Finding remote protein homologs with hidden Markov models

Kim Laurio

Department of Computer Science
University College of Skövde, Box 408
S-54128 Skövde, SWEDEN

HS-IDA-MD-97-08

Finding remote protein homologs with hidden Markov models

Kim Laurio

Submitted by Kim Laurio to the University of Skövde as a dissertation towards the degree of M.Sc. by examination and dissertation in the Department of Computer Science.

October 1997

I hereby certify that all material in this dissertation which is not my own work has been identified and that no material is included for which a degree has already been conferred upon me.

Finding remote protein homologs with hidden Markov models

October 1997

Abstract

Detecting remote homologs by sequence similarity gets increasingly difficult as the percentage of identical residues decreases. The aim of this work was to investigate if the performance of hidden Markov models could be improved by ignoring the subsequences that exhibit high variability, and only concentrate on the truly conserved regions. This is based on the underlying assumption that these high variability regions could be unnecessary, or even misleading, during search of remote protein homologs.

In this paper we challenge this assumption by identifying the high and low variability regions of multiple alignments and modifying models by focusing them on the conserved regions. The high variability regions are located with information theoretic measures and modeled by free insertion modules, which are special nodes that can be used to model arbitrarily long subsequences with a uniform probability distribution.

The results do not support a definitive conclusion since a few cases exhibit a performance increase, while the general trend is that the performance decreases when ignoring high variability regions. Two supplementary tests suggest that when there is a significant performance loss due to deletion of high variability nodes, a much smaller decrease occurs when the nodes are preserved but the position-specific amino acid distributions are removed. Taken together, these results support the hypothesis that there is some valuable information present in the high variability regions that enable the model to better discriminate between true and false homologs; and that other constructs for the high variability regions could perform better.

Table of contents

1 Introduction

Proteins are the machinery of the living cell [LBB+95], and their structure and function are of interest to pharmaceutical companies trying to find cures to various diseases [Coh96]. The function of a protein is inseparable from its structure [LBB+95]; therefore the determination of a protein's structure is crucial for an understanding of its characteristics [LBB+95], especially the location of the protein's active sites, which are the parts of the protein that interact with other molecules [LBB+95].

However, the determination of the structure of a protein is a complex process which may require several months of hard work by highly skilled researchers [BT91]. In contrast, finding the sequence of amino acids (the building blocks of proteins) is relatively easy [LBB+95], and the number of solved protein sequences is several orders of magnitude larger than the number of solved protein structures [SS91]. This, coupled with the increase in readily available computer power, has led to the emergence of a new discipline of scientific research: computational molecular biology [Tau96].

This new field of research is an interdisciplinary effort where computer science and molecular biology fuse together into an area of research that tries to find out how amino acid sequence determines protein structure and function [Pen96]. The ultimate goal is to be able to deduce the function of a new protein by knowing its sequence alone, or being able to create the necessary protein sequence of some desired functionality.

1.1 Sequence comparison

One approach to protein analysis is to compare the sequence similarity of two proteins, one with solved three-dimensional structure and one with unknown, and estimate how closely related the two proteins are, i.e. how homologous they are [THT94]. Analyzing multiple sequences of a protein family, i.e. a set of proteins sharing some degree of functional and structural similarity, reveals more information than a single sequence does [GLE90]. This is a consequence of how proteins evolve through time while preserving the basic functionality and structure [LBB+95]. Since the functionality of a protein is so closely tied to its structure, there will be parts of the proteins that are more conserved than others and locating these regions may provide important clues to the functionality of that protein [Alt91]. Various statistical techniques designed for identification and modeling of conserved regions have been proposed [BKM+96], and in this report we are mainly interested in the variant called hidden Markov models [Edd96].

A hidden Markov model (HMM) is a statistical model for sequential data. In this application the data represent the sequence of amino acids commonly occurring in a family of proteins [KBM+94], but HMMs can also be used for speech recognition [Rab89], or any other domain which can be represented as a series of measurements and where there is some underlying regularity. By training an HMM on a family of protein sequences, it becomes a ‘prototype’ which captures how the common parts of the family look like. That is, it is a model of a typical protein belonging to the modeled family and realizes a probability distribution over all protein sequences [KBM+94]. Given the model, it is possible to estimate how probable it is that any protein sequence could be modeled by that HMM, giving a relative estimate of how probable it is that the new sequence belongs to the modeled family [KBM+94].

Given a model and a set of sequences it is possible to produce a multiple alignment. A multiple alignment is a way to visualize how closely related the sequences are by aligning them to the common model, and it is produced by finding the enumeration of states in the given model that gives each sequence the highest probability [KBM+94].

However, modeling and creating multiple alignments of remote homologs introduces regions in the alignment with high variability which correspond to regions of the proteins that have diverged through evolutionary time. These high variability regions are problematic; even though they are not capturing a common pattern, they contribute to the calculation of the probability that a given sequence could have been produced by that HMM. Incorporating those regions into the model means that they will affect the score of the sequences, either positively or negatively depending on how well the current sequence matches the high variability region, and this decreases the predictability of these methods.

In the CASP2 contest, (the second Critical Assessment of Techniques for Protein Structure Prediction, initiated by John Moult from the University of Maryland's Center for Advanced Research in Biotechnology in Rockville [Pen96]), one of the conclusions was that the clear identification of (and removal of) regions of high variability in multiple alignments would have improved their evaluation of the data [KSB+97].

To deal with this situation more effectively, it was suggested that free insertion modules could be used for regions with high variability so that the HMM would focus on modeling the truly conserved regions [Sjö97]. A free insertion module (FIM) is a special construct of the HMM that does not have information about position-specific amino acid occurrences and realizes a 'uniform length function' of arbitrarily numbers of inserted amino acids. So, FIMs can be used to model regions of arbitrary length where the model building process cannot find a consensus pattern. Typically, they are used at the beginning and the end of a model to allow for variable positions of a conserved motif in different proteins [HK96].

One of the key features of HMMs is that they assign position-specific penalties based on statistical estimates [Edd96]. That is, if a sequence does not adhere to the model consensus in conserved regions it will be given a lower score than if it would had deviated from the model consensus in regions of high variability. FIMs take this principle a step further by capturing arbitrary long regions of high variability and in effect saying that those regions are unimportant, or even misleading, for the detection of proteins belonging to the current family.

1.2 Project aim

The aim of this project is to investigate if the power of hidden Markov models, as used in computational biology in the analysis of remote protein homologs, can be improved by using FIMs to eliminate regions of high variability from further consideration.

In general, this report will present our attempts at finding out whether high variability regions can be excluded when analyzing and modeling remote homologs. We proceed by removing information from the HMM in the nodes that are identified as trying to capture regions of high variability in multiple alignments, and evaluate the relative change in performance.

1.3 Structure of report

The rest of this report is organized as follows: Chapter 2 gives more details on proteins, their structure and what techniques are used to analyze them. It also includes an overview of HMMs, an explicit statement of the assumptions underlying this work and a concise look at related work. Chapter 3 presents the measures used to analyze multiple alignments with the goal of identifying high variability regions, and the method used to modify the HMMs. In chapter 4 the procedure used for validation of this approach is specified. The evaluation criteria used are presented and the data sets collected from protein sequence databases are introduced. In chapter 5 the results of the experiments are presented. It starts by describing the method for analyzing multiple alignments and modifying HMMs. Chapter 6 contains a discussion of the presented results and it is a preparation for the conclusion, which is stated in chapter 7.

2 Background

In this chapter we introduce proteins, protein structure(s) and their determination. We clarify how comparison of multiple proteins could give clues to their function and mention briefly a few techniques that have been developed for that purpose. The chapter ends with a brief look at closely related work.

2.1 Proteins - the workhorses of living organisms

The simultaneous interaction of billions of molecules is essential for the processes occurring in all living organisms. *Proteins* are responsible for many of the metabolic activities and the structural arrangements that creates the foundation to what we call life [LNC93]. Proteins are involved in the construction of cells, which in turn are used to create tissues, organs and whole organisms. They can function as storage of substances, e.g. oxygen and iron. They are also essential for the transportation of energy and waste products, as when oxygen is transported to muscles and carbon dioxide is removed (hemoglobins). Proteins also aid the construction and destruction of other proteins (chaperons and proteases, respectively), and help the organism to protect itself from alien or otherwise unwanted substances (immunoglobins or antibodies) [LNC93]. Other types of proteins can have a catalytic function in the organism and regulate the production of new proteins by increasing the production rate when there is a shortage and decreasing it when there are sufficient amounts of the desired product (enzymes). All of the above occur in a delicate balance and any malfunction of some aspect of these processes, e.g. too slow or fast reaction, are often observed as an abnormal state of being of the organism, i.e. it is 'ill' [LNC93]. If the malfunction is too severe then the organism cannot recover from it and eventually the life sustaining processes will terminate and the organism dies. One examples of what proteins that are not working properly can cause, is sickle-cell anemia which is the consequence of a change in the structure of the oxygen transporting protein hemoglobin [LNC93]. Since proteins are involved in so many different processes they are of tremendous interest to pharmaceutical companies and researchers in molecular biology who are trying to analyze and find cures for diseases and other malfunctions [Coh96, Wel97].

2.2 Protein structure

Proteins are complex molecules assembled from a set of 20 different building blocks, called *amino acids*, also commonly called *residues* when they occur in proteins. These molecules can be linked together into a chain by the *peptide bond*. See figure 2.1 for one example of an amino acid and the peptide bond.

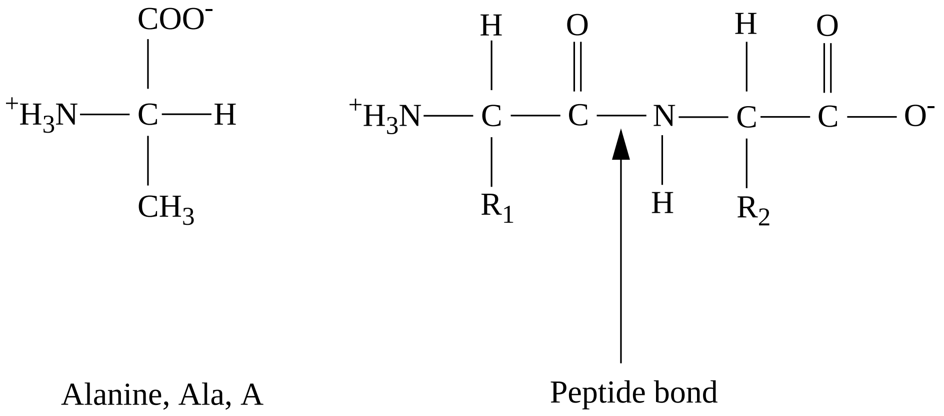


Figure 2.1: The structure of an amino acid and the peptide bond[1].

[1] Figure adapted from [LBB+95].

All the residues share a common core structure and differ only in their side chains, which determine their properties [LBB+95]. Some side chains are charged either positively or negatively and other are polar or nonpolar. One example of a nonpolar amino acid is Alanine, whose mnemonic is ’Ala’ and is denoted by the single letter ’A’. The 20 different amino acids have all been assigned single and three letter codes based on their names, e.g. the nonpolar amino acid Alanine has a three letter code, Ala, and a single letter code, A. Hence, a protein can be represented by the sequence of letters that denote its constituent residues. Short residue sequences are called *peptides* and longer sequences are called *polypeptides* [LBB+95]. Peptides are usually 20-40 residues long, while polypeptides can be up to 4000 residues long.

The structure of a protein can be regarded at four different levels [LBB+95]. The *primary structure* of a protein is the sequence of residues that it is built from. The *secondary structure* is the set of distinct substructures that the residue sequences can fold into, e.g. alpha helices, beta sheets and loops. Commonly occurring combinations of secondary structure elements are called *motifs*. One example is the helix-loop-helix combination which is common to proteins that bind calcium [LBB+95]. The *tertiary structure* is the overall three-dimensional shape of the protein and it is the highest level of organization for a protein consisting of a single residue chain. The *quaternary structure* is used to describe molecules that consist of several individual polypeptides, similar or dissimilar, that bond together to realize some functionality. Commonly occurring combinations of secondary structure elements are called *motifs*. One example of a motif is the helix-loop-helix sequence which is present in e.g. calcium-binding proteins [LBB+95].

The structure of a protein, i.e. the amino acid sequence and its three-dimensional shape, determines its function [LBB+95]. The structure and function of a protein is of interest to molecular biologists working at pharmaceutical companies and at research centers. A good example of the value of knowing the structure of a protein is finding effective drugs against HIV. Drugs designed to *inhibit*, i.e. counteract, the function of HIV's *protease* (a molecule that cleaves peptide bonds in proteins and is necessary for the correct assembly of the HIV molecule), have shown promising results in reversing the symptoms of AIDS [Coh96]. In order to be able to design effective substances with the desired effect, it was necessary to know the three dimensional shape of the protein and its function in cutting the HIV molecule to its proper size [Coh96]. This knowledge was used to design a drug which blocked the active site of the protease molecules.

However, determining the structure of a protein is a time consuming process which involves a lot of labor. In some cases it is not even possible, and when it is, it can take several years of hard work by highly skilled researchers [BT91]. At the same time, the rate of publication of newly sequenced proteins and genes is steadily increasing [GME87], and the number of proteins whose three-dimensional structure has been experimentally determined is just a small fraction of the number of available protein sequences [GLE90]. The ultimate dream would be to be able to predict the three-dimensional structure of a protein from the sequence of amino acids alone [Pen96].

