGEWHtiilasdkr
```

Figure II.9: **An example of a motif-based multiple alignment.** Motif regions appear in capital letters. Non-motif regions are unaligned. The sequences are lipocalins and are truncated after the first alignment row.

II.F Model training

Once a set of MEME motif models has been assembled into a single hidden Markov model, the model's emission and transition parameters can be trained via the Baum-Welch algorithm [112], a version of the expectation-maximization algorithm. Typically, this algorithm maximizes the likelihood of the data, given the model. However, like MEME, Meta-MEME employs Dirichlet mixture priors [32] in the training of emission probabilities, thereby maximizing the posterior probability of the model, given the data. In MEME, information about the occurrences of multiple motifs within a single sequence is unavailable. Hence, training the model parameters in the context of the HMM can be expected to improved both the motif models and the transitions among them.

II.G Multiple alignment

A hidden Markov model generates a multiple alignment of a set of sequences by first recovering the Viterbi path (i.e., the most likely sequence of states) corresponding to each given sequence and then aligning amino acids from different sequences that were generated by the same hidden state in the model. Because multiple alignments, as they are traditionally understood and used by biologists, are inherently linear, Meta-MEME only produces multiple alignments from models with linear topologies. An example of such an alignment is shown in Figure II.9.

The most striking feature of this alignment, as compared with traditional alignments, is that the Meta-MEME alignment is motif-based. The spacer regions between motifs (represented by lowercase letters) are not aligned at all. In the figure, although the contents of the spacer regions have been included, no attempt has been made to align even the most obvious similarities. There are three important points to be made about these motif-only alignments. First, the task of aligning the biologically significant motif regions is the most important part of the multiple alignment task, from a biological perspective [92]. An alignment that fails to properly align these regions is not very useful. Second, from a computational perspective, the task of aligning the spacer regions, insofar as they can be aligned, is much easier than the global task of aligning entire sequences. Meta-MEME has done the hard part, and the remaining, relatively small, unaligned regions could be passed to another sequence alignment algorithm for post-processing. Finally, there are situations, especially in phylogenetic inference, where it makes sense to throw out the spacer regions entirely, using only the motif regions.

II.H Phylogenetic inference

The multiple alignment task is often the precursor to the task of inferring the evolutionary relationships among a set of species. The usual means of building a phylogenetic tree that summarizes these relationships involves selecting a homologous protein from each species in the desired tree, producing a multiple alignment of those proteins, and then inferring the phylogenetic tree from the alignment, based upon some assumed model of evolution.

One reason to use a motif alignment, rather than a complete alignment, for phylogenetic inference is that the motif regions are far less likely to yield alignment errors. Especially for sequences that are widely divergent, a perfect multiple alignment may be impossible to create. Consequently, an alignment of the entire sequence will often contain errors that can lead to corresponding errors in the phylogenetic inference. Several authors [20, 86] have produced phylogenetic trees based upon constrained alignments from which the noisy regions have been discarded. Similarly, CLUSTALW [123] contains an option for eliminating from consideration all positions in the multiple alignment that contain insertions or deletions. Meta-MEME provides a theoretically justified means of deciding which regions of the sequences are highly conserved and hence can be trusted for producing a phylogenetic tree.

There is reason to believe, however, that motif-only multiple alignments may be useful even in situations in which the alignment is undisputed. Recent evidence [96] suggests that the best phylogenetic trees are generated by amino acids that are important in determining the three-dimensional structure of the protein. This evidence is discussed in more detail in Section IV.A.

II.I Homology detection

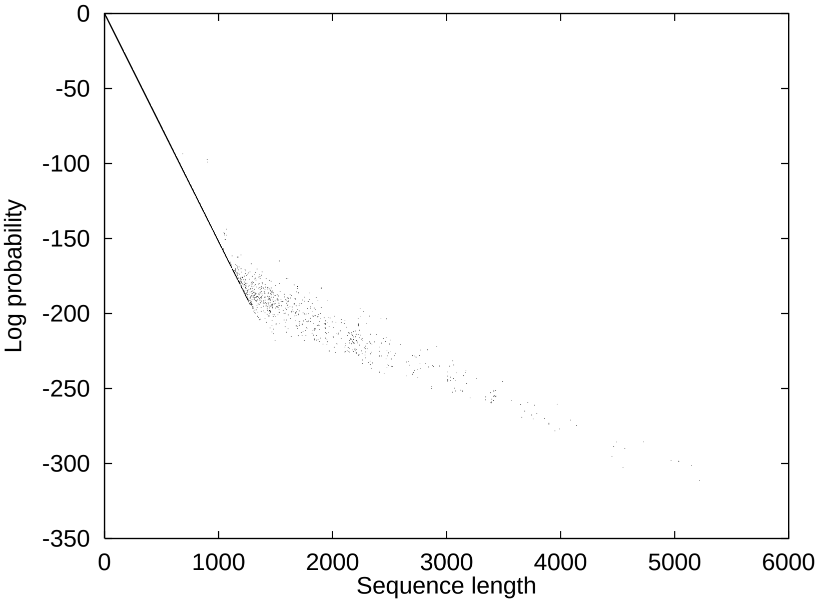
In addition to aligning known family members, a Meta-MEME model may be used to identify previously unrecognized homologs. Meta-MEME computes a score for each sequence in a protein database. If the model is accurate and the scores are computed properly, homologs of the training set will appear at the top of the the score-ranked list of proteins, possibly including sequences that were not previously known to be homologous to the modeled family. In order to be effective, this procedure requires that three distinct questions be addressed: how to score the database sequences properly; how to normalize those scores for the lengths of the sequences; and how to compute a threshold for statistically significant homologies.

There are two choices for the type of scores computed during homology detection: the Viterbi score and the total probability score returned by the forward algorithm. Both of these scores can be computed by exactly similar dynamic programming algorithms that differ only in the operation performed when a cell of the dynamic programming matrix is updated. The Viterbi algorithm computes a maximum; the forward algorithm computes a sum. Because of this difference, however, the Viterbi score can be calculated more quickly than the total probability score.

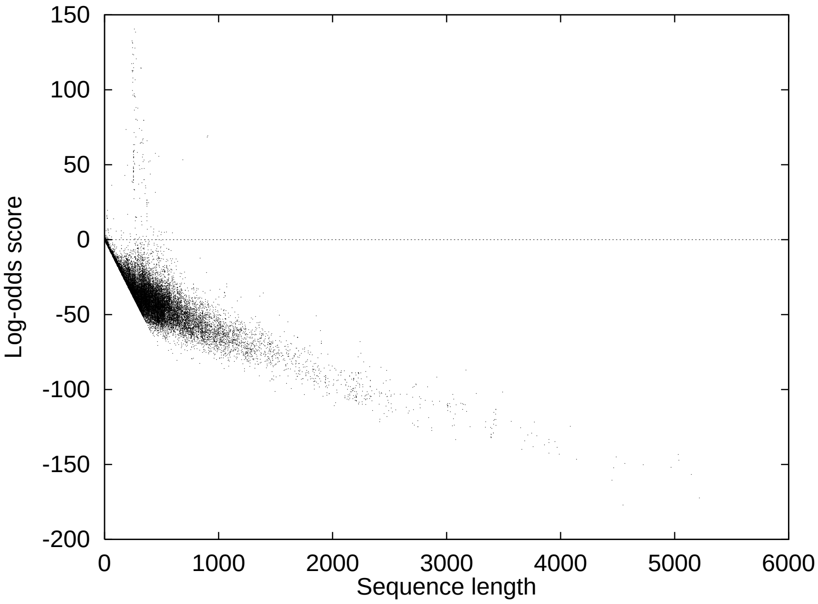
The final score for each method involves a long series of multiplications; therefore, these algorithms must be either dynamically scaled or carried out in log space in order to avoid precision problems. The *max* operator used in the Viterbi algorithm translates cleanly into log space. In contrast, the sum operation cannot be carried out in log space without a conversion involving an exponentiation and a logarithm. Because of this additional overhead, the total probability score generally takes longer to compute. However, in a Bayesian framework, the total probability score makes more sense, especially since the Baum-Welch training algorithm maximizes this total probability, rather than the Viterbi score.[2] An additional motivation for using total probability scoring in Meta-MEME is that the combination of multiple exponential distributions used by the multi-state spacer models does not occur when a single Viterbi path is used for scoring.

An important drawback to total probability scores is that they are strongly dependent upon the length of the scored sequence. This dependence is not precisely linear and cannot be easily computed a priori. Krogh *et al.* [83] have developed an empirical normalization procedure that fits a piecewise linear curve to the scores generated by a large database, excluding outliers. A more straightforward method, however, is to rely upon the implicit length normalization that occurs when the scores are compared to background scores generated by a simple background model. The usual background model consists of a linear hidden Markov model with length equal to the given sequence and no insert or delete states. The emission probability distribution at each match state in the model is set equal to an empirical frequency distribution from a large protein database. If the effect of the transition probability distributions upon the score is relatively small, then by multiplying together a series of background emission probabilities of the same length as in the foreground model, the background score depends upon the length of the sequence in the same way that the foreground model does. Thus, since both the foreground and background models are length dependent in a similar way, the ratio of the two, known as the *odds*

[2]There does exist a Viterbi approximation of the Baum-Welch algorithm [21, 94], but it is not implemented in Meta-MEME.



(a)



(b)

Figure II.10: **Length dependence of HMM total probability scores.** Figure (a) plots the log total probability score of a motif-based linear HMM as a function of sequence length for all sequences in SWISS-PROT version 28. In Figure (b), the scores have been converted to log-odds. The background model is a linear HMM of the same length as the given sequence. All transitions have probability 1.0 and emission probabilities are taken from the non-redundant protein database.

score, contains very little length dependence. In practice, the raw odds score has a large dynamic range, so Meta-MEME reports the logarithm in base 2 of the odds. This *log-odds score* is used in most of the experiments reported in later chapters. Figure II.10(a) shows the length dependence of log total probability scores for a standard HMM; Figure II.10(b) shows the same scores as log-odds.

An additional benefit of odds scoring is that the resulting scores have a well defined theoretical threshold. Using total probability scoring, the odds score S of a sequence s relative to foreground model λ and background model γ is

$$S = \frac{Pr(s|\lambda)}{Pr(s|\gamma)} \tag{II.15}$$

Ultimately, however, we are interested in the probability that the sequence belongs to the family ($Pr(\lambda|s)$), rather than the likelihood of the family given the sequence; i.e., we want to know whether

$$\frac{Pr(\lambda|s)}{Pr(\neg\lambda|s)} > 1.0 \tag{II.16}$$

A ratio of 1.0 implies equal probabilities that the sequence may or may not belong in the family. A larger value indicates family membership; a smaller value indicates non-family membership. If the size of the family is small relative to the size of the database being searched, then the background model γ of the entire database is approximately the same as a model of non-family members. Thus, the inequality II.16 becomes

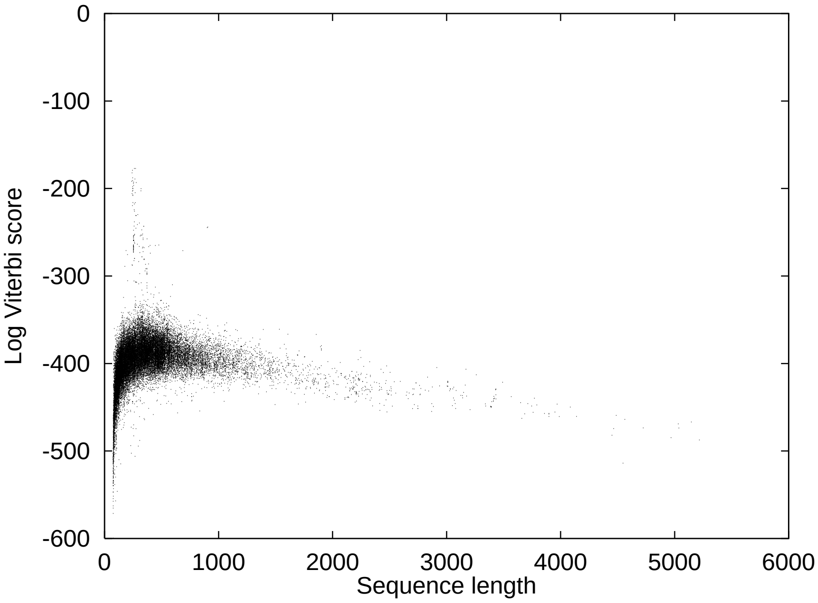
$$\frac{Pr(\lambda|s)}{Pr(\gamma|s)} > 1.0 \tag{II.17}$$

The left-hand side of this inequality can be converted using Bayes' Rule as follows:

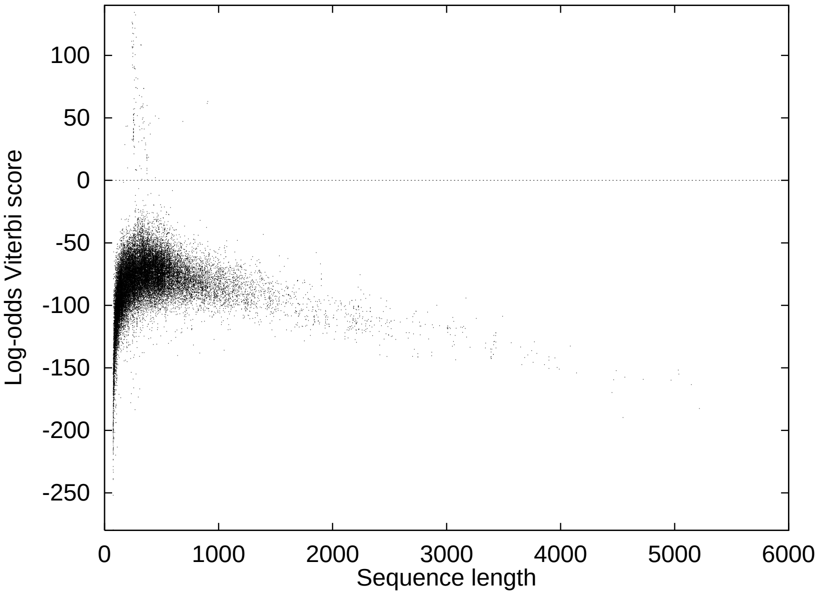
$$\begin{aligned} \frac{Pr(\lambda|s)}{Pr(\gamma|s)} &= \frac{Pr(s|\lambda)Pr(\lambda)}{Pr(s)} \frac{Pr(s)}{Pr(s|\gamma)Pr(\gamma)} && \text{(II.18)} \\ &= \frac{Pr(s|\lambda)}{Pr(s|\gamma)} \frac{Pr(\lambda)}{Pr(\gamma)} && \text{(II.19)} \end{aligned}$$

The first term is the odds score S, so inequality II.17 becomes

$$\begin{aligned} S\frac{Pr(\lambda)}{Pr(\gamma)} &> 1.0 && \text{(II.20)} \\ S &> \frac{Pr(\gamma)}{Pr(\lambda)} && \text{(II.21)} \end{aligned}$$



(a)



(b)

Figure II.11: **Scaling of Viterbi scores via log-odds.** Figure (a) plots the log Viterbi score of a motif-based linear HMM as a function of sequence length for all sequences in SWISS-PROT version 28. In Figure (b), the scores have been converted to log-odds. The background model is a linear HMM of the same length as the given sequence. All transitions have probability 1.0 and emission probabilities are taken from the non-redundant protein database.

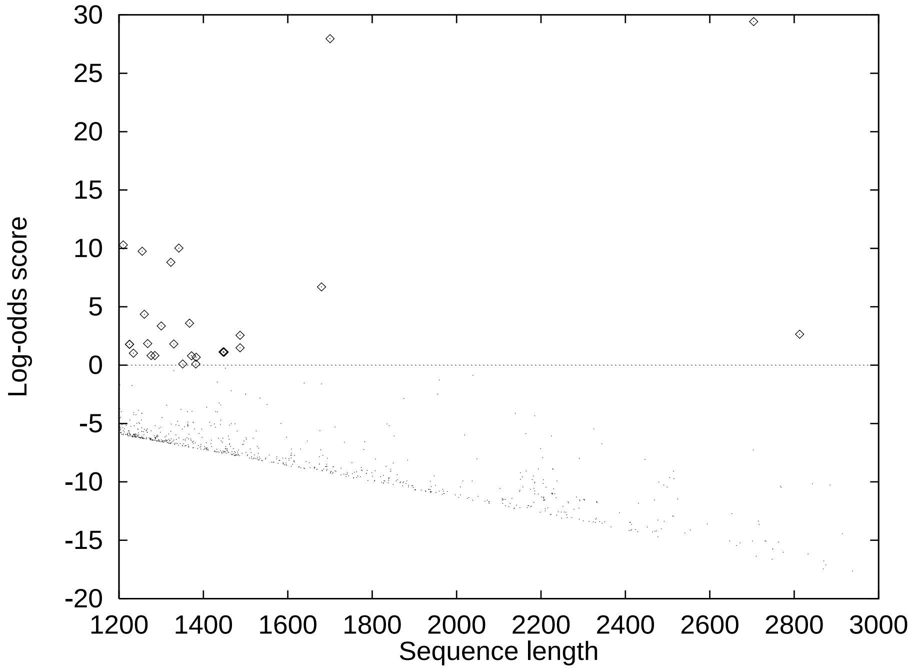


Figure II.12: **Long false positive matches to an HMM with a completely connected topology.** The figure plots the total probability log-odds score as a function of sequence length for a completely connected motif-based HMM of the 4Fe-4S ferredoxin family. All sequences in SWISS-PROT version 28 with lengths from 1200 to 3000 are included. The longest 4Fe-4S ferredoxin in this database has length 1171, so all data shown here are for non-family members. Sequences scoring above 0 are marked with larger points. Note in particular the four rightmost false positives. Details about these sequences are given in Table II.3.

This inequality reflects that, when searching a large database, false matches to the model are likely to appear by chance. If we approximate the prior probabilities of family membership and non-family membership using frequency data from the database, then the theoretical threshold for the odds score S just is the ratio of these frequencies. The scaling accomplished by computing odds is evident in Figure II.10(b). Figure II.11 shows that the same scaling occurs for Viterbi scores, even though the Viterbi score lacks the strong length dependence of the total probability score.

Sequence	Length	Total	Viterbi
G168_PARPR	2704	29.43	-56.76
BAR3_CHITE	1700	27.96	-44.36
FUR2_DROME	1680	6.70	-45.44
VWF_HUMAN	2813	2.64	-58.79

Table II.3: **Low Viterbi log-odds scores of long false positive matches.** The four longest sequences that receive positive total probability log-odds scores in Figure II.12 each receive a very low Viterbi log-odds score.

II.J Explicit length modeling

In practice, when searching a database for homologs using a motif-based HMM with a completely connected topology, many of the highest-scoring false positive matches to the model tend to be very long sequences. This problem arises because, as mentioned above, the implicit length normalization carried out by the odds scoring procedure assumes that the sequence score depends primarily upon the emission probability distributions in the given model. For a standard HMM, most sequences that match the model reasonably well do so by traversing primarily match states, with relatively few insertions and deletions. Thus, since the transitions between match states generally have high probabilities, the standard HMM score is determined primarily by the emission probabilities at the match states. For a motif-based HMM, on the other hand, the spacer states and their accompanying self-transitions account for much of the sequence score. Hence, the implicit length normalization is less effective for these models.

One means of improving the length normalization is to mimic the topology of the foreground model in the background model [26]. Thus, instead of consisting of a single chain of match states, the background model topology is identical to the foreground model. In the background model, every emission probability distribution is set equal to the global background distribution.

Although complex background models work well in practice, they are not easily interpretable. In a Bayesian framework, the background model describes all members of the domain; i.e., all known proteins. Giving the background model the same topology as the foreground obfuscates the background model's function.

Therefore, rather than relying upon a complex background model, we carry out length normalization by explicitly modeling the lengths of the sequences in the given family. Hence, we treat a single sequence as though it consists of two distinct features: the sequence s, and the sequence length ℓ. Inequality II.17 then becomes

$$\frac{Pr(\lambda|s,\ell)}{Pr(\gamma|s,\ell)} > 1.0 \tag{II.22}$$

Applying Bayes' Rule, as above, yields

$$\begin{aligned} \frac{Pr(\lambda|s,\ell)}{Pr(\gamma|s,\ell)} &= \frac{Pr(s,\ell|\lambda)Pr(\lambda)}{Pr(s,\ell)} \frac{Pr(s,\ell)}{Pr(s,\ell|\gamma)Pr(\gamma)} & \text{(II.23)} \\ &= \frac{Pr(s,\ell|\lambda)}{Pr(s,\ell|\gamma)} \frac{Pr(\lambda)}{Pr(\gamma)} & \text{(II.24)} \end{aligned}$$

If we assume that the length score and sequence score are independent, then

$$\frac{Pr(s,\ell|\lambda)}{Pr(s,\ell|\gamma)} = \frac{Pr(s|\lambda)Pr(\ell|\lambda)}{Pr(s|\gamma)Pr(\ell|\gamma)} \tag{II.25}$$

Substituting from equation II.25 into equation II.24 and the result into inequality II.22 yields

$$\frac{Pr(s|\lambda)Pr(\ell|\lambda)}{Pr(s|\gamma)Pr(\ell|\gamma)} \frac{Pr(\lambda)}{Pr(\gamma)} > 1.0 \tag{II.26}$$

Finally, to derive a score similar to the odds score described previously, let

$$S' = \frac{Pr(s|\lambda)Pr(\ell|\lambda)}{Pr(s|\gamma)Pr(\ell|\gamma)} \tag{II.27}$$

Then inequality II.26 becomes

$$S' > \frac{Pr(\gamma)}{Pr(\lambda)} \tag{II.28}$$

This odds score S' is analogous to the score S in inequality II.21 and has the same threshold for statistical significance.

In order to calculate the odds score S', we need to compute four terms: $Pr(s|\lambda)$, $Pr(s|\gamma)$, $Pr(\ell|\lambda)$ and $Pr(\ell|\gamma)$. We have already described how to compute the first two terms using the HMM and the background model for s. To compute

$Pr(\ell|\lambda)$, the likelihood of a sequence of length ℓ, we need a model of sequence lengths. The most straightforward such model is a normal distribution based upon the observed sequence lengths in the training set. Thus, the observed lengths are used to compute empirical estimates of the mean and variance of a normal distribution. Then, during homology detection, this distribution is used to evaluate $Pr(\ell|\lambda)$.

One complicating factor in such a model is that, when the training set is very small, the estimated variance of the distribution of sequence lengths can be very poor. As usual, when the training data is insufficient, we exploit prior information to provide a more reasonable model. In this case, the prior information takes the form of pseudocounts. If we assume that the observed training set mean is accurate, then we can compute pseudocount values such that the variance of the pseudocounts is equal to a prior σ^2 on the variance. A set of n pseudocounts with mean μ, each with value a, has variance

$$\sigma^2 \simeq \frac{\sum_n (a-\mu)^2}{n} \tag{II.29}$$

$$\sigma^2 \simeq (a-\mu)^2 \tag{II.30}$$

$$a \simeq \sigma - \mu \tag{II.31}$$

We calculate the variance prior by computing the average standard deviation of sequence lengths for all 1150 families in PROSITE [12]. The resulting value is $\sigma = 114.6$. Thus, the values of the pseudocounts are computed using equation II.31 with $\sigma = 114.6$ and μ equal to the empirical mean of the training set. Choosing the number n of pseudocounts corresponds to choosing the weight of the prior. Meta-MEME is conservative in using $n = 2$. These 2 pseudocounts are added to the observed set of sequence lengths prior to computing the variance of the length model.

The remaining term in the computation of S' is the likelihood of the length ℓ according to a background model of sequence lengths. This model is empirically derived from a large protein database, the NCBI non-redundant database [53]. A histogram of sequence length frequencies is computed across all observed sequence lengths up to a maximum of 5300 (see Figure II.13). Each bin in the histogram

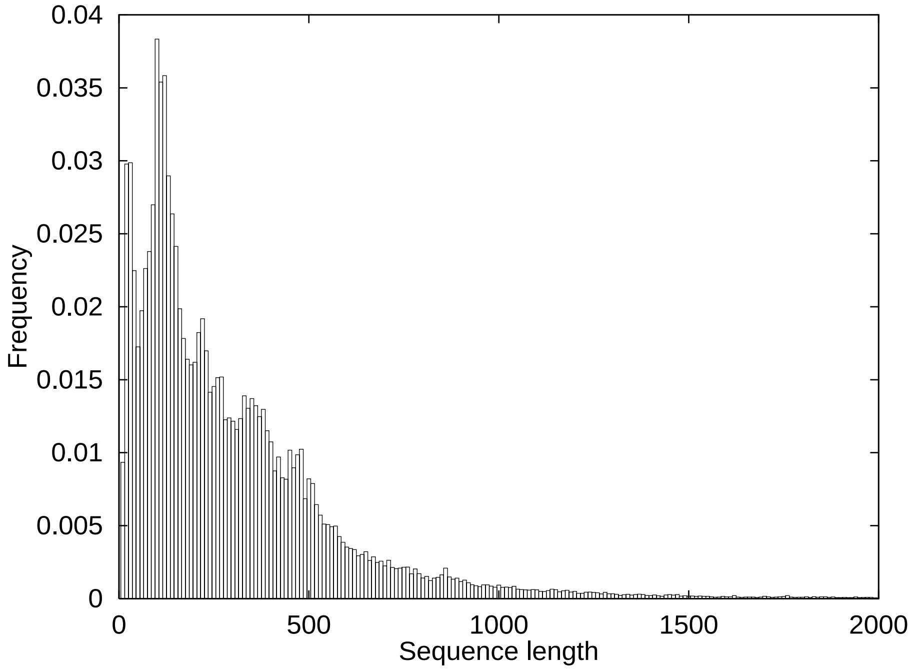


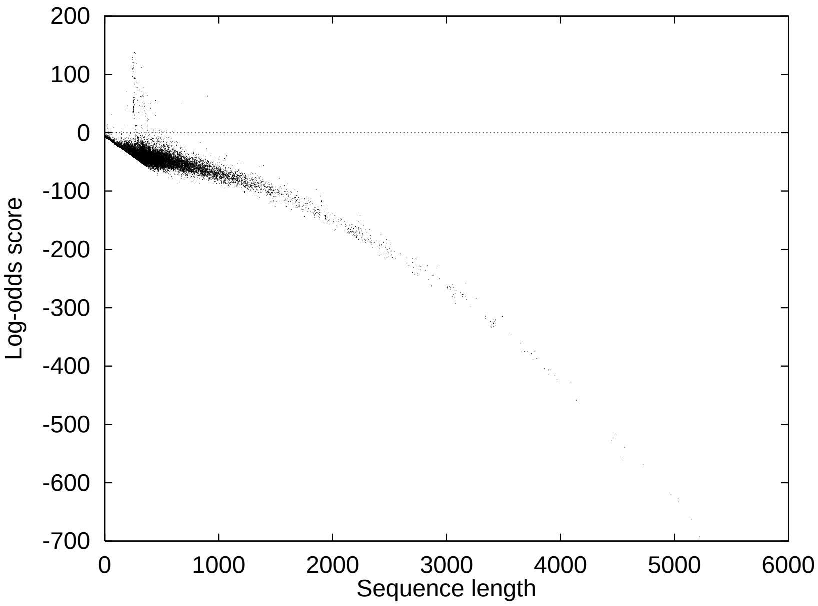
Figure II.13: **Empirical distribution of sequence lengths in the non-redundant protein database.**

contains sequences of length ± 5 and is initialized with a single pseudocount to avoid zero probabilities. This histogram is used directly to evaluate $Pr(\ell|\gamma)$. Any sequence of length greater than 5300 receives a length score of $1/N$, where $N = 292\ 459$ is the size of the database.

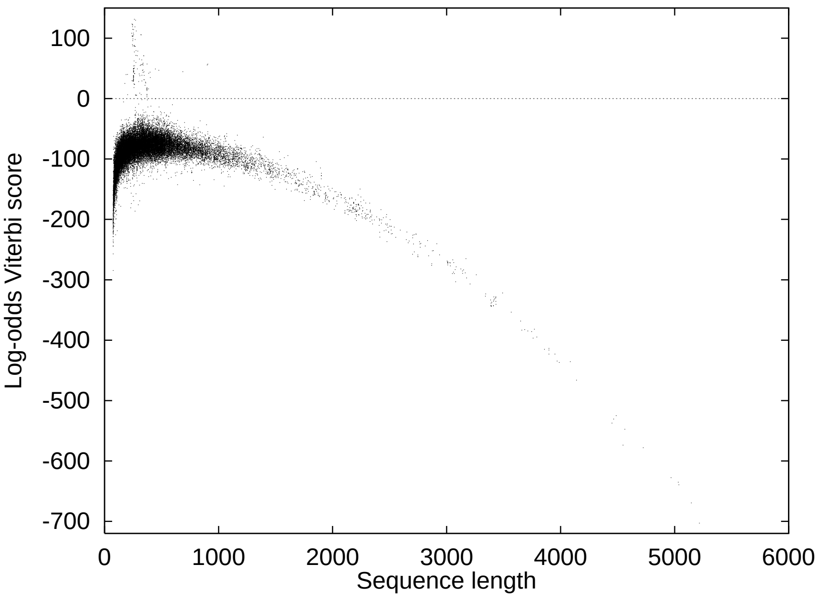
Figure II.14 shows the same data as Figure II.10(a) and II.11(b), but including the explicit length model in the odds calculation. In Figure II.15, the false positives evident in Figure II.12 are eliminated by the length model.

II.K Discussion

In many ways, Meta-MEME resembles the BLOCKS method for protein family classification [71, 67]. The BLOCKMAKER program discovers highly conserved regions of protein families by combining motifs found by either the MOTIF algorithm [117] or the Gibbs sampling algorithm [87]. Individual blocks may be represented as ungapped position-specific scoring matrices, similar to the motif models



(a)



(b)

Figure II.14: **Combining normally distributed length scores with total probability and Viterbi scores.**

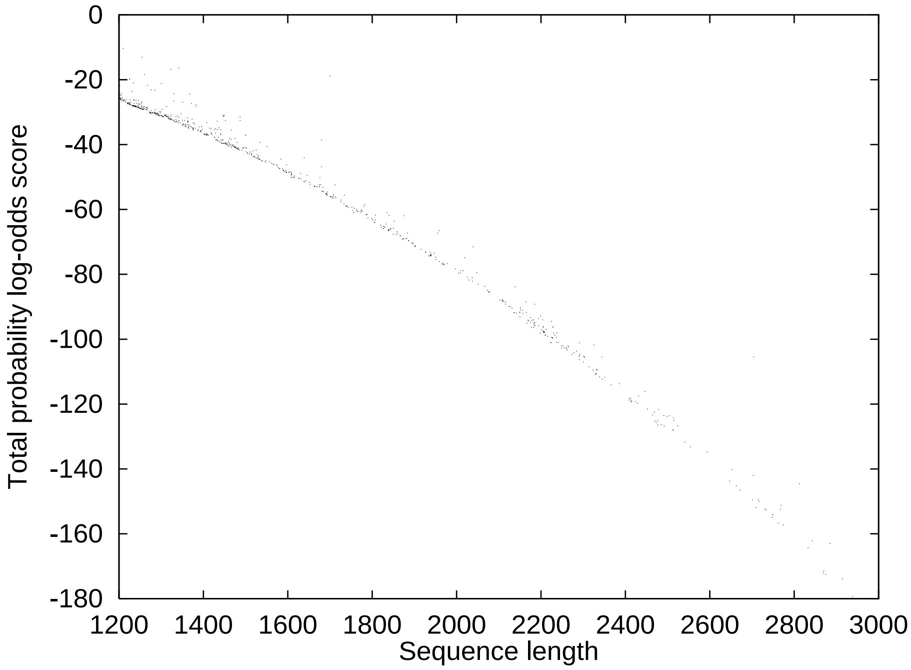


Figure II.15: **Elimination of long false positive matches via explicit length modeling.**

created by MEME. However, MEME is more likely than BLOCKMAKER to split a motif in two if any of the sequences contain an insertion or deletion, so MEME motifs tend to be shorter than BLOCKMAKER blocks. Since motifs (and blocks) are supposed to model ungapped regions, MEME generally produces more accurate models. The BLOCKS database [28] contains, for each known protein family, an ordered set of blocks along with the minimum and maximum observed spacings between the blocks in the training set. The BLIMPS program [73] searches this database using a single sequence as a query, thus taking into account the order and spacing of blocks. Clearly, Meta-MEME's linear models and the BLOCKS method share many features. In general, however, a hidden Markov model approach is more attractive because of its well-founded underlying probabilistic theory.

The motif-based hidden Markov models produced by Meta-MEME offer several important advantages over other multiple alignment and homology detection methods. First, Meta-MEME models focus on the biologically significant regions of the given protein family. By exploiting the prior knowledge that protein families tend to consist of conserved motif regions separated by noisy spacer regions, Meta-MEME produces alignments and detects homologs in a biologically motivated fashion. Furthermore, multiple alignments produced by Meta-MEME do a better job of aligning the important, motif regions. The models are also effective at discovering remote homologies. By modeling the spacer regions between motifs in a very simple way, Meta-MEME selectively discards information from the training set about the contents of spacer regions. This discarding of information is beneficial for distantly related sequences, because distant homologs typically show conservation only in functionally or structurally important portions of their sequences.

Another important advantage of Meta-MEME models is their small size. The HMMs produced by Meta-MEME are typically less than one-fifth as large as corresponding models of complete proteins. This reduction in size allows Meta-MEME models to be trained from relatively few known family members. This ability is important, since many families contain only a handful of known sequences. In addition to improving the trainability of the models, the reduced parameter set greatly increases the efficiency with which the models can be trained and used, sometimes producing up to a factor of 30 speedup relative to standard HMMs.

Finally, a motif-based HMM differs from a standard HMM in the assumptions it makes about molecular evolution. Point mutations are allowed within motif regions, but insertions and deletions are not. This constraint reflects the strong biological constraints imposed on the motif regions. Inter-motif spacer regions, on the other hand, are modeled less precisely. Contents of these regions are relatively unconstrained, and the lengths of the spacer regions are modeled only loosely. Furthermore, within the HMM the motifs may be completely connected, so that entire motifs may be repeated, skipped, or shuffled. Motif-based HMMs therefore allow for evolutionary mechanisms such as large-scale deletions and copying events. Thus, this generalized topology implies a domain-based model of molecular evolution [119].

Chapter III

Family Pairwise Search

Science may be described as the process of building models to explain natural phenomena. Although every scientific theory implies a corresponding model, some models are less explicit than others. An explicit model with an exact interpretation is desirable, since it effectively summarizes the important features of the target phenomenon, rendering them easily explicable. In the case of protein family characterization, a statistical model with a probabilistic interpretation, in addition to being useful for tasks such as multiple alignment and homology detection, can provide biological insight into the important functional or structural features of the modeled family.

Unfortunately, the most elegant model is not always the most useful. In this chapter, we introduce a simple, non-model-based algorithm, called Family Pairwise Search (FPS) which, for small training sets, outperforms several complex and theoretically justified protein modeling techniques on the homology detection task. The FPS algorithm involves computing, for each sequence in the database being searched, its average pairwise similarity score with the sequences in the family of known homologs comprising the query. These similarity scores may be computed using a sequence search algorithm such as BLAST [2]. For small query sets, the FPS algorithm outperforms a full-sequence hidden Markov Model approach (HMMER [46]) and a motif-based modeling approach (model construction by MEME [6] followed by search with MAST [11]) to homology detection.

The explanations for the relatively poor performances of these model-based techniques differ. For HMMER, the difficulty lies in the large number of model parameters relative to the size of the training set. When only a few sequences are available for training, the number of parameters in the model is on the order of the total size of the training set. Consequently, even with strong prior information, training these models accurately is difficult. MEME, on the other hand, reduces the number of trainable parameters by focusing only upon the motif regions of the training set. The result is a set of relatively well-trained motif models. However, MEME loses homology information by discarding the non-motif regions of the sequences [107], and this loss affects MAST's search performance.

Cobbling [72] is a hybrid modeling scheme that addresses both of these problems. A cobbled profile model of a protein family is constructed by converting a single, representative family member (the template sequence) into a profile [57] and then replacing the motif regions with profile representations of the motifs. All gap opening and extension penalties in the profile are set to the same values. The number of trainable parameters in the cobbled profile model is small, because models are only learned for the motif regions. The rest of the profile is constructed by simply replacing the letter in the template sequence with a column from a pairwise score matrix such as BLOSUM [69]. Thus, the cobbled model retains useful homology information in the inter-motif regions by embedding the motif models into the profile constructed from the inter-motif regions of the template sequence.

In this chapter we extend the FPS algorithm to searching with profiles. We show that when a small set of protein family members is available, searching with the BLAST algorithm using a single cobbled profile fails to detect homologs as well as the BLAST FPS algorithm using sequences. However, even better homology detection performance is attainable by applying the FPS algorithm to cobbled profiles. In our experiments, this cobbled profile FPS technique also provides significantly better homology detection performance than the purely model-based methods HMMER and

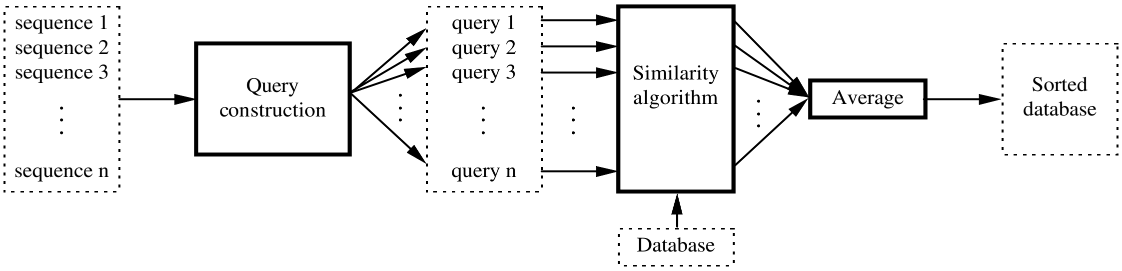


Figure III.1: **Schematic diagram of the Family Pairwise Search algorithm.**

MEME/MAST. The FPS algorithm does not require BLAST as the similarity step, and accordingly, we show that accuracy can be improved further by replacing BLAST with the Smith-Waterman algorithm.

III.A The Family Pairwise Search algorithm

The Family Pairwise Search algorithm is illustrated in Figures III.1 and III.2. The input to the algorithm is a query set of sequences that are known to be homologous to one another, as well as a sequence database to be searched. FPS outputs a version of the database sorted in order of decreasing homology with the query set. The algorithm proceeds in four steps. First, the sequences in the query set are each converted into a separate query. Second, the queries are input to a similarity algorithm and compared to each sequence in the search database. Third, each sequence in the search database is assigned a similarity score equal to its average similarity with the sequences in the query set. Fourth, the search database is sorted according to the average similarity score.

In this chapter, we study the variations of the FPS algorithm outlined in Table III.1. In the simplest form of the algorithm, the sequences in the query set are used directly to search the database. The current work additionally investigates two other query formats: cobbled profiles and sequences with their non-motif regions