2.2.1 Determination of protein structure

The quaternary structure of large molecules can be solved with electron microscopy and the primary structure can either be found directly, by chemical methods from the protein, or deduced from the encoding DNA and RNA [LNC93]. The two techniques that dominate when it comes to the determination of secondary and tertiary structures are *x-ray crystallography* and *nuclear magnetic resonance*, or NMR [BT91]. X-ray crystallography is done by analysis of the diffraction patterns of x-rays that are sent through a crystallization of a protein. This method requires a well-ordered crystal that will diffract x-rays strongly, and growing a crystal can take many months and may require several attempts where the controlling parameters are adjusted so that a useful crystal is produced. In addition to that, there is a problem with controlling the phase of the x-ray, and this requires several experiments where metal atoms, with known phase shifts, are inserted into the crystal. Thus, there is a number of steps which all have their own individual sources of error. To make the task even more difficult; above a certain level of resolution some chemical groups are indistinguishable and cannot be solved without prior access to the amino acid sequence. Therefore, knowledge of the amino acid sequence is essential for protein structure determination by x-ray crystallography. The following citation should give a good picture of just how difficult it is (from [LLB+95], chapter 3, p. 60):

> “The process is analogous to finding the precise shape of a rock from the ripples it creates in a pond.”

NMR, on the other hand, works by determining the structure of a protein from the magnetic properties of atomic nuclei. The technique involves placing a solution of the protein in a magnetic field and then finding individual hydrogen atoms that are close together in space by emitting and recording radio frequency pulses. The emission excites the atoms and when they revert to their original state, they send out a signal whose properties depends on the environment of and the identity of the current nucleus. However, atoms that are far apart in the sequence, but close in space, will interfere with each other to produce signals that could be hard to interpret without knowledge of the actual amino acid sequence. Therefore, knowledge of the primary structure is essential for determination of protein structure by NMR as well. A drawback of NMR is that it only works for relatively small molecules [BT91].

2.2.2 Analysis of protein primary structure

The function of a protein is inseparable from its structure and that in turn is determined by the sequence of amino acids that it is built from [LBB+95]. However, the fold of a protein is the result of a process that is not completely understood and a lot of research is devoted to revealing the secrets of protein folding [Pen96]. The final shape of a protein is believed to emerge from noncovalent[2] interactions between the amino acids at different positions of the protein, both nearby and far away [LBB+95]. Proteins can renature (or refold) by themselves and that indicates that the information necessary for the folding process is present in the amino acid sequence alone [LBB+95]. Therefore, if one could predict the structure of a protein from its amino acid sequence alone, the dependence on the tedious techniques described in the previous subchapter could be significantly reduced. One approach to protein analysis is to look at sequence similarity and estimate how closely related they are, i.e. how homologous they are, and from that estimation deduce that the three-dimensional shape of the molecules will resemble each other [SS91]. The underlying assumption here is that proteins that are related in some way, by structure and functionality, will have similar residue sequences. This is not true in all cases since proteins can be very similar structurally (and functionally), and yet have very different amino acid

[2] There are a number of different interactions between atoms and molecules. They can roughly be divided into either covalent or noncovalent. For more detail, the interested reader is referred to [LBB+95].

sequences [HS96a]. Still, any information that can be gained from analyzing the residue sequence alone is potentially useful, since the primary structure can be solved by sequencing robots and the rate of new sequences solved wastly exceeds the number of solved three-dimensional structures [GME87].

2.3 Remote homologs

Proteins evolve just as organisms do. However, to be able to function it has to retain the basic shape of and certain residues that constitute the *active site(s)*. An active site of a protein consists of those amino acids that cooperate in recognition of the target molecule and the catalyzation of some process [LBB+95]. These amino acids can be far apart sequence-wise, but usually are close together in space. If the shape of the molecule changes too much, then it probably cannot connect to the target molecule or looses the capability to perform its function. This means that, through evolutionary time, certain areas of proteins have been conserved and can be used to detect relationships between proteins [LNC93]. The more distantly related the proteins are, while still accomplishing the same basic functionality, the more the *conserved regions* will focus on the active sites and the structurally critical elements [LNC93]. Therefore, the analysis of *remote homologs*, i.e. proteins that have diverged through evolutionary time while still preserving the same overall function and structure, can give important clues as to where the active sites are located and which residues are important for the structure of that family.

The fact that proteins having a similar structure and possibly function can be encoded by quite different sequences of amino acids [HS96a], makes the identification of remote homologs particularly difficult. At the same time, the more remotely related they are, the more informative clues would they give to the functionality of that protein family. A *pro-*

tein family is defined as being a set of proteins that share at least some degree of structural and functional similarity [LNC93].

2.4 Protein sequence comparison techniques

Comparing protein sequences is complicated due to a number of reasons [SK83]. One is that even sequences coming from the same family are often of different length. This is due to the fact that when protein molecules evolve, they can have insertions and deletions relative to each other. Another factor is that the amino acids of which they are built can be divided into groups corresponding to different chemical properties, e.g. charge, size and whether they like to be in contact with water or not [LNC93]. This means that even if the two compared sequences do have different amino acids at a particular position, they can still be considered as very similar since the two amino acids share some properties. According to Joseph B. Kruskal (in [SK83]), some of the central themes of sequence comparison are:

- distance functions (e.g. when different amino acids sharing some chemical properties can be considered very similar);
- optimum correspondence between sequences (finding the best alignment of subsequences);
- dynamic programming algorithms for calculating distances and optimum correspondence between (sub)sequences.

Finding the optimum correspondence between two or more sequences is not a trivial task. The computational complexity of aligning two sequences of length n has complexity proportional to n^2, using a basic dynamic programming approach [SK83]. Considering that it is not uncommon to have dozens of sequences with lengths up to several hundred residues, it is easy to see that creating a *multiple alignment* (MA), i.e. an alignment of more than two sequences, quickly becomes intractable, even on a powerful computer. However, if all you have is one sequence and you want to find homologs to it, there are a number of algorithms published (and accompanying search tools) for the purpose of pairwise sequence comparisons. Two of the most notable are BLAST [AGM+90] and FASTA [PL88], which both use heuristics to speed up the similarity estimates of two sequences.

A multiple alignment of several sequences contains more information than a single sequence alone [GLE90], since it is possible to gain clues as to where the active sites are and which regions encode for structurally critical parts of the protein, by identifying the subsequences that have been conserved through evolutionary changes. In figure 2.1, with the aid of the color coding based on amino acid types, it is easy to detect the fully conserved columns and compare them with regions which exhibit a higher degree of variability.

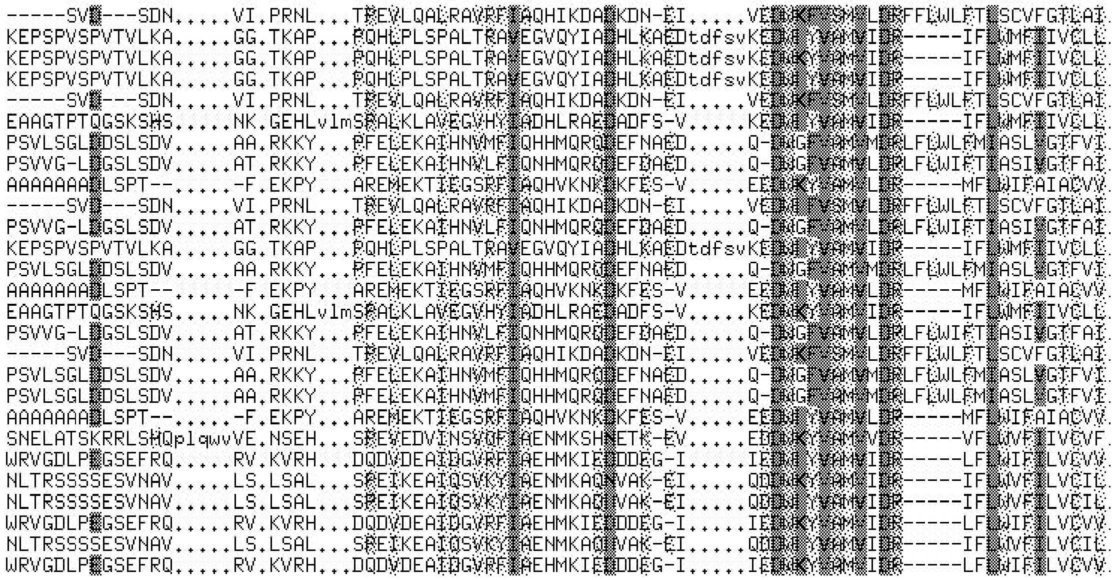


Figure 2.1: An example of a multiple alignment[3]. The different shades of grey represent color coding schemes set by residue type, e.g. hydrophobicity (residues which do not attract water molecules), and can be used as a visual guide for detecting conserved regions.

By somehow extracting the information present in a MA into a statistical model, that model can be used to search for other homologs by calculating their scores with respect to the model. A number of models have been proposed for this purpose, including consensus sequences, weight matrices, profiles and hidden Markov models [BKM+96]. A *consensus sequence* is a string which captures the most commonly occurring symbols in each position of the multiple alignment together with possible alternatives or some allowance for mismatches [BKM+96, Edd95]. A *weight matrix* is essentially a table with position-specific scores for each amino acid [BKM+96]. A *profile* can be seen as the combination of both the consensus sequence and the weight matrix techniques since it is a matrix of fixed length with position-specific scores for each amino acid [GLE90]. However, the main dif-

[3] Picture created with Belvu, a tool for visualization of multiple alignments.

ference between the profile and the previous methods is that profiles allow for deletions and insertions relative to the consensus [GME87]. A significant drawback of all of these methods is that they require an existing multiple alignment.

2.4.1 Hidden Markov models

Hidden Markov models (HMMs) are probabilistic models for sequential data. *Sequential data* can be either *continuous* (as speech signals) or *discrete* (as protein sequences). HMMs are stochastic machines that, using a hidden sequence of states, produces a string of symbols. An HMM consists of: a set of states; an alphabet of symbols that may be emitted from the states; a set of transitions between the states; a state transition probability distribution; a symbol emission probability distribution and the initial state distribution. The HMM used in this work is a *linear, left-right model* since for any enumeration of valid paths, the states visited are restricted so that they proceed from left to right (or remain in the same state) [Rab89]. See figure 2.2 for an example of the topology of an HMM.

The HMM used in this project has an inherent assumption that protein sequences are generated as random and independent samples from some underlying probability distribution. This is due to the *Markov property* which, simply put, states that all the information for deciding what to do at timestep *n+1* is present at timestep *n*, and that is all the information you need [Rab89]. That is, you do not need any information from *n-1, n-2* etc.

When using HMMs, the actual set of states visited is hidden from the user. That is why they are called *hidden* Markov models [Edd96].

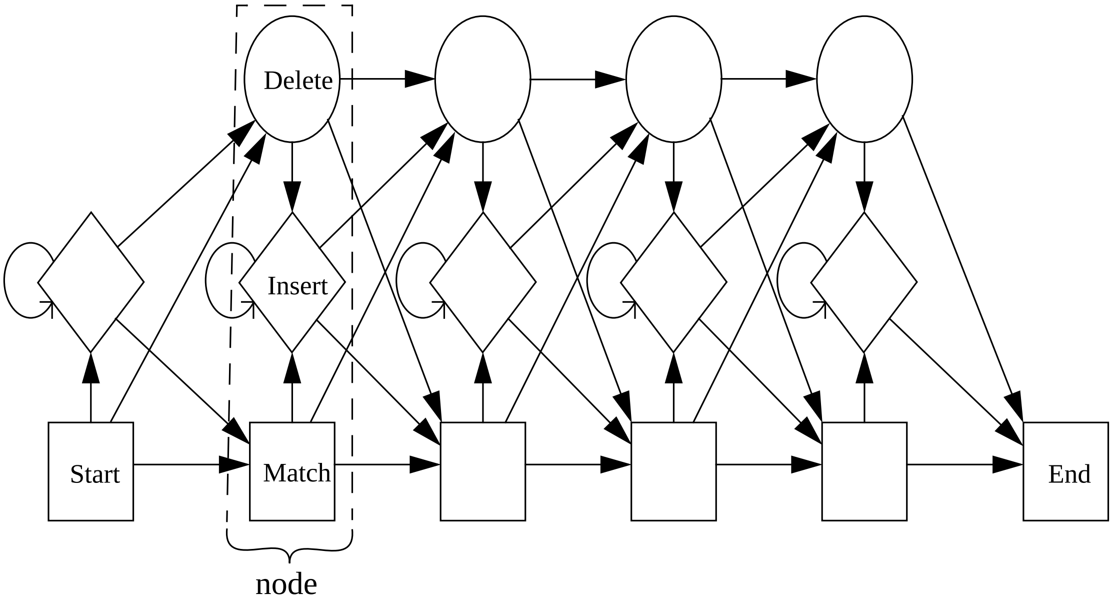


Figure 2.2: A visualization of the topology of a hidden Markov model[4].