```
procedure FPS (sequence_set, database, threshold)
  query_set ← Query_construction(sequence_set)
  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) ← Similarity_algorithm(query, target)
      if (evalue < threshold) then
        sum_of_scores = sum_of_scores + score
      end
    end
    scores[i] = sum_of_scores / size(query_set)
  end
  return scores
```

Figure III.2: **The Family Pairwise Search algorithm.**

Method	*Query format*	*Search*	*Averaging*
BLAST FPS	sequence	BLAST	Yes
motif-only BLAST FPS	motif regions	BLAST	Yes
cobbled profile BLAST FPS	cobbled profile	BLAST	Yes
PFS FPS	profile	PFS	Yes
cobbled profile PFS FPS	cobbled profile	PFS	Yes
BLAST	single sequence	BLAST	No
cobbled profile BLAST	single cobbled profile	BLAST	No
MAST	motif models	MAST	No
HMMER	sequence model	hmmsw	No

Table III.1: **Summary of homology detection methods investigated here.** Five query format types are considered: the original sequences, the motif regions of the sequences, motif models built from the sequences, a standard HMM of the sequences, or the cobbled profiles of the sequences. PFS refers to the Profilesearch [57] algorithm as implemented on the Bioccelerator [40].

removed. For comparison, we also study searching with a representative sequence selected from the query set, searching with a cobbled profile constructed using the representative sequences as a template, searching using one or more motif models, and searching using a standard HMM. These last four search methods do not involve the averaging step of the FPS algorithm.

To convert a sequence in the query set to a cobbled profile, we use a modified version of the Cobbler [72] algorithm to embed motif profiles into a profile constructed from the template sequence. Our modified version of the algorithm can output both log-odds profiles and frequency profiles. Log-odds profiles, for use with the Smith-Waterman algorithm, are built by replacing each letter in the sequence with the BLOSUM row for that letter; frequency profiles, for use with the BLAST algorithm, use the target letter frequencies corresponding to the BLOSUM row, rather than the log-odds scores. To convert a profile into a cobbled profile, motif models (built as described below) are converted either into log-odds position-specific score matrices (for log-odds profiles) or target frequency matrices (for frequency profiles) and are used to replace the appropriate positions in the profile as in the original Cobbler algorithm. Log-odds profiles are built using BLOSUM55, whereas frequency profiles use BLOSUM62 in order to be comparable with the standard BLAST algorithm. All gap opening and extension penalties in the log-odds profiles were set to 100.

Motifs need only be discovered once for each query set. Ungapped motifs are discovered and modeled using MEME version 2.2 [6] with the default parameter settings from the web interface [60]. These defaults include empirical Dirichlet mixture priors weighted according to the megaprior heuristic [9], 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 are discovered from each query set, and motif significance is judged using the majority occurrence heuristic [62]: motifs that do not appear in more than half of the query sequences are discarded. This heuristic excludes motifs that are specific to subfamilies of the given query set. For eight-sequence queries, the heuristic selects an average of 5.1 motifs. MEME outputs the motifs in BLOCKS [68] format for use as input to the modified Cobbler algorithm.

To test the effect of focusing on the motifs, we construct a set of queries with the inter-motif regions removed. The motif regions of a sequence are identified by aligning the sequence with a motif-based linear hidden Markov model using Meta-MEME. The intervening spacer regions are deleted, and the resulting concatenated sequence of motifs is treated as the query for the similarity step of the FPS algorithm.

To evaluate the benefit of the averaging aspect of the FPS algorithm, we compare FPS to the use of a single, representative sequence from the query set. We choose this representative sequence using the same method as the unmodified Cobbler algorithm. Essentially, the sequence which best matches the motifs for the family is chosen. This sequence is used to search the database, and the averaging step in the FPS algorithm is skipped. For comparison, we also include a test of the original Cobbler method. This involves using the same, representative sequence as the template for a cobbled profile. The database is searched using just this profile, and the averaging step is skipped.

Finally, we compare the FPS approach with two strictly model-based approaches. The MEME motifs discovered previously are provided as a single query to the MAST [11] homology search algorithm. For each sequence in the database, MAST compute a p-value for each motif in the query and combines these values assuming that motif occurrences are statistically independent. The resulting sequence-level p-value scores are used to rank the sequences in the database.

In addition, hidden Markov models of each query set are built using the HMMER software package version 1.8 [46]. Models are trained using expectation-maximization coupled with simulated annealing. The default geometric annealing schedule is used, and Dirichlet mixture priors allow the models to be trained with smaller training sets. Database searches are carried out using a modified form of the Smith-Waterman algorithm. This algorithm performs a local search for sequence-to-model matches, allowing partial matches to either the sequence or the model. For each database sequence, a log-odds score in bits is computed.

In the second step of the FPS algorithm, the queries are input to a similarity algorithm and compared to each sequence in the search database. Any algorithm suitable for comparing the given type of query with protein sequences may be employed in this second step of FPS. The current work investigates using the BLAST and Smith-Waterman algorithms for computing query-to-sequence similarities. For BLAST searches, we use the "bit score" [3] as the similarity score. For Smith-Waterman searches, we use the negative logarithm of the p-value of the Smith-Waterman score.[1]

We use gapped BLAST version 2.0 [2, 3], a heuristic approximation of a dynamic programming optimization of maximal segment pair scores. In order to use cobbled profiles as BLAST queries, we obtained a pre-release version of PSI-BLAST [3] that is capable of storing and reading binary checkpoint files. Since these files contain a frequency matrix representation of the query, converting our cobbled frequency profiles to the BLAST checkpoint format is straightforward. PSI-BLAST is run for one iteration with its default parameters. The filtering of low-complexity regions in the query sequence is turned off because this option is unavailable in conjunction with reading checkpoint files. For BLAST searches using sequences as the queries, we use the BLOSUM62 score matrix.

For Smith-Waterman searches, we use the Profilesearch (PFS) algorithm as implemented on the Bioccelerator [40]. We set the gap opening penalty to 8 and the extension penalty 0.3. In order to calculate p-values corresponding to Smith-Waterman scores, we calculated the score distribution by fitting the Karlin-Altschul [80] distribution to 10 000 random sequences of length 250 using linear regression. The estimated values of λ and K can then be used to calculate the p-value of any score.

In the third step of the FPS algorithm, the similarity scores for a given database sequence with each of the queries are averaged together to give the score

[1]We define the p-value of the Smith-Waterman score of a sequence as the probability that the score of a random sequence of the same length as the given sequence would be at least as high as the observed score for the sequence.

for comparing the sequence with the family. For convenience, we only include in this average the sequences most similar to the query. We assign all other sequences the lowest possible similarity score. For both similarity algorithms tested here, the lowest possible score is zero. When BLAST is used as the similarity algorithm, we compute all similarity scores that correspond to an E-value[2] smaller than 1000. When PFS is used, we compute scores for the 1000 highest-scoring sequences. Because all protein families in the database we search have far fewer than 1000 members, this approach should yield the same results as actually computing all similarity scores.

III.B Comparing homology detection methods

We use a collection of 73 protein families [10, 59] in our homology detection experiments (see Appendix A). These families were selected from the PROSITE database [12] for their difficulty, based upon the number of false positives reported in the PROSITE annotations. The families range in size from 5 to 109 sequences, and from 949 to 58 015 amino acids. The associated release of SWISS-PROT [13] contains 36 000 sequences and nearly 12.5 million amino acids.

Bias within the families is minimized via sequence weighting. Since many weighting schemes perform almost as well as one another [70], all the experiments reported here employ a simple, binary weighting scheme based upon BLAST similarity scores [87]. 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 is used. The sizes of the weighted PROSITE families range 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 and 16 sequences are randomly selected from the set of weighted sequences. This results in 73 query sets

[2]An E-value is just the p-value multiplied by the size of the database being searched.

of size 2, 57 sets of size 4, 35 of size 8, and 16 query sets of size 16.

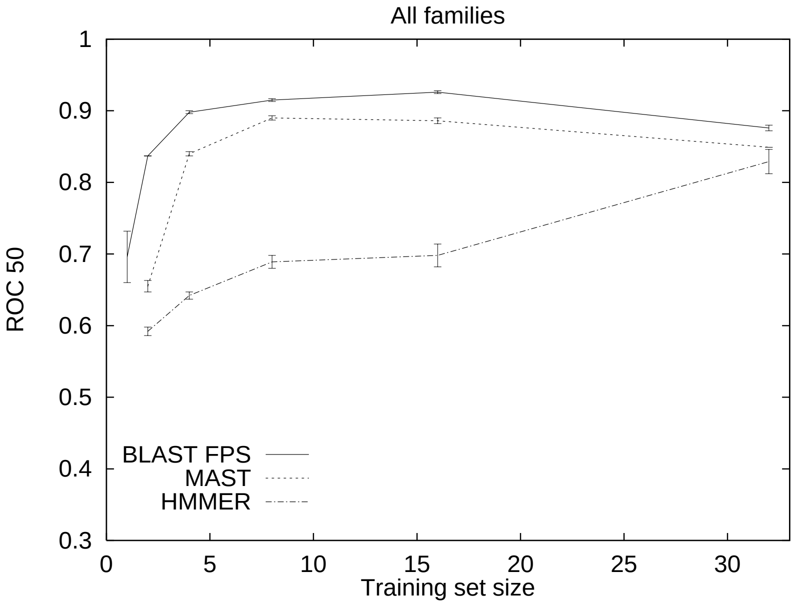
A modified version of the Receiver Operating Characteristic, called ROC_{50} [58], is used to compare the various search techniques. The ROC score is the area under a curve that plots true positives versus false positives for varying score thresholds. ROC analysis combines measures of a search's sensitivity 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. The lists of positive family members used in calculating ROC_{50} scores are taken from PROSITE. For MAST and PFS, sequences are ranked by p-value. For BLAST, sequences are ranked by average bit scores, and for HMMER, sequences are ranked by average log-odds scores.

III.C Results

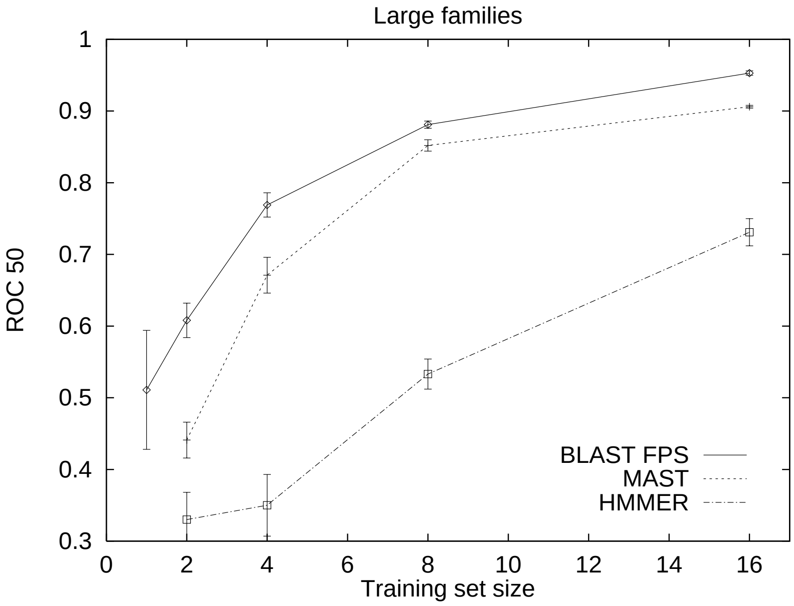
Our experiments show that, while the FPS algorithm that uses as queries the raw sequences is an effective homology detection method, it can be significantly improved by embedding motif models in the query sequences. We illustrate and explain this method through a series of experiments. In what follows, in comparing two search methods, a method is "significantly better" if the pairwise differences in ROC for the 73 families is significant at the 1% confidence level, based upon a paired t test.

III.C.1 Comparing BLAST FPS with model-based techniques

Figure III.3(a) shows the average ROC_{50} scores for all 73 protein families in the study. In the figure, for query sets of size 2, 4, 8 and 16, all differences between



(a)



(b)

Figure III.3: **BLAST FPS performs better than model-based techniques.** The figure shows average ROC_{50} scores as a function of query set size. 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.

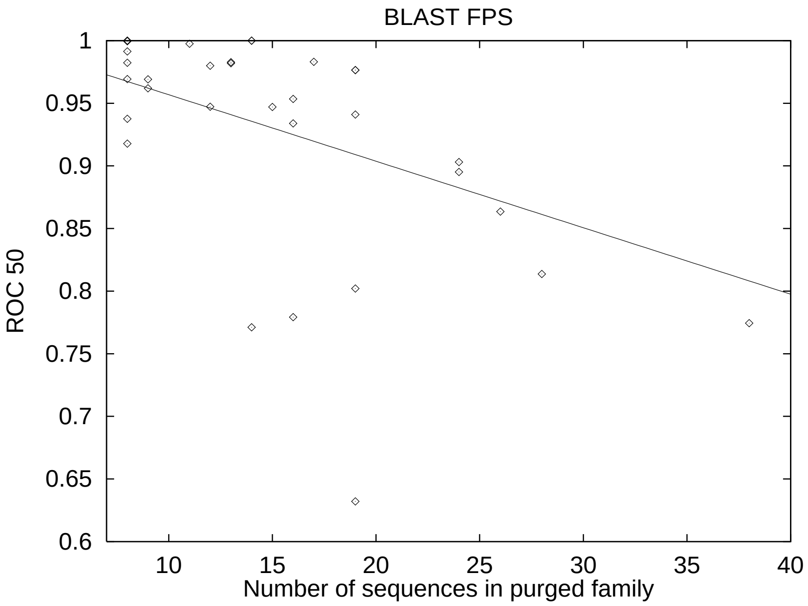
search techniques at a given query set size are significant. Thus, BLAST FPS uniformly outperforms the other two methods, and MAST outperforms HMMER. Only three families contain more than 32 weighted members, so the differences between techniques at that query set size are not significant.

One unexpected characteristic of Figure III.3(a) is the downward trend of the BLAST FPS and MAST 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 III.4, 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.

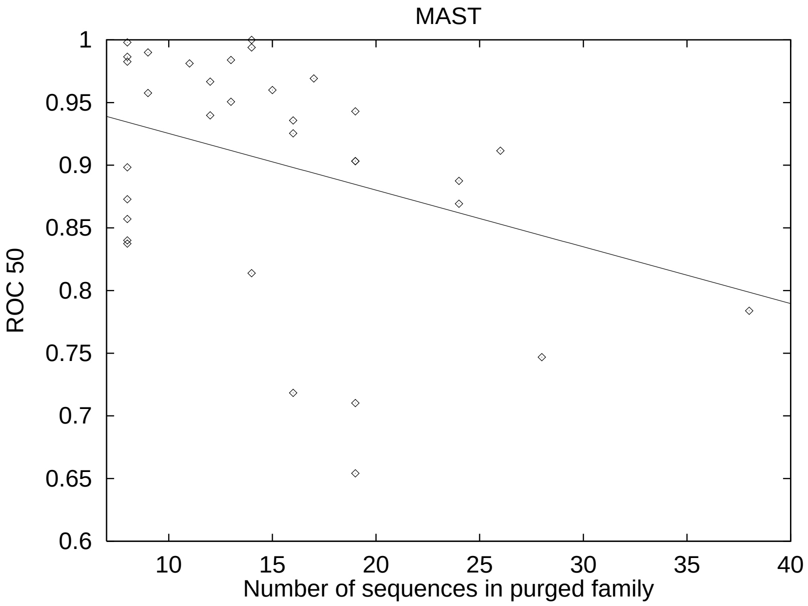
Figure III.3(b) corrects for differences in family size by including only families containing between 16 and 31 members. Here, the trend toward better performance with more query sequences is clearer. All three methods improve significantly at each increase in the query set size except for HMMER between queries of size 2 and 4.

Some protein families are difficult to recognize regardless of the homology detection method employed. Table III.2 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 FPS and MAST 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 FPS uncovered



(a)



(b)

Figure III.4: **The effect of family size upon recognition difficulty.** The figures show average ROC_{50} scores 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.

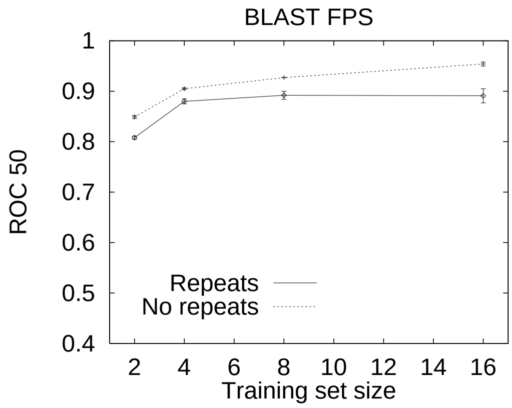
Family	BLAST FPS		HMMER		MAST		Total
	ROC_{50}	R	ROC_{50}	R	ROC_{50}	R	rank
Cytochrome c	0.562	1	0.043	2	0.548	1	4
ABC transporters	0.774	4	0.292	4	0.784	6	14
N-6 Adenine-specific DNA methylases	0.814	7	0.257	3	0.747	5	15
Aminoacyl-transfer RNA synthetases class-II	0.632	2	0.455	11	0.654	2	15
Binding-protein-dependent 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 III.2: **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 are ranked by increasing ROC_{50} score. The rank of each family with respect to each method is given in the columns labeled "R." The families are listed in order of increasing total rank.

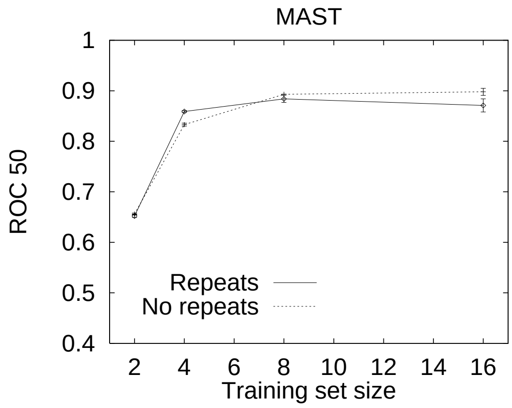
for the cytochrome c family contain six sequences for which the annotation includes the words `CYTOCHROME C`. Three of these false positive sequences are cytochrome C precursors; three more are listed as cytochrome c family members in a later version of SWISS-PROT (one as a potential member).

An important difference between BLAST FPS and MAST on the one hand and HMMER on the other is that the former two algorithms allow multiple local matches. The `hmmsw` program, although it performs a local search, allows only a single subsequence of the HMM to 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 III.5 illustrates this effect. PROSITE annotations were used to separate out those families containing repeated elements. For all query set sizes, HMMER performs significantly worse on families containing repeated domains. For MAST, 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 FPS performs better on families without repeated domains.

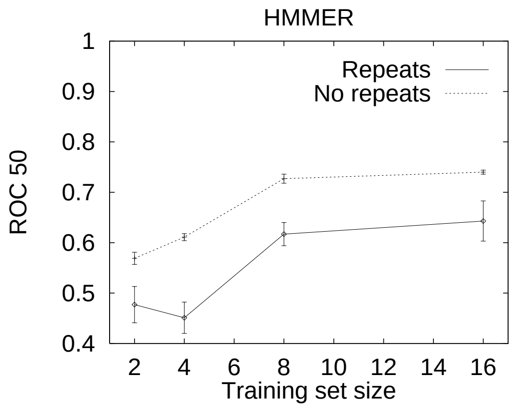
One important reason for using homology detection to infer protein function is speed. Most wet lab experiments are slow relative to a protein database search. However, not all computational methods are equally fast. In this respect, BLAST FPS clearly outperforms both MEME/MAST and HMMER. For example, Table III.3 shows typical timing data for one protein family. For an 8-member query set, BLAST FPS 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



(a)



(b)



(c)

Figure III.5: **Detecting homologs of families containing repeated elements.** The figures show average ROC_{50} scores 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.

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 III.3: **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.

$O(n^2)$ 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 SWISS-PROT takes nearly 2.5 hours on a fast workstation.

III.C.2 Adding motif models to the FPS algorithm

The experiments summarized in Figure III.6 show that using a single representative sequence from a protein family as the query in a BLAST search is the poorest homology detection method among those we examine here. Homology detection using the motif models as the query to the MAST algorithm is far better. Better still is the FPS algorithm using sequence queries and the BLAST similarity algorithm (BLAST FPS). Using only the motif regions of the sequences as queries to BLAST in the FPS algorithm (motif-only BLAST FPS) gives intermediate homology detection accuracy.

These results indicate that BLAST is taking advantage of homology information in the non-motif, spacer regions of the given sequences. When the inter-motif regions are eliminated from the queries, BLAST FPS's performance degrades significantly. On the other hand, MAST's significantly improved performance relative to motif-only BLAST FPS shows that MEME is producing useful models of the motif regions.

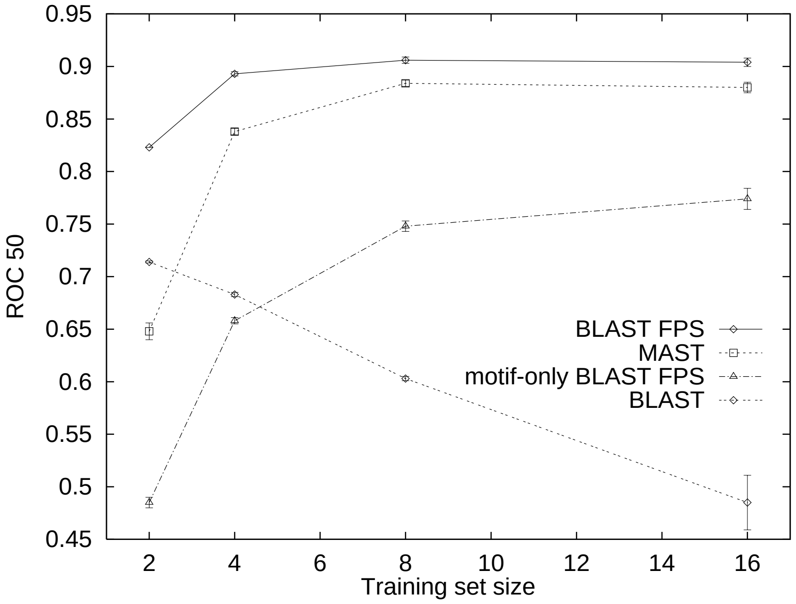


Figure III.6: **Superior search accuracy of Family Pairwise Search.** The figure also illustrates the value of using motif models (with MAST), as well as the importance of not discarding the inter-motif regions. The figure plots ROC_{50} score as a function of query size. Included are 73 query sets of size 2, 57 sets of size 4, 35 of size 8, 16 of size 16 and 3 sets of size 32. Error bars represent standard error.

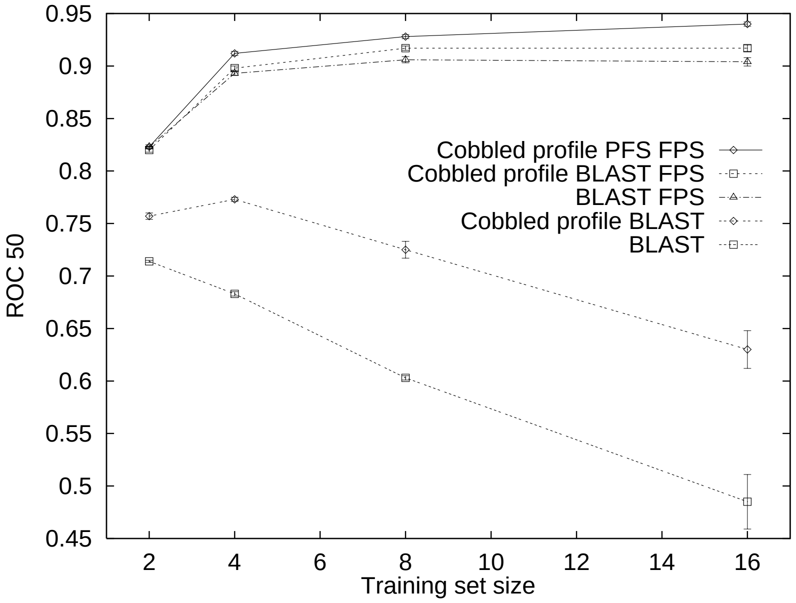


Figure III.7: **Benefits of combining cobbling with Family Pairwise Search.** The figure plots average ROC_{50} score as a function of query size. Error bars represent standard error.

Given that the model is helpful and that throwing out non-motif regions is detrimental, the apparent solution should be to build models of entire sequences, rather than only of the motif regions. However, we have already seen that, for the relatively small training set sizes under investigation here, a model of the entire sequence fails to provide accurate homology detection for these 73 families. This failure most likely results from the large number of parameters in a sequence-level model relative to the amount of noise in the non-motif regions of the sequences. For the less conserved regions of the training sequences, aligning positions may be difficult or impossible. Consequently, building models of these regions is also difficult.

One solution to this dilemma is to build models in which the motif regions are based upon the entire set of given sequences, and the non-motif regions are based only upon a single sequence in the family. This is the cobbled profile approach. Figure III.7 confirms that this approach improves homology detection accuracy. Cobbled profile BLAST has significantly greater accuracy than single sequence BLAST. The figure also shows, however, that using only a single cobbled profile built from a representative sequence, as in the original Cobbler paper, is tremendously inferior to the BLAST FPS algorithm. Combining the cobbled profile approach with the FPS algorithm, however, yields better search accuracy. Cobbled profile BLAST FPS is significantly better than BLAST FPS. Substituting the Smith-Waterman (PFS) algorithm for BLAST gives yet better performance. Cobbled profile Smith-Waterman FPS is significantly better than even cobbled profile BLAST FPS. This is not surprising, since the BLAST algorithm is a heuristic approximation of the Smith-Waterman algorithm. Cobbled profile Smith-Waterman FPS is also significantly better than searching with a single profile derived from a representative sequence (data not shown).

III.D Discussion

Our results show the benefits of building models of protein families. The first set of experiments indicate that, for small query sets, a non-model-based FPS algorithm (BLAST FPS) out-performs both sequence-level and motif-based models on the homology detection task. The second set of experiments reported here, however, show that statistical models, used appropriately, can be helpful even for very small query sets. The Cobbler approach [72] is an effective means of reducing the size of the models being trained while retaining homology information in noisy regions of the query sequence. The FPS algorithm takes Cobbler one step further by retaining the noisy regions of all query sequences, rather than a single representative. The result is an intelligent compromise, an algorithm that models only the regions of the sequence that are effectively model-able while retaining all of the information from the noisier regions. The version of this algorithm based upon the Smith-Waterman algorithm, rather than BLAST, performs best overall, probably due to the exhaustive search for local alignments performed by the Smith-Waterman algorithm.

The reasons for the FPS algorithm's excellent performance are two-fold. First, the average similarity 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 similarity score is therefore similar to the intermediate sequence approach suggested by Pearson [107] and Park *et al.* [102]. Second, FPS may perform well relative to motif-based methods because the similarity algorithms allow 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 [107].

The improved performance of both cobbled profile BLAST and BLAST FPS relative to BLAST can be explained in terms of the use of homology information in the query sequences. When multiple query sequences are available, searching for homologs using BLAST with a single representative sequence obviously discards important homology information from the rest of the query set. Cobbled profiles remedy this problem somewhat, since they include in the motif regions information from all the query sequences. BLAST FPS furthers the improvement by including all of the information from all of the query sequences.

This kind of explanation, however, fails to account for cobbled profile BLAST FPS's strong performance relative to BLAST FPS. Since the BLAST FPS algorithm already considers all of the information in the query sequences, the improvement that cobbled profiles add to the algorithm must derive from the motif models themselves, rather than because cobbled profile BLAST FPS considers more information in the query set.

Statistical models of the type built by MEME offer two important advantages over direct pairwise sequence similarity algorithms. First, a position-specific scoring matrix entails the assumption that amino acid occurrences at one position in a protein are statistically independent of amino acid occurrences at other positions. This site independence assumption allows a candidate protein to receive a high score even if that protein does not closely resemble a single query sequence but instead is comprised of a mix of sites similar to several proteins in the query set. Second, a statistical model can incorporate prior knowledge that effectively augments the information provided in the query set. For this purpose, MEME employs a set of empirically derived Dirichlet mixture priors [32]. These priors allow MEME to guess from very little evidence a biologically plausible amino acid distribution for each position in the motif model. Thus, cobbled profile BLAST FPS's improved homology detection performance relative to BLAST FPS illustrates the positive effect of the site independence assumption and of the use of prior information in detecting homologs.

An important consideration for any homology detection method is speed. Since the FPS algorithm involves comparing each of the n sequences in the protein family to the sequences in the search database, the algorithm requires approximately n times as long as searching with a single representative sequence. In practice, however, if binary weighting of the family is employed (*i.e.*, highly similar sequences are discarded), n is fairly small. In this study, for example, the average weighted family size is 10.7. In conjunction with a heuristic algorithm such as BLAST, or with special-purpose hardware such as the Bioccelerator, FPS is therefore quite efficient, especially considering the relatively large improvement in performance that the algorithm offers over single-sequence BLAST. Adding motif modeling to FPS incurs considerable overhead, since the MEME motif discovery algorithm is roughly $O(n^2)$ in the size of the training set. However, as we have shown, the model-based method offers the best homology detection performance. For large families, searching once using MAST is faster than searching with multiple cobbled profiles using either BLAST or PFS, and MAST is almost as accurate.

For fairness of comparison, the experiments reported here employ 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 [82] and implemented in Probe [98] and PSI-BLAST [3]. PSI-BLAST employs a non-motif-based, position-specific scoring matrix representation; Probe, on the other hand, employs motif models. The current results indicate that a hybrid Cobbler representation may be superior to either a sequence-level or motif-only model. We therefore intend to investigate the use of the cobbled profile BLAST FPS algorithm within an iterative framework.

The text of Chapter III, in part, is a reprint of the material as it appears in *Proceedings of the Second International Conference on Computational Molecular Biology* [59].

The dissertation author was the sole author listed on this publication.

Chapter IV

Experimental results

IV.A Motif-based multiple alignment

IV.A.1 Introduction

For the biologist, one of the most familiar means of illustrating the evolutionary relationships among a set of sequences is the multiple alignment. By aligning corresponding positions among a set of homologous sequences, a multiple alignment quickly illustrates the extent to which a set of sequences has been conserved over evolutionary time. Furthermore, regions of high conservation, as opposed to poorly conserved regions containing many insertions and deletions, indicate to the biologist regions of the sequences that may have particular biological significance.

In the first set of experiments reported here, we compare motif alignments created by Meta-MEME with multiple alignments generated by ten other multiple alignment methods. The quality of an alignment is judged, following McClure *et al.* [92, 91], according to its success in properly aligning a set of known, biologically significant regions. We find that, overall, Meta-MEME performs as well or better than all but one of the competing alignment methods. With respect to alignments derived from a standard linear HMM, Meta-MEME performs considerably better.

Meta-MEME alignments differ from standard alignments in that Meta-

MEME only aligns the motif regions of the given set of sequences. These motif-only alignments could be easily augmented using traditional sequence alignment methods. However, motif-only alignments are useful as inputs to phylogenetic inference algorithms. For particularly divergent data sets, a motif-only alignment is likely to contain fewer errors than an alignment of entire sequences. Furthermore, recent evidence suggests that motif-only alignments may be phylogenetically useful even when the alignment contains few errors.

In the second set of experiments reported here, we use Meta-MEME to analyze a set of mitochondrial proteins from fourteen chordate and five non-chordate species. This data set was used by Naylor and Brown [96] in a comparison of phylogenetic inference algorithms. Because these are species for which there exists an abundant fossil record, their phylogenetic relationships are undisputed. Hence, there is a known, true tree for this data set. Furthermore, because the mitochondrial sequences are so similar to one another, the alignment is relatively easy to create, so it too is undisputed. Finally, since the data set is large, containing thirteen complete proteins for each species, the phylogenetic inference algorithms should be able accurately to infer the true tree. However, Naylor and Brown report that none of the phylogenetic methods that they used, including equally-weighted parsimony, maximum likelihood and distance methods, succeeded in recovering the true tree, and most of the methods disagreed about what the incorrect tree is. Furthermore, their analysis of the positions in the multiple alignment that are consistent with the true tree with respect to a parsimony analysis suggest that a molecular phylogeny based upon “amino acids that seemed to be critical for determining the proteins’ three-dimensional structure” [23] provides the best match to morphological data. This hypothesis leads to our second experiment, in which we show that, for the data set used by Naylor and Brown, Meta-MEME, in conjunction with a parsimony-based phylogenetic inference package, is the only known method capable of finding the true tree for this data set.

IV.A.2 Methods

For the validation of Meta-MEME multiple alignments, four families of known homologs are aligned [92, 91]. For each family, twelve representative sequences are selected. The first set includes α- and β-globins from mammals and birds, myoglobins from mammals, and hemoglobins from insects, plants and bacteria. These globin sequences share between 10 and 70% pairwise sequence identity. The sequence sets for the other three families are more divergent, with a maximum pairwise sequence identity of 30%. These families are the eukaryotic kinases, including serine/threonine, tyrosine and dual specificity kinases from mammals, birds, fungi, retroviruses and herpes viruses; the eukaryotic aspartic acid proteases, including pepsins, chymosins and renins; and the RH domain of the RNA-directed DNA polymerase.

Following McClure *et al.*, biologically significant regions of each family are defined *a priori*. In most cases, these regions correspond to motif regions, although some of the biologically significant regions are too short properly to be called motifs (e.g., several kinase regions consist of a single amino acid). Nevertheless, for simplicity, we refer to these pre-defined regions as motifs. For the latter three families, the motif regions are defined based upon independent biological evidence. For the globins, there is no external measure of authenticity, but reasonable motif regions are selected *a priori* in order to provide a uniform testing paradigm.

The quality of an alignment is measured as the percentage of the predetermined motif regions that are successfully aligned in the multiple alignment. A successful alignment of a particular motif sequence is one that exactly matches that sequence with the corresponding motif in at least one other family member. The quality of a motif alignment improves if all motifs in the family are aligned to one another, rather than being aligned in two or more disjoint blocks. In addition, gapless motifs with no insertions or deletions are preferred over gapped motifs. In the reporting of results, quality is reported as the percentage of correctly aligned motif regions, with annotations indicating gapped motifs and motifs with misaligned subsets.

For each family, eight motifs are discovered using MEME version 2.0. Motifs

Species name	Common name	GenBank ID
Paracentrotus lividus	sea urchin	J04815
Strongylocentrotus purpuratus	sea urchin	X12631
Branchiostoma floridae	lancelet	
Xenopus laevis	frog	M10217
Gallus gallus	chicken	X52392
Didelphis virginiana	opossum	Z29573
Mus musculus	mouse	J01420
Bos taurus	cow	J01394
Cyprinus carpio	carp	X61010
Oncorhynchus mykiss	trout	L29771
Petromyzon marinus	lamprey	U11880
Balaenopterus physalus	fin-back whale	X61145
Balaenopterus musculus	blue whale	X72204
Rattus norvegicus	rat	X14848
Drosophila yakuba	fruit fly	X03240
Cepaea nemoralis	snail	U23045
Anopheles gambiae	mosquito	L20934
Ascaris suum	nematode	X54253
Caenorhabditis elegans	nematode	X54252

Table IV.1: **Species included in the mitochondrial data set.** The last five species serve as a collective outgroup that is used to root the phylogenetic tree.

are selected according to the majority occurrence heuristic (see Section II.B) for inclusion in a linear Meta-MEME model. The model is written in HMMER format, and the `hmma` program is used to align the training set sequences with the model.

HMMER models of each family are built using the HMMER program `hmmt`. This program implements a simulated annealing training algorithm. The program is run with its default parameters, including an exponential annealing schedule and add-one pseudocount priors. Again, `hmma` is used to create multiple alignments.

The data set for phylogenetic analysis consists of thirteen concatenated mitochondrial protein sequences from each of fourteen chordate species as well as five non-chordate species that serve as a collective outgroup. The names of the species are given in Table IV.1. The total size of the data set is 71 001 amino acids.

Fifty motifs are discovered four divergent sequences (snail, nematode, opossum and lancelet) using MEME 2.0. These sequences were selected using the `purge` program with a BLAST threshold of 1500 bits [98]. Of the fifty motif models, one is discarded because it appears in less than half of the nineteen sequences in the original data set. The remaining motif models are combined into a Meta-MEME linear hidden Markov model with single-state spacer models. The HMM is then used to align the mitochondrial sequences, throwing out non-motif regions. The resulting alignment is given as input to the protein parsimony program in Phylip [50]. The reported tree is the single most parsimonious tree found in ten random shuffles of the sequences.

IV.A.3 Results

Multiple alignments

Figures IV.1 and IV.2 show multiple alignments generated by HMMER and by Meta-MEME for the globin family. This is the least divergent of the four data sets; therefore, both multiple alignments succeed in accurately aligning all of the motif regions of the sequences.

The set of RH domains of the RNA-directed DNA polymerase is much more divergent than the globin sequences. The results indicate that, of the four families under consideration, this one is the hardest to align. For this data set, Meta-MEME performs much better than HMMER. Figures IV.3 and IV.4 show that Meta-MEME succeeds in aligning all four of the pre-defined motif regions, whereas the HMMER analysis of the same data yields a correct alignment of only the first two motifs.

The results in Tables IV.2 through IV.5 compare the alignment performance of Meta-MEME and HMMER to the performance of nine alignment methods tested by McClure *et al.* on the same four sets of sequences. These results are summarized in table IV.6. Overall, Meta-MEME performs as well or better than eight of the other ten multiple alignment methods examined, and only slightly worse than the second best method. Given the relatively *ad hoc* nature of these test, it is not possible to assign a threshold for statistical significance to these differences in performance. Nevertheless,