The linear, left-right HMM is built from *nodes*, which have three fundamental states: the match, the delete and the insert states. The match states include a position-specific probability distribution over the 20 amino acids, and can be seen as representing the backbone of the model. The delete states enable a sequence to skip a position relative to the consensus and the insert states capture the insertions that might occur between two match states. Without the insert and the delete states, the sequence of match states would represent a gap-less profile. In other words, the insert and the delete states handle the exceptions to the detected pattern. The insert states all share one identical amino acid distribution that models the background frequency of the training set. In addition to these there are special types of nodes, called *free insertion modules* (FIMs), that are added at the beginning and at the end of an HMM. FIMs consist of a delete state and an insert state (figure 2.3). In a FIM the

[4] This is the topology used in SAM, the Sequence Alignment and Modeling tool suite developed at UCSC.

probabilities are set to unity on the reflexive transition in the insert state and on the transition from the delete state to the insert state. That is, FIMs realize a flat distribution of insertion lengths and do not penalize sequences that enter the insert state of a FIM. The amino acid distribution in the insert state of a FIM is the background distribution for the training set. Typically, they are used at the beginning and the end of an HMM to allow for variable positions of the model in different protein sequences [HK96]. All the states have three outgoing edges, except for the node immediately preceding the FIM (in that node the states have two outgoing edges).The FIM has no match state (the rectangle) and all the sequences are forced to pass through the delete state (the circle). Those sequences that ‘select‘ to enter the insert state for some arbitrary number of inserts, will not be penalized since both the edges traversed are set to unity. The ‘x’ and the ‘y’ in figure 2.3 indicate that the probabilities are set to identical values on the outgoing edges.

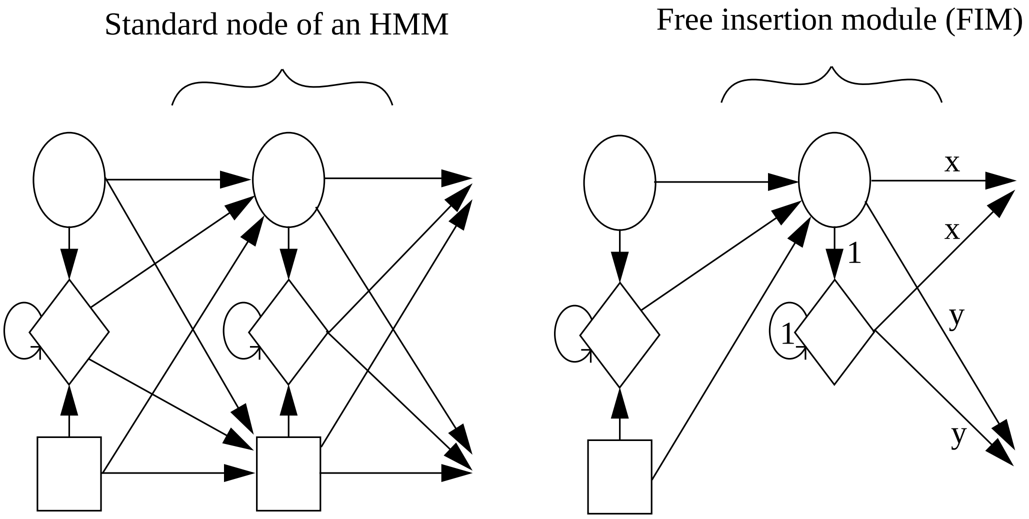


Figure 2.3: A FIM versus a standard node of an HMM.

The HMM can be trained on an unaligned set of sequences and it treats insertions and deletions in a formal probabilistic manner, whereas the previously mentioned profile methods use *ad hoc* techniques, e.g. trial and error techniques for setting deletion/insertion scores [Edd96]. The training process is an iterative process which strives to maximize the likelihood of the model with respect to the training sequences [KBM+94]. Creating a multiple alignment involves calculating the path through the model that yields the highest score for each and every sequence. The score for a sequence, given the model, is the product of the state transition probabilities and the symbol emission probabilities. Since the HMM is not modeling any higher order correlations between the amino acids, i.e. they are independent, the symbol emission probability is only dependent on the current state.

The basic score of an HMM, the *negative log-likelihood*[5] (-log(P(*s*|*m*)) score, or *NLL*, is very dependent on the length of both the sequence *s* and the model *m* [HBK96]. The solution is to compare the NLL to a *NULL model*, which is used as a ‘default’ approximation for all possible sequences, and calculate the log-odds score [HBK96]:

$$score(s) = \log\left(\frac{P_m(s)}{P_{NULL}(s)}\right)$$

This way the score becomes a measure of whether it is more probable that the sequence was modeled by the model *m*, than by the *NULL* model. To separate the true positives from the false ones, the score must be used in conjunction with a *significance threshold*,

[5] The base of the logarithm is in this report always assumed to be 2, unless explicitly stated otherwise. The unit of the result then is called bits.

which is set to give the best separation between sequences belonging to the family and those not [GHK+96].

The NULL model is basically a FIM with the background amino acid distribution estimated from the training sequences. Since the score of an HMM is with respect to the NULL model, the regions that are being represented by FIMs in the HMM will be ‘masked out’. That is, those subsequences that are modeled by FIMs are being ignored.

For a high level introduction to HMM, with a lot of pointers to further reading, see [Edd96]. For a more in depth tutorial on the use and properties of HMMs (and their application in speech recognition), see [Rab89].

2.5 Protein sequence databases

There is an increasing number of publicly available databases that have information about biomolecules. The ones relevant to this project will be described briefly.

SWISS-PROT contains protein sequences with accompanying documentation [BB94]. The information that is available includes e.g. comments on the function of the listed proteins, present domains and active sites, secondary and quaternary structure and similarities to other proteins. It is maintained by the Department of Medical Biochemistry of the University of Geneva and the EMBL Data Library.

Pfam is a database of multiple alignments of protein domains or conserved protein regions and their corresponding HMMs [SED97]. The method for creating multiple alignments is a compromise between human involvement and automatic techniques. Pfam includes ‘seed’ alignments created by an expert on nonredundant sequence families, and the ‘full’ alignments are created from all of SWISS-PROT with an HMM that is created with the ‘seed’ alignment. Pfam was originally developed by Erik Sonnhammer and Sean Eddy. It is maintained at the Sanger Centre in UK.

Expasy [ABH94] is a world wide web front-end to a number of databases that are maintained in Geneva, Switzerland. Some of the databases are: SWISS-PROT, Prosite, Swiss-2Dpage, Swiss-3Dimage, Enzyme, CD40Lbase and SeqAnalRef and Prosite.

Prosite is a database of commonly occurring sites, patterns and profiles, which is intended to support the determination of the function of a newly sequenced protein [Bai93]. It is maintained by the Geneva University Hospital and University of Geneva, Switzerland.

The Protein Data Bank (PDB) is a database of biological macromolecules (e.g. proteins, peptides and viruses) with experimentally determined three-dimensional structures [ABB87]. The documentation includes atomic coordinates, citations and information on the primary and the secondary structures.

The homology-derived secondary structure of proteins (HSSP) is a database of proteins where structural homology has been predicted from sequential homology with the use of an empirically derived significance threshold. This threshold is based on the percentage of identical residues required at varying protein sequence lengths [SS91]. That is, for each protein with known structure in PDB, the HSSP lists the proteins that have been inferred to be homologous to it, based purely on sequential similarity above the empirically defined threshold.

The fold classification based on structure-structure alignment of proteins (FSSP) [HS96a] is a continuously updated database with an all-against-all structural comparison of the proteins present in PDB. All structural similarities are based on alignments of the three-dimensional coordinates of the proteins.

2.6 Related work

Meta-MEME is a software tool that builds HMMs that focus on the conserved regions of a protein family [GBE+97a and GBE+97b]. It starts by identifying the motifs occurring in the data set, and ranks them so that the strongest motifs are identified. Then the tool assembles them into a linear HMM with ‘spacer regions’ between the motifs, and the resulting model can be used to search databases. The spacer regions give an exponential distribution on lengths between the motifs, but the authors of Meta-MEME plan to incorporate an explicit probability distribution for the output length of an insert state [GBE+97b].

In contrast to Meta-MEME, no information about the length of the region between motifs was to be included in this project, i.e. the reflexive loop of the insert state of the FIM was retained at unity, giving a uniform distribution over insertion lengths.

3 Method

In this chapter we define the methods used in our effort to find out whether the exclusion of high variability nodes result in a better model or not (see section 1.1). We also define the formulas used for estimating the information content in the columns of a multiple alignment and we introduce informally the heuristic used for modifying HMMs.

The general idea is to use the information present in a multiple alignment to modify the corresponding HMM so that it only captures the truly conserved regions. The underlying assumption is that regions with high variability do not capture information that is relevant to the detection of other protein sequences that belong to the modeled family.

3.1 Analyzing multiple alignments

To be able to make modifications on the underlying HMM based on the information detected in a multiple alignment, the first prerequisite is to have a method for estimating the information content. Without a measure of information content there is no way to automatically and objectively identify the regions with high and low variability. For this project we have identified two measures that can be used for this purpose, namely the *entropy* and the *encoding cost* measures, which will be described shortly.

Both of these measures need probability distributions or estimates to work on, and to calculate these from small data sets can produce inferior results. To avoid these problems we use algorithms taken from the SAM software suite ([HBK+96]) which use *prior probability estimates* on amino acid distributions, in the form of *Dirichlet density mixtures* [SKB+96]. Dirichlet mixtures are a method of using prior information about amino acid distributions usually found in different contexts in proteins, e.g. based on their hydrophobicity, and it has been shown to improve performance when training models on small data sets [BHK+93]. The priors, i.e. the individual densities of a mixture, encode for various chemical properties of amino acids (size, charge etc.) and they are estimated from previously created HMMs or multiple alignments. The mixture is combined with a pseudocount method which gradually shifts the weight from the prior beliefs to the actually observed amino acid frequencies when the data sets get bigger, i.e. it interpolates smoothly from the prior to the observations. In practice they prevent the model from overfitting when estimating parameters on small data sets. The mixture used in this project consists of nine components, where each component is a density representing an amino acid distribution in some context. For more details, see Sjölander et al. [SKB+96].

3.1.1 Prerequisites for information content estimations

Given a count vector $\vec{n}$ of length 20 (where each element is the number of observations done of each of the 20 amino acids), the formula for the posterior probability estimation of seeing amino acid *i* in this particular context, is [SKB+96, p. 335]:

$$p_i = \sum_{j=1}^{l} Prob(\alpha_j|\vec{n}, \Theta) \times ((n_i + \alpha_{j,i})/(|\vec{n}| + |\alpha_j|)) \qquad \text{(F1)}$$

The posterior probability estimation p_i of amino acid *i* is a sum over all the components α_j of the Dirichlet mixture Θ. Each term is a product formed by the posterior probability of the component and the pseudocounts. That is, the first part implements a weighting scheme where the component of the prior which best matches the actually observed count vector returns the highest probability. The second part adds the pseudocounts $\alpha_{j,i}$ to the observed number of occurrences n_i of amino acid *i* and normalizes the sum. This formula is necessary for the entropy measure which requires a probability distribution on the alphabet we are using to be able to estimate the amount of information found in a single column of the multiple alignment.

The prerequisite for the encoding cost measure is a likelihood estimate of the observed amino acid distribution, with respect to the priors. This likelihood is calculated by the following formula [SKB+96, p. 335]:

$$Prob(\vec{n}|\Theta, |\vec{n}|) = \sum_{k=1}^{l} q_k \times Prob(\vec{n}|\vec{\alpha}_k, |\vec{n}|) \qquad \text{(F2)}$$

The probability estimation is a sum over all of the components of the Dirichlet mixture Θ. Each of the terms is a product of a mixture coefficient q_k, and the estimated probability of the observed count vector $\vec{n}$ given the component α_k and the total number of observations $|\vec{n}|$. So, the second part implements, similarly to formula F1, a weighting scheme where the observed distribution receives a probability estimate for each of the nine components of the prior. The first part is the mixture coefficient which implements a fixed weighting of the components, based on the behaviour of the data sets that the prior has been estimated from.

Now when we have dealt with the prerequisites, it is time to introduce the entropy and the encoding cost measures used for estimating the information content in each of the columns of the multiple alignment.

3.1.2 Entropy

The entropy measure presented by Shannon (introduced originally in the context of channel capacity) can be interpreted as a measure of the surprisal of some event. In this case, the value of a discrete random variable [CT91]. The entropy of a vector representing a probability distribution over some finite alphabet of discrete variables is defined by the following formula [CT91, p. 13]:

$$ENT(\vec{p}) = -\sum_{i=1}^{20} p_i \times \log\ p_i \qquad \text{(F3)}$$

This lends itself quite easily to the analysis of multiple alignments, where each column is supposed to carry information about the probability distribution over the 20-letter alphabet of the proteins. The input to this formula is an estimated probability distribution which we already discussed in formula F1 in the previous section.

3.1.3 Encoding cost

The encoding cost is a ‘distance’ measure [SK83] of the observed column, i.e. in this case the observed distribution of amino acids with respect to the prior. The encoding cost is defined to be the negative logarithm of a likelihood estimate for the observed column, given the prior [SKB+96, p. 336]:

$$ENC(\vec{n}) = -\log\ Prob(\vec{n} \mid \Theta, |\vec{n}|) \qquad \text{(F4)}$$

3.2 Modifying HMMs

With the quantifiable measures of information content in a column available, the next issue is how to modify the HMM that was used to create the multiple alignment in the first place. The simplest approach is to measure the mean and the variance of the entropy and encoding cost measures, and that is the method we use. With these in place it is straightforward to define a threshold with respect to the mean, in units of standard deviations, which will be used for demarcation between conserved and highly variable regions.

However, it is not desirable to chop the HMM into small discontinuous fragments, each with a FIM in between, since that would most likely destroy the capability of the HMM to identify the conserved regions. This is to ensure that we follow the assumptions made, that we want to focus on the conserved regions by ignoring the ones which exhibit high variability. The solution was to define a heuristic that ensured that the identified region with high variability had to be at least of length *l* contiguous columns before it is considered to be a candidate for modeling by a FIM.

So, to repeat, the steps are (see Fig. 3.1):

- Calculate a measure of information content for the columns in the multiple alignment (entropy or encoding cost);
- Calculate mean and variance of the selected measure;
- Use a threshold value and a minimum length limit to scan for regions with high variability;
- Delete the corresponding match states from the HMM and replace them with FIMs.

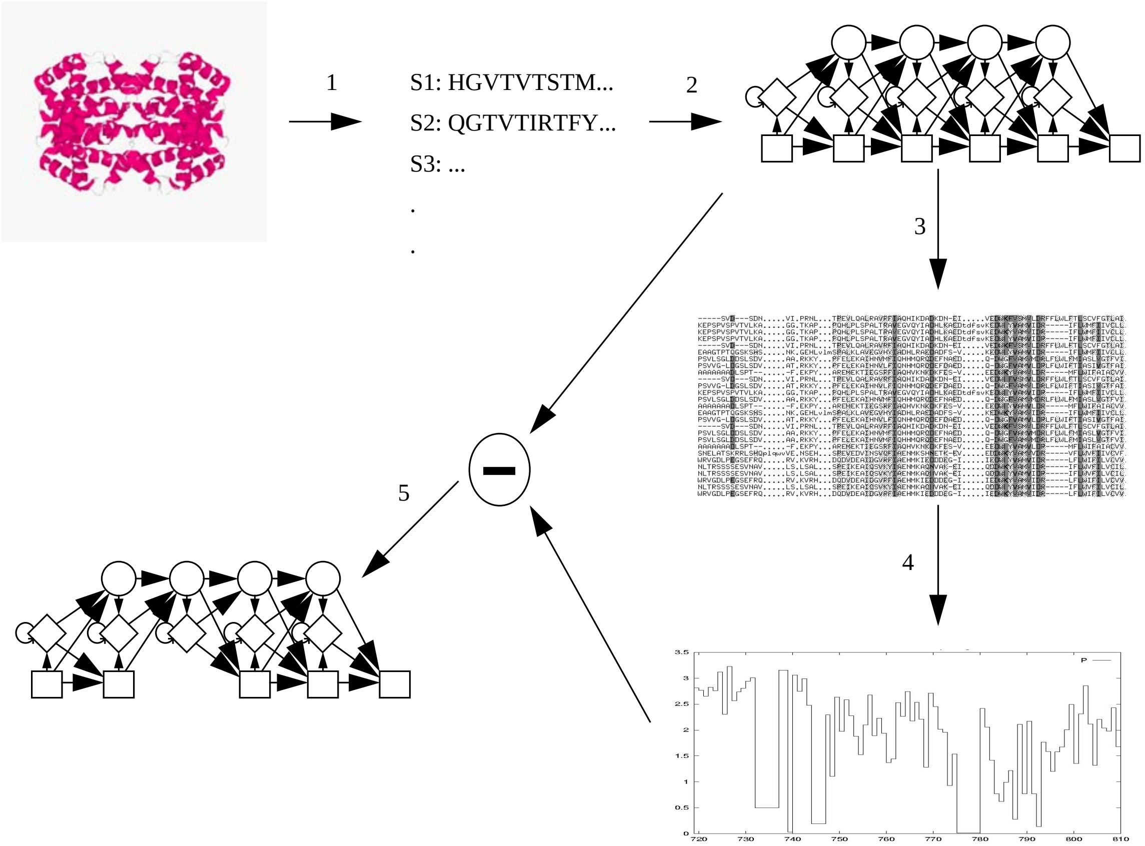


Figure 3.1: Visualization of the process described. Proteins are represented as strings (1); a model is trained on these strings (2); the strings are aligned to the model, creating a multiple alignment (3); the multiple alignment is analyzed to locate the regions with high variability (4), a new, and hopefully better, model is created by deleting the nodes that were capturing high variability regions and inserting FIMs (5).

The problem of evaluating this approach remains, and the method of choice is experimental validation, i.e. compare the relative performance of the ‘original’ HMM with the modified one and see which one performs best at remote homolog detection. However, to really elucidate the power/weakness of the FIM approach and to test the hypothesis, a set of experiments will be performed which gradually shift from data sets that, according to our assumptions, should be advantageous for the original method and over to other data sets with which the modified model should perform better.

4 Experimental validation

Here we present the evaluation criteria that we have defined for the experimental evaluation of the FIM-based approach. These criteria are mapped down to metrics, which can be studied individually or together.

4.1 Evaluation criteria

The basic task that we are trying to accomplish is to detect as many as is possible of the remote homologs to the modeled family, and clearly reject those that are not true homologs (Fig. 4.1). The desired result of the FIM-based approach is to cover a larger portion of the string family, while still rejecting strings that do not represent true homologs.

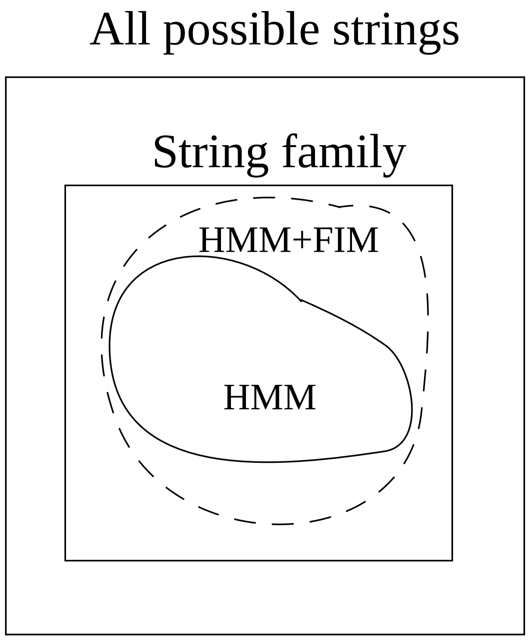


Figure 4.1: We want to cover as much as possible of the string family, while not including any sequences that lie outside. The universe of discourse is the set of all possible proteins. The string family denotes the protein family that is being analyzed and modeled.

Considering the above situation, it is sensible to define two metrics, which we call *sensitivity* and *specificity*, to evaluate the performance on these two dimensions. These two metrics are essentially the same as used by Gribskov and Robinson in Receiver Operating Characteristic (ROC) analysis [GR96].

Sensitivity is a measure of how well the model recognizes the remote homologs, i.e. how sensitive it is to sequence similarities. Specificity, on the other hand, is a measure of how well the model rejects unrelated sequences, i.e. how good it is at drawing a clear demarcation line between true and false family members. To test the generalization capability of the models, both of these metrics are calculated from scoring performance on unseen sequences. That is, for all experiments done, the model is trained on one set of protein sequences and tested on another set, and the intersection of these two is the empty set.

The raw data needed for evaluation purposes is produced by calculating the optimal path through an HMM for each of the sequences in some protein database. This produces a file where the highest scoring proteins are listed first and in order of decreasing similarity to the model[6]. Depending on how focused the model is on the modeled family and how many sequences there are in the database, it is necessary to define a *critical value* which is to be used as a *threshold* for separating true positives from false ones [GHK+96]. Even if an appropriate threshold level has been identified (e.g. 99% of the database is estimated to be false hits), that does not give any detailed information on how the true and false positives are distributed above the threshold. For example, models A and B can be very different depending on how well they cluster the positive sequences from the false ones. Using a fixed threshold value A can be thought of as being as good as B, while in reality it is not.

[6] This description applies to SAM.

To avoid these issues we use Receiver Operating Characteristic analysis to evaluate the performance of HMMs [GR96].

ROC curves are constructed by plotting the fraction of positives as a function of the fraction of negatives [GR96]. Files with positive sequences mixed randomly with negative ones will produce a line of slope one, indicating that the model has no capability to discriminate between homologous and non-homologous sequences. The better the model performs, the more the curve will be pushed to the upper left corner. The area below the ROC curve is the probability of a positive sequence getting a higher score than a negative [GR96].

In figure 4.2 the behaviour of the ROC curves for different score distributions is visualized. On the left two result files are depicted. They consist of an ordered enumeration of all the protein names from a protein sequence database. The ordering is based on the scores that the sequences receive when they are aligned to some model. The black dots represent true homologs and the white dots denote the non-homologous protein. The ROC curves on the right are produced by starting with the highest ranking protein and plotting the fraction of positives that it represents as a function of the fraction of negatives detected. The relative performance of the two models is estimated from the difference in the area below the ROC curve. Analyzing the result files this way gives an indication of the scoring distribution of the searched protein sequences, without having to worry about critical values necessary for some significance level. Another technique for estimating this property would be to produce histograms and see how clear the separation is between positives and negatives. However, ROC curves give a quantifiable measure of this separation which is easy to calculate and the area has direct relevance to the probability of correct classification of new sequences [GR96].

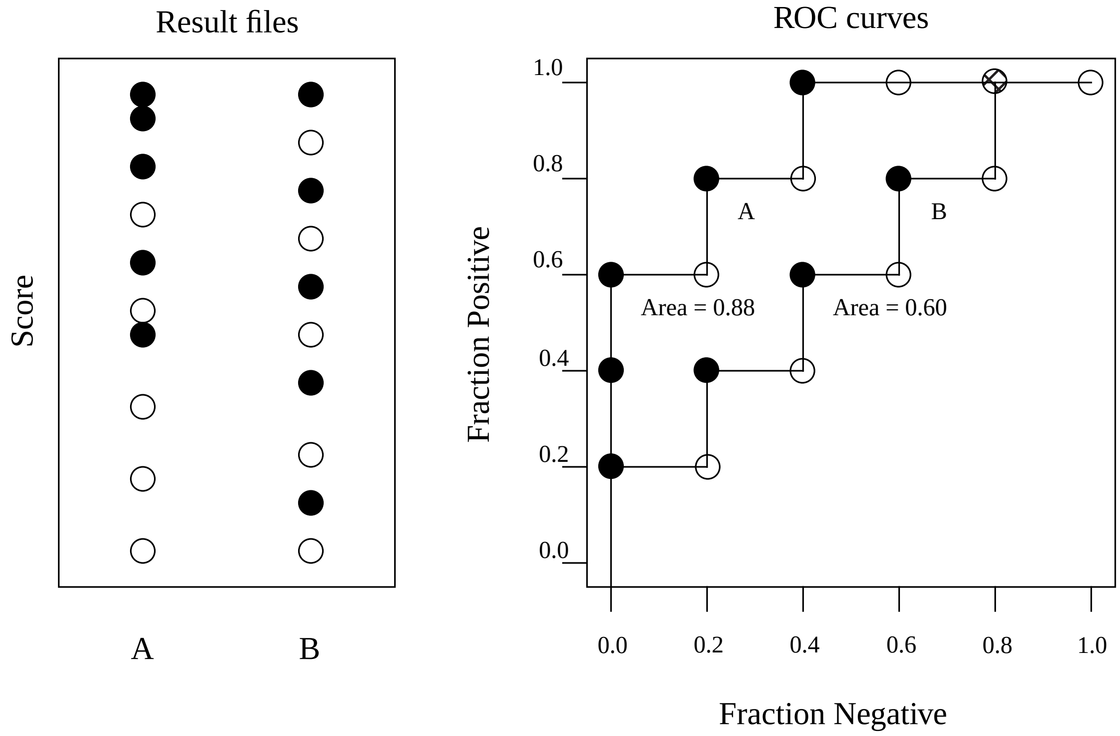


Figure 4.2: Example of an ROC curve and how good (A) and bad (B) distributions look like and how they affect the area under the curve. The distributions represent the protein sequences ordered by the score they receive for some model. This approach for visualization of the behaviour of ROC curves is similar to the one that Gribskov and Robinson use in [GR96]. See the text for further explanations.

When searching large databases, usually the number of positives is wastly outnumbered by the negatives. This gives that for most models the area under the ROC curve approaches one fairly quickly and the precision has to be increased to be able to analyze the performance differences. Given that the results from a database search are not considered reliable after the detection of several tens of false hits, Gribskov and Robinson define a slightly modified performance metric that stops after seeing 50 false hits, called ROC_{50} [GR96]. ROC_{50} is the evaluation metric used in the present work.