```
        xxxxxxxxxxx.xxxxxxxxx.xxxxx..........xxxxxx.xxxxxxx.xxxxxxxxxx.xxxxxxxxx
HAHU    V.LSPADKTN..VKAAWGKVG.AHAGE..........YGAEAL.ERMFLSF..PTTKTYFPH.FDLS.HGSA
HAOR    M.LTDAEKKE..VTALWGKAA.GHGEE..........YGAEAL.ERLFQAF..PTTKTYFSH.FDLS.HGSA
HADK    V.LSAADKTN..VKGVFSKIG.GHAEE..........YGAETL.ERMFIAY..PQTKTYFPH.FDLS.HGSA
HBHU    VHLTPEEKSA..VTALWGKVN.VDEVG...........G.EAL.GRLLVVY..PWTQRFFES.FGDL.STPD
HBOR    VHLSGGEKSA..VTNLWGKVN.INELG...........G.EAL.GRLLVVY..PWTQRFFEA.FGDL.SSAG
HBDK    VHWTAEEKQL..ITGLWGKVNvAD.CG...........A.EAL.ARLLIVY..PWTQRFFAS.FGNL.SSPT
MYHU    G.LSDGEWQL..VLNVWGKVE.ADIPG..........HGQEVL.IRLFKGH..PETLEKFDK.FKHL.KSED
MYOR    G.LSDGEWQL..VLKVWGKVE.GDLPG..........HGQEVL.IRLFKTH..PETLEKFDK.FKGL.KTED
IGLOB   M.KFFAVLALCiVGAIASPLT.ADEASlvqsswkavsHNEVEIlAAVFAAY.PDIQNKFSQFaGKDLASIKD
GPUGNI  A.LTEKQEAL..LKQSWEVLK.QNIPA..........HS.LRL.FALIIEA.APESKYVFSF.LKDSNEIPE
GPYL    GVLTDVQVAL..VKSSFEEFN.ANIPK...........N.THR.FFTLVLEiAPGAKDLFSF.LKGSSEVPQ
GGZLB   M.L.DQQTIN..IIKATVPVLkEHGVT...........ITTTF.YKNLFAK.HPEVRPLFDM.GRQ..ESLE

        xxxxx.xxxxxxxxxxxxx..xxxxxxxxxxxxxxx..xxxxxxx.xxxxxxx...xxxxxxxxxxxxxxxx
HAHU    QVKGH.GKKVADA.LTN......AVA.HVDDMPNA...LSALS.D.LHAHKL....RVDPVNF.KLLSHCLL
HAOR    QIKAH.GKKVADA.L.S......TAAGHFDDMDSA...LSALS.D.LHAHKL....RVDPVNF.KLLAHCIL
HADK    QIKAH.GKKVAAA.LVE......AVN.HVDDIAGA...LSKLS.D.LHAQKL....RVDPVNF.KFLGHCFL
HBHU    AVMGNpKVKAHGK.KVLGA..FSDGLAHLDNLKGT...FATLS.E.LHCDKL....HVDPENF.RL.LGNVL
HBOR    AVMGNpKVKAHGA.KVLTS..FGDALKNLDDLKGT...FAKLS.E.LHCDKL....HVDPENFNRL..GNVL
HBDK    AILGNpMVRAHGK.KVLTS..FGDAVKNLDNIKNT...FAQLS.E.LHCDKL....HVDPENF.RL.LGDIL
MYHU    EMKASeDLKKHGA.TVL......TALGGILKKKGHH..EAEIKPL.AQSHATK...HKIPVKYLEFISECII
MYOR    EMKASaDLKKHGG.TVL......TALGNILKKKGQH..EAELKPL.AQSHATK...HKISIKFLEYISEAII
IGLOB   T.GA...FATHATRIVSFLseVIALSGNTSNAAAV...NSLVSKL.GDDHKA....R.GVSAA.QF..GEFR
GPUGNI  NNPK...LKAHAAVIFKTI...CESATELRQKGHAVwdNNTLKRL.GSIHLK....N.KITDP.HF.EVMKG
GPYL    NNPD...LQAHAG.KVFKL..TYEAAIQLEVNGAVAs.DATLKSL.GSVHVS....K.GVVDA.HF.PVVKE
GGZLB   Q......PKALAM.TVL......AAAQNIENLPAIL..PAVKKIAvKHCQAGVaaaH.YPIVGQEL.LGAIK

        xxxxxxxxx.xxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxx..x
HAHU    VT.LAA.H..LPAEFTPA..VHASLDKFLASV.STVLTS..KY..R
HAOR    VV.LAR.H..CPGEFTPS..AHAAMDKFLSKV.ATVLTS..KY..R
HADK    VV.VAI.H..HPAALTPE..VHASLDKFMCAV.GAVLTA..KY..R
HBHU    VCVLAH.H..FGKEFTPP..VQAAYQKVVAGV.ANALAH..KY..H
HBOR    IVVLAR.H..FSKDFSPE..VQAAWQKLVSGV.AHALGH..KY..H
HBDK    IIVLAA.H..FTKDFTPE..CQAAWQKLVRVV.AHALAR..KY..H
MYHU    QV.LQSKHPgDFGADAQGA.MNKALELFRKDM.ASNYKELGFQ..G
MYOR    HV.LQSKHSaDFGADAQAA.MGKALELFRNDM.AAKYKEFGFQ..G
IGLOB   TA.LVA.Y..LQANVSWGDnVAAAWNKA.LDN.TFAIVV..PR..L
GPUGNI  ALLGTIKEA.IKENWSDE..MGQAWTEAYNQLVATIKAE..MK..E
GPYL    AILKTIKEV.VGDKWSEE..LNTAWTIAYDELAIIIKKE..MKdaA
GGZLB   EVLGDAAT..DDILDAWGK.AYGVIADVFIQVEADLYAQ..AV..E
```

Figure IV.1: **HMMER alignment of the globins.** The pre-defined motif regions are indicated by underlining. Capital letters correspond to match states in the HMM; lowercase letters are generated by insert states and are therefore unaligned.

```
        .........................xxxxxxxx..............xxxxxxxxxxxxxxxxx.......
HAHU    vlspa................dktnVKAAWGKVgahag.....eygaEALERMFLSFPTTKTYFphfd...
HAOR    mltda................ekkeVTALWGKAaghge.....eygaEALERLFQAFPTTKTYFshfd...
HADK    vlsaa................dktnVKGVFSKIgghae.....eygaETLERMFIAYPQTKTYFphfd...
HBHU    vhltp...............eeksaVTALWGKVnvde.......vggEALGRLLVVYPWTQRFFesfgdls
HBOR    vhlsg...............geksaVTNLWGKVnine.......lggEALGRLLVVYPWTQRFFeafgdls
HBDK    vhwta...............eekqlITGLWGKVnvad.......cgaEALARLLIVYPWTQRFFasfgnls
MYHU    glsdg................ewqlVLNVWGKVeadip.....ghgqEVLIRLFKGHPETLEKFdkfkhlk
MYOR    glsdg................ewqlVLKVWGKVegdlp.....ghgqEVLIRLFKTHPETLEKFdkfkglk
IGLOB   mkffavlalcivgaiaspltadeaslVQSSWKAVshn.........evEILAAVFAAYPDIQNKFsqf....
GPUGNI  altek................qealLKQSWEVLkqnip.....ahslRLFALIIEAAPESKYVFsflkds.
GPYL    gvl....................tdVQVALVKSsfeefnanipknthRFFTLVLEIAPGAKDLFsflkgs.
GGZLB   mldqqtin..........iikatvpVLKEHGVTit...........tTFYKNLFAKHPEVRPLFdmg....

        ......xxxxxxxxxxxx..............xxxxxxxxx.........xxxxxxxxxxxxxxx......
HAHU    ...lshGSAQVKGHGKKVadalt.....navaHVDDMPNALsa......lSDLHAHKLRVDPVNFk.....
HAOR    ...lshGSAQIKAHGKKVadals.....taagHFDDMDSALsa......lSDLHAHKLRVDPVNFk.....
HADK    ...lshGSAQIKAHGKKVaaalv.....eavnHVDDIAGALsk......lSDLHAQKLRVDPVNFk.....
HBHU    tpdavmGNPKVKAHGKKVlgafs.....dglaHLDNLKGTFat......lSELHCDKLHVDPENFr.....
HBOR    sagavmGNPKVKAHGAKVltsfg.....dalkNLDDLKGTFak......lSELHCDKLHVDPENFn.....
HBDK    sptailGNPMVRAHGKKVltsfg.....davkNLDNIKNTFaq......lSELHCDKLHVDPENFr.....
MYHU    sedemkASEDLKKHGATVltalggi..lkkkghHEAEIKPLAq........SHATKHKIPVKYLEF......
MYOR    tedemkASADLKKHGGTVltalgni..lkkkgqHEAELKPLAq........SHATKHKISIKFLEY......
IGLOB   ...agkDLASIKDTGAFAt..............HATRIVSFLse......vIALSGNTSNAAAVNSlvsklg
GPUGNI  .neipeNNPKLKAHAAVIfkticesatelrqkgHAVWDNNTLkr......lGSIHLKNKITDPHFEvmk...
GPYL    .sevpqNNPDLQAHAGKVfklt......yeaaIQLEVNGAVasdatlkslGSVHVSKGVVDAHFPvvk...
GGZLB   ...rqeSLEQPKALAMTVlaa.........aqNIENLPAILpav...kkiAVKHCQAGVAAAHYPi.....

        ............xxxxxxxxxxxxxxxxxxxxxxxxxxx...........xxxxxxxxxx
HAHU    ...........lLSHCLLVTLAAHLPAEFTPAVHASLDKfl........asVSTVLTSKYR
HAOR    ...........lLAHCILVVLARHCPGEFTPSAHAAMDKfl........skVATVLTSKYR
HADK    ...........fLGHCFLVVVAIHHPAALTPEVHASLDKfm........caVGAVLTAKYR
HBHU    ...........lLGNVLVCVLAHHFGKEFTPPVQAAYQKvv........agVANALAHKYH
HBOR    ...........rLGNVLIVVLARHFSKDFSPEVQAAWQKlv........sgVAHALGHKYH
HBDK    ...........lLGDILIIVLAAHFTKDFTPECQAAWQKlv........rvVAHALARKYH
MYHU    ............ISECIIQVLQSKHPGDFGADAQGAMNKalelf..rkdmaSNYKELGFQG
MYOR    ............ISEAIIHVLQSKHSADFGADAQAAMGKalelf..rndmaAKYKEFGFQG
IGLOB   ddhkargvsaaqfGEFRTALVAYLQANVSWGDNVAAAWNKal.........dNTFAIVVPRL
GPUGNI  ..........gaLLGTIKEAIKENWSDEMGQAWTEAYNQl...........VATIKAEMKE
GPYL    ..........eaILKTIKEVVGDKWSEELNTAWTIAYDEla.........iIIKKEMKDAA
GGZLB   ............VGQELLGAIKEVLGDAATDDILDAWGKaygviadvfiqvEADLYAQAVE
```

Figure IV.2: **Meta-MEME alignment of the globins.** The pre-defined motif regions are indicated by underlining. Motifs discovered by MEME and included in the model are in capital letters. Amino acids in the inter-motif regions are in lowercase and are unaligned.

```
         xxxxxxxxxxxxxx.xxxxxxx..xxxxx.xxxx..x...xxxx........xxxxxxxx....xxxxx..x
HTLV-II  LDTAPCLFSDGSPQ.KAAYVLW..DQTIL.QQDI..T...PLP.........SH.ETHS.......AQ...K
SRV-I    LNNALLVFTDGSS....TGMAA....YTL.A..D..T...TIKF........QT.NLNS.......AQ...L
RSV      PVPGPTVFTDASSS.THKGVVV....W.R.E.GP..R...WEIK........EIADLGA.....SVQQ....
HIV-II   IPGAETFYTDGSCN.RQSKEGK..AGY.V.T.DR..G...KDKV........KKLEQTT......NQQ....
MoMLV    PDADHTWYTDGSSL.LQEGQRK..AGAAV.T.TE..T...EVIW........AK.ALDAG....TSAQ...R
Ingi     PREHYKLWTDGSVS.LGEKLGA...AALL.H.RN..N...TLIC........AP.KTGA.....GELSCsyR
CAMV     PEEKLIIETDASDD.YWGGMLK..AIKIN.EGTN..T...ELICryasgsfkAA.EKNY.....HSND....
17.6     FTKKFTLTTDASDVaLGAVLSQ..DGHPL.S.YIs.R...TLN.........EH.EINY......STI...E
MAUP     FNNSTNLQEP.SDS.RLLYRKG...SWVN.I.RFaaY...LYSK........LSEEKHGLvpkfLEKL...R
HBV      RPGLCQVFADATP..TGWGLVM...GHQR.M..R..G...TFSA........PL.PIHT.....AELL....
Copia    FENKIIGYVDSDWA.GSEIDRKstTGYLFkMFDF..N...LICW........NTKRQNSVa...ASST...E
E.coli   MLKQVEIFTDGSCL.GNPGPGG..YGAIL.R.YR..GrekTFSA........GYTRTTNNr...MELM....

         xxxxxxx......xxxxxx.xxxxxxx.xxx.xxxxxxx..xxxx.xxxxxxxxxxxx....xxxxxxxx..
HTLV-II  GELLAL.......ICGLRAaKPWPSL..NI..FLD.S....KYLI.KYL.HSLAIGAFl...GTSAHQ....
SRV-I    VELQAL.......IAVLSA.FPNQPL..NI..YTD.S....AYLA.HSI.PLLETVAQikhiSETAKL....
RSV      LEARAV.......AMAL...LLWPTTP.TNV.VTDSAF...VAKM.LLK.MGQEGV.P....STAAAF....
HIV-II   AELEAF.......AMAL...TDSGPKV.NI..IVD.S....QYVM.GIS.ASQPTESE....SKIVNQ....
MoMLV    AELIALTqa...lKMAEGK.KLNVYTD.SRYaFATAHI...HGEI.YRR.RGLLTS.E....GKEIKN....
Ingi     AECVALE......IGLQRL.LKWLPRYrST..PSRLSIFs.DSLS.MLT.ALQTGPLA....VTDPIL....
CAMV     KETLAV.......INTI.K.KFSIYL..TPVhFLI.R....TDNT.HFK.SFVNLN.Y....KGDSKL....
17.6     KELLAI.......VWAT.K.TFRHYLL.GRH.FEISSD...HQPL.SWL.YR.M.KD......PNSKL....
MAUP     EINFALDkvdvteIDSKLS.RLMKFSV.SA..AYDEVGTl.ALKS.LFKFRNSERESI....KASFKQLRen
HBV      AACFARSr.....SGANIIgTDNSVVL.SR..KYTSFPWllGCAA.NWILRGTSFV.Y....VPSALNPA..
Copia    AEYMAL.......FEAVREaLWLKFLL.TS..INIKL....ENPI.KIYEDNQGCISIa..nNPSCHKRAk.
E.coli   AAIVALE......ALKEHC.EVILSTD.SQ..YVRQGI...TQWIhNWKKRGWKTADK....KPVKNV....

         .....xxxxxxxxxxxxxxxxx.xxxxxx.xxxxxxxxx..xxxxxxxxxxxx...xxxx.xxx...xxxx
HTLV-II  .....TLQA..ALPPLLQGKT....IYLH.HVRSHT......N.LPDPISTFN...EYTDsLIL...APL.
SRV-I    .....FLQCQ.QLIYNRSIPF....YIGH.V.RAHS......G.LPGPIAHGN...QKAD.LATktvASN.
RSV      .....ILED..ALS..QRSAM...AAVLH.V.RSHSEVPgfFT.EGNDVADS....QAT..F.....QAY.
HIV-II   .....IIEEM.IKK..EAIYV...AWVPA.H.KGIGG.....NQEV.D.HLV....SQG...IR...QVL.
MoMLV    .....KDEIL.ALLKALFLPKRlSIIHCPgHQKGHSAEAr.GN.RMADQAARKaaiTET..PDT...STLL
Ingi     .....RRLWR.LLLQVQRRKI...RIRLQ.FVFDHCGVK..RN.EVCD.EMA....KKA..ADL...PQL.
CAMV     .....GRNIRWQAW..LSHYS...FDVEH.I.KGTD......N.HFAD.FLS....REF..N.....KVNS
17.6     .....TRWRV.KLS..EFDFD....IKY..I.KGKE......N.CVAD.ALSR...IKL..E.....ETY.
MAUP     gkiaeFSEAR.RLW.FEILKL....IRLD.L.FNAS......S.LACD.DLL....SHL..QDR...RSI.
HBV      .....DDPSRGRLG..LSRPL....LRLP.F.RPTTGR....TSLYAD.SPSV...PSHL.PD....RVH.
Copia    ....hIDIKYHFAR.EQVQNN...VICLE.Y.IPTE......N.QLADIFTK....PLP..AAR...FVE.
E.coli   .....DLWQRLDAALGQHQIKW.EWVKGH.AGHPENER....CDELAR.AAAM...NPTL.EDTg.yQVEV
```

Figure IV.3: **HMMER alignment of the RH domains of the RNA-directed DNA polymerase.** The pre-defined motif regions are indicated by underlining. Capital letters correspond to match states in the HMM; lowercase letters are generated by insert states and are therefore unaligned.

```
        ..............xxxxxxxx.........................................xxxxxxxx
HTLV-II ldt.........apCLFSDGSPqkaayvlwdqtilqqd..........itplpshethsaqkgELLALICG
SRV-I   lnn.........alLVFTDGSStgmaaytladtti................kfqtnlnsaqlvELQALIAV
RSV     pvp.........gpTVFTDASSsthkgvvvwregprw...........eikeiadlgasvqqlEARAVAMA
HIV-II  ipg.........aeTFYTDGSCnrqskegkagyvtdrg..........kdkvkkleqttnqqaELEAFAMA
MoMLV   pda.........dhTWYTDGSSllqegqrkagaavttet.......eviwakaldagtsaqraELIALTQA
Ingi    pre.........hyKLWTDGSVslgeklgaaallhrnntl......icapktgagelscsyraECVALEIG
CAMV    pee.........klIIETDASDdywggmlkaikinegtntelicryasgsfkaaeknyhsndkETLAVINT
17.6    ftk.........kfTLTTDASDvalgavlsqdghplsyi........srtlneheinystiekELLAIVWA
MAUP    fnnstnlqepsdsrLLYRKGSWvnirfaay..........................lysklseEKHGLVPK
HBV     rpg.........lcQVFADATPtgwglvmghqrmr................gtfsaplpihtaELLAACFA
Copia   fen.........kiIGYVDSDWagseidrksttgylfkmfdf.nlicwntkrqnsvaassteaEYMALFEA
E.coli  mlk.........qvEIFTDGSClgnpgpggygailryrg.......rektfsagytrttnnrmELMAAIVA

        x....................xxxxxxxxxx........................................
HTLV-II Lraak............pwpsLNIFLDSKYLikylhslaigaflgtsahqtlqaalp..............
SRV-I   Lsafp.............nqpLNIYTDSAYLahsiplletvaqikhisetaklflqcqq.............
RSV     Lllwp.............ttpTNVVTDSAFVakmllkmgqegvpstaaafiledal...............
HIV-II  Ltds..............gpkVNIIVDSQYVmgisasqpteseskivnqiie...................
MoMLV   Lkmae.............gkkLNVYTDSRYAfatahihgeiyrrrglltsegkeiknkdeil.........
Ingi    Lqrllkwlp....ryrstpsrLSIFSDSLSMltalqtgplavtdpilrrlwrll.................
CAMV    Ikkfsiy.........ltpvhFLIRTDNTHFksfvnlnykgdsklgrnir.....................
17.6    Tktfrhy.........llgrhFEISSDHQPLswlyrmkdpnskltrwr.......................
MAUP    Flek..............lreINFALDKVDVteidsklsrlmkfsvsaaydevgtlalkslfkfrnseres
HBV     Rsrs...............gaNIIGTDNSVVlsrkytsfpwllgcaanwilrgtsfvyvpsa.........
Copia   VrealwlkflltsiniklenpIKIYEDNQGCisiannpschkrakhidiky....................
E.coli  Lealk............ehceVILSTDSQYVrqgitqwihnwkkrgwktadkkpvknvd............

        .......................................xxxxxxxxx.................
HTLV-II .............pllqgktiylhhvrshtnlpdpistfNEYTDSLIlapl..............
SRV-I   ............liynrsipfyighvrahsglpgpiahgNQKADLATKtvasn............
RSV     ..............sqrsamaavlhvrshsevpgfftegNDVADSQATfqay.............
HIV-II  ..................emikkeaiyvawvpahkgiggNQEVDHLVSqgirqvl..........
MoMLV   .........allkalflpkrlsiihcpghqkghsaeargNRMADQAARkaaitetpdtstll...
Ingi    ................lqvqrrkirirlqfvfdhcgvkrNEVCDEMAKkaadlpql.........
CAMV    ....................wqawlshysfdvehikgtdNHFADFLSRefnkvns..........
17.6    ......................vklsefdfdikyikgkeNCVADALSRikleety..........
MAUP    ikasfkqlrengkiaefsearrlwfeilklirldlfnasSLACDDLLShlqdrrsi.........
HBV     .........lnpaddpsrgrlglsrpllrlpfrpttgrtSLYADSPSVpshlpdrvh........
Copia   ...................hfareqvqnnvicleyipteNQLADIFTKplpaarfve........
E.coli  ............lwqrldaalgqhqikwewvkghaghpeNERCDELARaaamnptledtgyqvev
```

Figure IV.4: **Meta-MEME alignment of the RH domains of the RNA-directed DNA polymerase.** The pre-defined motif regions are indicated by underlining. Motifs discovered by MEME and included in the model are in capital letters. Amino acids in the inter-motif regions are in lowercase and are unaligned.

	1	2	3	4	5	Total
META-MEME	12	11	11*	11*’	12*	57
HMMER	11	12*’	11*	10*’	11*’	55
AMULT	12	12	12	12	12	60
ASSEMBLE	12	11	12	12	12	59
CLUSTAL V	12	11	12	12	12	59
DFALIGN	12	12	12	12	12	60
GENALIGN	11*’	12	12	10’	11	56
MULTAL	12	11	12	12	12	59
MACAW	9	11	9	8	8	45
PIMA	12	12	12	12	12	60
PRALIGN	8	8*	9*	8*	10	43

Table IV.2: **Comparison of multiple alignment methods on the globin family.** Each column contains, for one pre-defined motif region, the number of motif occurrences properly aligned by each method. An asterisk (*) indicates that the motif was correctly aligned in two or more misaligned subsets of the test sequences. A dagger (†) indicates that a gap was inserted into the motif. Data from all but the first two rows of this table are from [92]

	1	2	3	4	5	6	7	8	Total
META-MEME	12	11	10	12	12	12	12	11	92
HMMER	12	12*	12*	12	12	12	12	12	96
AMULT	12	10	11	12	12	12	12	12	93
ASSEMBLE	11	7	10	12	12	12	12	12*	87
CLUSTALV	12	11	11*	12	12	12	12	12*	94
DFALIGN	12	12	12	12	12	12	12	12	96
GENALIGN	12’	9*	10	12	12	12	12*	11*	90
MULTAL	12	9*	10*	12	12	12*	12	12	91
MACAW	8	0	9	12	12	10	12	0	63
PIMA	12	11	11	12	12	12	12	12	94
PRALIGN	12	10*	6*	4	9*	9*	4	4	58

Table IV.3: **Comparison of multiple alignment methods on the kinase family.** See caption on p. 86.

	1	2	3	Total
META-MEME	12	4*'	9*	25
HMMER	11	8	6*	25
AMULT	11	7	10	28
ASSEMBLE	—	—	—	0
CLUSTALV	12	9*	6*	27
DFALIGN	12	12*	12	36
GENALIGN	11	8*'	7*	26
MULTAL	10	7*	9*	26
MACAW	12	4	8	24
PIMA	12	5*	5*	22
PRALIGN	8*	4*	8*	20

Table IV.4: **Comparison of multiple alignment methods on the proteases.** See caption on p. 86. A dash (—) indicates that the method failed to produce an alignment.

	1	2	3	4	Total
META-MEME	11	9	11	12	43
HMMER	11	10*	5*'	7*	33
AMULT	11	9*	8*	7*	35
ASSEMBLE	—	—	—	—	0
CLUSTALV	12	9	9*	9*	39
DFALIGN	12	12	10	12	46
GENALIGN	12*'	7	8*'	9*'	36
MULTAL	11*	11*	9*	10	41
MACAW	7	5	7	3	22
PIMA	10	9	8*	11*	38
PRALIGN	9	8*	6*	3	26

Table IV.5: **Comparison of multiple alignment methods on the RH domains.** See caption on p. 86. A dash (—) indicates that the method failed to produce an alignment.

	Globins	Kinases	Proteases	RHs	Average
DFALIGN	60	96	36	46	59.5
CLUSTALV	59	94	27	39	54.8
META-MEME	57	92	25	43	54.3
MULTAL	59	91	26	41	54.3
AMULT	60	93	28	35	54.0
PIMA	60	94	22	38	53.5
HMMER	55	96	25	33	52.3
GENALIGN	56	90	26	36	52.0
MACAW	45	63	24	22	38.5
PRALIGN	43	58	20	26	36.8
ASSEMBLE	59	87	0	0	36.5

Table IV.6: **Summary of multiple alignment methods comparison.** Each column lists, for one family, the overall percentage of motifs that were correctly aligned by each multiple alignment method. Methods are ranked according to the average percentage of motifs correctly aligned, which is listed in the right-most column.

the cluster of scores around 54% indicates that Meta-MEME is a member of this high-scoring group of alignment methods.

With respect to HMMER, Meta-MEME produces better alignments for two of the four families and an equally good alignment for a third family. Only for the kinases does HMMER produce a better alignment, and the difference in quality for this family is small (four motif occurrences). The difference between Meta-MEME and HMMER is greatest for the most difficult family of the four, the RH domains. Indeed, for this family, Meta-MEME performs second best overall, indicating that Meta-MEME is particularly good at aligning highly divergent sequence sets.

Phylogenetic analysis

The tree shown in Figure IV.5 is identical to the widely accepted true tree for this data set. Thus, Meta-MEME provides the only known method of finding the true phylogenetic relationships for this data set.

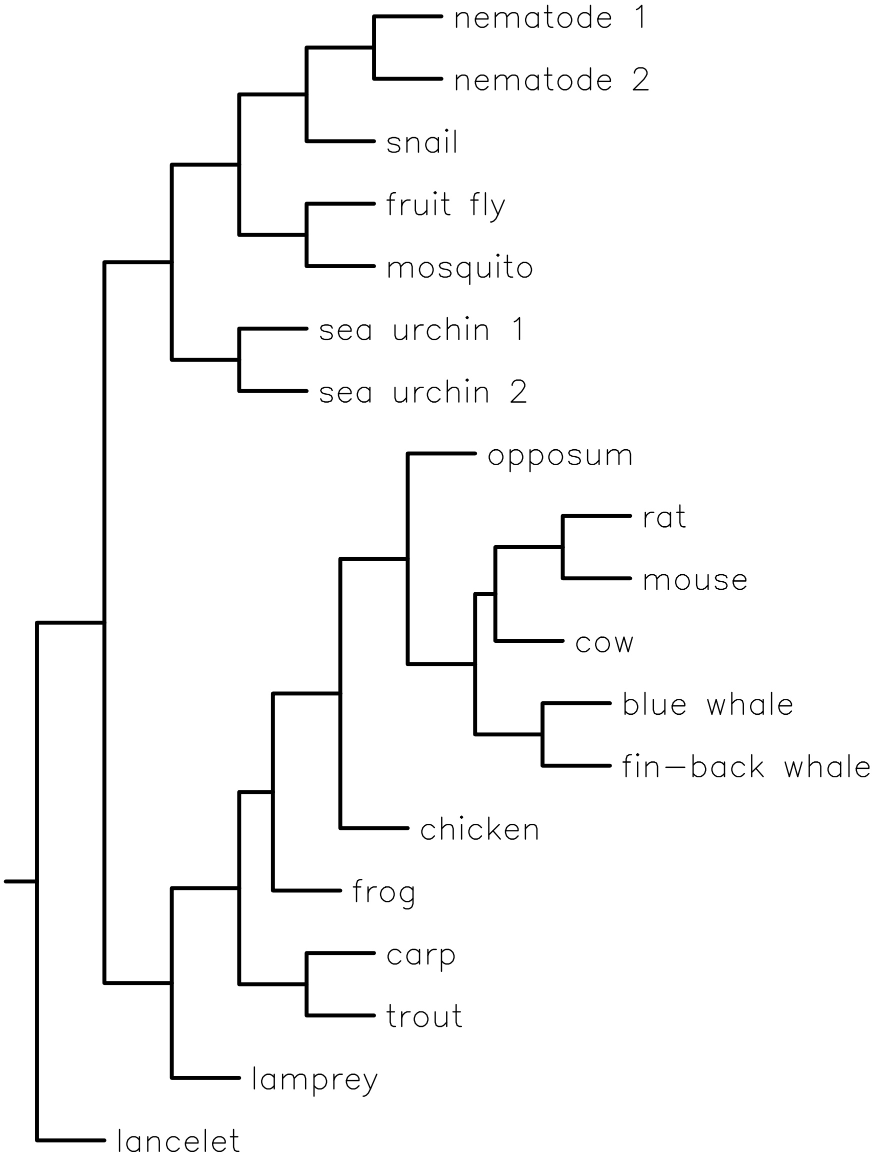


Figure IV.5: **The correct chordate phylogeny recovered by Meta-MEME.** This phylogeny is based upon a 49-motif alignment. It is the single most parsimonious tree found by the protein parsimony program in Phylip [50].

IV.A.4 Discussion

The first set of experiments reported here show that Meta-MEME produces multiple alignments that are comparable in quality to the best alignments currently available, and that are better than alignments produced by a non-motif based hidden Markov modeling package.

The comparisons reported here are biased against Meta-MEME and HMMER in several ways. First, each of the multiple alignment methods tested by McClure *et al.* was tested using a wide range of parameter settings. For each method, the setting that yielded the best alignment was selected. In addition, the data sets were sent to the authors of each multiple alignment software package, and the authors were invited to use their own software to produce the best possible alignment. In contrast, both Meta-MEME and HMMER were run with their default parameter settings. Both methods could almost certainly perform better if their parameters were optimized for these data sets. Finally, the DFALIGN method of Feng and Doolittle likely performs best because its algorithm was modified in order to cope with problems presented by these particular data sets [90].

Given these considerations, Meta-MEME's performance is quite impressive. That performance could likely be improved even further if the Meta-MEME models were trained via expectation-maximization.

The phylogenetic tree presented in the second experiment suggests that motif-based alignments are useful in inferring phylogenetic relationships. Clearly, several important questions remain open, including the degree to which the results are sensitive to various parameter settings, such as the number of motifs and the selection of training sequences. In addition, any general claim that Meta-MEME alignments provide the basis for superior phylogenetic inference must await testing on additional data sets. Nonetheless, the fact that Meta-MEME alone finds the true tree for this data set is impressive. Note that this performance is not the result of data snooping: the tree reported here is the first tree generated by Meta-MEME. None of the parameter settings were varied in order to find this tree. The next step in this research will be to perform sensitivity analysis, systematically varying the training sequences and the number of motifs to determine how frequently Meta-MEME succeeds in recovering the true tree.

IV.B Homology detection using linear models

IV.B.1 Introduction

The experiments described in this section illustrate the effectiveness of modeling protein families using motif-based linear hidden Markov models. We examine two well-studied families, the short chain alcohol dehydrogenases and the 4Fe-4S ferredoxins. Few members of the dehydrogenase family contain shuffled or repeated domains. Hence, a linear topology should be sufficient for modeling this family. Many of the ferredoxins, on the other hand, contain repeated elements, making the linear topology less appropriate for this family. In order to compare Meta-MEME's motif-based strategy with the standard topology, we create Meta-MEME models in a format readable by the standard HMM package HMMER [46]. This allows us to perform database searches using the same searching software for both the HMMER and the Meta-MEME models.

The results of these searches show that Meta-MEME outperforms the standard HMM on the homology detection task for both of the families and for all training set sizes examined here. The difference in performance is particularly large for small training sets.

IV.B.2 Methods

The first data set consists of a group of dehydrogenases that includes mammalian 11β-hydroxysteroid and 17β-hydroxysteroid dehydrogenase and their homologs in the short chain alcohol dehydrogenase family. We chose this data set because it is large and phylogenetically diverse [108, 17, 18]. The thirty-eight sequences

2BHD_STREX	20-β-hydroxysteroid dehydrogenase
3BHD_COMTE	3-β-hydroxysteroid dehydrogenase
ACT3_STRCO	Putative ketoacyl reductase
ADH_DROME	Alcohol dehydrogenase
AP27_MOUSE	Adipocyte P27 protein (AP27)
BA72_EUBSP	7-α-hydroxysteroid dehydrogenase
BDH_HUMAN	D-β-hydroxybutyrate dehydrogenase precursor
BEND_ACICA	Cis-1,2-dihydroxy-3,4-cyclohexadiene-1-carboxylate dehydrogenase
BPHB_PSEPS	Biphenyl-2,3-dihydro-2,3-diol dehydrogenase
BUDC_KLETE	Acetoin (diacetyl) reductase
CSGA_MYXXA	C-Factor
DHB2_HUMAN	Estradiol 17 β-dehydrogenase 2
DHB3_HUMAN	Estradiol 17 β-dehydrogenase 3
DHCA_HUMAN	Carbonyl reductase (NADPH)
DHES_HUMAN	Estradiol 17 β-dehydrogenase
DHGB_BACME	Glucose 1-dehydrogenase B
DHII_HUMAN	Corticosteroid 11-β-dehydrogenase
DHMA_FLAS1	N-acylmannosamine 1-dehydrogenase
ENTA_ECOLI	2,3-dihydro-2,3-dihydroxybenzoate dehydrogenase
FABG_ECOLI	3-oxoacyl-[acyl-carrier protein] reductase
FABI_ECOLI	Enoyl-[acyl-carrier-protein] reductase (NADH)
FIXR_BRAJA	FixR protein
FVT1_HUMAN	Follicular variant translocation protein 1 precursor
GUTD_ECOLI	Sorbitol-6-phosphate 2-dehydrogenase
HDE_CANTR	Hydratase-dehydrogenase-epimerase (HDE)
HDHA_ECOLI	7-α-hydroxysteroid dehydrogenase
HMTR_LEIMA	H region methotrexate resistance protein
LIGD_PSEPA	C α-dehydrogenase
MAS1_AGRRA	Agropine synthesis reductase
NODG_RHIME	Nodulation protein G
PCR_PEA	Protochorophyllide reductase precursor
PGDH_HUMAN	15-hydroxyprostaglandin dehydrogenase (NAD(+))
PHBB_ZOORA	Acetoacetyl-coa Reductase
RFBB_NEIGO	dTDP-glucose 4,6-dehydratase
RIDH_KLEAE	Ribitol 2-dehydrogenase
YINL_LISMO	Hypothetical 26.8 Kd protein in Inla 5'region (ORFA)
YRTP_BACSU	Hypothetical 25.3 Kd protein In Rtp 5'region (ORF238)
YURA_MYXXA	Hypothetical protein in Uraa 5'region (Fragment)

Table IV.7: **SWISS-PROT identifiers and descriptions of the 38 dehydrogenase training set.**

FER1_AZOVI	FER2_RHOCA	FER2_RHORU	FER_MYCSM
FER_STRGR	FER_PSEPU	FER_PSEST	FER_THETH
FER_CLOBU	FER_CLOPA	FER_CLOPE	FER_CLOSP
FER_CLOTM	FER_CLOTS	FER_MEGEL	FER_PEPAS
FER_BUTME	FER_CHLLT	FER1_CHLLI	FER2_CHLLI
FER_METBA	FER_METTL	FER_THEAC	FER2_DESDN
FER1_DESVM	FER_ENTHI	FERX_ANASP	FERN_AZOCH
FDXN_RHILT	FERN_RHIME	FERN_BRAJA	FER1_RHOCA
FER_SULAC	FER1_RHOPA	FERN_AZOVI	FER3_ANAVA
FER3_RHOCA	FER_CLOTH	FER_DESGI	FER1_DESDN
FER_THELI	FER_THEMA	FIXX_RHILP	FIXX_RHILE
FIXX_RHILT	PSAC_ANTSP	PSAC_CHLRE	PSAC_CUCSA
PSAC_MAIZE	PSAC_MARPO	PSAC_PEA	PSAC_PINTH
PSAC_TOBAC	PSAC_WHEAT	PSAC_CYAPA	PSAC_ANASP
PSAC_FREDI	PSAC_SYNEN	PSAC_SYNP2	PSAC_SYNP6
PSAX_SYNY3	DHSB_BACSU	DHSB_ECOLI	FRDB_ECOLI
FRDB_PROVU	YFRA_PROVU	FRDB_WOLSU	FDHB_METFO
FIXG_RHIME	RDXA_RHOSH	PHFL_DESVH	PHFL_DESVO
DMSB_ECOLI	DMSB_HAEIN	YFFE_ECOLI	FDNH_ECOLI
FDXH_HAEIN	FDHB_WOLSU	HMC2_DESVH	HMC6_DESVH

Table IV.8: **SWISS-PROT identifiers of the 4Fe-4S ferredoxins.** See caption for Table IV.9

used in the training set are listed in Table IV.7. Pairwise alignments of almost all of these sequences are less than 30% identical after using gaps and insertions to maximize identities. Many sequences are less than 20% identical after use of gaps and insertions. These thirty-eight sequences represent a small portion of the approximately 650 known dehydrogenases in GenBank release 95 [55].

We also search for homologs of a set of 159 4Fe-4S ferredoxins [101]. These sequences comprise all known 4Fe-4S ferredoxins in SWISS-PROT release 33 [13]. Family members were selected using PROSITE 13.1 [12]. Ten additional members were added to the family by Michael Gribskov, based upon ROC analysis and sequence comparisons. The SWISS-PROT identifiers for all 159 sequences, as well as the justifications for including the ten additional sequences, are given in Table IV.9. Nested training sets were selected at random from all 159 sequences.

GLPC_ECOLI	GLPC_HAEIN	HYCB_ECOLI	HYCF_ECOLI
PHSB_SALTY	PSRB_WOLSU	NRFC_ECOLI	NRFC_HAEIN
NAPF_HAEIN	NAPG_ECOLI	NAPG_HAEIN	NAPH_ECOLI
YGL5_BACST	YJES_ECOLI	YA43_HAEIN	DHSB_USTMA
DHSB_SCHPO	DHSB_HUMAN	DHSB_RAT	DHSB_DROME
MBHT_ECOLI	PHF1_CLOPA	ASRC_SALTY	NUIC_MAIZE
NUIC_ORYSA	NUIC_TOBAC	NUIC_WHEAT	NUIC_PLEBO
NUIM_BOVIN	NUIM_RHOCA	NQO9_PARDE	NUOI_ECOLI
YJJW_ECOLI	FER1_DESAF	FIXX_AZOCA	FIXX_BRAJA
NARH_ECOLI	NARY_ECOLI	NIFJ_ANASP	NIFJ_KLEPN
FER_METTE	PSAC_ODOSI	YEIA_ECOLI	FER_BACTH
DHSB_CHOCR	DHSB_CYACA	NARH_BACSU	YWJF_BACSU
FER_SACER	FER_CLOAC	FER_CLOST	FER1_RHORU
FER_CHRVI	FER3_DESAF	FERV_AZOVI	FER_ALIAC
FER3_PLEBO	FER2_DESVM	FIXX_RHIME	PSAC_EUGGR
PSAC_SPIOL	PSAC_ANAVA	PSAC_SYNY3	FRDB_HAEIN
FRHG_METTH	COOF_RHORU	FDOH_ECOLI	ASRA_SALTY
HYDN_ECOLI	NAPF_ECOLI	NAPH_HAEIN	DHSB_YEAST
DHSB_ARATH	NUIC_MARPO	NUIC_SYNY3	DCMA_METSO
ISP1_TRYBB	YAAT_ECOLI	FER_BACST	

Table IV.9: **SWISS-PROT IDs for the 159 4Fe-4S ferredoxins (continued).** Ten of the sequences listed here are not included in the PROSITE 13.1 listing for this family. DHSB_CHOCR, DHSB_CYACA, FER_METTE, and PSAC_ODOSI are included here based on homology to PROSITE annotated families in this group, and ROC analysis. ISP1_TRYBB, excluded from this group by PROSITE, appears to be closely related to NADH oxidoreductases in this group as shown by ROC and sequence comparisons (NQQ9, NUIM, NUOI, HYCF, NUIC). NARH_BACSU, NARH_ECOLI and NARY_ECOLI, while showing lower ROC, have excellent 4Fe-4S sequences highly similar to those in DMSB, PHSB, FDNH, HYCB, etc. YEIA_ECOLI is a possible type III ferredoxin and has a very strong ROC. YWJF_BACSU is included in the positives because of high ROC, significant similarity to glycerol-3-phosphate dehydrogenase subunits (GLPC) which are ferredoxins, and clear presence of two appropriate 4Fe-4S binding sequences.

Motifs are discovered in each family using MEME version 2.0 [6] with its default parameters, as specified on the web site [5]. Specifically, we use the ZOOPS motif occurrence model, which stands for "zero or one occurrence per sequence." We use Dirichlet mixtures for prior probabilities, modified by the megaprior heuristic [9]. The minimum width of a motif is specified as 12 (although the motifs returned may be shorter than this, due to a shortening heuristic in MEME), and the maximum width is 55.

Linear motif-based HMMs are constructed by Meta-MEME using motifs selected by the majority occurrence heuristic (described in Section II.B). Each inter-motif spacer is represented by a single state with a self-loop. The order and spacing of motifs in the model is determined by the MAST motif occurrence diagram from the database sequences that receives the lowest MAST e-value.

The standard linear HMMs used for comparison with Meta-MEME are constructed using the default settings of the HMMER program *hmmt*, version 1.8. The training algorithm begins with a uniform model with length equal to the average length of sequences in the training set. The model is trained via expectation-maximization, using a simulated annealing protocol to avoid local optima. The initial Boltzmann temperature is 5.0, with a temperature decrease of 5% at each iteration.

Numerous algorithms exist for searching a database using a hidden Markov model. HMMER offers four such programs, which vary in the way they match sequences against models. The first, *hmmsw*, performs a local Smith/Waterman search for matches of a partial sequence to a partial model; *hmms* matches a complete model against complete sequences; *hmmls* matches a complete model against one or more partial sequences; and *hmmfs* matches fragments of a model to multiple non-overlapping partial sequences. Informal experiments with these programs yielded consistently better results using *hmmsw*.

In the best case, a database search with an HMM would return sequence scores which ranked all of the family members above all of the non-family members. However, all of the HMMER programs suffer from the presence in the database of intermediate-scoring sequence fragments. When a sequence fragment exists in the database, it will match only a portion of the model, giving a relatively low score. Then, even though the fragment is a member of the family, it may be ranked among the non-family members.

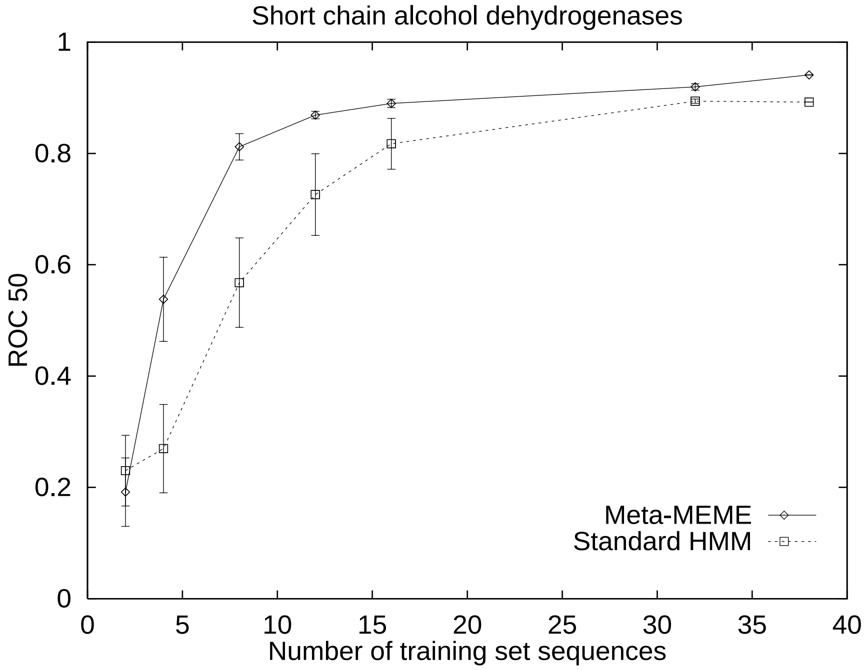
Because sequence fragments are a deficiency of the database rather than of the search method, and because many fragments are redundant with the whole sequences included in the database, we opted to filter such fragments from the database. Rather than use a fixed threshold for all models, we calculated from the canonical motif signature the minimum length of a sequence containing two motifs and two spacers. All sequences in the database shorter than this value are filtered out. The filtered database is then used for both the Meta-MEME search and the standard HMM search.

IV.B.3 Results

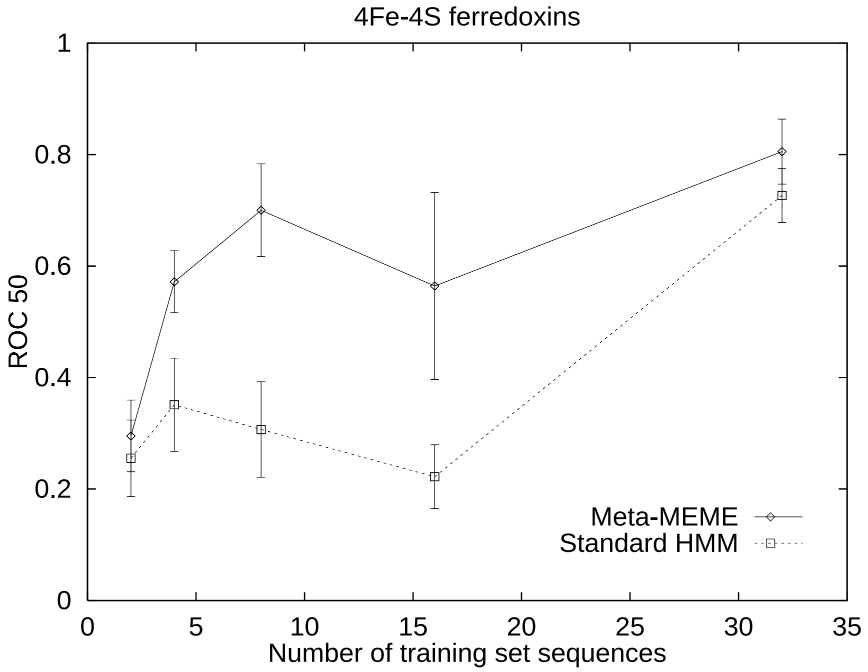
Short-chain alcohol dehydrogenases

Figure IV.6(a) shows that Meta-MEME outperforms standard linear HMMs for most subsets of the dehydrogenase training set, with the most striking difference between the two methods appearing for smaller data sets. Each series in the figure represents the average of ten successions of training and testing runs, using randomly selected, nested subsets of the 38-sequence training set. Searches are evaluated according to their ROC_{50} score (described in Section III.B). Error bars represent standard error. For each subset of sequences, a standard and a motif-based HMM were built and were used to search Genpept 95. Not only does Meta-MEME consistently score better than the standard linear HMMs, the motif-based HMMs appear to be more robust across different random subsets, as evidenced by the relative smoothness of the Meta-MEME curve.

Figure IV.7 shows an alignment of four different motif-based HMMs, built from nested subsets of the dehydrogenase training set. These motifs illustrate the



(a)



(b)

Figure IV.6: **Comparison of Meta-MEME and standard linear HMMs in recognizing (a) short-chain alcohol dehydrogenases and (b) 4Fe-4S ferredoxins.** Each point represents an average of ten separate runs, except for the ferredoxin runs using 16-sequence training sets, for which only three runs completed (see the discussion below). Error bars represent standard error.

```
38 sequences   9-[2]-64-[1]-12-[6]-17-[4]-9-[3]-73
16 sequences   5-[2]-61-[1]-42-[4]-12-[3a]-5-[3b]-33-[5]-13
 8 sequences   11-[2]-65-[1]-64-[3a]-22-[3b]-26-[7]-28
 4 sequences   13-[1]-18-[6]-37-[3a]-22-[3b]-41

38 ---------LVTGAASGIG---------------------------------------------------------
16 ****-----LVTGASRGIG****-----------------------------------------------------
 8 -----------TGASSGIG---------------------------------------------------------
 4 **********************************************************************------

38 ------VDVLVNNAG*------------EDWDRVIxVNLTGVF*----------------GRIVNVSSVAG-----
16 -------DVLVNNAG****-----------------------------------------GRIVNVSS--------
 8 -------DVLVNNAG**-----------------------------------------------------------
 4 -------DALINNAG------------------VFHINVVGPIR--------------------------------

38 ----YSASKAAVxGLTRSLALELAPxGIRVNVVAPG----------------------------------------
16 ----YSASKAALxGLTRSLALE-----IRVNAVAPGFVxTDM-------------------------------FL
 8 ----YAASKAAL-----------------------PGxIxTDM------------------------IPIGRMGQP
 4 ----YxMSKAAL-----------------------PGWVxTDM---------------------------------

38 --------------------------------
16 ASDEASYIT-------------**********
 8 EEIA---------------------------*
 4 ------**************************
```

Figure IV.7: **Comparison of four motif-based HMMs built from a nested series of random subsets of the 38-sequence dehydrogenase training set.** The canonical schema for each model is shown at the top, with the lengths of spacers alternating with motif numbers in brackets. In the models, motifs are represented by their consensus sequence. Hyphens ("-") represent the expected length of spacers generated by insert nodes, and asterisks ("*") are gaps inserted into this diagram in order to align the models.

biological basis for the sensitivity of Meta-MEME. Motifs 1 and 2 are part of the nucleotide cofactor binding site [30, 129, 130]; motif 3 is part of the catalytic site. A protein sequence that had, for example, motifs 1 and 3 interchanged would not have the same 3D structure and could not function as a steroid dehydrogenase. By scoring protein similarity and dissimilarity on the basis of motif order and spacing, Meta-MEME effectively models spatial information in the 3D structure of the canonical dehydrogenase. This information differentiates homologs from unrelated proteins which contain isolated fragments resembling sequences in the training set. Comparison of protein 3D structures is the most sensitive method for determining homology [37]. This explains Meta-MEME's excellent ability to recognize alcohol dehydrogenase homologs as seen in Figure IV.6(a).

The motifs discovered using smaller training sets correspond strongly to the original motifs found using the largest training set. In the figure, motifs are numbered consecutively according to the order in which they were discovered. Any motif from one training set which overlaps with a motif from a previous training set is assigned the same number as the first. Using the largest training set, MEME finds five motifs which appear in more than half of the training set. The third of these motifs, however, is very long (32 residues); in subsequent analyses using smaller data sets, motif 3 gets split into two halves (marked 3a and 3b). Furthermore, motif 5, which was discarded because of the majority occurrence heuristic in the 38-sequence analysis, is found and included in the HMM based upon sixteen sequences. Motif 6 is lost when the training set is reduced from thirty-eight to sixteen sequences but is recovered when the training set size reaches 4 sequences. Motifs 4 and 5 are lost between sixteen and eight sequences, and motif 2 is lost when four sequences are used. Only one new motif (marked 7) is introduced in the smaller training sets; other candidates are discarded because of the majority occurrence heuristic.

The order and spacing of the motifs within the different models is also conserved. In all four models, the order of motifs is identical. Furthermore, spaces between motifs are consistent across the four models. In the figure, hyphens represent spacer states in the model, whereas asterisks represent "gaps," which were inserted into the figure in order to align the motifs. Very few asterisks were required in order to generate a perfect alignment. Only the last model, based upon four training sequences, contains a significant missing portion.

The motif-based HMMs are considerably smaller than their standard HMM counterparts. For the dehydrogenase family, the average Meta-MEME model contains 58 states; the standard models average 264 states. Assuming six motifs per model, the average Meta-MEME model therefore contains $(19 * 58) + 6 + 1 = 1109$ trainable parameters. The standard HMM, by contrast, averages $25 * 264 = 6600$ parameters. The standard model is therefore 6.0 times as large as the motif-based model.

4Fe-4S ferredoxins

A similar set of experiments was conducted using the 4Fe-4S ferredoxin data set. In addition to using a different, considerably smaller family, the ferredoxin searches were carried out on a different database, SWISS-PROT 33 instead of Genpept 95. Nonetheless, Meta-MEME again consistently outperforms the standard HMMs, as shown in Figure IV.6(b). The degree of separation between the two series is even greater than for the dehydrogenases. The standard HMMs of the ferredoxin family are on average 5.1 times as large as the average motif-based HMM.

Although Meta-MEME outperforms standard HMMs, both methods perform more poorly for ferredoxin data sets of size 16 than for smaller, 8- or 4-sequence data sets. This anomaly results from the interaction of two of the heuristics described above. For many of the 16-sequence data sets, the majority occurrence heuristic selected a relatively large number of motifs. Unfortunately, it was often impossible for MAST to locate a single sequence containing all of these motifs. Consequently, a canonical motif occurrence schema was found for only three of the runs. As a result, neither Meta-MEME nor HMMER completed the other runs, since the filtering of the database depends upon the canonical schema. This adverse interaction of heuristics only occurred with the ferredoxin data set and only with training sets of size 16. A variant of our heuristics would overcome this problem; however, our emphasis in this work is to demonstrate the general utility of motif-based HMMs. Rather than fine-tuning heuristics, future work will replace these heuristics by, for example, completely connecting the motifs and learning the occurrence schema from the given data.

IV.B.4 Discussion

Results from Meta-MEME are encouraging. As expected, motif-based HMMs discriminate better than their standard linear counterparts for the two protein families we investigated, yet due to their small size, motif-based HMMs require fewer training sequences in order to be trained to precision. Furthermore, since HMM search algorithms are generally linear in the size of the model, motif-based HMMs can search a database 5-6 times faster than a standard model. By focusing its models on highly conserved regions of the training set, Meta-MEME effectively ignores noisy portions of the data, thereby allowing the software to recognize distant homologs.

Meta-MEME's performance may be affected by biases in the training set. In the experiments reported here, the dehydrogenase training set was hand-selected so as to fairly uniformly represent a particular protein family. However, in the ferredoxin experiments, randomly selected training sets containing several closely related sequences may have biased some of the trained ferredoxin models. These biases may explain the relatively large standard error bars in Figure IV.6(b). Such biases could have been reduced by first removing highly similar sequences using a program such as `purge` [97]. This approach is followed in subsequent experiments.

IV.C A case study: short chain alcohol dehydrogenases

IV.C.1 Introduction

In this section, we examine in detail Meta-MEME's ability to model a family of short chain alcohol dehydrogenases [15, 109, 122, 84, 79, 16]. This family includes 11β-hydroxysteroid and 17β-hydroxysteroid dehydrogenase, enzymes that are important in actions of steroids that affect blood pressure, reproduction and development and also the growth of some cancers of breast and prostate. In addition to its medical importance, we chose this family for testing our method because it is large and phylogenetically diverse.

Using a dataset of thirty-seven dehydrogenases, Meta-MEME identifies at least 350 members of this family in Genpept 96 and clearly separates these sequences from non-homologous proteins. In addition, we show that concatenated MEME motifs can be used to construct reliable phylogenetic trees for distantly related sequences. Concatenated motifs can be aligned unambiguously, unlike entire sequences. This is an important consideration when constructing a multiple alignment of many distantly related sequences because the alignment may be degraded by mutations suggested spuriously by ambiguities in assigning insertions and deletions. Others have dealt with this problem and have constructed useful phylogenetic trees by ignoring the highly divergent segments containing insertions and deletions [20, 86]. We find that concatenated MEME motifs also yield useful trees, with the advantage that the analysis is unbiased and automated.

IV.C.2 Methods

The training set of alcohol dehydrogenases consists of the thirty-eight sequences listed in Table IV.7, except for dTDP-Glucose 4,6-Dehydratase. Pairwise alignments of almost all of these sequences are less than 30% identical after using gaps and insertions to maximize identities [109, 17, 18]. Many sequences are less than 20% identical after use of gaps and insertions.

The six strongest motifs in the set of thirty-seven divergent dehydrogenase sequences are determined using MEME version 2.0. MEME is run with the ZOOPS model, with the minimum motif width set at 12 amino acids, and the Dirichlet mixture prior [8, 60]. Next, Genpept release 96 is searched with all six motifs using MAST. The motif occurrence diagram from the highest-scoring protein provides the framework for a linear Meta-MEME model incorporating all six motifs. This motif-based HMM is used by a modified Smith-Waterman algorithm [46] to search Genpept 96 for homologs. The output score for each sequence is expressed as log-odds scores in bits (i.e., log_2).

In order to construct a phylogenetic tree, the sequences of the first six motifs from the MEME analysis of each dehydrogenase homolog were collapsed into a single string. These motif-only strings were analyzed using the protein parsimony analysis program from the Phylip software package [50]. The analysis was repeated 30 times, using at each iteration a random reordering of the sequences, and selecting the most parsimonious tree from all iterations.

IV.C.3 Results

MEME analysis

Figure IV.8 displays the six motifs of the dehydrogenase dataset along with the entropy plot, which is a measure of the information content at each position. The motifs are mapped onto the primary sequence of 20β-hydroxysteroid dehydrogenase in Figure IV.9. Also shown in Figure IV.9 is the secondary structure determined from X-ray crystallographic analysis [56]. The secondary and tertiary structure of this enzyme is very similar to homologs such as dihydropteridine reductase [127], 17β-hydroxysteroid dehydrogenase-type 1 [31], enoyl reductases [114, 24], and *E. coli* 7α-hydroxysteroid dehydrogenase [121] despite having pairwise sequence similarities

```
MOTIF 1
         bits 6.8
              6.1
              5.4
              4.7
Information   4.1
content       3.4         *
(26.1 bits)   2.7    *   **
              2.0 *  * * ** *
              1.4 * ** ******
              0.7 ***********
              0.0 -----------

Multilevel        GRVDVLVNNAG
consensus           L I I
sequence
```

```
MOTIF 2
         bits 6.8
              6.1
              5.4
              4.7
Information   4.1
content       3.4
(25.1 bits)   2.7    **     *
              2.0   ***   ***
              1.4 ******  ***
              0.7 ***********
              0.0 -----------

Multilevel        ALVTGASSGIG
consensus         VII  GA  L
sequence                G
```

```
MOTIF 3
         bits 6.8
              6.1
              5.4
              4.7
Information   4.1 *
content       3.4 *   *
(31.3 bits)   2.7 *   *
              2.0 *  **      *
              1.4 * *****   **  **
              0.7 ******** *******
              0.0 ----------------

Multilevel        YSASKAAVxGLTRSLA
consensus            A FGL  FSK
sequence
```

```
MOTIF 4
         bits 6.8
              6.1
              5.4
              4.7
Information   4.1
content       3.4
(18.2 bits)   2.7        *
              2.0 * ***  *
              1.4 * ******
              0.7 ********
              0.0 --------

Multilevel        GRIVNVSS
consensus            I I
sequence
```

```
MOTIF 5
         bits 6.8
              6.1
              5.4
              4.7
Information   4.1
content       3.4
(28.1 bits)   2.7         *
              2.0  *      **   *
              1.4 ** ** * **   * *
              0.7 ***** **** * ***
              0.0 ----------------

Multilevel        GIRVNAVxPGxVxTDM
consensus          VT   I    I
sequence
```

```
MOTIF 6
         bits 6.8
              6.1
              5.4
              4.7
Information   4.1
content       3.4        *
(35.7 bits)   2.7        *
              2.0 *      *
              1.4 *     **  * *         *
              0.7 ********* ******* *** *
              0.0 -----------------------

Multilevel        WDRVIxVNLTGVFxLTRAVLPxM
consensus         F     I   S V G Q     L
sequence
```

Figure IV.8: **Motifs from MEME analysis of short chain alcohol dehydrogenases.** The entropy plot is a measure of the information content at each position of the motif. The consensus sequence below the entropy plot shows sites where specific amino acids are present with a probability of at least 20%.