The next question is how to define the positives and the negatives. In this project the protein sequences divide into two slightly different test procedures. The first set (three families) is created by downloading all the Prosite [Bai93] identified family members from SWISS-PROT [BB94], creating models from a subset of those sequences and then searching a local copy of SWISS-PROT. All the sequences not listed by Prosite were assumed to be negatives, i.e. not belonging to the modeled protein family. The second set (two families) is created by using the HSSP [SS91] identified sequences for training and the FSSP [HS96a] identified sequences for testing in searches of a local copy of PDB [ABB87]. The remaining sequences in PDB, i.e. the ones not listed in either HSSP or FSSP, were used as negatives.

4.2 Data sets

The data sets were selected so as to contain both sequence families where the standard HMM was believed to perform well and other where the HMM with FIMs was believed to perform better (according to our prior assumptions). By visual inspection of multiple alignments we selected the following five families:

- Globin family
- Ferredoxins, 4Fe-4S subfamily
- ADH short-chain dehydrogenases/reductases family
- Remote homologs of 1try
- Remote homologs of 1hurA

Each of these families will be described very briefly. None of the families was selected because of some specific functionality that it may exhibit.

4.2.1 Globins

The globin family consists of several subfamilies and they are occurring in a large number of different organisms. They are involved in the transportation and binding of oxygen molecules[7]. The globin family is a well studied set and the protein sequences have evolved relatively uniformly over all of the sequence lengths, which was reflected by the fact that the initial model developed for this family contained 140 nodes while the average length of the sequences in the training set was 146. That is, the model of the globin family contained few inserts with respect to the length of the ‘consensus’ sequence. This family has been used for the evaluation of HMM performance before, see for example [HBK96, KBM+94 or BKM+96].

The downloaded set consisted of 685 sequences, of which 500 were selected randomly for the training set. No manual scanning of the protein sequences was done to remove duplicates or fragments.

4.2.2 Ferredoxins

The structure of the 4Fe-4S family includes two distinct domains consisting of twenty six amino acids each. Both of these domains have four very conserved cysteines that bind to a 4Fe-4S center[8]. The ferredoxin group is involved in electron transfer in a number of processes.

[7] Description adapted from Prosite document PDOC00793.

[8] Description adapted from Prosite document PDOC00176.

The total set used consisted of 150 sequences, of which 125 were randomly selected for the training set. No manual scanning of the sequences was done.

4.2.3 ADH short-chain dehydrogenases/reductases

This is a very large family of enzymes where most members are protein sequences with lengths around 250 to 300 residues[9]. Visual inspection of the multiple alignment at Pfam suggested that this family is a good mix of both high and low variability regions with varying lengths in between the conserved positions.

The total set used counted 183 sequences and out of those 129 were randomly selected for the training set. No manual scanning of the sequences was done.

4.2.4 Remote homologs to 1try

The trypsin family belongs to the proteases, which are proteins that degrade other proteins [LBB+95]. The regions around the active sites are well conserved and if a protein includes both of the two specified active site signatures, then the probability of that protein belonging to the serine protease family is 100%[10]. The 1try family was selected because it was identified in the CASP2 experiment (by the jury) as being the structural homolog of one of the target sequences that the teams had to produce predictions on. It was included to be used for comparison of the capability of the original HMM and the modified HMM to mark it as homologous to the target sequence.

The HSSP-identified homologs of 1try consisted of 213 sequences and 180 of those were used for the training of the initial model. A total of 200 protein sequences were marked as being homologous to 1try in the FSSP database. They were all included in the test set. One

[9] Prosite document PDOC00060.

[10] Prosite document PDOC00124.

of the target sequences from the CASP2 experiment (T0031) was also included in the test set.

4.2.5 Remote homologs to 1hurA

1hurA (human ADP-ribosylation factor 1) is involved with protein trafficking[11]. 1hurA has been used previously for evaluating the performance of HMMs in remote homolog recognition, and that was the reason for this family being included in the experiments [HBK96]. The HSSP-identified homologs of 1hurA consisted of 125 sequences and 73 of those were selected for training. The FSSP database identified 155 structurally homologous sequences, which were used for the test set.

4.3 Procedure

The basic approach is to build an initial model; use that model to create a multiple alignment and use the information revealed in that multiple alignment to modify the initial model by removing the states that are capturing high variability regions. However, to get a better estimate of how the FIM-based method works in different situations, we implement a sequence weighting scheme, which is employed to create models at different levels of *generalization*.

4.3.1 Levels of generalization

When creating models from protein sequences the most frequently occurring pattern will dominate over the less frequent ones [GHK+96]. This means that if the training set is skewed, the model will be overfitting on the subfamily which was overrepresented. To counteract this, we will create four models at different levels of generalization; at 0.3, 0.5,

[11] Prosite document PDOC0781

0.7 and 1.0 bits of encoding saving. The bits of generalization denote the *average encoding saving in each match state of the model with respect to the background distribution of amino acids*. So, the higher the encoding saving that we specify, the bigger the 'distance' is between the background distribution and the match state distribution [GHK+96]. This means that low values of encoding saving makes the model more general or 'fuzzy', i.e. the match state distributions are heavily influenced by the prior that we are using in training; and high encoding saving means that we let the match state distribution be closer to the frequency distribution of the training set [GHK+96][12]. It is called encoding *saving*, since the description of a random distribution requires more data (or bits if we are using the binary system) than a distribution with inherent regularities, patterns or other characteristics which makes it less random.

This generalization procedure also tests where the performance changes occur, i.e. if they come from 'fuzzy' models, indicating usefulness when training on small data sets, or from 'focused' models, indicating that the method works best for larger data sets.

4.3.2 Selecting threshold levels for the entropy and encoding cost measures

Due to time constraints a three-step procedure was selected for setting the user-controlled threshold values used when analyzing multiple alignments. Since the range of the entropy and the encoding cost measures are very different, the procedure is slightly different for the two measures.

For the entropy measure, the first step used either the default value or, if no nodes were deleted, the threshold was adjusted until a deletion did occur. The default value is always

[12] For more details on the sequence weighting scheme (unpublished), contact Kevin Karplus at UCSC.

the mean over all of the columns of the multiple alignment. The defaults for the two remaining thresholds were set to one and two standard deviations below the mean. The exceptions are the 1hurA and the ADH families where the defaults were felt to be too dramatic (i.e. major performance losses), and instead the thresholds were set so that they represented increasing numbers of deletions.

The encoding cost measures are different from the entropy measures and therefore the default values were selected differently. The first threshold was set to the mean of the values. The other two were set to one half and one standard deviation above the mean. In the trypsine and the ferredoxine experiments a threshold of one standard deviation below the mean was used instead of the mean. The second and the third thresholds were set to the mean and one standard deviation above the mean. This was done to see how the model would perform when almost all the information was removed. The selected threshold values are summarized in table 5.1.

4.3.3 Testing procedure for the globin, ferredoxin and adh families

The complete testing procedure was as follows: The multiple alignment for the selected families were inspected at the Pfam [SED97] site and the corresponding sequences were downloaded from SWISS-PROT [BB94] using Prosite [Bai93], at the Expasy Molecular Biology Server [ABH94]. The sequences were partitioned into non-overlapping training and test sets. A initial HMM was created using the default settings for SAM version 1.3. A multiple alignment was created of the training sequences with the initial model. Four generalized versions of the initial HMM were created at 0.3, 0.5, 0.7 and 1.0 bits of generalization. The multiple alignment was analyzed with both the entropy and encoding cost measures at three different threshold levels. The resulting set of models (a total of 24) were used to score all of a locally installed copy of SWISS-PROT (release 34). The resulting files, with a score for each of the sequences in SWISS-PROT, were scanned to calculate the two metrics defined in section 4.1. Finally, the relative performance was obtained by subtracting the sensitivity and specificity scores of the generalized model from the scores of the entropy- and encoding cost-modified ones.

4.3.4 Testing procedure for the 1try and 1hurA remote homologs

The testing procedure was as follows: The HSSP-identified homologs to the two selected families were downloaded and initial models were created. Generalized versions of the original models were created at 0.3, 0.5, 0.7 and 1.0 bits of generalization. Multiple alignments were created for the training sequences. The multiple alignments were analyzed with both the entropy and the encoding cost measures. A number of modified models were created to a total of 48, i.e. 24 each for the 1try and 1hurA families. A local copy of PDB was appended with those of the HSSP- and FSSP-identified homologs that were missing. The superset of PDB was then scored with all the created models and evaluation metrics were calculated.

5 Results

The first step was to design, implement and test the created software for analyzing multiple alignments (AMA). AMA was designed according to the ideas and methods described in section 1.2 and chapter 3. Testing the software was done by visual inspection of the columns of the multiple alignment that the algorithm identified as belonging to regions of high and low variability. The documentation present in protein sequence files was used to verify that structurally and functionally important residues were preserved. Next the modification of HMMs was scrutinized to see that it really did identify the nodes corresponding to high variability regions and that the introduced changes did not disrupt the basic functionality of the HMM.

The minimum length of consecutive residues that had to be identified as belonging to a high variability region was set to six. This was done to protect against extreme fragmentation of HMMs.

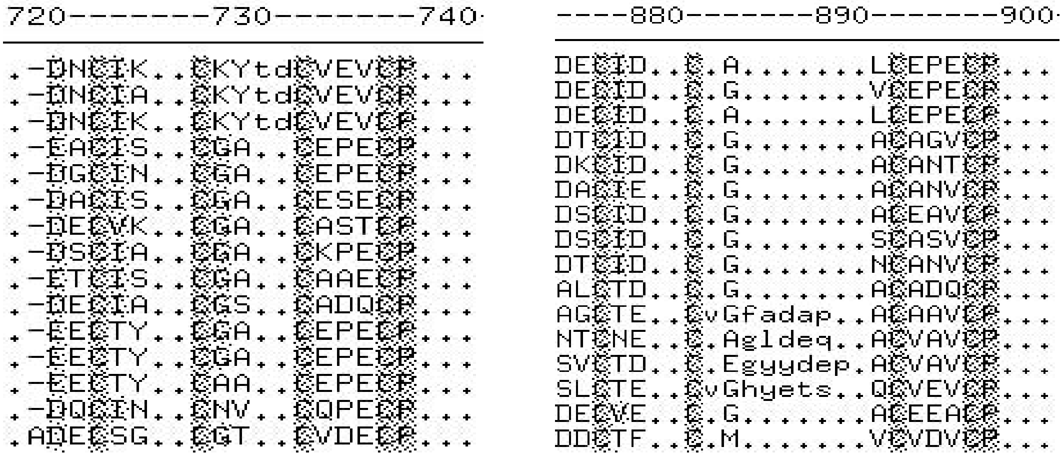
5.1 Analyzing multiple alignments

The 4Fe-4S subfamily of ferredoxins will be used for visualization of the behaviour of the algorithm. As mentioned in section 4.3, the ferredoxins are involved in electron transfers in a number of metabolic reactions. The active sites of this family consist of two domains with four cysteine residues[13].

To see that the developed software did behave accordingly with respect to what is known about the 4Fe-4S family, an HMM was trained on a subset of the downloaded protein

[13] See e.g. the FER1_AZOVI entry in SWISS-PROT and Prosite document PDOC00176.

sequences and a multiple alignment was constructed by aligning the sequences to that model.



```
720-------730-------740·
.-DNCIK..CKYtdCVEVCP...
.-DNCIA..CKYtdCVEVCP...
.-DNCIK..CKYtdCVEVCP...
.-EACIS..CGA..CEPECP...
.-DGCIN..CGA..CEPECP...
.-DACIS..CGA..CESECP...
.-DECVK..CGA..CASTCP...
.-DSCIA..CGA..CKPECP...
.-ETCIS..CGA..CAAECP...
.-DECIA..CGS..CADQCP...
.-EECTY..CGA..CEPECP...
.-EECTY..CGA..CEPECP...
.-EECTY..CAA..CEPECP...
.-DQCIN..CNV..CQPECP...
.ADECSG..CGT..CVDECP...
```

Active site 1

```
----880-------890-------900·
DECID..C.A.......LCEPECP...
DECID..C.G.......VCEPECP...
DECID..C.A.......LCEPECP...
DTCID..C.G.......ACAGVCP...
DKCID..C.G.......ACANTCP...
DACIE..C.G.......ACANVCP...
DSCID..C.G.......ACEAVCP...
DSCID..C.G.......SCASVCP...
DTCID..C.G.......NCANVCP...
ALCTD..C.G.......ACADQCP...
AGCTE..CvGfadap..ACAAVCP...
NTCNE..C.Agldeq..ACVAVCP...
SVCTD..C.Egyydep.ACVAVCP...
SLCTE..CvGhyets..QCVEVCP...
DECVE..C.G.......ACEEACP...
DDCTF..C.M.......VCVDVCP...
```

Active site 2

Figure 5.1: Two fragments from the multiple alignment that are modeling the regions in the vicinity of the two active sites in the 4Fe-4S family of ferredoxins[14]. The shaded columns represent a color coding scheme by residue type.

From the multiple alignment it is clear that the trained HMM does capture some of the structurally and functionally important regions of the protein sequences. Figure 5.1 contains the columns of the multiple alignment which were found to be modeling the regions in the vicinity of the active sites. In both of the regions the four cysteine residues are clearly marked as being common to this family.

Applying the developed software on the whole multiple alignment results in an estimation of the information content or the variability in each of the columns. A typical result for the entropy measure can be seen in figures 5.2 and 5.3.