```
                β-strand-A   Turn        α-helix-B
               |-----------|-----|----------------------------|
                             •       •   •
  1 M N D L S G K T V I I T G G A R G L G A E A A R Q A V A A
                    2 2 2 2 2 2 2 2 2 2 2

      β-strand-B          α-helix-C
    |-----------------|-----------------------|         |----
 30 G A R V V L A D V L D E E G A A T A R E L G D A A R Y Q H

   β-strand-C          α-helix-D                β-strand-D
    ---------|  |-----------------------------|   |-------------
 59 L D V T I E E D W Q R V V A Y A R E E F G S V D G L V N N
                                        1 1 1 1 1 1 1 1 1

     β-strand-D                 α-helix-E
    -----------|     |---------------------------------
 88 A G I S T G M F L E T E S V E R F R K V V D I N L T G V F
    1 1                         6 6 6 6 6 6 6 6 6 6 6 6 6

       α-helix-E              β-strand-E
    -------------------|     |-------------|
117 I G M K T V I P A M K D A G G G S I V N I S S A A G L M G
    6 6 6 6 6 6 6 6 6 6             4 4 4 4 4 4 4 4

                     α-helix-F
          |-----------------------------------|
146 L A L T S S Y G A S K W G V R G L S K L A A V E L G T D R
                3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3             5

        β-strand-F
     |-------------------|
175 I R V N S V H P G M T Y T P M T A E T G I R Q G E G N Y P
    5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