[14] The figure was created with Belvu, a multiple alignment viewer developed by Erik Sonnhammer.

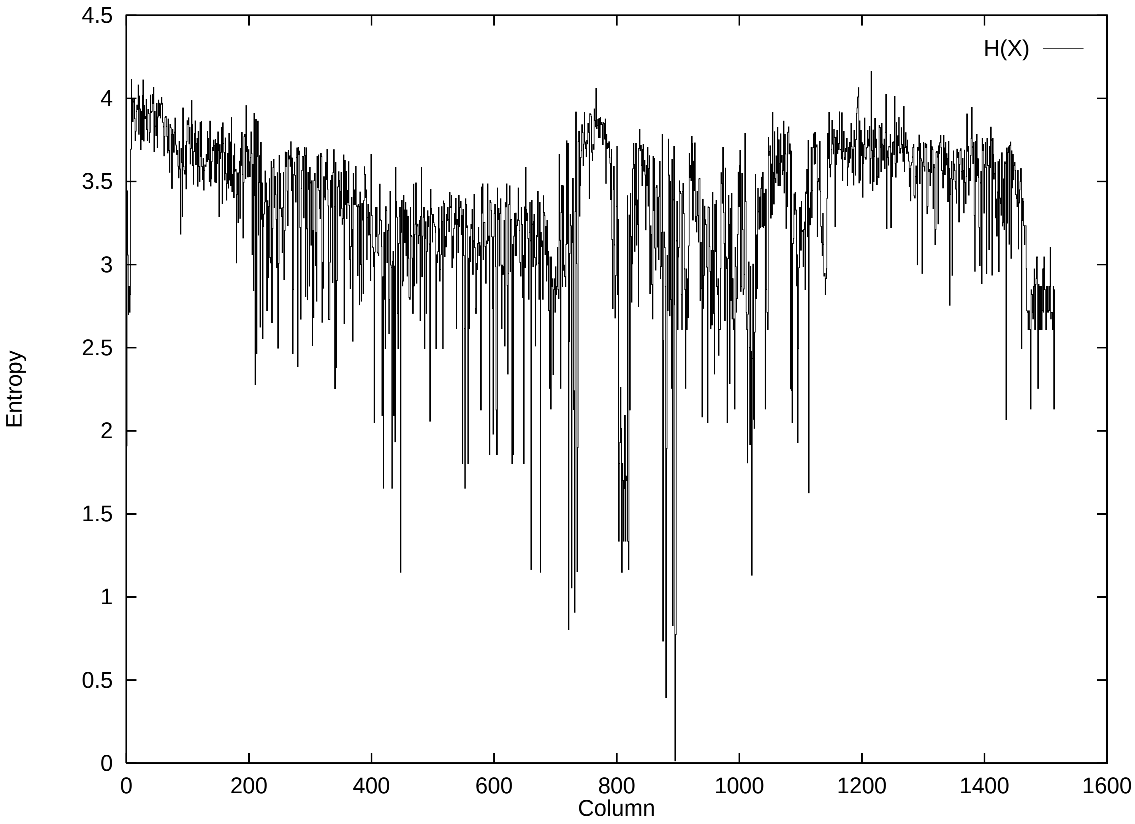


Figure 5.2: The estimated entropy values for the multiple alignment of the 4Fe-4S family.

The estimated entropy values in figure 5.2 have a mean of 3.3 and a standard deviation of 0.5. The maximum possible value achievable is ^{2}log 20, representing a completely uniform amino acid distribution. Only a few of the columns are given relatively small values of entropy and not all of them represent residues that are being modeled by the match states of the HMM. For example, the spikes at the positions around columns 450, 660 and 680 have only two of the 125 sequences sharing identical residues, these are being modeled by insert states in the HMM. The low entropy values are due to the Dirichlet density mixture; one of its components favors fully conserved positions with only Proline, Glycine, Tryptophan or Cysteine residues and those are exactly the amino acids that are conserved at these columns. The other spikes between columns 700 and 900 belong to positions that are modeled by match states, and these are the positions of, and in the vicinity of, the active sites.

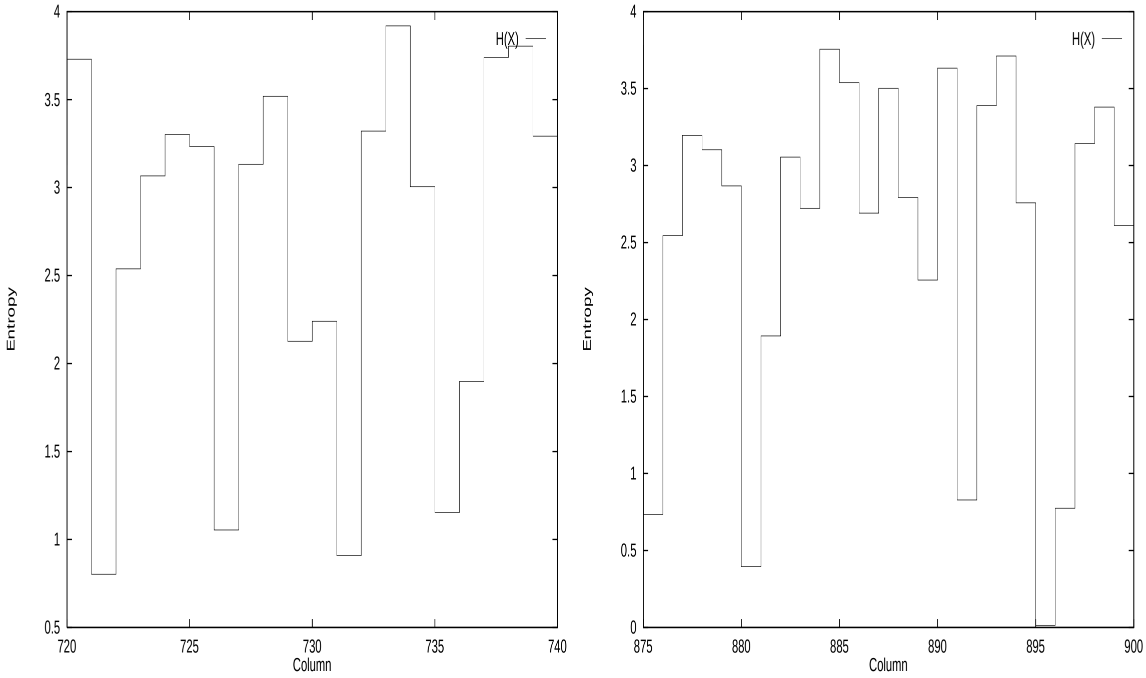


Figure 5.3: A closeup of the active sites reveals that they have been assigned the lowest entropy values.

The four cysteine residues of each active site have the lowest entropy in that region. Closer inspection of the entropy values might indicate that the second active site is better conserved according to the Dirichlet mixture parameters. However, since the model is created in an iterative procedure using simulated annealing, it could just be a side effect of the HMM being trapped in a local maxima.

The presented results and other similar data (not shown) indicate that the algorithm can deliver satisfactory results. The regions that are marked as structurally important in protein sequence documentation generally are given low entropy values.

5.2 Verifying model modification

By visual inspection of the created models it was confirmed that the developed algorithm did mark the nodes that were capturing high variability regions as candidates for deletion. That the basic functionality of the HMM was preserved (ability to detect the presence of conserved subsequences and align protein sequences to each other), was confirmed by creating multiple alignments and seeing that the conserved patterns from the original model were not destroyed.

5.3 Performance evaluation

When searching large databases, usually the number of negatives (the proteins that should not be considered to be true homologs) is much larger than the number of positives. For these reasons the specificity measure is more important than the sensitivity measure, since if large numbers of false positives are reported, the true hits will be drowned in the noise. This is the reason why we have decided to plot the sensitivity as a function of the specificity in the graphs; to clearly see how the changes in sensitivity and specificity are correlated.

Another factor which has to be investigated is how the level of generalization affects the performance of the modified HMMs. Therefore we have decided to present the relative performance in a set of graphs for the 0.3, 0.5, 0.7 and 1.0 bits of generalization.

The third factor which has to be clarified is the behaviour of the two different measures used, i.e. if there are any fundamental differences in performance when using the entropy measure versus the encoding cost measure.

The following subchapters present the results from the tests made on all the families: globins, ferredoxins, adh short-chain dehydrogenases and remote homolog scoring for 1try and 1hurA. First a set of graphs are presented where the sensitivity and the specificity are visualized by ROC curves. Second the relative performance gains/losses at different levels of generalization will be presented; both for the entropy and the encoding cost measures. Third the results of some complementary tests where the position-specific amino acid distributions were removed will be presented.

All of the following graphs represent results from *single runs*, i.e. they are not averaged over several experiments. In a few cases, one additional test was performed to see whether the good/bad performance could be caused by the HMM being stuck in a local maximum.

All of the results are calculated from runs where the training and the test sets were disjoint, i.e. there was no overlap between these two sets.

In the presented ROC_{50} curves the fraction of positives represent the fraction of the test set (the unseen sequences either belonging to the modeled family or marked as being homologous to it) that have been detected in the result file. The fraction of negatives reaches one when 50 sequences that were not included in the training and test sets have been detected in the result file.

5.3.1 Globins

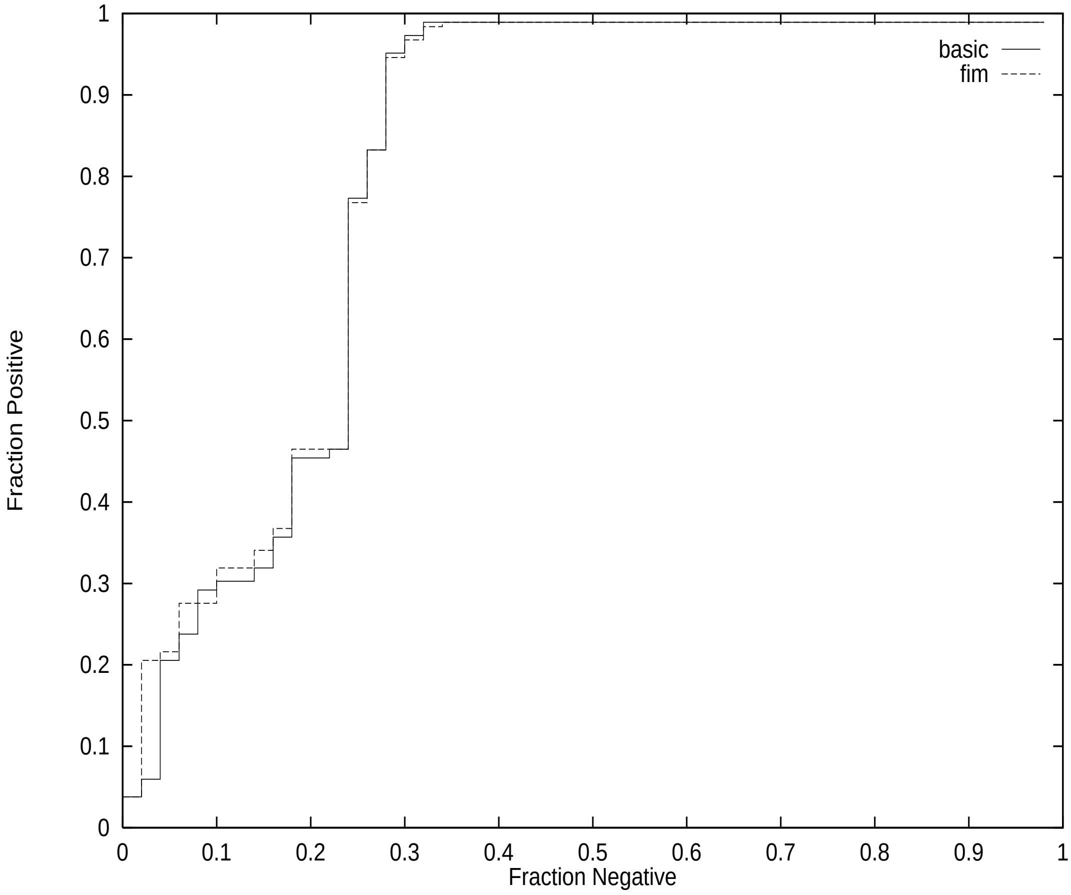


Figure 5.4: The performance of the model before (basic) and after (FIM) the modification by deletion of nodes and addition of FIMs.

In figure 5.4 the best result of all the 24 different tests is visualized. The area below the ROC curve for the basic case is 0.7932 and for the modified case 0.7981, i.e. roughly an increase by a half percent in the probability of getting a correct classification. This improvement was achieved by deletion of four nodes out of a total of 141. The general trend of the experiments (see data in appendix A) done for the globin family indicate that there is no significant performance increase when deleting high variability nodes and using FIMs. Deleting more than these four nodes resulted in performance losses of increasing severity.

5.3.2 Ferredoxins

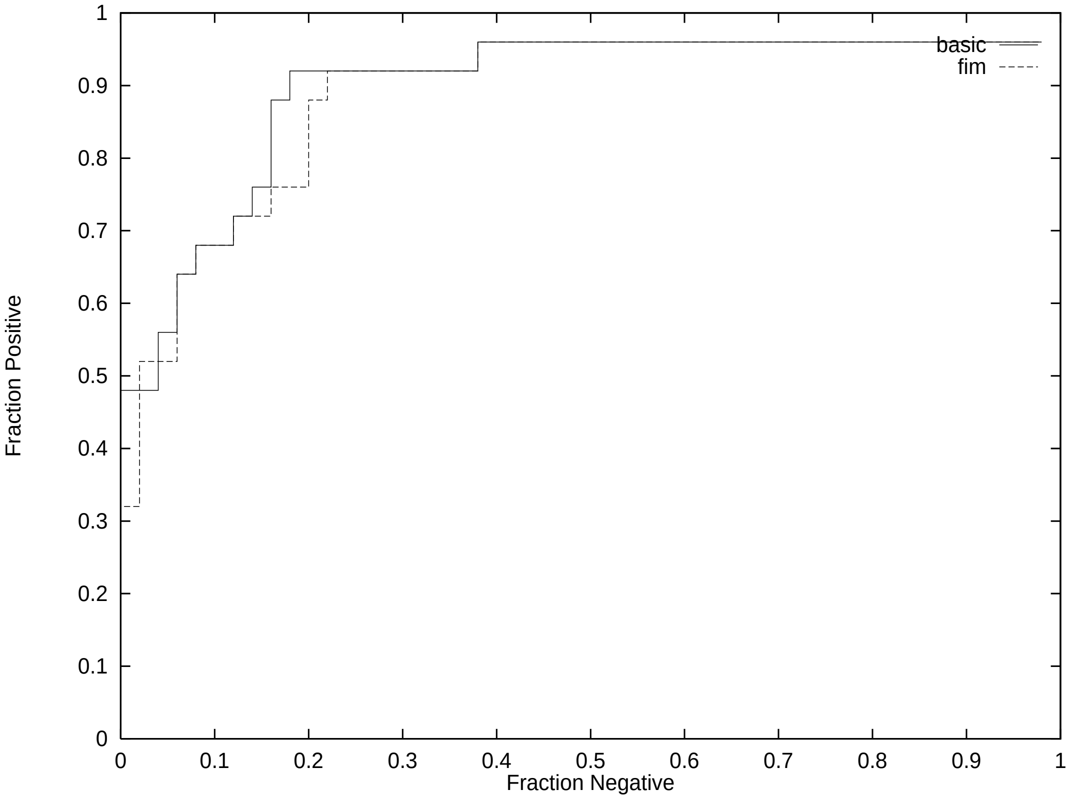


Figure 5.5: Relative performance of the modified models in the ferredoxin experiments. The fraction of negatives reaches 1 when 50 negative protein sequences have been detected.

Figure 5.5 depicts the best case of the original 24 experiments, and it shows a performance decrease of one percent. That deletion of 3 nodes out of a total of 106 for the whole model gives a significant performance decrease is a strong indication of the fact that even the regions that have relatively high variability do contain information that is absolutely crucial for the model.

The two active sites and their vicinity both contain 26 residues according to Prosite documentation, suggesting that a total of 52 nodes could be near optimal for the detection of this family. Therefore, one additional test was done where a new model was trained on the same training set and the developed software was applied on the resulting multiple alignment. This time the model had length 78, quite a difference to the first case. The length indicated that in this experiment the basic HMM was already highly focused on the two conserved regions and any attempt at removing nodes resulted in a performance decrease. Another sign of the level of focus is that the threshold had to be pushed one standard deviation below the mean before it started removing any nodes. That the new model only had 78 nodes should indicate that the first model, of length 106, contained a number of nodes that should have been suitable for deletion. Especially since the average performance level of the unmodified model improved with this shorter topology (see appendix A). However, similar to the first test, no performance improvement was achieved by deletion of nodes and addition of a FIM.

5.3.3 ADH

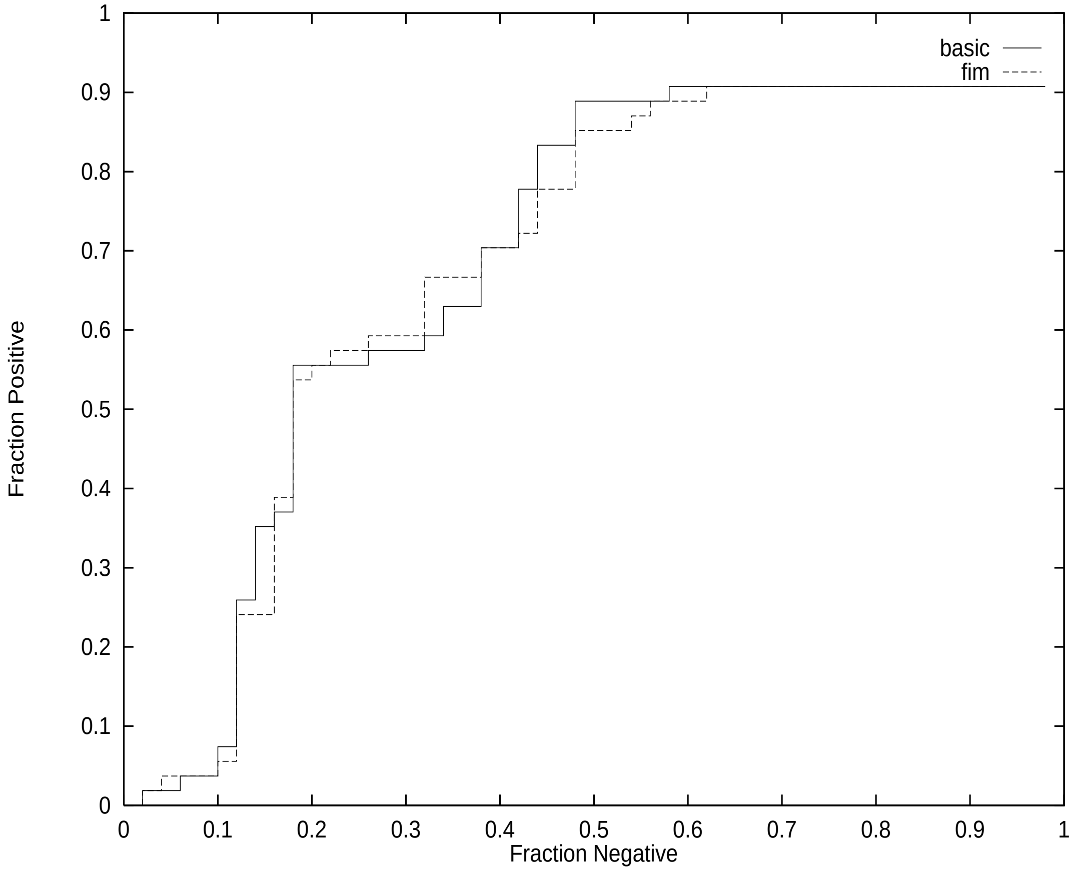


Figure 5.6: Results from the experiments done on the adh short-chain alcohol dehydrogenase family.

No performance improvements were detected for this family. The best case (figure 5.6) was a decrease of 0.4% when the algorithm deleted nine nodes out of a total of 267.

5.3.4 1try

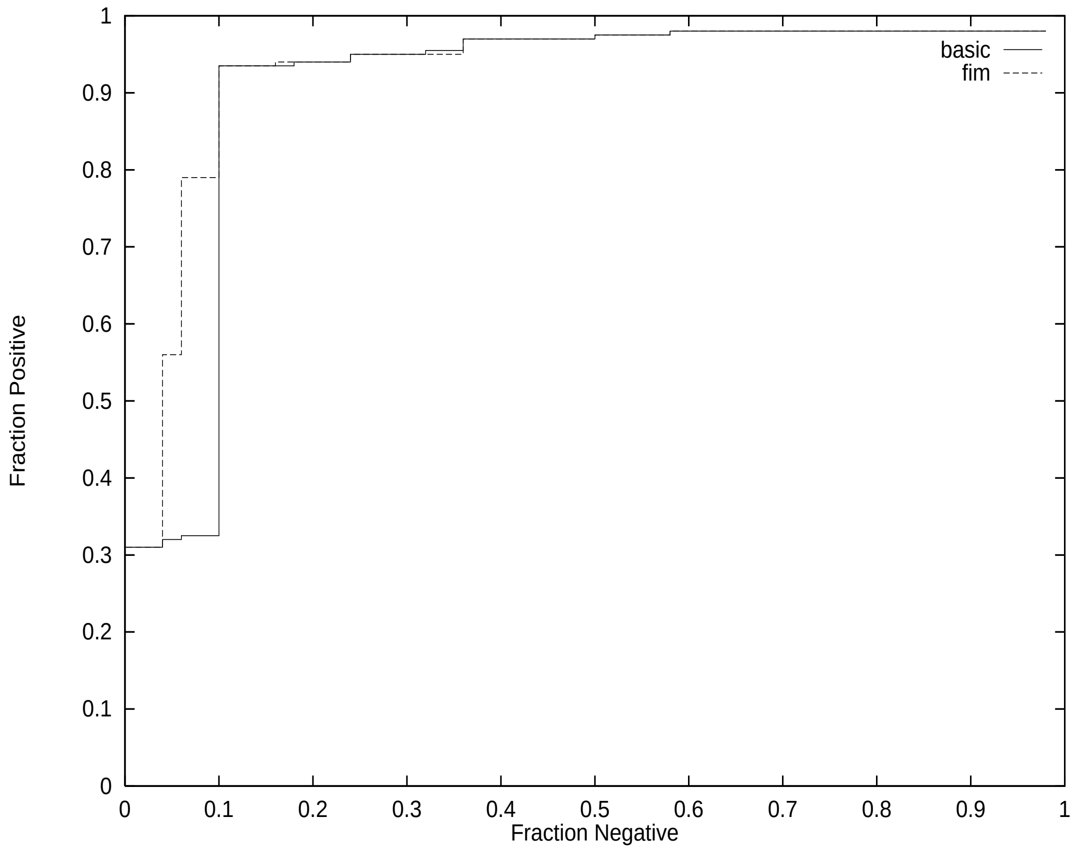


Figure 5.7: Results from remote homolog tests on the trypsine family of serine proteases.

In this case a good improvement was detected. The area under the ROC curve increased by 2%. This occurred when the algorithm removed two nodes out of a total of 225.

A new run (new model and new tests, see appendix A) failed to show the same level of improvement. The new model, with length 262, could only show a performance increase by 0.6%. However, it occurred when 39 nodes were removed.

5.3.5 1hurA

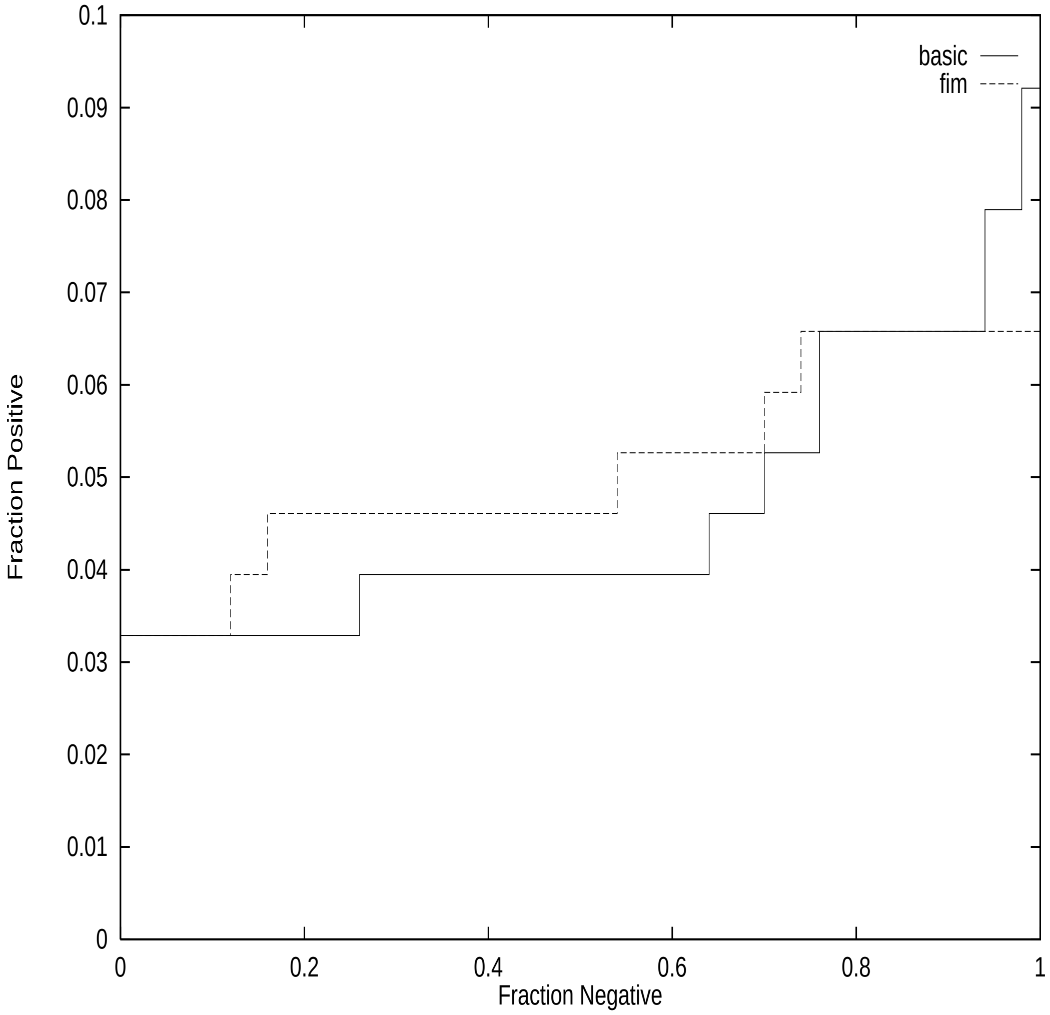


Figure 5.8: Results from remote homolog tests on human ADP-ribosylation factor 1.