                             α-helix-G
                        |-------------------|
204 N T P M G R V G N E P G E I A G A V V K L L S D T S S Y V

      β-strand-G
     |-----------|
233 T G A E L A V D G G W T T G P T V K Y V M G Q 255
```

Figure IV.9: **Alignment of MEME motifs on** *Streptomyces hydrogenans* **20β-hydroxysteroid dehydrogenase.** Each motif as determined by MEME is shown below the sequence of *S. hydrogenans* 20β-hydroxysteroid dehydrogenase. The secondary structure was determined from the X-ray analysis of crystals of *S. hydrogenans* 20β-hydroxysteroid dehydrogenase [56], and has a similar fold to that of its homologs [127, 31, 114, 24, 121]. The boxed segment at the beginning of motif 3 contains the conserved tyrosine and lysine residues at the catalytic site.

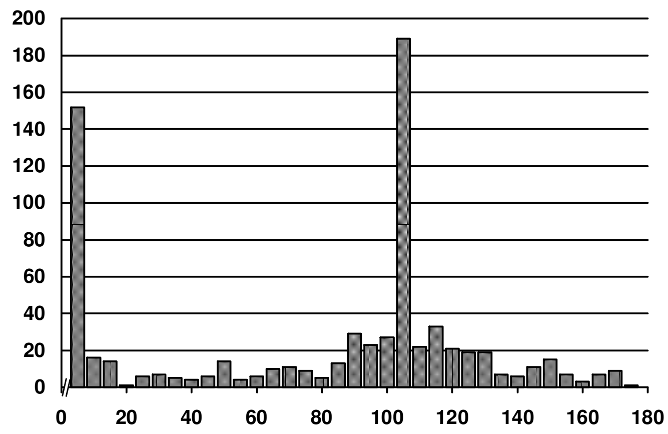


Figure IV.10: **Meta-MEME analysis of Genpept 96.** The output histogram has a minimum at 20 bits, demonstrating the selectivity of the HMM analysis. Sequences with negative scores are not shown. The peaks at 105 and 115 bits are due to *Drosophila* alcohol dehydrogenase sequences.

of 15% to 22%. The six motifs map onto structurally important domains, some of which have been shown to be functionally important by site-specific mutagenesis studies [100, 1, 34, 111, 131, 35] and structural analysis [126, 125]. Beginning at the amino terminus, the order of the motifs is (2)-(1)-(6)-(4)-(3)-(5). Their combined length is 85 amino acids, and they span 183 residues of 20β-hydroxysteroid dehydrogenase.

Meta-MEME analysis

These six motifs were combined in their proper order into a single hidden Markov model for analysis. This model was then used to search Genpept 96 for homologs. Figure IV.10 shows the histogram of the output of this search, and Table IV.10 shows selected sequences from the output. The distribution is bimodal with a clear minimum at 20 bits, demonstrating excellent separation of dehydrogenase homologs from the rest of the database.

The high scoring sequences contain the full 85 residues in the template, which spans 180 to 188 amino acids in most of the proteins. This is consistent with an absence of extra loops in these proteins and a common 3D structure. An interesting exception is carbonyl reductase, in which the six motifs span 228 residues

Score	Seq	Mod	ID	Species	Description
178.7	8-188	1-85	145881	*E. Coli*	3-ketoacyl-acyl carrier protein reductase
174.8	9-194	1-85	153142	*S. coelicolor*	actIII protein
170.9	5-184	1-85	790552	*R. meliloti*	acetoacetyl CoA reductase
170.4	8-190	1-85	1203984	*H. sapiens*	NAD+-dependent 15-hydroxyprostaglandin dehydrogenase
170.2	9-189	1-85	46308	*R. meliloti*	nodG gene product
169.3	6-186	1-85	1222069	*H. influenzae*	3-oxoacyl-[acyl-carrier protein] reductase
149.4	10-191	1-85	309860	*C. testosteroni*	β-hydroxysteroid dehydrogenase
149.1	14-196	1-85	912437	*E. coli*	7α-hydroxysteroid dehydrogenase
148.0	10-192	1-85	1419053	*M. tuberculosis*	unknown
145.5	325-504	1-85	695398	*C. tropicalis*	hydratase-dehydrogenase-epimerase
133.1	6-193	1-85	975895	*H. sapiens*	17-β-hydroxysteroid dehydrogenase
127.7	8-235	1-85	181037	*H. sapiens*	carbonyl reductase
116.6	37-222	1-85	179475	*H. sapiens*	11-β-hydroxysteroid dehydrogenase
115.4	32-213	1-85	1054531	*B. taurus*	11-cis-retinol dehydrogenase
90.6	86-188	14-82	304662	*D. immigrans*	alcohol dehydrogenase
65.8	118-244	12-85	957251	*A. thaliana*	oxidoreductase

Table IV.10: **Selected Meta-MEME output from from an analysis of Genpept 96.** The table (continued on the next two pages) shows some high scoring sequences that contain all 85 residues in the six motifs. Column 1 gives the log-odds score in bits. Columns 2 and 3 show the correspondence between amino acids in the sequence and states in the model. The last three columns contain the Genpept ID, species name and sequence description. Analysis of proteins with scores from 23.2 to 8.5 bits reveal that the first protein that is not a member of the short chain dehydrogenase family is malate dehydrogenase with a score of 8.9 bits, followed by ribulose bisphosphate carboxylase/oxygenase with a score of 8.5 bits. The sequences of several homologs, such as halohydrin epoxidase [135] and the sugar epimerases [77, 85, 19], have diverged from the signature motif used in PROSITE [12], which has made identification of their ancestry difficult.

Score	Seq	Mod	ID	Species	Description
23.2	138-195	46-85	46868	*S. coelicolor*	ORF3 protein
21.3	32-203	2-58	861340	*C. elegans*	similar to ribitol dehydrogenase
19.7	15-101	1-22	603171	*E. coli*	unknown
19.1	12-45	58-85	453866	*A. thaliana*	tropinone reductase homologue
18.8	8-18	1-11	145888	*E. coli*	ORF3
18.5	4-41	54-85	699381	*M. leprae*	glucose 1-dehydrogenase
18.0	3-157	1-60	473600	*S. fradiae*	dTDP-glucose dehydratase
17.7	1-33	59-85	1053075	*P. mirabilis*	ORF1; similar to E. coli EnvM
17.6	128-184	47-85	641817	*Corynebact. sp.*	halohydrin epoxidase A
17.3	19-168	1-77	641819	*Corynebact. sp.*	halohydrin epoxidase B
17.3	4-14	1-11	415277	*E. coli*	unknown
16.7	1-13	73-85	887852	*E. coli*	ORF_f67p
16.0	1-26	66-85	1234827	*L. pneumophila*	ORF1; similar EnvM
15.6	262-313	51-85	237650	*B. napus*	enoyl-acyl carrier protein reductase
15.3	85-149	27-67	1332595	*Synecho. sp.*	dNDP-glucose dehydratase
14.5	28-147	2-40	618456	*A. parasiticus*	norsolornic acid
13.9	10-20	1-11	471145	*S. paucimobilis*	ORFUP
13.7	1-13	73-85	1166429	*C. elegans*	K08F4.9
13.4	217-298	25-85	1055124	*C. elegans*	coded for by cDNA yk62b4.3
13.0	98-173	27-67	1314581	*Sphingo. S88*	dTDP-D-glucose-4,6-dehydratase

Table IV.11: **Selected Meta-MEME output from from an analysis of Genpept 96.** See caption on p. 107

Score	Seq	Mod	ID	Species	Description
13.0	89-143	29-59	1359482	*A. mediterranei*	dNDP-glucose dehydratase
12.8	97-171	27-67	398120	*X. campestris*	TDP-glucose oxireductase
12.4	9-117	1-37	1143392	*A. thaliana*	uridine diphosphate glucose epimerase
12.2	113-186	50-85	203979	*R. norvegicus*	dihydropteridine reductase
12.2	116-189	50-85	181553	*H. sapiens*	dihydropteridine reductase
12.0	101-174	27-67	1001273	*Synecho. sp.*	hypothetical protein
10.8	2-22	68-85	666992	*D. mojavensis*	alcohol dehydrogenase
10.4	6-164	2-67	413996	*B. subtilis*	ipa-72d gene product
10.3	25-200	1-19	506333	*H. roretzi*	HrEpiB
9.9	8-116	1-37	1173555	*P. sativum*	UDP-galactose-4-epimerase
9.6	3-93	1-27	567874	*S. erythraea*	thymidine diphospho-glucose 4,6-dehydratase
9.6	1-15	71-85	516105	*Synecho. sp.*	aklaviketone reductase
9.4	23-64	37-59	699306	*M. leprae*	hypothetical protein
9.3	2-29	1-18	1294775	*H. influenzae*	ADP-L-glycero-D-manno-heptose-6-epimerase
8.9	3-111	1-41	1429254	*B. subtilis*	UDP-glucose 4-epimerase
8.9	4-154	1-58	406095	*N. meningitidis*	UDP-glucose 4-epimerase
8.9	3-62	1-15	294198	*Photobact. sp.*	malate dehydrogenase
8.6	38-48	1-11	466869	*M. leprae*	gpdB; B1496_F1_31
8.5	87-198	13-85	407314	*M. tuberculosis*	inhA peptide (AA 1-269)
8.5	87-198	13-85	1155270	*M. bovis*	enoyl ACP reductase
8.5	58-95	33-56	1381396	rhodophyte BOm1	ribulose bisphosphate carboxylase/oxygenase large subunit

Table IV.12: **Selected Meta-MEME output from from an analysis of Genpept 96.** See caption on p. 107

due to an insertion of 41 residues between motifs 4 and 2 [128]. This insertion does not compromise the analysis. Meta-MEME output is useful in identifying the region where a distantly homologous protein has diverged from the dataset. For example, *Drosophila immigrans* alcohol dehydrogenase has a score of 90.6 bits based on residues 14-85 of the template. Evidently, the segment corresponding to motif 2 in this alcohol dehydrogenase has diverged from the dataset. A similar analysis holds for an oxidoreductase (score of 65.7 bits) required for shoot apex development in *Arabidopsis thaliana*.

We examined the sequences with scores below twenty bits using citations in Entrez and SWISS-PROT and, in some cases, a BLAST search to determine which sequences were homologous to short chain dehydrogenases. All sequences above 8.9 bits are homologs. The first non-homologous protein is malate dehydrogenase at 8.9 bits; the next is ribulose bisphosphate carboxylase/oxygenase at 8.5 bits.

Phylogeny

One consequence of the projects to sequence genomes in phylogenetically diverse organisms is a wider use of phylogenetic analysis to assist in understanding the evolution of structure and function. We were interested in how well the motifs generated by MEME could be used for a phylogenetic analysis. We therefore combined the first six motifs for each protein into a single sequence, which by virtue of the MEME analysis can be aligned with the other thirty-six proteins. Two equally parsimonious phylogenies were discovered by Phylip [50]. One of these two is shown in Figure IV.11; the other phylogeny was similar. Phylogenies using the entire sequences of 11β-hydroxysteroid dehydrogenase-type 1, 17β-hydroxysteroid dehydrogenase-types 1, 2, and 3, and β-hydroxybutyrate dehydrogenase [18], as well as bacterial steroid dehydrogenases [17] have been determined previously [18] and are in general agreement with that from the motifs. In particular, the type 1 11β- and 17β-hydroxysteroid dehydrogenases cluster together on a branch separate from 17β-hydroxysteroid dehydrogenase-type 2, which clusters with β-hydroxybutyrate dehy-

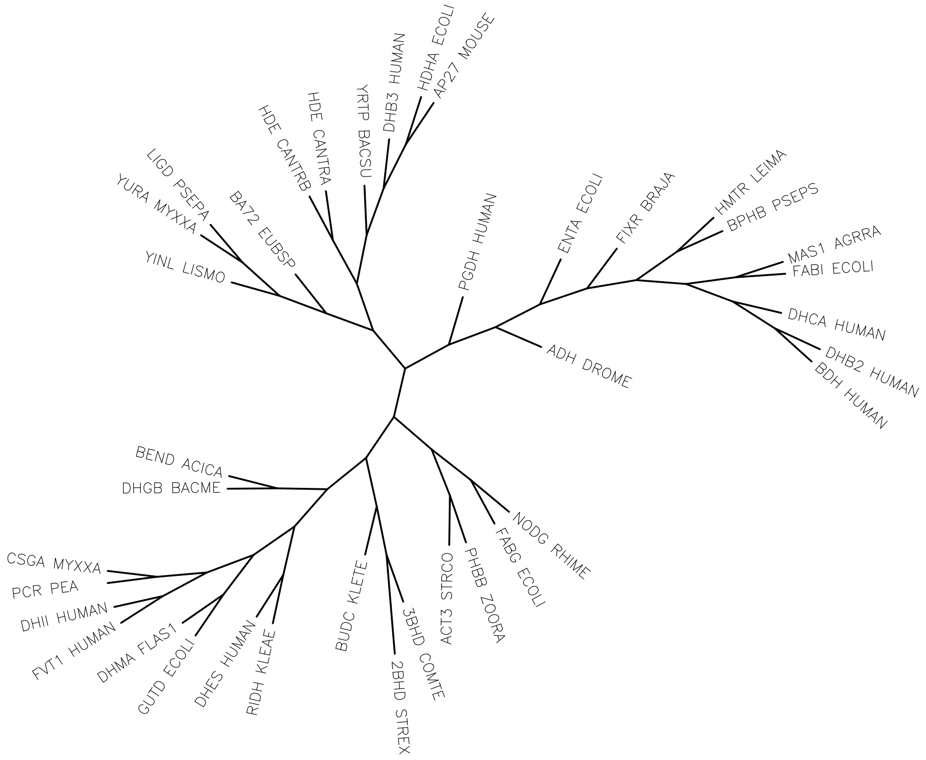


Figure IV.11: **Phylogenetic analysis of the dehydrogenase dataset.** The sequences of the first six motifs from the MEME analysis of each protein were collapsed into a single sequence and analyzed by parsimony analysis [50]. The 11β-hydroxysteroid and 17β-hydroxysteroid dehydrogenases-type 1 cluster together on a branch separate from 17β-hydroxysteroid dehydrogenases-type 2 and 3, which are on separate branches. The motif phylogeny is in agreement with a phylogenetic analysis of the entire sequences of the steroid dehydrogenases [18].

drogenase. On a separate branch is 17β-hydroxysteroid dehydrogenase-type 3. Thus, the information in the eighty-five residues in the first six motifs gives a useful phylogeny for the steroid dehydrogenases.

IV.C.4 Discussion

There is a strong biological basis for the sensitivity of Meta-MEME. Motifs 1 and 2 are part of the nucleotide cofactor binding site [30, 129, 130]; motif 3 contains the catalytic site. A protein sequence that had motifs 1 and 3 interchanged would not have the same 3D structure and could not function the way the steroid dehydrogenases do. By scoring protein similarity and dissimilarity on the basis of motif order and spacing, the HMM method is using the spatial information in the 3D structure of the canonical dehydrogenase to identify homologs from the noise of unrelated proteins that have islands of amino acid sequence similarity to the dataset. Comparisons of protein 3D structures is the most sensitive method for determining homology [37], which we propose explains the excellent ability of HMM to separate homologs from noise as seen in Figure IV.10.

In summary, Meta-MEME provides a sensitive and selective method for homology searches to identify distantly related proteins. This facilitates collecting large and diverse collections of homologous proteins for motif analysis for use in elucidating the relationship between structure, function and evolution.

IV.D Modeling families containing repeated elements

IV.D.1 Introduction

Several mechanisms of molecular evolution can result in subsequences of the chromosome being copied and reinserted once or multiple times into a single gene. Some of these copying events result in proteins with small tandem repeats, such as are found in the ice-nucleating proteins [133, 64]; other proteins contain repeated elements, such as zinc fingers or kringle domains, located in varying positions throughout the protein.

Meta-MEME is the only computational tool currently available that explicitly models these repeated elements. A motif-based hidden Markov model with a completely connected topology can accurately characterize a protein family in which single family members contain multiple occurrences of one or more motifs.

In the experiments reported here, we investigate Meta-MEME's ability to model two well-known families that contain repeated subsequences: the 4Fe-4S ferredoxins and the kringle domain proteins. For these families, we find that Meta-MEME's total probability log-odds scores are accurate, but the corresponding Viterbi scores tend to underestimate the likelihood that a given sequence belongs to the family in question. Training the Meta-MEME model results in improved characterization of an independent test set of family members. However, using this trained model to search a database of sequences for homologs results in slightly degraded performance relative to that of the original, untrained model. A tentative explanation for these results is given in Section IV.D.4.

IV.D.2 Methods

The 4Fe-4S ferredoxins are a primarily bacterial subfamily of the ferredoxin family of iron-sulfur proteins [101]. The ferredoxins mediate electron transfer in a wide variety of metabolic reactions. The 4Fe-4S group is characterized by a 26-amino-acid domain containing four cysteine residues that bind to four iron and four sulfur atoms. PROSITE version 13.1 lists 149 4Fe-4S ferredoxins. For these experiments, ten additional sequences were added to this set, based upon independent analyses carried out by Michael Gribskov. The resulting set of 4Fe-4S ferredoxins is given in Table IV.9.

The second protein family we examine is defined by the kringle domain signature [33]. The kringle domain is a triple-looped, disulfide cross-linked domain

APOA_HUMAN	APOA_MACMU	FA12_BOVIN	HGFA_HUMAN
HGFL_HUMAN	HGFL_MOUSE	HGF_RAT	PLMN_BOVIN
PLMN_CANFA	PLMN_MACMU	PLMN_MOUSE	PLMN_PETMA
THRB_BOVIN	THRB_HUMAN	THRB_MOUSE	UROK_CHICK
UROK_HUMAN	UROK_MOUSE	UROK_RAT	UROT_HUMAN
UROT_MOUSE	URT2_DESRO	URTB_DESRO	URTG_DESRO
FA12_CAVPO	FA12_HUMAN	HGF_HUMAN	HGF_MOUSE
PLMN_HORSE	PLMN_HUMAN	PLMN_PIG	PLMN_RAT
THRB_RAT	UROK_BOVIN	UROK_PAPCY	UROK_PIG
UROT_RAT	URT1_DESRO		

Table IV.13: **SWISS-PROT identifiers for the 38 kringle domain proteins.**

which repeats as many as 38 times in a single sequence. An exhaustive list of the 38 sequences in SWISS-PROT 33 containing kringle domains was created by Michael Gribskov using Smith-Waterman searches with kringle regions from the sequences PLMN_HORSE and UROK_CHICK. See Table IV.13 for the list of SWISS-PROT identifiers in this family.

As in previous experiments (see Section. III.B) binary sequence weighting is employed to reduce redundancy within the protein families. After eliminating eight sequence fragments from the set of 4Fe-4S ferredoxins, the `purge` program [87] is used to eliminate highly similar sequences, using a similarity threshold of 200 bits. This procedure results in a set of 70 divergent sequences. The kringle domain proteins are more closely related to one another, so the weighting procedure reduces the original set of 38 sequences to only five divergent sequences.

From each purged family, five series of nested training sets are randomly selected. For the 4Fe-4S ferredoxins, these training sets are of sizes 2, 4, 8, 16 and 32 sequences, resulting in a total of 25 training sets. For the kringle proteins, the training sets are of sizes 2 and 4, yielding only 10 distinct training sets. Finally, for each series of nested training sets, an independent test set is constructed, consisting of all sequences that do not appear in the training sets. For the 4Fe-4S ferredoxins, these test sets are drawn from the set of 70 purged family members; for the kringle

Motif discovery (MEME)	
Number of motifs	6
Minimum motif width	12
Maximum motif width	55
Motif occurrence model	TCM
Prior	Mega-prior
Motif order and spacing (MAST)	
Database	training set
p-value threshold	0.0001
Model building (mhmm and mhmmt)	
Number of motifs	6
Topology	complete
States per spacer	1 (Viterbi) or 3 (total probability)
Spacer emission distribution	Frequencies from NRDB
Training	none, emissions, transitions or both
EM iterations	20
Homology detection (mhmms)	
Scoring	Viterbi or total probability log-odds
Background model	Frequencies from NRDB
Explicit length modeling	With trained models only

Table IV.14: **Meta-MEME parameter settings.** See text for a more complete description.

domain proteins, the test sets are drawn from the entire set of 38 proteins.

For each training set, a completely connected motif-based hidden Markov model is constructed. Six motifs are discovered by MEME using the most general model of motif occurrences, which allows a motif to occur multiple times in a single sequence. Motifs lengths can range between 12 and 55 amino acids, and the probability distributions at each position are estimated using Dirichlet mixture priors [32] modified by the Mega-prior heuristic [9]. The motif-based HMM is constructed from all six motifs, and motif occurrence information from a MAST analysis of the training set is used to initialize the transition matrix in the HMM, as described in Section II.C. All motif occurrences with p-values less than 0.0001 are included in this analysis.

Each inter-motif spacer region is modeled with three tied HMM states, using emission probabilities taken from the NCBI non-redundant protein database [53]. These parameters are summarized in Table IV.14.

Homology detection is performed on the SWISS-PROT database, using either version 33 or version 28. Version 33 contains 52 205 sequences comprising 18.5 million amino acids, including 159 4Fe-4S ferredoxins and 38 kringle domain proteins; version 28 contains 36 000 sequences, including 86 4Fe-4S ferredoxins and 30 kringle domain proteins. Each sequence is scored using either the total probability log-odds score or the Viterbi log-odds score. Models that have been trained include a Gaussian model of sequence length, so the homology scoring for these models includes the length component, as described in Section II.J.

Homology detection results from different training sets are compared using the ROC_{50} score [58]. This score, which was described in Section III.B, is the area under a curve that plots, for various classification thresholds, the true positives versus the false positives, up to the first fifty false positives. ROC_{50} scores are normalized to range from 0 to 1, with 1 corresponding to perfect separation of family members from non-family members. When computing the ROC_{50} scores, sequences that are members of the original training set are discarded.

The statistical significance of differences in performance is measured by a paired t test. In the results that follow, a difference is called *significant* if it reaches a 1% confidence level, *slightly significant* if it reaches a 5% confidence level, and *not significant* if it fails to reach a 5% confidence level. Unless otherwise stated, the significance tests are conducted using all training sets and so have 24 degrees of freedom for the 4Fe-4S ferredoxins and 9 degrees of freedom for the kringle domain proteins.

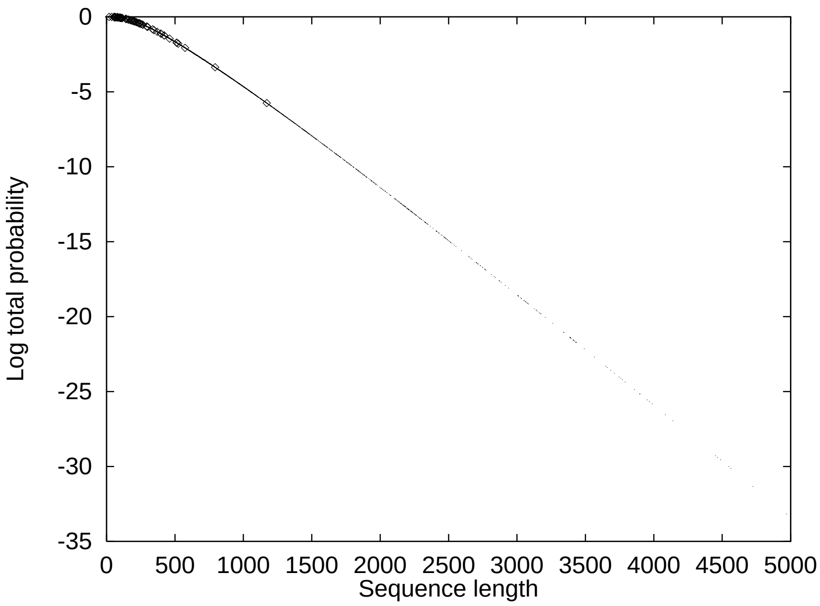
IV.D.3 Results

We find that Meta-MEME succeeds in accurately learning the features of both the 4Fe-4S ferredoxin and kringle domain proteins, but that this learning does not necessarily lead to improved homology detection performance. We begin by examining the log-odds scoring of the 4Fe-4S ferredoxins and show that total probability log-odds and Viterbi log-odds scores provide reasonable separation of family members from non-family members, but that the total probability log-odds scores are more accurate with respect to the theoretical classification threshold described in Section II.I. Next we show that training the model parameters results in improved total probability log-odds scores of an independent test set of positive examples. However, using the trained models to search the SWISS-PROT database for homologs results in decreased performance relative to the untrained model. Indeed, for the kringle domain proteins, the performance of Meta-MEME is worse than that of MAST, which does not exploit information about motif order and spacing.

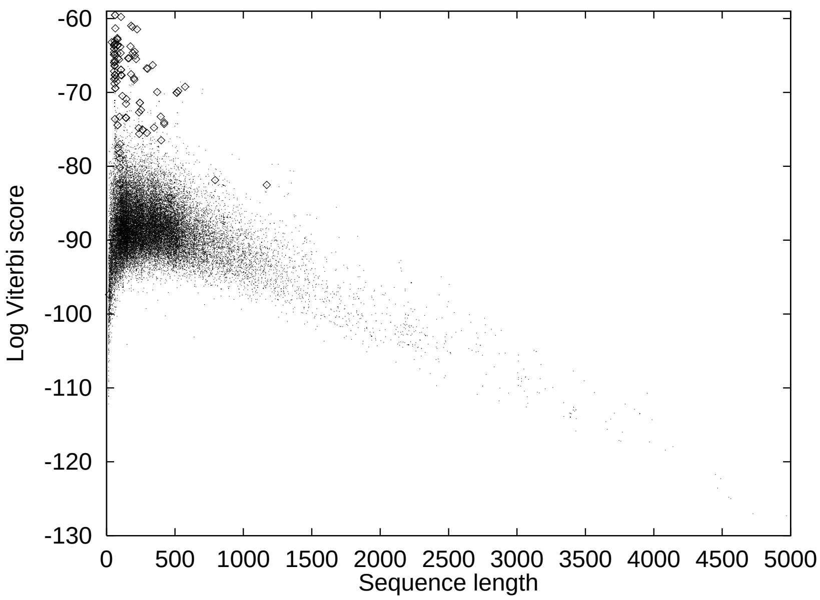
Log-odds scores

Figure IV.12 illustrates the strong length dependence of the raw total probability and Viterbi scores returned by a hidden Markov model. The figure was generated using a single HMM trained on 32 randomly selected 4Fe-4S ferredoxin sequences. Each point in the figure represents the score generated by one sequence in the SWISS-PROT 28 database. Members of the 4Fe-4S ferredoxin family are marked with larger points. The length dependence of the total probability scores is most evident. The family and non-family members cluster so closely together that simply correcting for the length dependence would fail to separate family members from non-family members. The separation provided by Viterbi scoring is better but still not good. Note also that all of the scores reported are less than zero, and that the Viterbi scores are particularly low.

The scaling and separation of total probability scores improves considerably after the scores have been converted to log-odds. Figure IV.13(a) shows the total probability log-odds scores from the same model and database. Although a number of clear false positive and false negative sequences appear, there is a separation between family members and non-family members that was not apparent from the raw total

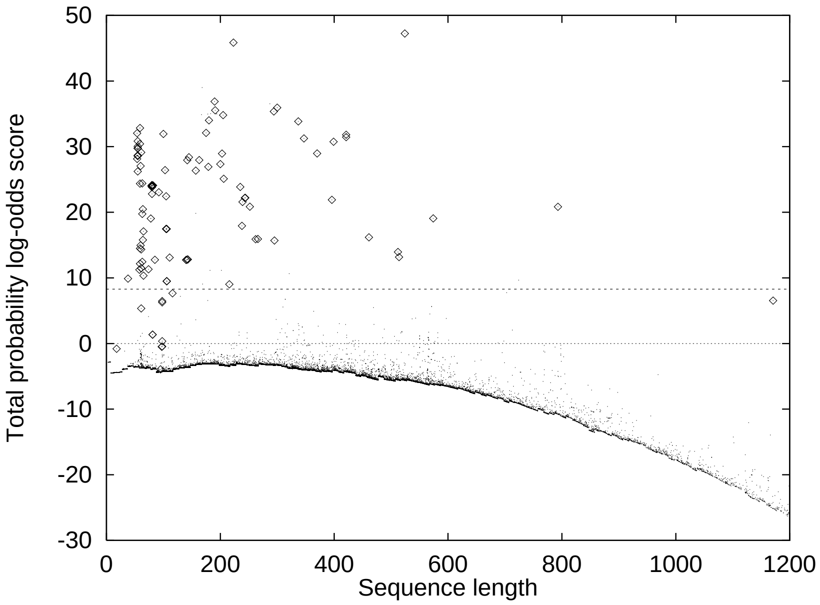


(a)

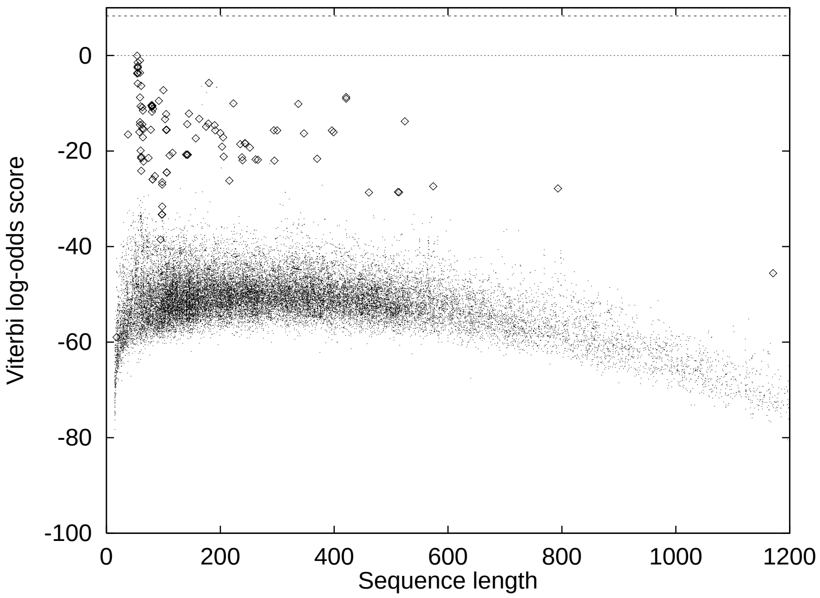


(b)

Figure IV.12: **Length dependence of HMM total probability and Viterbi scores for the 4Fe-4S ferredoxins.** Each point corresponds to one sequence in the SWISS-PROT 28 database. Members of the 4Fe-4S ferredoxin family are marked with larger points.



(a)



(b)

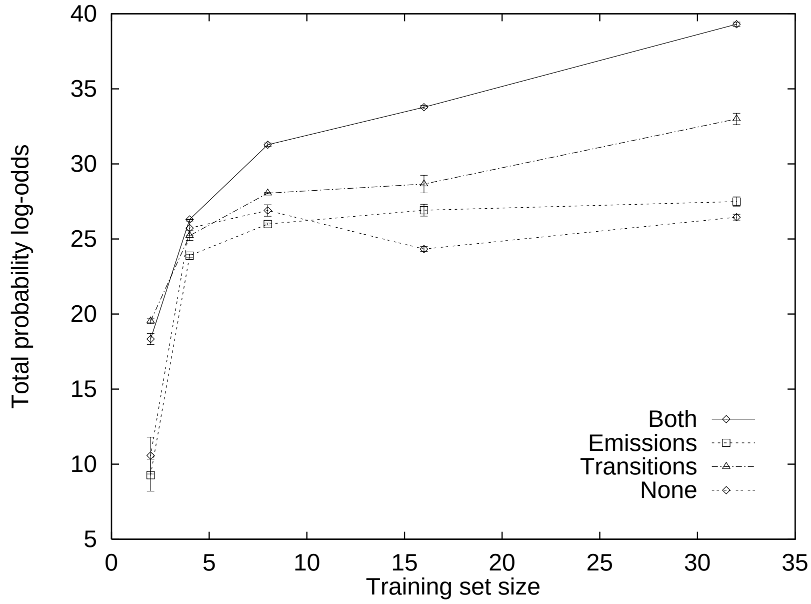
Figure IV.13: **Decreased length dependence and improved scaling of log-odds scores.** These data are similar to those in Figure IV.12, except that the scores have been converted to log-odds using a uniform background model, as well as foreground and background length models. The theoretical classification threshold is shown as a horizontal line at 8.71 bits.

probability scores. The curved baseline of the scores is a result of the Gaussian length modeling. As shown previously in Figure II.15, this length modeling eliminates a number of long false positive sequences, although it does introduce a false negative sequence at length 1171. The theoretical classification threshold for this family is given by Equation II.28. The database contains 36 000 sequences, of which 116 belong to this family, leading to a threshold of $log_2(36\ 000/116) = 8.28$ bits. This threshold is shown in the figure, and leads to a classification with 12 false negatives and 16 false positives. The equivalence score [105], which is the classification threshold yielding an equal number of false positives and false negatives, falls at 9.49 bits and gives 13 false positives.

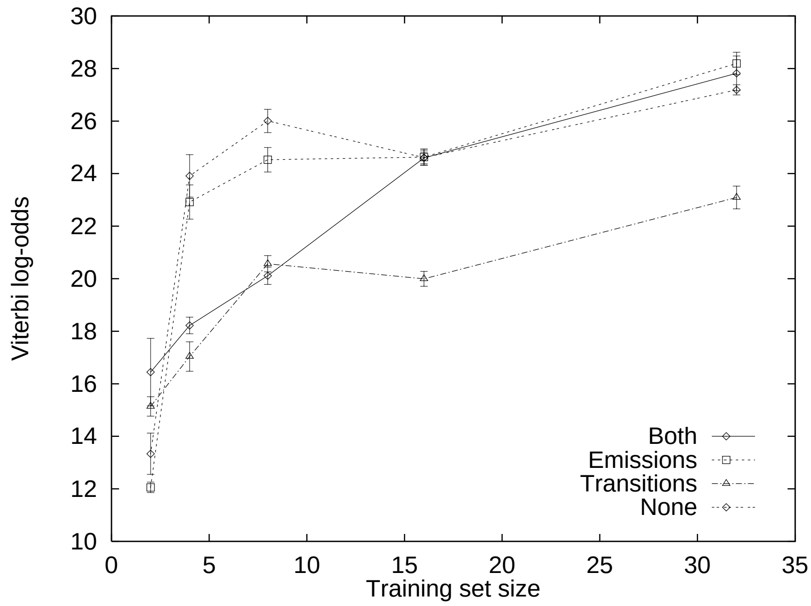
The scaling of Viterbi scores via the log-odds calculation is much less successful than was the scaling of total probability scores. As shown in Figure IV.13(b), no sequence receives a Viterbi log-odds score greater than zero, although these scores are much greater than the raw scores shown in Figure IV.12(b). On the other hand, the separation provided by the Viterbi log-odds scores is only slightly worse than that provided by the total probability log-odds scores. The equivalence score of -27 bits yields 11 false positives. This separation is also better than that of the raw Viterbi scores, which, with a threshold of -71.1 bits, yield 33 false positives.

Training the HMM

Figure IV.14(a) verifies that training a Meta-MEME model improves the model's ability to characterize previously unseen family members. The figure plots the average total probability log-odds score of a series of independent sets of 38 4Fe-4S ferredoxins as a function of training set size. Not surprisingly, the ability of the models to characterize the test set increases as the training set size increases. Furthermore, models that have been trained by Meta-MEME show improved characterization of the test set relative to the untrained models. The bottom series in the figure, labeled "None," represents the scores generated by HMMs built directly from MEME motifs with no training by Meta-MEME. After training either the emission probabil-



(a)



(b)

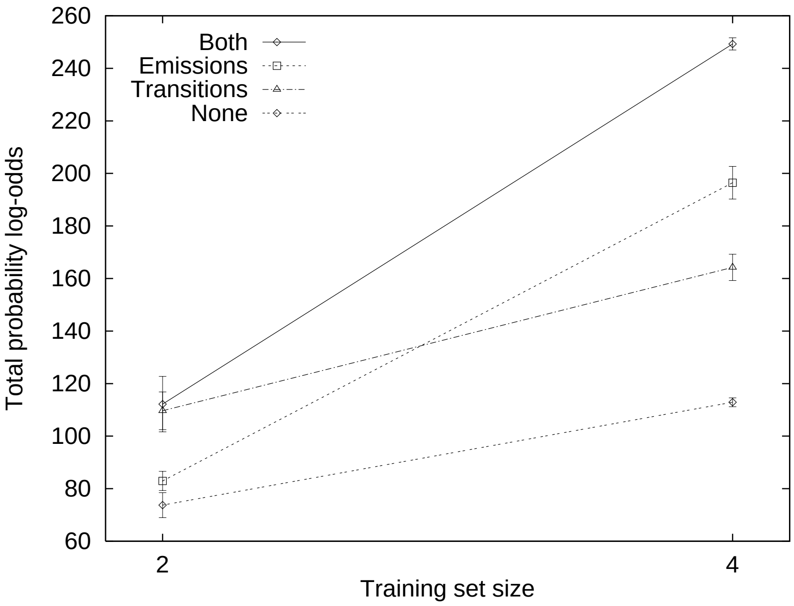
Figure IV.14: **Characterization of 4Fe-4S ferredoxins after HMM training.** The figure plots the average (a) total probability log-odds score and (b) Viterbi log-odds score of a series of independent test sets of 38 4Fe-4S ferredoxins. Scores are computed with respect to motif-based HMMs trained on nested ferredoxin training sets of various sizes. Each point represents an average over five randomly selected training sets, and error bars represent standard error. The series labels indicate which parameters of the HMM were trained: both sets of probability distributions, emission probabilities only, transition probabilities only, or no HMM training.

ity distributions or the transition probability distributions, the scores of the test set increase. This increase is even more pronounced when the two types of probability distributions are trained at once. For training sets of size 32, all differences apparent in Figure IV.14(a) are statistically significant at the 1% confidence level, according to a paired t test with four degrees of freedom. Thus, during training the model acquires features of the training set that characterize the family as a whole. Furthermore, the benefit provided by Meta-MEME training increases as the training set size increases.

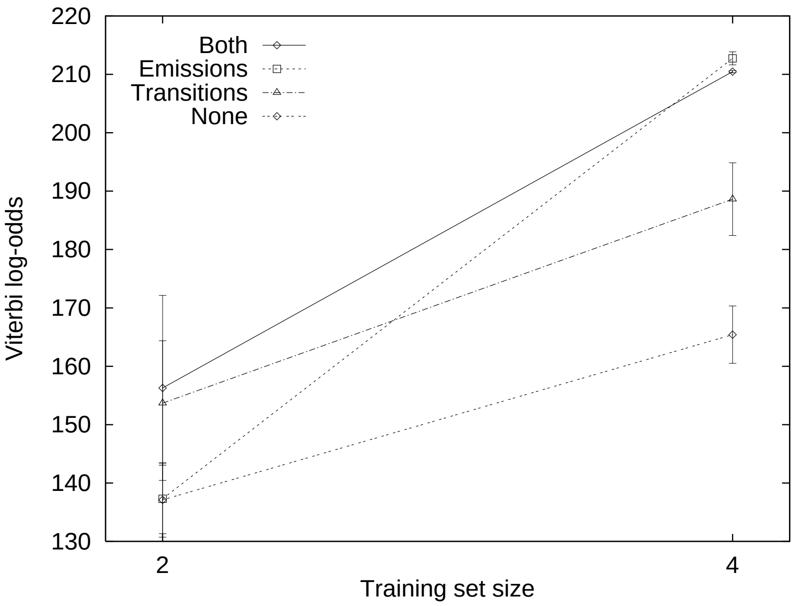
Because the Baum-Welch training algorithm maximizes the total probability of the model, given the data, it is unsurprising that training causes an increase in the total probability of the test set. Figure IV.14(b) shows, however, that this increase is not accompanied by a corresponding increase in Viterbi score. The figure plots average Viterbi log-odds score for the same series of independent test sets used to generated Figure IV.14(a). Training the transition probabilities, in particular, causes a decrease in the probability of the Viterbi path: the scores generated by the transition-trained models are significantly lower than scores generated by any of the other types of models. The other trained models score no better, and for some training set sizes significantly worse, than the untrained models.

For the kringle domain proteins, the same increase in total probability log-odds scores after training is observed. In Figure IV.15(a), models of kringle domain proteins are used to score independent test sets. Once again, training either set of probability distributions results in improved total probability log-odds scores. The improvement is most striking, however, when both sets of distributions are trained. Unlike for the 4Fe-4S ferredoxins, however, this improvement carries over to the Viterbi log-odds scores. Figure IV.15(b) shows that the untrained models produce lower Viterbi log-odds scores than do any of the trained models. The difference in performance is significant with respect to the models with trained transitions and slightly significant with respect to the other two types of trained models.

The increase in total probability log-odds scores after training can be explained in two ways: either the models are acquiring features of the family through



(a)



(b)

Figure IV.15: **Improved characterization of kringle domain proteins after HMM training.** The figure plots the average (a) total probability log-odds and (b) Viterbi log-odds score of a series of independent test sets of 34 kringle domain proteins. Scores are computed by motif-based HMMs trained on nested training sets of various sizes. Each point represents an average over five randomly selected training sets, and error bars represent standard error. The series labels indicate which parameters of the HMM were trained: both sets of probability distributions, emission probabilities only, transition probabilities only, or no HMM training.

	Score	None	Emissions	Transitions	Both
Fer4	total	20.9	26.2	32.4	38.0
Non-fer4	total	-6.4	-6.7	-19.7	-16.5
Difference	total	27.3	32.9	52.1	54.5
Fer4	Viterbi	-16.4	-13.6	-12.4	-8.7
Non-fer4	Viterbi	-53.0	-52.4	-52.3	-46.4
Difference	Viterbi	36.6	38.8	39.9	37.7

Table IV.15: **Improvement of average scores assigned to family members versus scores assigned to non-family members.** The rows marked "Fer4" show the average score assigned to members of the 4Fe-4S ferredoxin family; the "Non-fer4" rows show average scores for all other sequences in SWISS-PROT version 28. "Difference" rows contain the difference between the previous two rows. Scores labeled "total" are total probability log-odds scores; scores labeled "Viterbi" are Viterbi log-odds scores. All scores include an explicit length model and are generated by variously trained versions of a single, motif-based HMM trained on the same set of 32 randomly selected 4Fe-4S ferredoxins.

training, or the models are acquiring features of proteins in general. The latter case would result in trained models that are "flatter" than the original models and hence assign higher scores to all sequences. Clearly, such behavior is not desirable. In order to show that this kind of flattening is not occurring, we examine a particular model (the same one used to generate Figures IV.12-IV.13), looking at the average score that the model assigns to family members and non-family members in the database. Table IV.15 summarizes the results. For both types of scores, the average score assigned to family members increases with model training; for non-family members, on the other hand, this increase does not occur. Instead, in nearly every case, the average score of non-family members remains approximately the same or decreases. For both types of scores, the difference between family and non-family member average scores is greater for trained models than for untrained models, indicating that training is increasing the discriminative ability of the model.

Training may also cause the total probability scores to increase because the models are learning features of a larger superfamily of proteins, of which the training set represents only a small part. Such a hypothesis is reasonable, especially for large-

Family	Score	None	Emissions	Transitions	Both
4Fe-4S ferredoxin	total	20.9	26.2	32.4	38.0
2Fe-2S ferredoxin	total	-4.0	-2.6	-15.5	-12.2
Adrenodoxin	total	-3.8	-3.7	-13.5	-12.0
Flavodoxin	total	-10.3	-10.9	-26.0	-21.7
Rubredoxin	total	-6.2	-6.2	-21.6	-18.3
Photosystem 1	total	-18.0	-19.4	-24.8	-26.7
Photosystem 2	total	-6.2	-5.9	-18.8	-15.2
4Fe-4S ferredoxin	Viterbi	-16.4	-13.6	-12.4	-8.7
2Fe-2S ferredoxin	Viterbi	-46.4	-37.9	-50.0	-43.8
Adrenodoxin	Viterbi	-46.5	-45.2	-45.3	-40.4
Flavodoxin	Viterbi	-57.6	-57.5	-58.7	-51.9
Rubredoxin	Viterbi	-51.4	-49.2	-56.0	-48.5
Photosystem 1	Viterbi	-65.0	-63.8	-58.3	-59.6
Photosystem 2	Viterbi	-54.4	-51.2	-51.6	-45.2

Table IV.16: **Change in average scores of proteins families related to the 4Fe-4S ferredoxins.** Lists of sequences for each family are taken from PROSITE version 13.0. The two photosystem families represent two different signature motifs.

scale features such as the order and spacing of motifs, since these features may be conserved among very remotely related homologs. In order to determine whether training is learning features of a larger set of proteins, we examine the average scores of five families that are known to be related to the 4Fe-4S ferredoxins. These families include the 2Fe-2S ferredoxins, the adrenodoxin subfamily of the 2Fe-FS ferredoxins, two families that are functionally interchangable with ferredoxins (flavodoxins and rubredoxins), and the photosystem I complex, which contains a 4Fe-4S iron-sulfur center. Table IV.16 shows how the average score for each of these families changes after training.

For each related family, completely training the model causes a decrease in average total probability log-odds score and an increase in average Viterbi log-odds score. This trend agrees with the data in Figure IV.14, which suggests that training leads to improved characterization by total probability scores but not by Viterbi scores. Overall, therefore, we expect that the homology detection performance of trained models should increase relative to untrained models when total probability

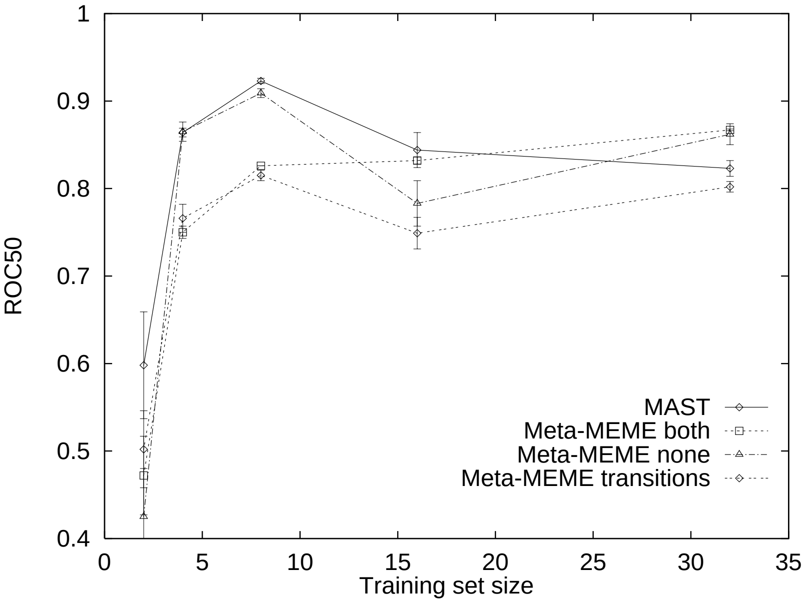


Figure IV.16: **Homology detection performance on 4Fe-4S ferredoxins using total probability log-odds scoring.** The figure plots ROC_{50} score as a function of training set size for homology detection searches in the SWISS-PROT database, version 33. Each point represents an average over five randomly selected training sets, and error bars represent standard error. The three Meta-MEME series represent results from untrained models, models with trained transition probabilities, and completely trained models (i.e., trained transition and emission probabilities).

log-odds scoring is employed, but that the converse will occur when Viterbi log-odds scoring is used. In the following section, we examine this hypothesis.

Homology detection

Despite the apparent improvement in Meta-MEME's models after training, as measured by total probability log-odds scores, the trained models fail to detect homologs as well as the untrained models using either type of scoring. Figures IV.16 and IV.17 summarize the results of homology detection experiments using trained and untrained Meta-MEME models. The variance in performance for the smallest training sets is too large to allow differentiation between techniques. However, for larger training sets, the untrained models outperform models in which the transition probabilities have been trained, as well as models in which all of the parameters have been trained. Using total probability log-odds scoring (Figure IV.16), this difference

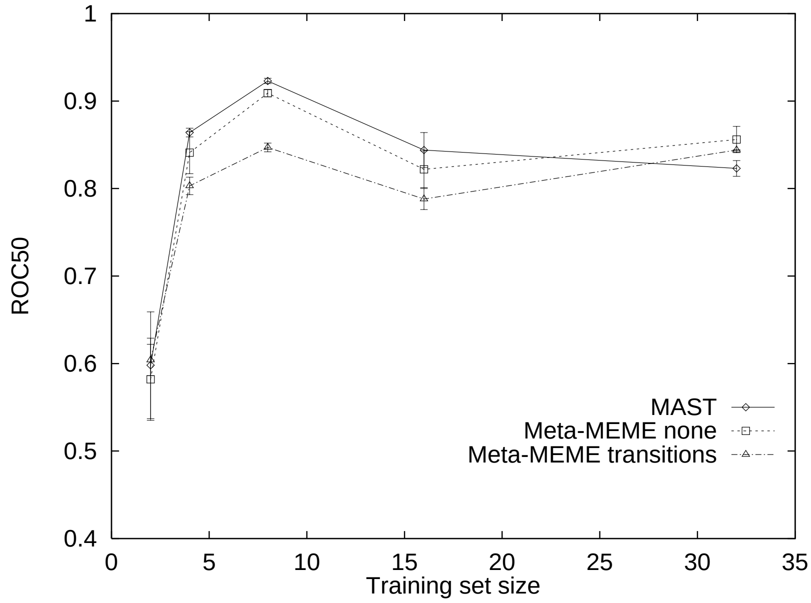


Figure IV.17: **Homology detection performance on 4Fe-4S ferredoxins using Viterbi log-odds scoring.** The figure plots ROC_{50} score as a function of training set size for homology detection searches in the SWISS-PROT database, version 33. Each point represents an average over five randomly selected training sets, and error bars represent standard error.

in performance is slightly significant using 4-sequence training sets and significant using 8-sequence training sets. The differences in performance between the two types of trained models are not consistent and are not statistically significant. Using Viterbi log-odds scoring (Figure IV.17), the difference in performance between the trained and untrained models, although consistently in favor of the untrained models, is not statistically significant, except when using training sets of size 8. Overall, it appears that training the HMMs decreases the models' ability to discriminate between family members and non-family members.

This decrease in performance is a result, in part, of the decreased discriminative ability of the trained motifs. The motif models from a completely trained Meta-MEME HMM were extracted and used as inputs to MAST. The MAST ROC_{50} score from the original motif models was 0.767. The score of the same models after training by Meta-MEME is 0.541, considerably lower than the original score. This indicates that the training algorithm results in motif models with decreased discriminative capacity.

Figures IV.16 and IV.17 also illustrate the benefit of Viterbi log-odds scoring versus total probability log-odds scoring. The performance of untrained Meta-MEME models using Viterbi scoring is slightly significantly better than that given by total probability scoring.

Finally, Figures IV.16 and IV.17 show that Meta-MEME fails to provide improved homology detection performance relative to that of MAST. One of Meta-MEME's primary goals is to exploit information about the order and spacing of motifs within a family, thereby improving homology detection performance relative to a tool such as MAST, which treats motifs independently. The figures indicate that this goal is not being attained. The difference in performance between Meta-MEME's untrained models and MAST is not significant when either type of log-odds scoring is used. After Meta-MEME training, MAST outperforms Meta-MEME. Of the three experiments involving trained HMMs in Figures IV.16 and IV.17, MAST performs significantly better than Meta-MEME using either type of scoring, although the difference in performance for Viterbi scoring is only slightly significant. Thus, for the 4Fe-4S ferredoxins, Meta-MEME's homology detection performance is as good as MAST's only when untrained Meta-MEME models are used.

Similar results are provided by the homology detection experiments using the kringle domain proteins. Figure IV.18 shows that, as for the 4Fe-4S ferredoxins, training the Meta-MEME models results in deterioration of the homology detection performance, as measured by ROC_{50} scores. This deterioration is not significant, due to the small number of samples available. In addition, neither the trained nor the untrained models performs as well as does MAST. MAST performs significantly better than the trained models and slightly significantly better than the untrained models.

A major difficulty involved in testing any homology detection program arises in defining the gold standard list of family members. Precisely defining a particular protein family sometimes requires that fairly arbitrary decisions be made about

Total	Viterbi	Length	ID	Description
166.27	-39.76	822	ANP_NOTCO	Antifreeze glycopeptide polyprotein
66.63	2.81	194	FRXB_PLEBO	FRXB protein
57.70	-5.32	168	YOJG_ECOLI	Hypothetical
54.08	17.10	55	FER_RHORU	Ferredoxin
54.04	-10.24	193	NUYM_SYNY3	NADH-plastoquinone oxido-reductase subunit NDHI
54.03	-7.99	167	FRXB_TOBAC	FRXB protein
53.94	-16.08	252	YCR2_BACTK	Hypothetical
46.37	-9.13	178	FRXB_ORYSA	FRXB protein
45.54	-10.49	183	FRXB_MARPO	FRXB protein
44.86	-10.97	176	FRXB_WHEAT	FRXB protein
35.39	-47.39	416	APEG_XENLA	APEG protein precursor (fragment)
31.02	-49.67	865	CPN_DROME	Calphotin
30.70	-20.79	157	YOJB_ECOLI	Hypothetical
25.46	-40.94	287	YEJZ_ECOLI	Hypothetical
22.95	-45.12	420	ZG58_XENLA	Gastrula zinc finger (fragment)
22.49	-45.16	337	ZG26_XENLA	Gastrula zinc finger (fragment)
19.68	-20.27	66	FER_PYRFU	Ferredoxin
19.65	-43.95	543	SRTX_ATREN	Sarafotoxins precursor
17.58	-44.39	453	ZO6_XENLA	Oocyte zinc finger protein (fragment)
16.29	-31.20	130	YOJA_ECOLI	Hypothetical

Table IV.17: **False positive 4Fe-4S ferredoxin sequences using total probability log-odds scoring.** Listed are the twenty non-4Fe-4S ferredoxin sequences from SWISS-PROT version 28 that receive the highest total probability log-odds scores from a completely connected Meta-MEME model that has been completely trained using a set of 32 randomly selected divergent sequences.

Total	Viterbi	Length	ID	Description
54.08	17.10	55	FER_RHORU	Ferredoxin
66.63	2.81	194	FRXB_PLEBO	FRXB protein
57.70	-5.32	168	YOJG_ECOLI	Hypothetical
54.03	-7.99	167	FRXB_TOBAC	FRXB protein
46.37	-9.13	178	FRXB_ORYSA	FRXB protein
54.04	-10.24	193	NUYM_SYNY3	NADH-plastoquinone oxido-reductase subunit NDHI
45.54	-10.49	183	FRXB_MARPO	FRXB protein
44.86	-10.97	176	FRXB_WHEAT	FRXB protein
53.94	-16.08	252	YCR2_BACTK	Hypothetical
19.68	-20.27	66	FER_PYRFU	Ferredoxin
30.70	-20.79	157	YOJB_ECOLI	Hypothetical
2.09	-22.10	93	GLHA_MURCI	Glycoprotein hormones α chain
1.77	-24.43	154	YR7E_ECOLI	Hypothetical
-1.94	-26.13	171	ATDA_HUMAN	Diamine acetyltransferase
1.39	-26.34	48	SIA2_SORBI	Small protein inhibitor of insect α-amylases 2
4.84	-26.40	540	KER2_CHICK	C-ERBB proto-oncogene tyrosine kinase
3.36	-26.47	60	CX1_NAJHA	Cytotoxin 1 (toxin V-II-1)
15.67	-26.94	169	ZG62_XENLA	Gastrula zinc finger protein (fragment)
2.92	-26.99	60	CX1_NAJNI	Cytotoxin 1 (toxin V-II-1)
-3.25	-27.19	171	ATDA_MESAU	Diamine acetyltransferase

Table IV.18: **False positive 4Fe-4S Ferredoxin sequences using Viterbi log-odds scoring.** Listed are the twenty non-4Fe-4S ferredoxin sequences from SWISS-PROT version 28 that receive the highest Viterbi log-odds scores from a completely connected Meta-MEME model that has been completely trained using a set of 32 randomly selected divergent sequences.

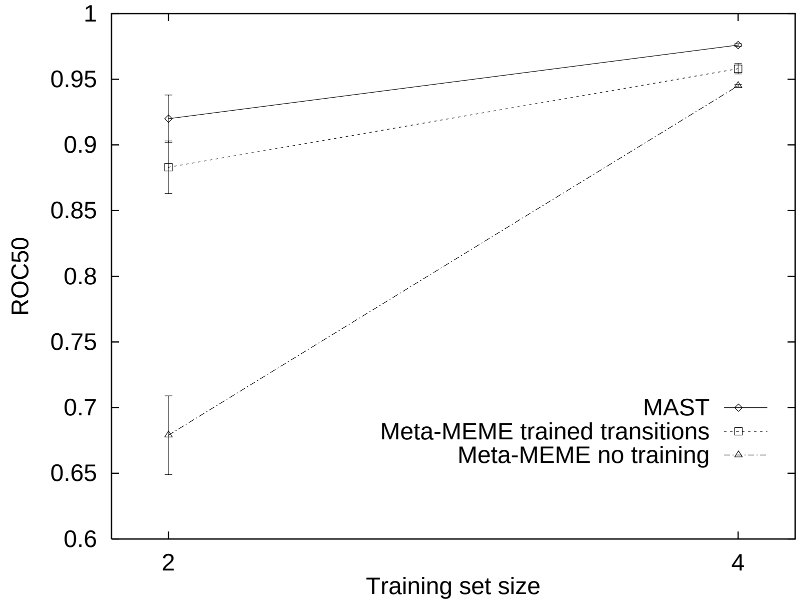


Figure IV.18: **Homology detection performance on kringle proteins using total probability log-odds scoring.** The figure plots ROC_{50} score as a function of training set size for homology detection searches in the SWISS-PROT database, version 33. Each point represents an average over five randomly selected training sets, and error bars represent standard error.

the boundaries of the family. In the case of the 4Fe-4S ferredoxins, the defining characteristic—having a 4Fe-4S binding site—is clear; however, there is no guarantee that the non-binding-site features of the 4Fe-4S ferredoxins will be equally clearly delineated. Thus, the distance in feature space between some pairs of 4Fe-4S ferredoxins may be greater than the corresponding differences between some pairs of 4Fe-4S ferredoxins and 2Fe-2S ferredoxins. Tables IV.17 and IV.18 list the twenty highest-scoring false positive sequences from a single, completely trained 4Fe-4S ferredoxin model, as measured by total probability log-odds scores and Viterbi log-odds scores. The two lists of false positives overlap by ten sequences, many of which are clearly related to the 4Fe-4S ferredoxins. For example, FER_PYRFU is described as a 3Fe-4S ferredoxin [48]. Also, the FRXB proteins contain iron-sulfur centers similar to those of 'bacterial-type' 4Fe-4S ferredoxins [48]. Perhaps these sequence properly belong in the 4Fe-4S ferredoxin family, in which case the discrimination provided by Meta-MEME is nearly perfect.

IV.D.4 Discussion

The experiments described in this section explore Meta-MEME's ability to model families containing repeated elements. The results show that Viterbi log-odds scoring, although it does not scale well with respect to the theoretical classification threshold, provides improved discrimination relative to total probability log-odds scoring. Unfortunately, this discrimination does not improve with training, most likely because the expectation-maximization training algorithm maximizes the total probability rather than the Viterbi score.

The improper scaling of Viterbi log-odds scores can be explained by considering the scoring performed by the background model. These experiments clearly show that log-odds scoring provides improved separation of family members from non-family members relative to raw scores, especially for total probability scoring. The log-odds scores are also fairly accurately distributed with respect to the theoretical classification threshold, but this accuracy only holds for total probability scoring. When Viterbi scoring is employed, the scores of family members are uniformly much less than zero (on the order of -20 bits). This reflects a huge difference in score, indicating that the foreground model of the family is 2^{20} times less likely to have generated the sequence than the generic background model.

The foreground model is a motif-based HMM, consisting of transition probability distributions and emission probability distributions. The background model, in contrast, only contains emission probability distributions. The probability associated with a path of length n through this background model, therefore, is the product of n individual emission probabilities. For the foreground model, if the path traverses motifs of total length ℓ, then the probability of the path is the product of n emission probabilities as well as $n-\ell$ transition probabilities. Therefore, even if the foreground path is highly probable, the additional $n - \ell$ probabilities that are included in the computation of this path's probability will decrease the magnitude of that probability. In essence, the skewed Viterbi log-odds scores arise because of a difference in scaling of the foreground model scores and the background model scores. This difference, in turn, arises because only the foreground model includes transition probabilities.

The skew is not as apparent for total probability scores because they are greater than Viterbi scores. The total probability score is a sum of probabilities over all paths through the model. This sum necessarily includes the Viterbi path, so the total probability must be greater than or equal to the probability of the Viterbi path. For the foreground model, many such paths are possible, so the total probability is much greater than the probability of the Viterbi path. The background model, however, has length equal to the given sequence. Hence, there exists only a single path through the background model. Consequently, the Viterbi score and the total probability score generated by the background model are equal. The skew apparent in the Viterbi log-odds scores disappears when total probability log-odds scores are used because, in the calculation of log-odds, the numerator of the fraction (the foreground model score) has increased while the denominator (the background model score) remains constant.

Despite this scaling problem, the experiments described here show that Viterbi log-odds scoring provides improved discrimination, as measured by Meta-MEME's performance on the homology detection task. The difference in performance between Viterbi and total probability scoring reflects the intuition that a well-trained model should find a single, correct path corresponding to the evolutionary history of the matched sequence. In much the same way that a multiple alignment reflects the evolutionary commonalities among a set of sequences, a Viterbi match between a sequence and a model should indicate the sequence's history with respect to the model. A total probability score, because it is the sum of a large number of paths, does not admit of an evolutionary explanation in the same way that the Viterbi score does.

These considerations suggest that the Baum-Welch algorithm may not be the most appropriate training algorithm for protein modeling. The results given here show that Baum-Welch training leads to a deterioration in the discriminative ability of Meta-MEME's models. This deterioration shows that improving the total probability of the sequences, given the model, leads to a decrease in the probability of the most likely path. If, however, the Viterbi score provides superior discrimination, then a training algorithm that maximizes this score [22], rather than the total probability score, would be more appropriate.

The experiments reported here have provided insight into Meta-MEME's ability to model families with repeated elements. The results are somewhat encouraging, indicating that training improves the model's ability to characterize members of the given family, as measured by total probability scores. However, this improvement may come at the cost of decreasing the probability of the Viterbi path, and hence in decreasing the discriminative power of the model. However the results are interpreted, they are not necessarily generalizable because these experiments only examine two protein families. Accordingly, in the following, final experiment, a larger collection of protein families is analyzed in order to determine Meta-MEME's overall effectiveness at modeling protein families.

IV.E A comparison of homology detection methods

IV.E.1 Introduction

In this final set of experiments, we examine more fully Meta-MEME's ability to detect homologs in a large sequence database. Using a set of 73 PROSITE families, we first compare the performance of Meta-MEME's search tool with that of the tool offered by HMMER, which was used in the first version of Meta-MEME (see Section IV.B). We find that the two search tools offer comparable levels of homology detection performance, despite the relative simplicity of the Meta-MEME search tool algorithm.

Second, we examine the two types of Meta-MEME model topologies—linear and completely connected—as well as the two available scoring schemes—Viterbi log-odds scoring and total probability log-odds scoring. The two topologies offer comparable levels of homology detection performance; however, Viterbi log-odds scoring yields significantly better discrimination than total probability log-odds scoring.

Next, we train the Meta-MEME models using expectation-maximization and apply the trained models to the homology detection task. The trained models perform nearly identically to the untrained models, indicating that training has little effect.

Finally, we compare Meta-MEME's performance with that of several other homology detection methods, including standard hidden Markov modeling (HMMER), motif modeling (MEME and MAST), and the Family Pairwise Search (FPS) algorithm described in Chapter III. Unfortunately, Meta-MEME fails to improve significantly upon the performance of the motif modeling software MEME and MAST. Furthermore, although both Meta-MEME and MAST perform significantly better than standard HMMs, none of these three methods performs as well as Family Pairwise Search.

IV.E.2 Methods

For all these experiments, the collection of 73 protein families [10] described in Section III.B is used. These families were selected from the PROSITE database [12] release 13.0 for their difficulty, based upon the number of false positives reported in the PROSITE annotations. The PROSITE IDs and sizes of these families are listed in Appendix A. The associated release of SWISS-PROT [13] (version 28) contains 36 000 sequences and nearly 12.5 million amino acids. As in previous experiments (see Section III.B) binary sequence weighting is carried out with the `purge` program [87]. The sizes of the purged families are given in Appendix A.

For each family, one series of nested subsets is randomly selected, containing 2, 4, 8, 16 and 32 sequences, limited by the total number of divergent sequences in the family. This results 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. In addition, for each family a single, independent

Motif discovery (MEME)	
Number of motifs	10
Minimum motif width	12
Maximum motif width	55
Motif occurrence model	ZOOPS
Prior	Mega-prior
Motif order and spacing (MAST)	
Database	SWISS-PROT 28
p-value threshold	0.0001
Model building (mhmm and mhmmt)	
Number of motifs	see text
Topology	linear or complete
States per spacer	1 (Viterbi) or 3 (total probability)
Spacer emission distribution	Frequencies from NRDB
Training	none, emissions, transitions or both
EM iterations	20
Homology detection (mhmms)	
Scoring	Viterbi or total probability log-odds
Background model	Frequencies from NRDB
Explicit length modeling	With trained models only

Table IV.19: **Meta-MEME parameter settings.** See text for more complete description.

test set is constructed, consisting of all family members not contained in the query sets, with no purging.

For each training set, MEME discovers a set of motifs, and these motif models serve as the basis for several Meta-MEME HMMs. HMMs with a linear topology are built using all motifs that appear with a MAST p-value less than 0.0001 in more than half the training sequences, up to a maximum of ten motifs. This "majority occurrence heuristic" eliminates from the model motifs that represent subfamilies of the training set. For models with a completely connected topology, such subfamily motifs are not necessarily detrimental, since sequences that don't include one or more of the motifs can match to the model via an alternate path. Therefore, all of the first six motifs discovered by MEME are included in the completely connected models.

Two versions of each model are built, one with three states representing each spacer, and one with a single state representing each spacer. The multi-state spacers, as described in Section II.D, approximate a normal distribution of spacer lengths by summing the exponential distributions from each state. However, because the Viterbi algorithm considers only a single path through the model, this summing does not occur during the computation of Viterbi scores. Therefore, the model with three-state spacers is only used for computing total probability scores; single-state models are used to compute Viterbi scores.

Finally, each model is trained in three different ways: training the transition probability distributions, the emission probability distributions, or both sets of distributions. Including the untrained models, this procedure results in a set of sixteen models (two topologies, two spacer representations, and four types of training) for each training set of sequences. These parameters are summarized in Table IV.19.

Meta-MEME's performance is compared with three other homology detection methods: HMMER [46], MAST [11] and the Family Pairwise Search (FPS) algorithm described in Chapter III. For each training set, HMMER version 1.8 is used to train a standard hidden Markov model via expectation-maximization coupled with simulated annealing. The default geometric annealing schedule is used, and Dirichlet mixture priors are used in order to allow the models to be trained with smaller training sets. Database searches are conducted using `hmmsw`, which implements 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. For each database sequence, the program returns a log-odds scores in bits. MAST searches are carried out using the first six motifs discovered by MEME. Each motif model is used to score each database sequence, and these scores are subsequently combined into an overall E-value for that sequence. The Family Pairwise Search algorithm uses gapped BLAST [2] to compute a bit score for each sequence in the database with respect to each training set sequence. The final score of the database sequence is the average of the individual training set scores.

Each homology detection experiment returns a score-labeled version of the database. The database is then sorted according to these scores, and each sequence in the sorted database is marked with a "1" or a "0," indicating whether that sequence appears in the PROSITE listing for the current family. In order to test the ability of the homology detection algorithms to generalize from the query set, all family members that do not appear in the independent test set are eliminated from the sorted list. The resulting, purged sequence of bits represents the homology detection algorithm's ability to separate novel family members from non-family members. Perfect performance corresponds to a series of 1s followed by a series of 0s.

This bit sequence is subjected to two forms of analysis. The first is a modified version of the Receiver Operating Characteristic, called ROC_{50} [58]. The ROC score is the area under a curve that plots true positives versus false positives 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 to 1, with 1 corresponding to the most sensitive and selective search.

In addition to ROC_{50} analysis, each homology detection method is evaluated using the normalized equivalence number [105]. The equivalence number is the number of false positives given by a database search when the classification threshold is set so that the number of false positives equals the number of false negatives. To compute the equivalence number from the sequence of bits described above, a mark is moved along the sequence until the number of 0s to the left of the mark equals the number of 1s to the right. Perfect separation corresponds to an equivalence number of 0, and the maximum possible equivalence number is the size of the family. In the results reported here, equivalence numbers are scaled to range from 0 to 1 by dividing by the size of the family. This allows equivalence numbers from homology searches for variously sized families to be combined.

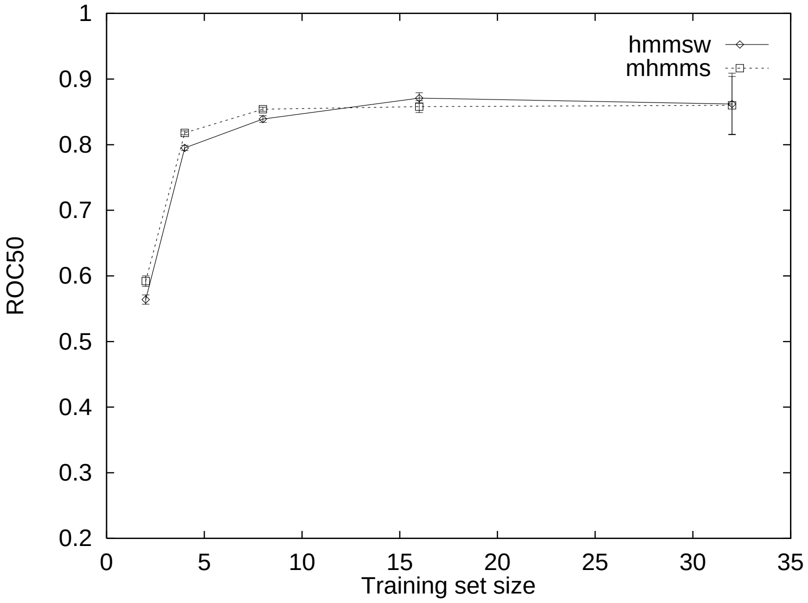
The statistical significance of differences in performance is measured by a paired t test. In the results that follow, a difference is called *significant* if it reaches a 1% confidence level and *not significant* if it fails to reach a 5% confidence level. Unless otherwise stated, the significance tests are conducted using all 184 training sets and so have 183 degrees of freedom.

IV.E.3 Results

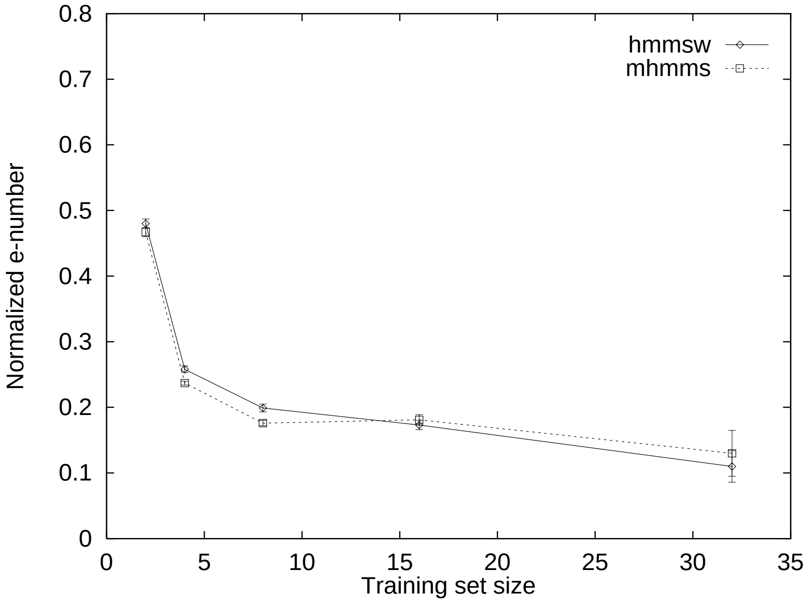
The Meta-MEME search tool

The first version of Meta-MEME, described in Section IV.B, created linear, motif-based hidden Markov models that can be represented as constrained versions of the standard linear HMM topology (see Figure II.2). This topology allowed Meta-MEME to create models in a format readable by the HMMER software [46], thereby providing Meta-MEME with access to the suite of sophisticated search tools available in HMMER. These tools include, most importantly, the program `hmmsw`, which implements a modified version of the Smith-Waterman algorithm tailored to hidden Markov models. Because the Smith-Waterman algorithm is a local search technique, `hmmsw` can assign relatively high scores to sequence fragments, even if the fragment matches only a portion of the hidden Markov model. This ability is very useful in searching real protein databases, since such databases generally contain a large percentage of fragmentary sequences. Indeed, informal experiments with the four search tools offered by HMMER consistently indicate that `hmmsw` offers the best homology detection performance.

In creating Meta-MEME version 2.0, it was necessary to depart from the HMMER HMM format in order to allow models with a completely connected topology. Thus, for these models, Meta-MEME 2.0 loses the ability to use the HMMER search tools. Meta-MEME's own search tool, `mhmms`, implements relatively simple algorithms—the Viterbi algorithm or the forward algorithm (see Section II.I)—neither



(a)



(b)

Figure IV.19: **Relative performance of the Meta-MEME and HMMER search tools.** Figure (a) plots average ROC_{50} score as a function of training set size for all 73 families in the study. Figure (b) plots average normalized e-number for the same families. Error bars represent standard error.

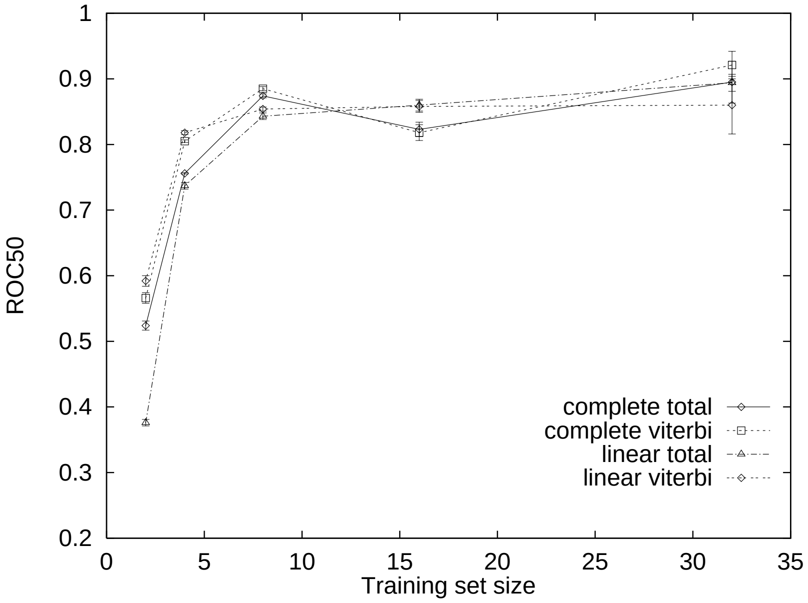
of which explicitly attempts to account for sequence fragments in the database.

Figure IV.19 shows that the change from `hmmsw` to `mhmms` does not significantly impact Meta-MEME's performance. The figure uses both the ROC_{50} metric and the normalized equivalence number to compare the homology detection performance of `hmmsw` with that of the Meta-MEME search tool `mhmms`. For each series the same set of untrained, linear, motif-based hidden Markov models is employed. Using either metric, neither search tool consistently outperforms the other, although both metrics assign `mhmms` a small but statistically significant improvement over `hmmsw`. The statistical significances of the differences between homology detection methods for these and the following experiments is summarized in Tables IV.20-IV.22. The current results indicate that the Meta-MEME search tool is only slightly more effective than the HMMER search tool in locating homologous sequences.

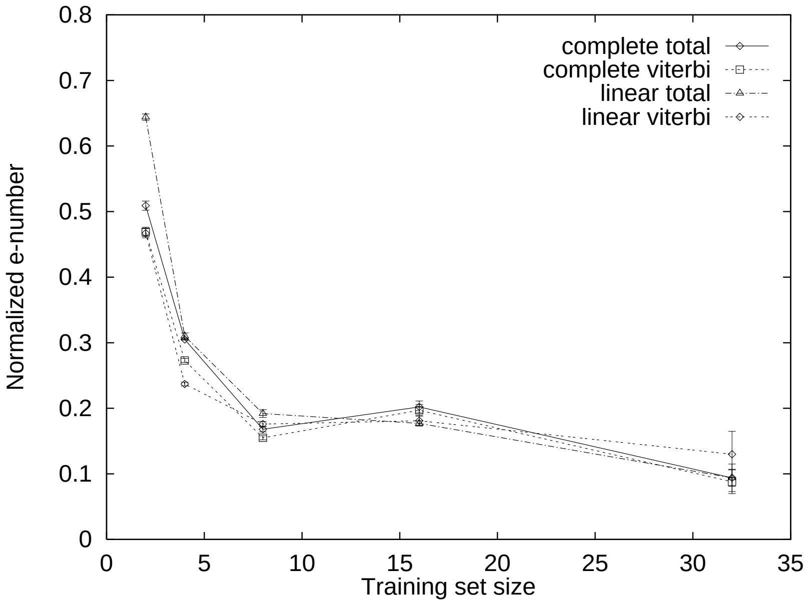
Model topologies and scoring schemes

Having established the effectiveness of `mhmms`, we next compare the performance of the two model topologies offered by Meta-MEME. Figure IV.20 compares the performance of linear and completely connected HMMs using Viterbi log-odds scoring and total probability log-odds scoring. No clear trend emerges. For Viterbi log-odds scoring, although the linear models provide a slight but significant performance advantage, as measured by either ROC_{50} or by normalized equivalence number, this advantage is not consistent across various training set sizes. Similarly, the corresponding difference for total probability log-odds scoring is also significant, with the completely connected topology performing better than the linear topology. However, the bulk of these differences results from the poor performance of the linear models at a training set size of two sequences. If these training sets are not considered, then the difference in performance between linear and completely connected models is no longer significant for either type of scoring or for either performance metric.

As discussed in Section III.C, the shape of the learning curves in Figure IV.20 is distorted by the relatively large number of small training sets. Figure IV.21 corrects

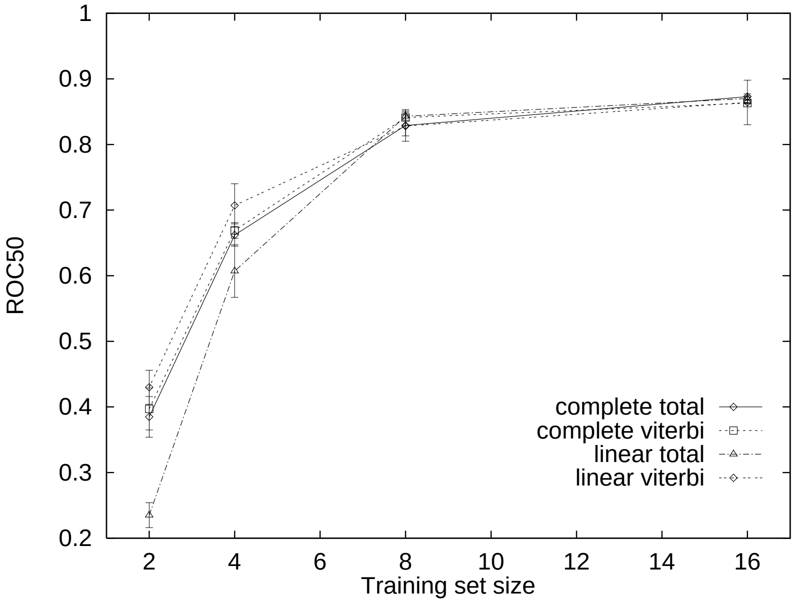


(a)

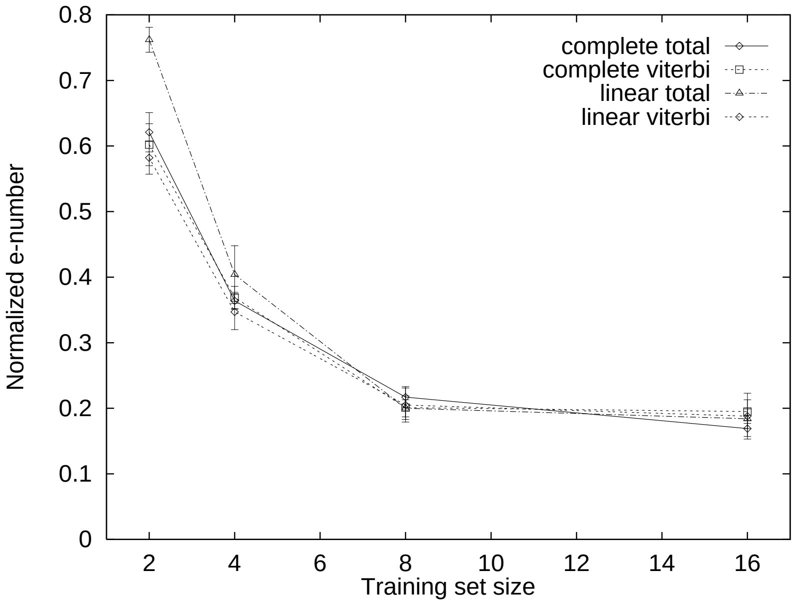


(b)

Figure IV.20: **Relative homology detection performance of completely connected and linear HMMs using Viterbi and total probability scoring.** Figure (a) plots average ROC_{50} score as a function of training set size for all 73 families in the study. Figure (b) plots average normalized e-number for the same families. Error bars represent standard error.



(a)



(b)

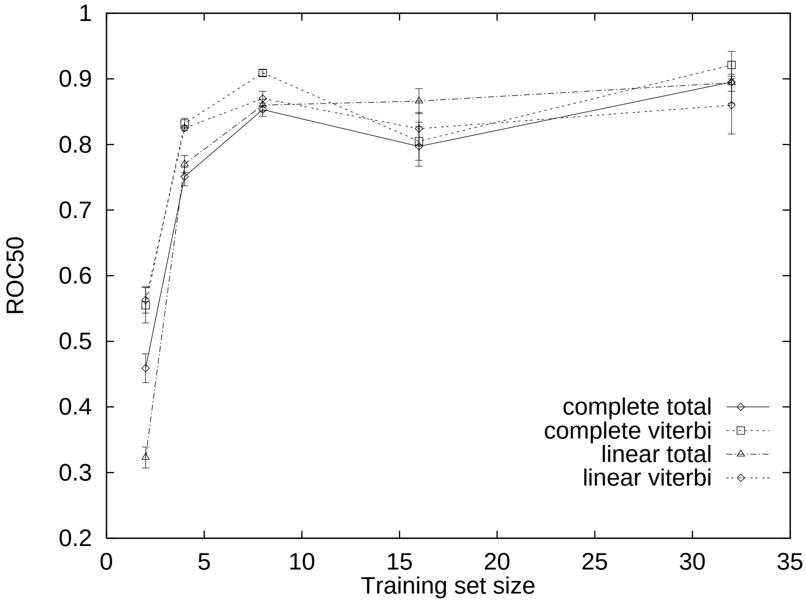
Figure IV.21: **Relative homology detection performance of completely connected and linear HMMs on large families.** Figure (a) plots average ROC_{50} score as a function of training set size for the thirteen families containing between 16 and 31 divergent members. Figure (b) plots average normalized e-number for the same families. Error bars represent standard error.

this distortion by including only those families containing between 16 and 31 divergent sequences. Thus, each point in Figure IV.21 represents an average over the same number (13) of training sets. Here, the difference in performance between the two topologies using Viterbi scoring is not significant. Using total probability scoring, both metrics report a slightly significant improvement from the completely connected topology.

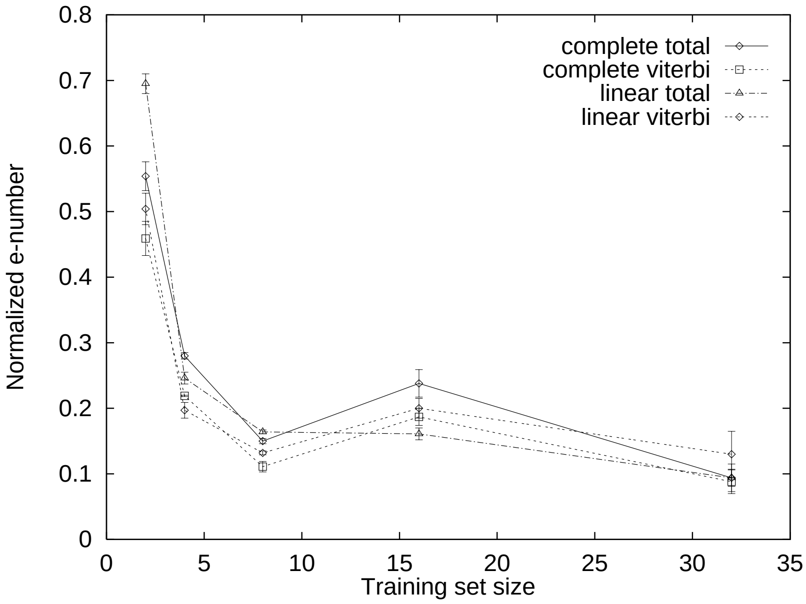
There is reason to believe that the completely connected topology may not be appropriate for modeling every protein family. The increased number of parameters in a completely connected model represents a cost that is only worthwhile if it is paid back via improved modeling ability. For a family in which all of the motifs always appear in the same order, the additional parameters in the completely connected topology do not buy improved modeling ability. Perhaps, then, the improved performance of the completely connected topology will only be evident if we look at families containing repeated elements. Figure IV.22 shows this hypothesis to be false, at least for these 73 families. The figure plots homology detection performance as a function of training set size for the 21 families whose PROSITE documentation indicates that they contain proteins with repeated elements. The results are even less differentiated than the results when all 73 families are considered: using either type of scoring, no significant difference between the performances of the two topologies appears. Thus, even for families containing repeated elements, the completely connected topology does not appear to offer a significant advantage in detecting homologs.

The data in Figure IV.20 can also be used to compare the performance of Viterbi log-odds scoring and total probability log-odds scoring. For both model topologies, Viterbi log-odds scoring performs significantly better than total probability log-odds. The mean difference in ROC_{50} score is 0.11 for linear models and 0.03 for completely connected models. The corresponding equivalence number mean differences are -0.09 and -0.03.[1] These differences are notably larger than the small

[1] Recall that the best possible ROC_{50} score is 1, whereas the best normalized equivalence number is 0. Hence a positive mean difference of ROC_{50} and a negative mean difference of normalized equivalence number both indicate improved performance.



(a)



(b)

Figure IV.22: **Relative homology detection performance of completely connected and linear HMMs on families containing repeated elements.** Figure (a) plots average ROC_{50} score as a function of training set size for the 21 families whose sequences contain repeated elements. Figure (b) plots average normalized e-number for the same families. Error bars represent standard standard error.

difference induced by different model topologies, indicating that the type of scoring used employed is a much more important factor than the model topology.

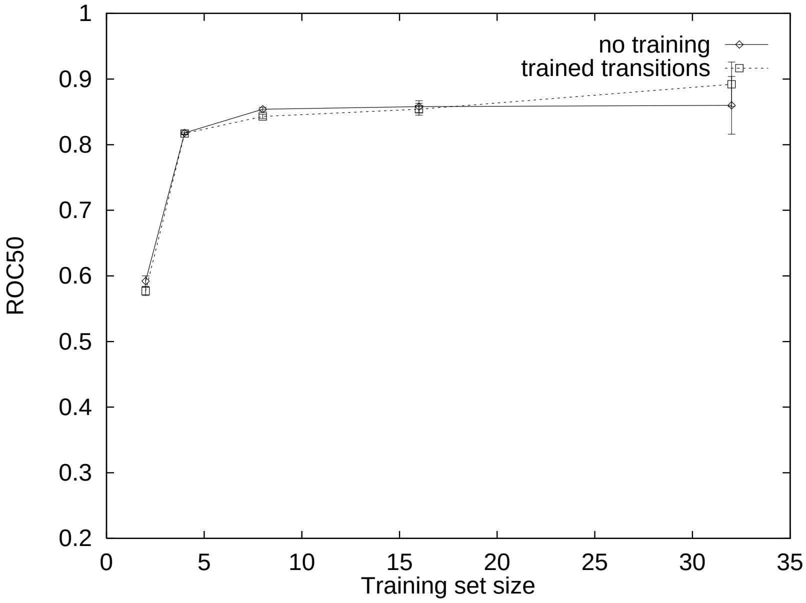
Model training

Next we consider the effect of training upon the homology detection performance of Meta-MEME models. Figures IV.23-IV.24 show that training the transition probabilities in the models has little effect on their performance. Indeed, this is the only set of homology detection results for which the analyses by ROC_{50} score and by normalized equivalence numbers differ: according to ROC_{50} scores, the trained models are slightly better than the untrained models; the converse is true when normalized equivalence numbers are considered. When only the larger families are considered (Figure IV.24), the difference between trained and untrained model performance is uniformly in favor of the untrained models, but this difference is not statistically significant. Thus, training the transitions in the HMM has little effect on the model's homology detection performance.

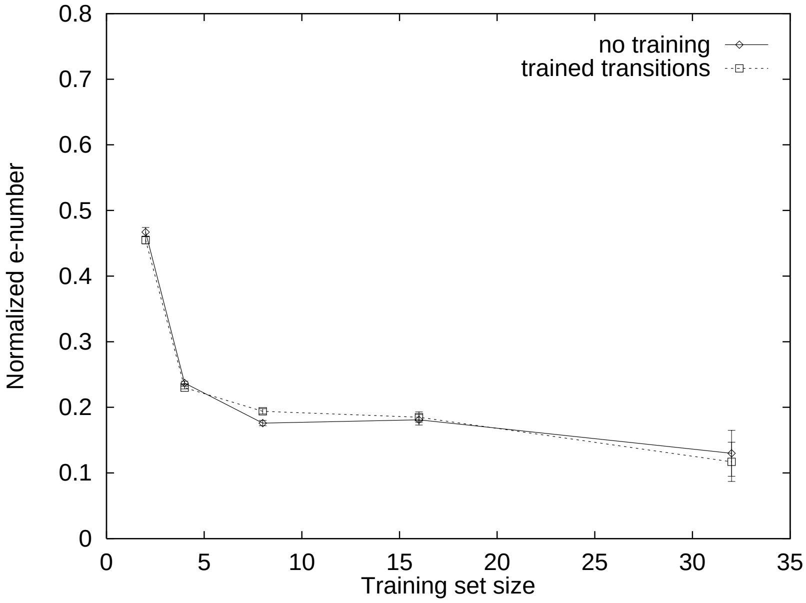
Comparison with FPS, MEME and HMMER

Finally, we compare Meta-MEME's homology detection performance with that of three other homology detection methods. In Chapter III, we compared three techniques: the standard hidden Markov model package, HMMER, the motif-based modeling and searching tools MEME and MAST, and the BLAST-based Family Pairwise Search algorithm. The results, as shown in Figure IV.25, clearly rank these three techniques, with HMMER performing the worst and FPS performing the best overall. Using the trained linear model as a representative example of Meta-MEME's performance places it in the middle of this ranking, since Meta-MEME's performance is only very slightly (but still significantly) different from that of MAST.

Because of the large number of homology detection experiments performed here, every comparison of techniques yields statistically significant differences in performance, even when that difference is very small. These statistical significances are

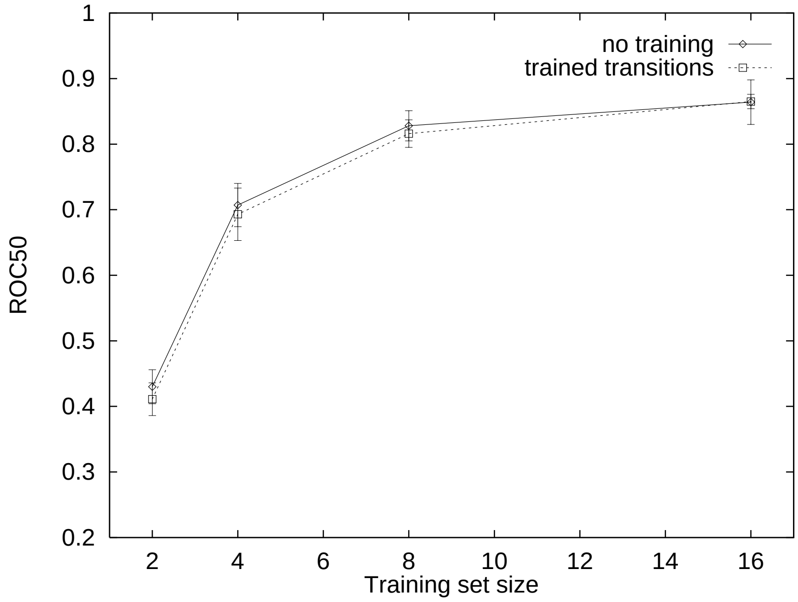


(a)

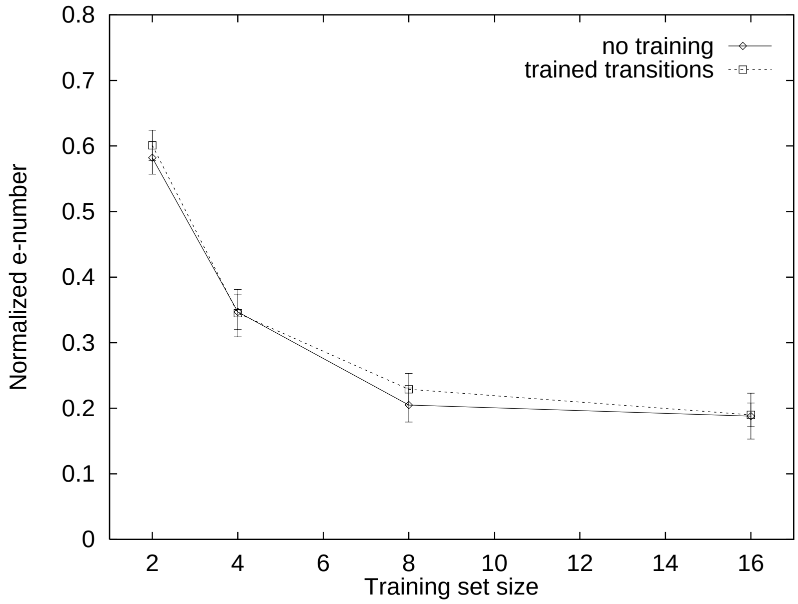


(b)

Figure IV.23: **Relative homology detection performance of untrained and trained linear HMMs.** Figure (a) plots average ROC_{50} score as a function of training set size for all 73 families. Figure (b) plots average normalized e-number for the same families. Error bars represent standard standard error.

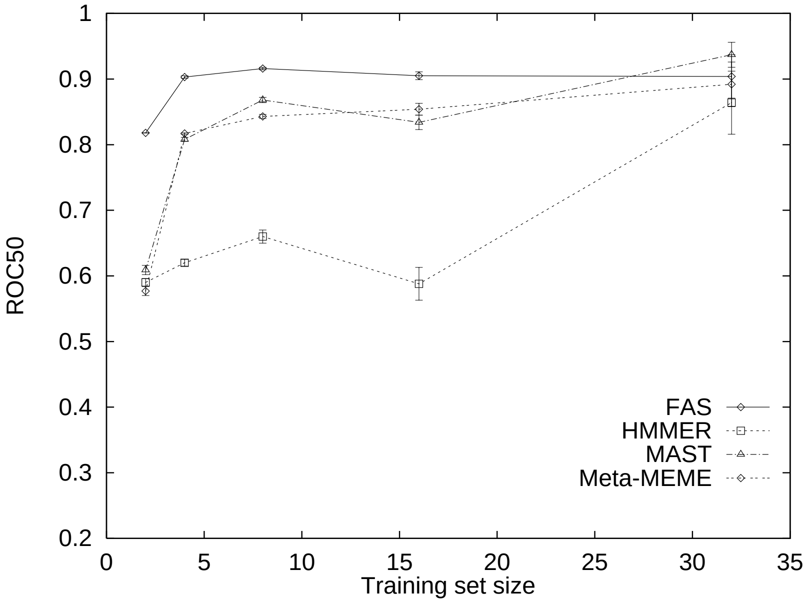


(a)

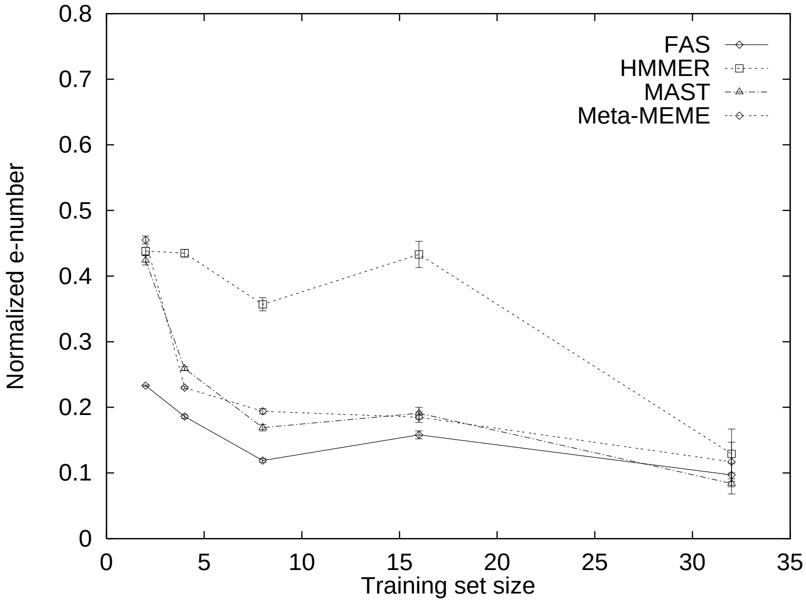


(b)

Figure IV.24: **Relative homology detection performance of untrained and trained linear HMMs on large families.** Figure (a) plots average ROC_{50} score as a function of training set size for all the thirteen families containing between 16 and 31 divergent sequences. Figure (b) plots average normalized e-number for the same families. All homology detection was performed using Viterbi log-odds scores. Error bars represent standard standard error.

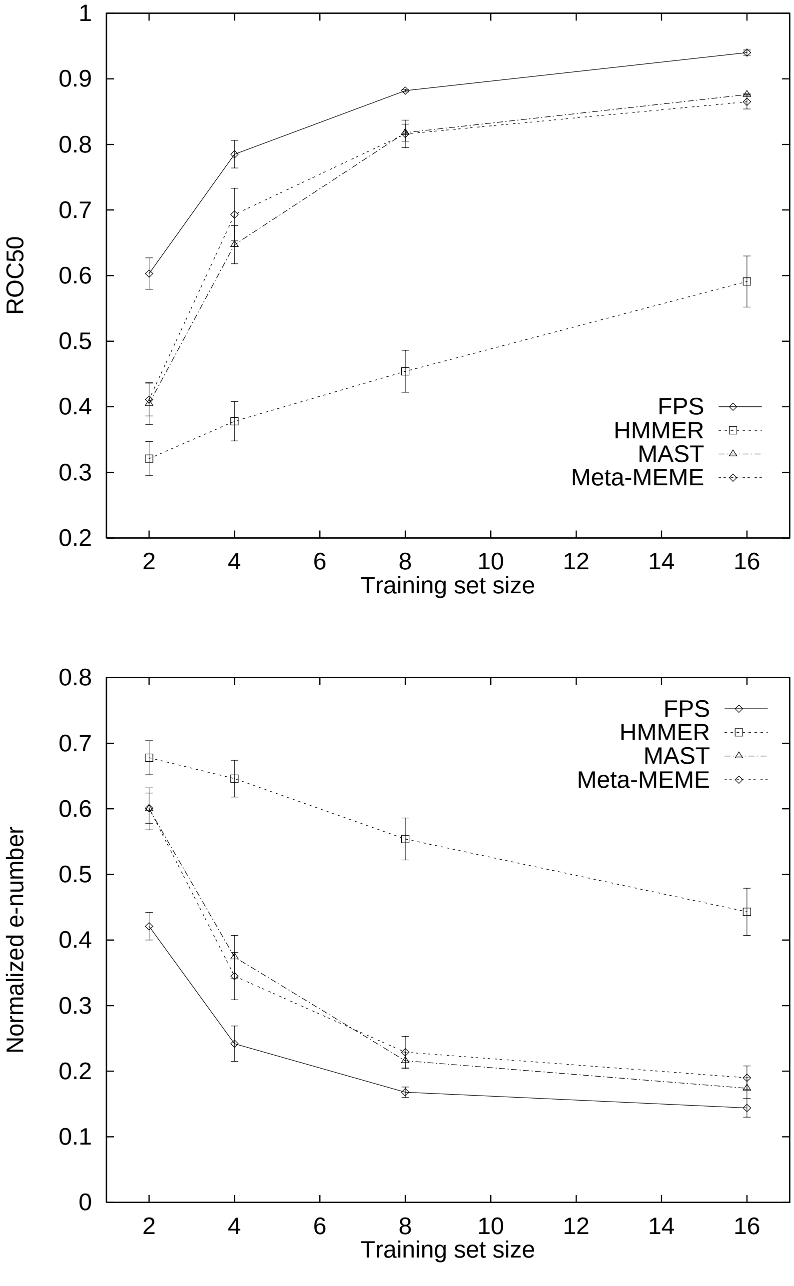


(a)



(b)

Figure IV.25: **Relative homology detection performance of FPS, HMMER, MAST and Meta-MEME.** Figure (a) plots average ROC_{50} score as a function of training set size for all 73 families. Figure (b) plots average normalized e-number for the same families. Error bars represent standard standard error.



(a)

(b)

Figure IV.26: **Relative homology detection performance of FPS, HMMER, MAST and Meta-MEME on large families.** Figure (a) plots average ROC_{50} score as a function of training set size for all the thirteen families containing between 16 and 31 divergent sequences. Figure (b) plots average normalized e-number for the same families. Error bars represent standard standard error.

	Method
1	BLAST FPS bits ave
2	HMMER
3	MAST
4	Meta-MEME hmmsw
5	linear Meta-MEME viterbi none
6	linear Meta-MEME viterbi trans
7	linear Meta-MEME total none
8	complete Meta-MEME viterbi none
9	complete Meta-MEME total none

	1	2	3	4	5	6	7	8	9
1	—	0.26	0.13	0.15	0.13	0.14	0.24	0.14	0.18
2	-0.26	—	-0.13	-0.10	-0.12	-0.11	-0.01	-0.11	-0.08
3	-0.13	0.13	—	0.03	0.01	0.01	0.12	0.02	0.05
4	-0.15	0.10	-0.03	—	-0.02	-0.01	0.09	-0.01	0.02
5	-0.13	0.12	-0.01	0.02	—	0.01	0.11	0.01	0.05
6	-0.14	0.11	-0.01	0.01	-0.01	—	0.10	0.00	0.04
7	-0.24	0.01	-0.12	-0.09	-0.11	-0.10	—	-0.10	-0.07
8	-0.14	0.11	-0.02	0.01	-0.01	-0.00	0.10	—	0.03
9	-0.18	0.08	-0.05	-0.02	-0.05	-0.04	0.07	-0.03	—

Table IV.20: **Differences in homology performance, as measured by $\mathbf{ROC_{50}}$.** Performance results of the nine different homology detection methods examined in this section, including six variants of the Meta-MEME algorithm, are summarized here. A positive value in a particular row and column indicates that the method corresponding to that row performs significantly better than the method corresponding to that column, as measured by a paired t test of ROC_{50} scores with 183 degrees of freedom, and vice versa. The magnitude of the value is the mean difference between the two techniques' average ROC_{50} scores. All differences are significant at the 1% confidence level. Meta-MEME experiments are listed according to model topology (linear vs. completely connected), type of scoring (Viterbi log-odds vs. total probability log-odds), and type of model training (none vs. trained transition probabilities).

	1	2	3	4	5	6	7	8	9
1	—	-0.23	-0.11	-0.14	-0.12	-0.12	-0.22	-0.13	-0.16
2	0.23	—	0.12	0.09	0.11	0.11	0.01	0.10	0.07
3	0.11	-0.12	—	-0.03	-0.01	-0.01	-0.11	-0.02	-0.05
4	0.14	-0.09	0.03	—	0.01	0.02	-0.08	0.01	-0.02
5	0.12	-0.11	0.01	-0.01	—	0.00	-0.09	-0.01	-0.04
6	0.12	-0.11	0.01	-0.02	-0.00	—	-0.10	-0.01	-0.04
7	0.22	-0.01	0.11	0.08	0.09	0.10	—	0.09	0.06
8	0.13	-0.10	0.02	-0.01	0.01	0.01	-0.09	—	-0.03
9	0.16	-0.07	0.05	0.02	0.04	0.04	-0.06	0.03	—

Table IV.21: **Differences in homology performance, as measured by normalized equivalence number.** See caption for Table IV.20. Note that, because the range of normalized equivalence numbers is reversed with respect to ROC_{50} scores, a positive mean difference here indicates a decrease in performance, rather than an improvement.

	Method	Mean difference	
		E	ROC_{50}
1	BLAST FPS	-0.11	0.13
2	MAST	-0.01	0.00
3	linear Meta-MEME viterbi none	-0.00	0.01
4	linear Meta-MEME viterbi trans	-0.00	0.00
5	complete Meta-MEME viterbi none	-0.01	0.01
6	Meta-MEME hmmsw	-0.02	0.03
7	complete Meta-MEME total none	-0.05	0.06
8	linear Meta-MEME total none	-0.01	0.07
9	HMMER		

Table IV.22: **Total ordering on performance of homology detection methods, as measured by ROC_{50} and normalized equivalence numbers.** The last two columns give the mean difference between the performance of the method on the current row and the method on the following row, according to the normalized equivalence number and the ROC_{50} score. For an explanation of the different methods, see caption to Table IV.20. The ordering of “linear Meta-MEME viterbi none” and “linear Meta-MEME viterbi trans” implied by the normalized equivalence number is the reverse of what is shown above.

Family	N_w	MAST	Meta-MEME	Difference
PS00190	73	0.048	0.743	0.694791
PS00402	19	0.662	0.814	0.152174
PS00339	19	0.401	0.550	0.149091
PS00343	16	0.858	0.953	0.095555
PS00038	26	0.952	0.958	0.005136
PS00678	17	0.980	0.984	0.004000
PS00095	16	0.999	0.999	0.000000
PS00659	19	1.000	1.000	0.000000
PS00340	16	0.995	0.994	-0.000952
PS00061	24	0.943	0.937	-0.006060
PS00211	38	0.971	0.958	-0.012522
PS00639	19	0.887	0.848	-0.038696
PS00640	19	0.887	0.848	-0.038696
PS00030	24	0.968	0.913	-0.055349
PS00198	53	0.928	0.792	-0.135644
PS00092	28	0.861	0.438	-0.423158

Table IV.23: **Performance comparison of MAST and Meta-MEME using sixteen-sequence training sets.** Listed are the sixteen families containing at least sixteen divergent sequences. The columns labeled “MAST” and “Meta-MEME” contain ROC_{50} scores. The Meta-MEME scores are for untrained, linear models using Viterbi log-odds scoring. N_w is the number of sequences in the family after binary sequence weighting. The final column contains the difference between the two ROC_{50} values. The families are ranked by this value.

summarized in Table IV.20 and IV.21. For both of the performance metrics, these data imply a total ordering on the nine homology detection methods examined in this chapter. Overall, these total orderings are in almost complete agreement and are given in Table IV.22. They place Meta-MEME's performance below that of both FPS and MAST, but above the performance of HMMER.

In general, the difference in performance between MAST and Meta-MEME is small, and for a number of families Meta-MEME succeeds in providing superior discrimination. Table IV.23 compares MAST and Meta-MEME ROC_{50} scores for the sixteen families that contain sixteen or more divergent sequences. Meta-MEME out-performs MAST on eight of these families.

In any evaluation of homology detection methods, the problem of unanno-

Score	ID	Description
22.75	MTSM_SERMA	Modification methylase SMAI
17.82	MTC1_CITFR	Modification methylase CFRBI
16.42	MTB2_BACAM	Modification methylase BAMHII
15.80	MTHZ_METTF	Modification methylase MTHZI
15.29	MTC9_CITFR	Modification methylase CFR9I
14.91	MTX1_XANCC	Modification methylase XCYI
0.04	MTP2_PROVU	Modification methylase PVU II
-0.29	MSP_ASCLU	Major sperm protein
-0.29	MSP1_ASCSU	Major sperm protein, isoform α
-1.13	YNI1_METTL	Hypothetical protein

Table IV.24: **Meta-MEME false positive sequences from the N-6 adenine-specific DNA methylases.** The table lists the Viterbi log-odds scores, IDs and descriptions of the first twelve false positive sequences generated by Meta-MEME using an untrained, linear model from a set of sixteen sequences. The family is PS00092 (see Table IV.23).

tated family members arises. Table IV.24 lists the first twelve false positives from family PS00092, the family for which Meta-MEME's performance was poorest relative to MAST's. The table includes eight sequences labeled as modification methylases. These same eight sequences also appear near the top of the list of false positives created by MAST. Since this family consists of N-6 adenine-specific DNA methylases, and since most of the family members are annotated as modification methylases, it is likely that these eight sequences belong in the family. Accordingly, the other four Meta-MEME experiments listed in Table IV.20 yield, for this family, ROC_{50} scores of 0.73, 0.78, 0.85 and 0.86, much greater than the 0.44 shown in Table IV.23. Thus, for this family, the large difference in ROC_{50} value may result from the noise introduced by these unannotated family members.

IV.E.4 Discussion

In the experiments reported here, we have examined Meta-MEME's performance on the homology detection task, while varying characteristics such as the search algorithm, model topology, score computation, and training algorithm. Overall, Meta-MEME's performance is very stable: relatively large algorithmic changes produce only small differences in performance. This stability requires further explanation.

The small change in performance afforded by the more complex search algorithm implemented in the HMMER search tool `hmmsw` is somewhat surprising. Unfortunately, although the `hmmsw` algorithm is described [47] as a hidden Markov model implementation of Smith-Waterman local alignment, the details of the algorithm are unpublished. As a local search algorithm, `hmmsw` allows a subsequence of the database sequence to match a subsequence of the states in the linear HMM. This ability should allow for matches to sequence fragments as well as incomplete homologies, in which a single domain in a multi-domain protein is homologous to the training set. The relatively good performance of `mhmms`, which makes no attempt to explicitly allow for such matches, indicates either a deficiency in `hmmsw` or that sequence fragments and incomplete homologies do not cause problems for a global alignment algorithm such as the Viterbi or forward algorithms.

The experiments reported here show that Viterbi log-odds scoring provides better homology detection performance than does total probability log-odds scoring. As discussed in Section IV.D.4, this difference in performance most likely arises because the Viterbi path of a family member corresponds to the actual evolutionary history of that sequence. The total probability, on the other hand, does not have a straightforward interpretation in terms of the evolutionary model implicit in the HMM topology.

The dependence of performance upon the type of scoring employed explains why training the HMMs has little effect on the discriminative ability of the models. The Baum-Welch training algorithm maximizes the total probability of the sequences, given the model. We would expect, therefore, that Meta-MEME's performance on the homology detection task using total probability scoring would improve after training. However, since we have already shown that Viterbi scoring provides better discrimination overall, the experiments reported here employed Viterbi scoring with the trained models. The difference in performance after training was relatively small because the training algorithm does not explicitly maximize the Viterbi score.

One of Meta-MEME's primary goals is to model sequences containing repeated or shuffled elements. This goal is accomplished using models with a completely connected topology. The results described above, however, indicate that completely connected models do not characterize families containing repeated elements any better than do models with a linear topology. This failure to improve upon the linear topology may be explained in two ways. First, the failure may result from the cost of training the additional parameters in the completely connected topology. Second, the failure may arise because of non-Markov properties at the motif level. For example, a family may contain exactly four copies of a single motif in every family member. A motif-based HMM, because it is Markov, cannot keep track of the number of occurrences of a given motif.

Overall, Meta-MEME fails to improve upon MAST's homology detection performance. Meta-MEME operates under the assumption that information about the order and spacing of motifs within a family can provide important homology detection information. Meta-MEME's failure to improve upon MAST does not indicate that this assumption is false. Rather, the failure of Meta-MEME's training to significantly improve the homology detection performance of the motif-based HMMs suggests the need for a better training algorithm. An improved version of Meta-MEME based upon a training algorithm that maximizes the Viterbi scores of the training sequences will more fully exploit meta-information about the order and spacing of motifs within a protein family.

The text of Section IV.B, in part, is a reprint of the material as it appears in *Computer Applications in the Biosciences* [62]. The dissertation author was the primary author, and the co-authors listed in this publication directed and supervised the research which forms the basis for this chapter.

The text of Section IV.C, in part, is a reprint of the material as it appears in

Biochemical and Biophysical Research Communications [61]. The dissertation author was the primary author, and the co-authors listed in this publication directed and supervised the research which forms the basis for this chapter.

Chapter V

Conclusion

Technological advances in the second half of the twentieth century have been dominated by two fields: computer science and molecular biology. The genome sequencing projects currently underway represent the first major convergence of these two fields and have led to the birth of the new subdiscipline, bioinformatics. Thirteen completed genomes are publicly available on the web [124] as of May, 1998. In the post-genomic era, when complete genomes are known for many species and even for many individual organisms within a species, the volume of sequence data available will require increasingly sophisticated computational analyses.

The field of artificial intelligence has traditionally been home to some of the hardest problems in computer science. Over the past forty years, the public's perception of the field has oscillated: optimism in the early 60s about problems such as speech recognition, planning and natural language understanding diminished as the complexity of those problems became more apparent. A similar wave of optimism in the late 70s was quelled in the mid-80s with the widely publicized “failure” of AI. In the past two years, one of AI's first major successes has reached the shelves of nearly every computer store. A useful, speaker-dependent speech recognition system can be purchased for less than $100 and used with relatively low error rates to take dictation. Every commercially available speech recognition system on the market today uses hidden Markov modeling as the basis for its signal processing.

Although HMMs were developed in the late 1960s, their probabilistic interpretation coincides with the current trend in AI toward Bayesian reasoning [66]. Traditional statistics avoided the Bayesian formalism primarily because computing the complete joint probability distribution is so computationally expensive. During the past decade, however, advances in computational power and in probability theory have led to the first working Bayesian inference and learning engines. One AI system based upon the Bayesian formalism diagnoses lymph-node diseases more accurately than some of the world's leading pathologists [65].

This dissertation applies these proven, state-of-the-art statistical techniques from AI to one of science's most important problems, that of discovering the functional meaning of the genome. The transfer of technology from speech recognition to protein modeling is initially unintuitive. The correspondence between the two "hidden" processes—the production of speech in the vocal tract and the evolution of biomolecules—is not immediately evident. Nonetheless, the mapping from proteins onto HMMs is effective and illustrates the general power of these statistical models.

As a computer science dissertation addressing a biological problem, this work necessarily crosses the borders between disciplines. Undoubtedly, the computational analyses reported here are more sophisticated than the relatively straightforward biological insights offered. Nonetheless, this work has important implications for the way biologists view proteins. Meta-MEME's assumption that proteins can best be modelled as a collection of motifs with a specified order and spacing is belied by the excellent performance of the Family Pairwise Search algorithm. The improved performance offered by FPS indicates that important functional information resides in the spacer regions between motifs. On the other hand, the improvement of FPS's performance when motif models are incorporated shows that statistical modeling of the kind employed by Meta-MEME can provide an advantage over non-model-based algorithms. The challenge for any Bayesian reasoning system is to balance the complexity of the model employed against the amount of knowledge—both prior knowledge and data—available to be modeled. A more accurate protein family model will likely be motif-based, but less strongly than Meta-MEME, a Bayesian counterpart to the hybrid motif model/spacer sequence approach adopted by cobbled FPS.

The Meta-MEME software toolkit is a work in progress. The experiments reported in Chapter IV point toward several improvements. Chief among these is the inclusion of a Viterbi-based training algorithm [21, 94]. Currently, a prototype Meta-MEME web server is being developed at the San Diego Supercomputer Center [63]. This server will make available Meta-MEME's modeling, alignment and homology detection capabilities to the larger biological community. Meta-MEME will succeed only if its analytical power is available, interpretable and usable by molecular biologists. The web server will guarantee availability and will elicit feedback from biologists about the usability and usefulness of the results that Meta-MEME produces.

In the long run, Meta-MEME's usefulness will continue only as long as its models can be employed in solving the new problems facing molecular biology in the post-genomic era. As technology and our understanding of the genome advance, emphasis will shift from interpreting the function of specific proteins to understanding the complex mechanisms by which groups of proteins interact with and regulate one another. The Bayesian formalism represents an advantage when new problems arise, since the laws of probability provide a theoretical foundation within which to combine diverse sources of knowledge. Meta-MEME combines multiple motif models into a single model of a protein. In the future, multiple protein models of the type employed by Meta-MEME may well form the basis of still larger models of multi-protein interactions.

Appendix A

73 PROSITE families

ID	Family	n	n_w	R
PS00030	Eukaryotic putative RNA-binding region RNP-1	59	24	Y
PS00037	Myb DNA-binding domain 1	18	4	Y
PS00038	Myc-type, ‘helix-loop-helix’ dimerization domain	90	26	N
PS00043	Bacterial regulatory proteins, gntR	10	8	N
PS00060	Iron-containing alcohol dehydrogenases 2	7	3	N
PS00061	Short-chain alcohol dehydrogenase	82	24	Y
PS00070	Aldehyde dehydrogenases cysteine active site	34	8	N
PS00075	Dihydrofolate reductase	33	14	N
PS00077	Cytochrome c oxidase subunit I, copper B binding region	53	2	N
PS00079	Multicopper oxidases 1	12	7	Y
PS00092	N-6 Adenine-specific DNA methylases	35	28	Y
PS00095	C-5 cytosine-specific DNA methylases C-terminal	33	16	N
PS00099	Thiolases active site	14	3	N
PS00118	Phospholipase A2 histidine active site	110	9	N
PS00120	Lipases, serine active site	36	14	N
PS00133	Zinc carboxypeptidases, zinc-binding region 2	19	5	N
PS00141	Eukaryotic and viral aspartyl proteases active site	50	13	Y
PS00144	Asparaginase / glutaminase active site 1	8	3	N
PS00180	Glutamine synthetase 1	55	7	N
PS00185	Isopenicillin N synthetase 1	10	2	N
PS00188	Biotin-requiring enzymes attachment site	15	8	N
PS00190	Cytochrome c family heme-binding site	223	73	Y
PS00194	Thioredoxin family active site	48	15	Y
PS00198	4Fe-4S ferredoxins, iron-sulfur binding region	109	53	Y
PS00209	Arthropod hemocyanins / insect LSPs 1	14	4	N

ID	Family	n	n_w	R
PS00211	ABC transporters	119	38	Y
PS00215	Mitochondrial energy transfer proteins	39	12	Y
PS00217	Sugar transport proteins 2	46	14	N
PS00225	Crystallins beta and gamma 'Greek key' motif	47	6	Y
PS00281	Bowman-Birk serine protease inhibitors	22	9	Y
PS00283	Soybean trypsin inhibitor (Kunitz) protease inhibitors	30	13	N
PS00287	Cysteine proteases inhibitors	32	11	Y
PS00301	GTP-binding elongation factors	110	8	N
PS00338	Somatotropin, prolactin and related hormones 2	86	12	N
PS00339	Aminoacyl-transfer RNA synthetases class-II 2	38	19	N
PS00340	Growth factor and cytokines receptors 2	37	16	N
PS00343	Gram-positive cocci surface proteins 'anchoring' hexapeptide	25	16	N
PS00372	PTS EIIA domains phosphorylation site 2	7	4	N
PS00399	ATP-citrate lyase and succinyl-CoA ligases active site	4	2	N
PS00401	Prokaryotic sulfate-binding proteins 1	5	2	N
PS00402	Binding-protein-dependent transport systems inner membrane component sign	39	19	N
PS00422	Granins 1	12	3	N
PS00435	Peroxidases proximal heme-ligand	41	8	N
PS00436	Peroxidases active site	40	8	N
PS00490	Prokaryotic molybdopterin oxidoreductases 2	9	6	N
PS00548	Ribosomal protein S3 1	18	3	N
PS00589	PTS HPR component serine phosphorylation site	10	5	Y
PS00599	Aminotransferases class-II pyridoxal-phosphate attachment site	21	8	N
PS00606	Beta-ketoacyl synthases active site	17	4	Y
PS00624	GMC oxidoreductases 2	9	5	N
PS00626	Regulator of chromosome condensation (RCC1) 2	6	2	Y
PS00637	CXXCXGXG dnaJ domain	9	5	N
PS00639	Eukaryotic thiol (cysteine) proteases histidine active site	62	19	N
PS00640	Eukaryotic thiol (cysteine) proteases asparagine active site	62	19	N
PS00643	Respiratory-chain NADH dehydrogenase 75 Kd subunit 3	5	2	N

ID	Family	n	n_w	R
PS00656	Glycosyl hydrolases family 6 2	5	4	N
PS00659	Glycosyl hydrolases family 5	40	19	N
PS00675	Sigma-54 interaction domain ATP-binding region A	36	6	N
PS00676	Sigma-54 interaction domain ATP-binding region B	36	6	N
PS00678	Beta-transducin family Trp-Asp repeats	26	17	Y
PS00687	Aldehyde dehydrogenases glutamic acid active site	33	7	N
PS00697	ATP-dependent DNA ligase AMP-binding site	11	6	N
PS00700	Ribosomal protein L6 2	13	4	N
PS00716	Sigma-70 factors 2	36	8	N
PS00741	Guanine-nucleotide dissociation stimulators CDC24	6	5	N
PS00760	Signal peptidases I lysine active site	8	5	N
PS00761	Signal peptidases I 3	8	5	N
PS00831	Ribosomal protein L27	6	3	N
PS00850	Glycine radical	4	3	N
PS00867	Carbamoyl-phosphate synthase subdomain 2	20	3	Y
PS00881	Protein splicing	3	3	Y
PS00904	Protein prenyltransferases alpha subunit	4	3	Y
PS00933	FGGY family of carbohydrate kinases 1	11	5	N

PROSITE IDs of the 73 families used in Chapter III and in Section IV.E. n is the total number of sequences in the family, and n_w is the number of sequences remaining after binary sequence weighting. The final column (R) indicates whether the family contains repeated elements. Two families from the original set of 75 [10] were discarded because they contain a single sequence after binary sequence weighting.

Bibliography

[1] R. Albalat, R. Gonzalez-Duarte, and S. Atrian. Protein engineering of *drosophila* alcohol dehydrogenase. the hydroxyl group of Tyr152 is involved in the active site of the enzyme. *FEBS Letters*, 308(3):235–239, 1992.

[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. K. Attwood, M. E. Beck, and A. J. Bleasby. PRINTS/SMITE documentation. `http://bmbsgi11.leeds.ac.uk/bmb5dp/prints_doc.html`, November 1993.

[5] T. L. Bailey. MEME – multiple EM for motif elicitation. `http://www.sdsc.-edu/MEME`, 1998.

[6] 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.

[7] T. L. Bailey and C. P. Elkan. Unsupervised learning of multiple motifs in biopolymers using EM. *Machine Learning*, 21(1–2):51–80, October 1995.

[8] T. L. Bailey and C. P. Elkan. The value of prior knowledge in discovering motifs with MEME. In C. Rawlings, D. Clark, R. Altman, L. C. Hunter, and L. C. Rawlings, editors, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 21–29. AAAI Press, 1995.

[9] 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.

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

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

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

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

[14] J. K. Baker. The Dragon system — an overview. *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-23(1):24–29, February 1975.

[15] M. E. Baker. Genealogy of regulation of human sex and adrenal function, prostaglandin action, snapdragon and petunia flower colors, antibiotics, and nitrogen fixation: Functional diversity from two ancestral dehydrogenases. *Steroids*, 56(7):354–360, 1991.

[16] M. E. Baker. Myxococcus xanthus c-factor, a morphogenetic paracrine signal, is similar to *escherichia coli* 3-oxoacyl-[acyl-carrier-protein] reductase and human 17 β-hydroxysteroid dehydrogenase [letter]. *Biochemical Journal*, 300:605–607, 1994.

[17] M. E. Baker. Sequence analysis of steroid and prostaglandin metabolizing enzymes: Application to understanding catalysis. *Steroids*, 59:248–258, 1994.

[18] M. E. Baker. Unusual evolution of mammalian 11β- and 17β-hydroxysteroid and retinol dehydrogenases. *Bioessays*, 18:63–70, 1996.

[19] M. E. Baker and R. Blasco. Expansion of the mammalian 3 β-hydroxysteroid dehydrogenase/plant dihydroflavonol reductase superfamily to include a bacterial cholesterol dehydrogenase, a bacterial UDP-galactose-4-epimerase, and open reading frames in vaccinia virus and fish lymphocystis disease virus. *FEBS Letters*, 301(1):89–93, 1992.

[20] S. L. Baldauf, J. D. Palmer, and W. F. Doolittle. The root of the universal tree and the origin of eukaryotes based on elongation factor phylogeny. *Proceedings of the National Academy of Sciences of the United States of America*, 93:7749–7754, 1996.

[21] P. Baldi and Y. Chauvin. Smooth on-line learning algorithms for hidden Markov models. *Neural Computation*, 6(2):307–318, 1994.

[22] 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.

[23] M. Balter. Morphologists learn to live with molecular upstarts. *Science*, 276:1032–1034, 1997.

[24] A. Banerjee, E. Dubnau, A. Quemard, V. Balasubramanian, K. S. Um, T. Wilson, D. Collins, G. de Lisle, and W. R. Jacobs. inhA, a gene encoding a target for isoniazid and ethionamide in *mycobacterium tuberculosis*. *Science*, 263(5144):227–230, 1994.

[25] M. Baron, D. G. Norman, and L. D. Campbell. Protein modules. *Trends in Biochemical Sciences*, 16:13–17, 1991.

[26] C. Barrett, R. Hughey, and K. Karplus. Scoring hidden Markov models. *Computer Applications in the Biosciences*, 13(2):191–199, 1997.

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

[28] Blocks WWW server. `http://www.blocks.fhcrc.org`, 1997.

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

[30] C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland, 1991.

[31] R. Breton, D. Housset, C. Mazza, and J. C. Fontecilla-Camps. The structure of a complex of human 17β-hydroxysteroid dehydrogenase with estradiol and nadp+ identifies two principal targets for the design of inhibitors. *Structure*, 4(18):905–915, 1996.

[32] 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.

[33] F. J. Castellino and J. M. Beals. The genetic relationships between the kringle domains of human plasminogen, prothrombin, tissue plasminogen activator, urokinase, and coagulation factor XII. *Journal of Molecular Evolution*, 26(4):358–369, 1987.

[34] Z. Chen, J. C. Jiang, Z. G. Lin, W. R. Lee, M. E. Baker, and S. H. Chang. Site-specific mutagenesis of *Drosophila* alcohol dehydrogenase: Evidence for involvement of tyrosine-152 and lysine-156 in catalysis. *Biochemistry*, 32(13):3342–3346, 1993.

[35] S. W. Chenevert, N. G. Fossett, S. H. Chang, I. Tsigelny, M. E. Baker, and W. R. Lee. Amino acids important in enzyme activity and dimer stability for *Drosophila* alcohol dehydrogenase. *Biochemical Journal*, 308:419–423, 1995.

[36] C. Chothia. Proteins—1000 families for the molecular biologist. *Nature*, 357:543–544, 1992.

[37] C. Chothia and A. M. Lesk. The relation between the divergence of sequence and structure in proteins. *EMBO Journal*, 5:823–826, 1986.

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

[39] G. A. Churchill. Stochastic models for heterogeneous DNA sequences. *Bulletin of Mathematical Biology*, 51:79–94, 1989.

[40] Compugen, Ltd. BIOCCELERATOR *Manual*. `http://www.compugen-us.com`, 1996.

[41] Jr. D. G. Forney. The viterbi algorithm. In *Proceedings of the IEEE*, volume 61, pages 268–278, 1973.

[42] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–22, 1977.

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

[44] R. F. Doolittle. Reconstructing history with amino acid sequences. *Protein Science*, 11:191–200, 1992.

[45] R. L. Dorit, L. Schoenback, and W. Gilbert. How big is the universe of exons? *Science*, 250:1377–1382, 1990.

[46] 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.

[47] S. R. Eddy. personal communication, May 1998.

[48] Entrez. `http://www.ncbi.nlm.nih.gov/Entrez`, 1998.

[49] J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 1981.

[50] J. Felsenstein. PHYLIP — phylogeny inference pacakge (version 3.2). *Cladistics*, 5:164–166, 1989.

[51] W. M. Fitch. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology*, 20:406–416, 1971.

[52] R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bultd, J-F. Tomb, B. A. Dougherty, and J. M. Merrick et al. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science*, 269(5223):496–512, 1995.

[53] National Center for Biotechnology Information. Non-redundant protein database, including GenBank CDS translations, PDB, SwissProt and PIR. `ftp://ncbi.nlm.nih.gov/blast/db/nr.Z`.

[54] C. M. Fraser, J. D. Gocayne, O. White, M. D. Adams, R. A. Clayton, R. D. Fleishman, C. J. Bult, A. R. Kerlavage, G. Sutton, and J. M. Kelley et al. The minimal gene complement of *Mycoplasma genitalium*. *Science*, 270(5235):397–403, 1995.

[55] GenBank overview. `http://www.ncbi.nlm.nih.gov/Web/Genbank/index.-html`, 1997.

[56] D. Ghosh, Z. Wawrzak, C. M. Weeks, W. L. Duax, and M. Erman. The refined three-dimensional structure of 3 α,20 β-hydroxysteroid dehydrogenase and possible roles of the residues conserved in short-chain dehydrogenases. *Structure*, 2(7):629–640, 1994.

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

[58] 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.

[59] W. N. Grundy. Family-based homology detection via pairwise sequence comparison. In *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, 1998.

[60] 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.

[61] W. N. Grundy, T. L. Bailey, C. P. Elkan, and M. E. Baker. Hidden Markov model analysis of motifs in steroid dehydrogenases and their homologs. *Biochemical and Biophysical Research Communications*, 231(3):760–766, 1997.

[62] 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.

[63] W. N. Grundy and C. P. Elkan. Meta-meme version 2.0. `http://www.sdsc.-edu/METAMEME`.

[64] D. Gurian-Sherman and S. E. Lindow. Bacterial ice nucleation: significance and molecular basis. *FASEB Journal*, 7(14):1338–1343, 1993.

[65] D. Heckerman. *Probabilistic Similarity Networks.* MIT Press, 1991.

[66] Leslie Helm. Improbable inspiration. *Wall Street Journal*, 1996.

[67] J. G. Henikoff and S. Henikoff. Blocks database and its applications. *Methods in Enzymology*, 266, 1996.

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

[69] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89:10915–10919, 1992.

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

[71] S. Henikoff and J. G. Henikoff. Protein family classification based on searching a database of blocks. *Genomics*, 19:97–107, 1994.

[72] S. Henikoff and J. G. Henikoff. Embedding strategies for effective use of information from multiple sequence alignments. *Protein Science*, 6(3):698–705, 1997.

[73] S. Henikoff, J. G. Henikoff, W. J. Alford, and S. Pietrokovski. Automated construction and graphical presentation of protein blocks from unaligned sequences. *Gene-COMBIS, Gene*, 163(GC):17–26, 1995.

[74] B. Henrissat. A classification of glycosyl hydrolases based on amino acid sequence similarities. *Biochemical Journal*, 280:309–16, 1991.

[75] The human genome project. `http://www.ornl.gov/TechResources/Human_Genome/project/project.html`, 1997.

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

[77] L. Holm, C. Sander, and A. Murzin. Three sisters, different names [letter]. *Nature Structural Biology*, 1(3):146–147, 1994.

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

[79] H. Jornvall, B. Persson, M. Krook, S. Atrian, R. Gonzalez-Duarte, J. Jeffry, and D. Ghosh. Short-chain dehydrogenases/reductases (sdr). *Biochemistry*, 34(18):6003–6013, 1995.

[80] S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the United States of America*, 87:2264–2268, 1990.

[81] E. V. Koonin, A. R. Mushegian, and K. E. Rudd. Sequencing and analysis of bacterial genomes. *Current Biology*, 6(4):404–416, 1996.

[82] 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.

[83] 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.

[84] Z. Krozowski. 11 β-hydroxysteroid dehydrogenase and the short-chain alcohol dehydrogenase (scad) superfamily. *Molecular and Cellular Endocrinology*, 84(1–2):C25–C31, 1992.

[85] G. Labesse, A. Vidal-Cros, J. Chomilier, M. Gaudry, and J. P. Mornon. Structural comparisons lead to the definition of a new superfamily of NAD(P)(H)-accepting oxidoreductases: the single-domain reductases/epimerases/dehydrogenases (the 'RED' family). *Biochemical Journal*, 304:95–9, 1994.

[86] V. Laudet, C. Hanni, J. Coll, F. Catzelflis, and D. Stehelin. Evolution of the nuclear receptor gene superfamily. *EMBO Journal*, 11:1003–1013, 1992.

[87] 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.

[88] S. E. Levinson. Continously variable duration hidden Markov models for speech recognition. *Computer, Speech and Language*, 1(1):29–45, March 1986.

[89] W. Li and D. Graur. *Fundamentals of molecular evolution*. Sinauer Associates, Inc., 1991.

[90] M. A. McClure. Personal communication, June 1997.

[91] M. A. McClure, C. Smith, and P. Elton. Parametrization studies for the SAM and HMMER methods of hidden Markov model generation. 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 155–164. AAAI Press, 1996.

[92] M. A. McClure, T. K. Vasi, and W. M. Fitch. Comparative analysis of multiple protein-sequence alignment methods. *Molecular Biology Evolution*, 11(4):571–592, 1994.

[93] MEME – multiple EM for motif elicitation. `http://www.sdsc.edu/MEME`, 1997.

[94] N. Merhav and Y. Ephraim. Maximum likelihood hidden Markov modeling using a dominant sequence of states. *IEEE Transactions in Signal Processing*, 39(9):2111–2115, 1991.

[95] G. L. G. Miklos and H. D. Campbell. The evolution of protein domains and the organizational complexities of metazoans. *Current Opinion in Genetic Development*, 2:902–906, 1992.

[96] G. J. P. Naylor and W. M. Brown. Amphioxus mitochondrial DNA, chordate phylogeny, and the limits of inference based on comparisons of sequences. *Systematic Biology*, 47(1):61–76, 1998.

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

[98] 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.

[99] 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.

[100] J. Obeid and P. C. White. Tyr-179 and Lys-183 are essential for enzymatic activity of 11 β-hydroxysteroid dehydrogenase. *Biochemical and Biophysical Research Communications*, 188(1):222–227, 1992.

[101] E. Otaka and T. Ooi. Examination of protein sequence homologies: IV. twenty-seven bacterial ferredoxins. *Journal of Molecular Evolution*, 26:257–267, 1987.

[102] 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.

[103] L. Patthy. Modular exchange principles in proteins. *Current Opinion in Structural Biology*, 1:351–361, 1991.

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

[105] W. R. Pearson. Comparison of methods for searching protein sequence databases. *Protein Science*, 4:1145–1160, 1995.

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

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

[108] B. Persson, M. Krook, and H. Jornvall. Characteristics of short chain alcohol dehydrogenases and related enzymes. *European Journal of Biochemistry*, 200(53):7–543, 1991.

[109] B. Persson, M. Krook, and H. Jornvall. Characteristics of short-chain alcohol dehydrogenases and related enzymes. *European Journal of Biochemistry*, 200(2):537–543, 1991.

[110] S. J. Press. *Bayesian Statistics: Principles, Models, and Applications.* John Wiley & Sons, 1989.

[111] T. J. Puranen, H. Poutanen, H. E. Peltoketo, P. T. Vihko, and R. K. R. K. Vihko. Site-directed mutagenesis of the putative active site of human 17 β-hydroxysteroid dehydrogenase type 1. *Biochemical Journal*, 304:289–293, 1994.

[112] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1995.

[113] L. R. Rabiner and B. Juang. *Fundamentals of speech recognition.* Prentice Hall, 1993.

[114] J. B. Rafferty, J. W. Simon, C. Baldock, P. J. Artymiuk, P. J. Baker, A. R. Stuitje, A. R. Slabas, and D. W. Rice. Common themes in redox chemistry emerge from the x-ray structure of oilseed rape (*brassica napus*) enoyl acyl carrier protein reductase. *Structure*, 3(9):927–938, 1995.

[115] SAM: sequence alignment and modeling system. `http://www.cse.ucsc.edu/-research/compbio/sam.html`, 1997.

[116] K. Sjolander, K. Karplus, M. 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*, 1996.

[117] H. O. Smith, T. M. Annau, and S. Chandrasegaran. Finding sequence motifs in groups of functionally related proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 87:826–830, 1990.

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

[119] E. Sonnhammer and D. Kahn. The modular arrangement of proteins as inferred from analysis of homology. *Protein Science*, 3:482–492, 1994.

[120] C. M. Starks, K. Back, J. Chappell, and J. P. Noel. Structural basis for cyclic terpene biosynthesis by tobacco 5-epi-aristolochene synthase. *Science*, 277(5333):1815–20, 1997.

[121] N. Tanaka, T. Nonaka, T. Tanabe, T. Yoshimoto, D. Tsuru, and Y. Mitsui. Crystal structures of the binary and ternary complexes of 7 α-hydroxysteroid dehydrogenase from *escherichia coli*. *Biochemistry*, 35(24):7715–7730, 1996.

[122] G. M. Tannin, A. K. Agarwal, C. Monder, M. I. New, and P. C. White. The human gene for 11 β-hydroxysteroid dehydrogenase: Structure, tissue distribution, and chromosomal localization. *Journal of Biological Chemistry*, 266(25):16653–16658, 1991.

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

[124] TIGR database (TDB) - the institute for genomic research. `http://www.tigr.org/tdb/tdb.html`, 1998.

[125] I. Tsigelny and M. E. Baker. Structures important in mammalian 11 β- and 17 β-hydroxysteroid dehydrogenases. *Journal of Steroid Biochemistry and Molecular Biology*, 55(5–6):589–600, 1995.

[126] I. Tsigelny and M. E. Baker. Structures stabilizing the dimer interface on human 11 β-hydroxysteroid dehydrogenase types 1 and 2 and human 15-hydroxyprostaglandin dehydrogenase and their homologs. *Biochemical and Biophysical Research Communications*, 217(3):859–868, 1995.

[127] K. I. Varughese, N. H. Xuong, P. M. Kiefer, D. A. Matthews, and J. M. Whiteley. Structural and mechanistic characteristics of dihydropteridine reductase: a member of the tyr-(xaa)3-lys-containing family of reductases and dehydrogenases. *Proceedings of the National Academy of Sciences of the United States of America*, 91(12):5582–5586, 1994.

[128] B. Wermuth. NADP-dependent 15-hydroxyprostaglandin dehydrogenase is homologous to NAD-dependent 15-hydroxyprostaglandin dehydrogenase and other short-chain alcohol dehydrogenases. *Prostaglandins*, 44(1):5–9, 1992.

[129] R. K. Wierenga, M. C. De Maeyer, and W. G. J. Hol. Interaction of pyrophosphate moieties with α-helices in dinucleotide binding proteins. *Biochemistry*, 24:1346–1357, 1985.

[130] R. K. Wierenga, P. P. Terpstra, and W. G. J. Hol. Prediction of the occurrence of the ADP-binding β-α-β-fold in proteins using an amino acid sequence fingerprint. *Journal of Molecular Biology*, 187:101–107, 1986.

[131] H. M. Wilks and M. P. Timko. A light-dependent complementation system for analysis of nadph:protochlorophyllide oxidoreductase: identification and mutagenesis of two conserved residues that are essential for enzyme activity. *Proceedings of the National Academy of Sciences of the United States of America*, 92(3):724–728, 1995.

[132] R. L. Winkler. *Introduction to Bayesian Inference and Decision*. Holt, Rinehart and Winston, 1972.

[133] P. Wolber and G. Warren. Bacterial ice-nucleation proteins. *Trends in Biochemical Sciences*, 14(5):179–182, 1989.

[134] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young. Large vocabulary continuous speech recognition using HTK. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 125–128. IEEE, 1994.

[135] F. Yu, T. Nakamura, W. Mizunashi, and I. Watanabe. Cloning of two halohydrin hydrogen-halide-lyase genes of *Corynebacterium sp.* strain n-1074 and structural comparison of the genes and gene products. *Bioscience, Biotechnology, Biochemistry*, 58(8):1451–1457, 1994.