Figure 5.8 displays the best case of the experiments done on this family. Although the overall performance is very low, the application of the developed software on the model and the multiple alignment did result in a small increase.

5.3.6 Levels of generalization and different measures

The relative performance changes for the two measures used are depicted in figure 5.9. The picture is divided into four rows, one for each level of generalization (0.3, 0.5, 0.7 and 1.0 bits), where 0.3 is ‘fuzzy’ and 1.0 is ‘sharp’. In each row there are six columns, three for each of the two measures used. The first three are for the threshold levels used for the entropy measure and the next three are for the encoding cost measure (see section 4.3.2). In each graph, the five bars represent the five protein families that the experiments were performed on. The families are, from left to right in each graph: globin, ferredoxin, adh, 1try and 1hurA. The height of the bar represents the change in the area under the ROC curve with respect to the original model.

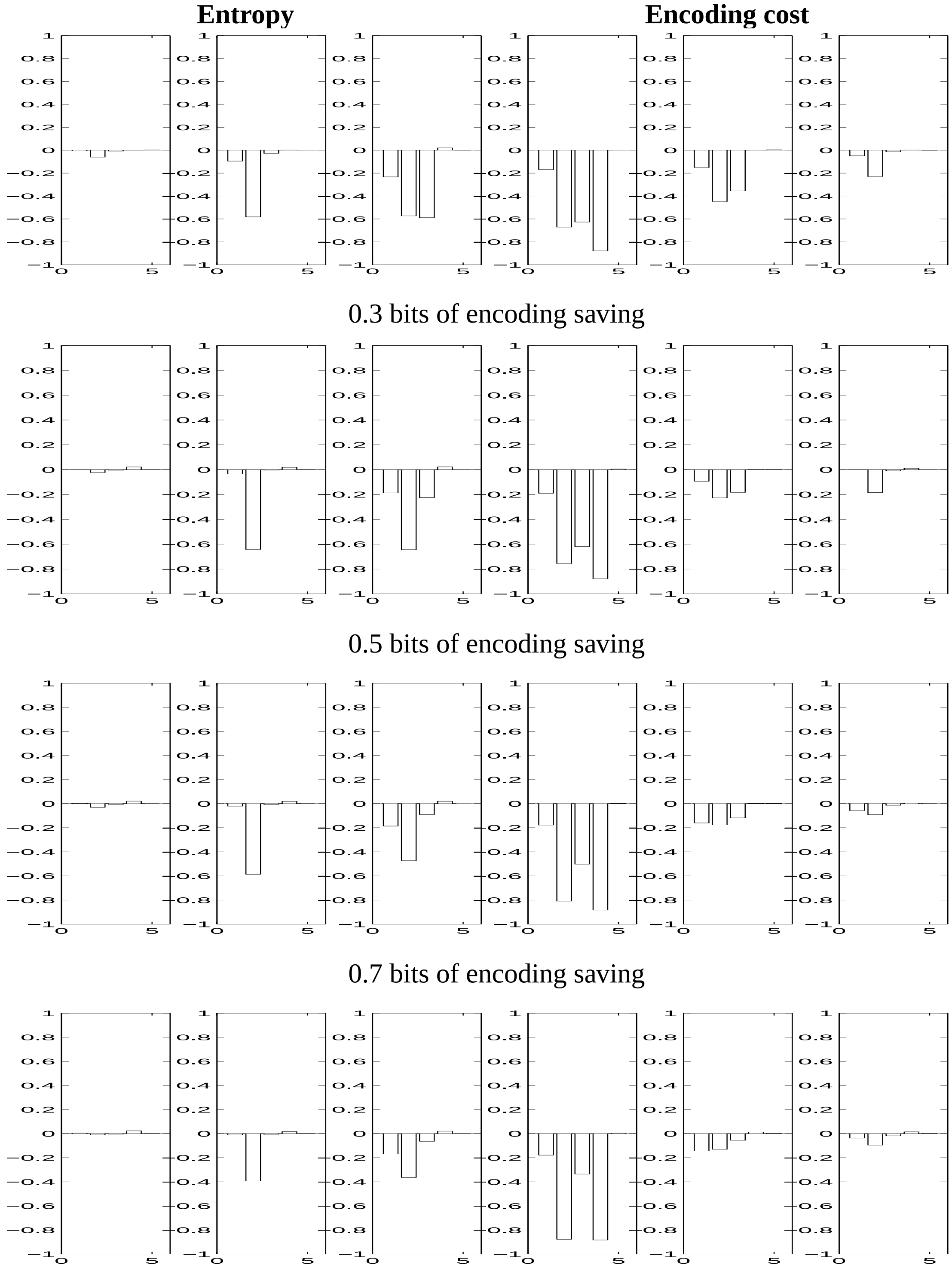
Looking at the graphs one can detect some general patterns that are more or less common to them all. For instance, looking at the columnwise relative height of the bars it is clear that the performance decrease is smaller when the model is ‘focused’, i.e. it has high encoding saving. This means that the model is less tolerant of deletions when it has few bits of encoding saving. This is not so surprising considering that when the average encoding saving of each match state is small, the model needs many nodes to add up to a score that is distinctly different from the NULL model. Recall from section 2.4.1 that the scores produced of an HMM is the log-odds ratio of the created model and a NULL model which is used as a default model for all the sequences.

The improvements, when they do occur, are generally small, which indicates that the original model already was quite good at detecting the remote homologs and the unseen sequences. Generally, for the protein families that have a sequence similarity above 25%, the models have a performance around 80-90%, while the true test of remote homolog detection capability, the 1hurA family, has a sequence similarity of 15% or less. The original model for the 1hurA family did not do particularly well and the modified ones were only slightly better.

The threshold values selected for each of the five families and the two measures are presented in table 5.1. The order of the families (globin, ferredoxin, ADH, 1try and 1hurA) corresponds to the five bars in each of the graphs in figure 5.9. The measures and thresholds used correspond to the order in figure 5.9, i.e. first the results from the three thresholds for the entropy measure are presented and then the same for the encoding cost measure. The values for both of the measure are with respect to the mean of all the columns of the multiple alignment. For explanations of the selected values, see section 4.3.2.

Measure	**Entropy**	**Entropy**	**Entropy**	**Encoding**	**Encoding**	**Encoding**
Threshold	**First**	**Second**	**Third**	**First**	**Second**	**Third**
Globin	-0.5	-1.0	-2.0	+0.0	+0.5	+1.0
Ferredoxin	-0.0	-1.0	-2.0	-1.0	+0.0	+1.0
ADH	-0.2	-0.6	-1.2	+0.0	+0.5	+1.0
1try	-0.0	-1.0	-2.0	-1.0	+0.0	+1.0
1hurA	-0.1	-0.6	-1.2	+0.0	+0.5	+1.0

Table 5.1: The selected threshold values for each of the five protein families and the two measures used.



Entropy **Encoding cost**

0.3 bits of encoding saving

0.5 bits of encoding saving

0.7 bits of encoding saving

1.0 bits of encoding saving

Figure 5.9: Bars of the relative performance changes. Each bar represents the performance change in one protein family with respect to the original model.

5.4 Extended results

To understand and analyze in more detail the effect of replacing deleted nodes with FIMs, we here describe the differences between these two node types (table 5.2).

HMM	**FIM**
The number of nodes encodes for length information in the model.	No corresponding information. Only one FIM for arbitrarily many deleted nodes.
The match states include position-specific information on amino acid distributions.	No corresponding information. No match state present in a FIM.
The probability on the reflexive transition in the insert state implements an exponential length distribution between match states.	No corresponding information. The probability is set to unity, giving a uniform length distribution.

Table 5.2: A comparison between standard nodes of an HMM and the FIM.

If the sequences in the training set share a region with high variability between the conserved regions, the number of nodes in the HMM that is modeling that high variability region encodes for its length. The heuristics used in SAM does not remove nodes from these regions, it considers only the number of sequences that use a match state or an insert state and adjust the model length accordingly, i.e. if few sequences use a match state it is deleted and if many sequences use an insert state, a new match state is inserted. [HKB+96]. This information is lost when nodes are deleted and a FIM is inserted (figure 5.10).

The 20 parameters of a match state (one for each amino acid) are adjusted in an iterative procedure which strives to maximize the probability of the model, given the training sequences. When the training sequences are aligned to the model, the protein sequences are mapped onto the model so that it yields the highest probability of all of the possible paths. Since FIMs do not include a match state, this information is lost as well.

The basic score of a sequence is the product of the transition and the symbol emission probabilities of the states that it passes through. If a subsequence is modeled by an insert state, the only information that can make a difference is the transition probability, since the symbol emission probabilities are 'masked out' by the NULL model with which the basic score is compared. Remember that the NULL model is basically just a FIM and that the amino acid distributions of the insert states in a standard node and in a FIM are identical. The consequence of this is that when standard nodes are deleted and replaced with a FIM, the length information is discarded.

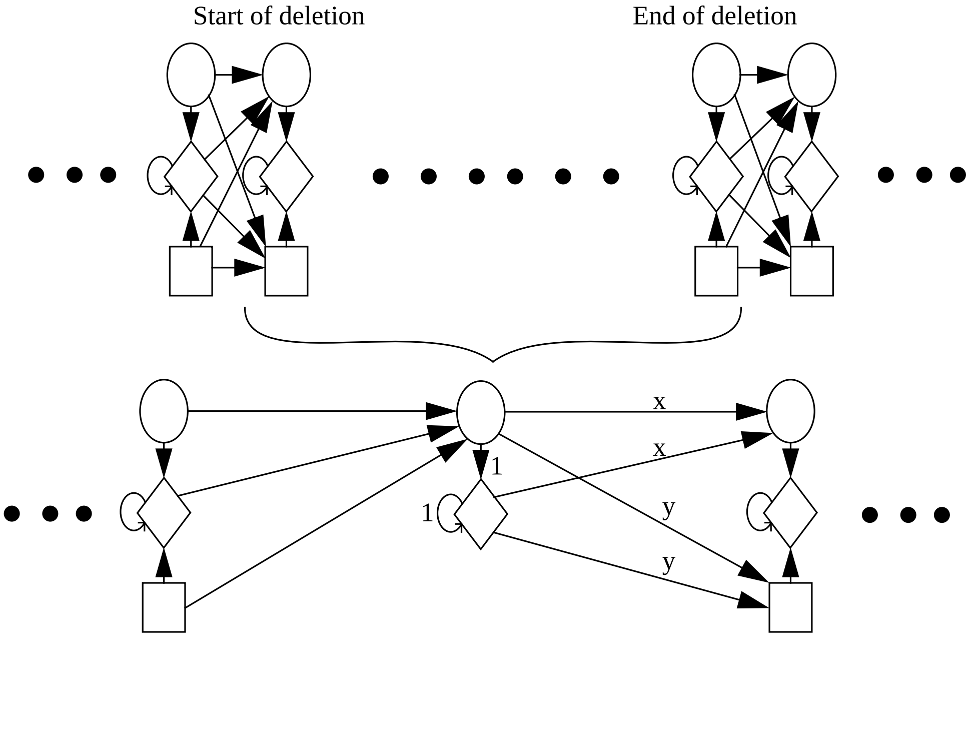


Figure 5.10: Visualization of the process of multiple node deletion and addition of one FIM. The FIM has probability set to unity on the transition from the delete state to the insert state and on the reflexive transition in the insert state. The 'x' and 'y' indicate that these transition probabilities are set to identical values.

The removal of all of the above information was too much in most cases. The capability of the HMMs to reject unrelated sequences vanished as more and more FIMs were inserted.

The difficulties of the FIM-based approach triggered another set of experiments which were intended to more clearly determine the cause of the varying performance. Two experiments were done to get a hint on if it was the length information or the position-specific amino acid distribution information of the deleted nodes that was crucial for the performance advantage of the original HMM.

5.4.1 Removing position-specific amino acid distribution information

Instead of deleting the nodes that the algorithm identified as trying to capture high variability regions, we removed the amino acid distribution information from the match states. This was done by manually modifying all the identified match states so that they used the same background distribution of amino acids as the insert states do. The original values of the transition probabilities were retained. The experiments were done on the ferredoxin and the trypsine families (see appendix B). New models were created and modified according to the output of the analysis of the multiple alignments. The models were *not generalized*, i.e. all the parameters were retained as they were after the first model creation process.

In the trypsine case the modifications gave a small improvement when the number of modified nodes is small, and then turns into a performance decrease when the number goes up. Interestingly, when the number of changed nodes goes up, the performance becomes even worse than when the generalized models were modified by deletion (see appendices A and B). That is, the model performed better with FIMs than with retained but changed nodes.

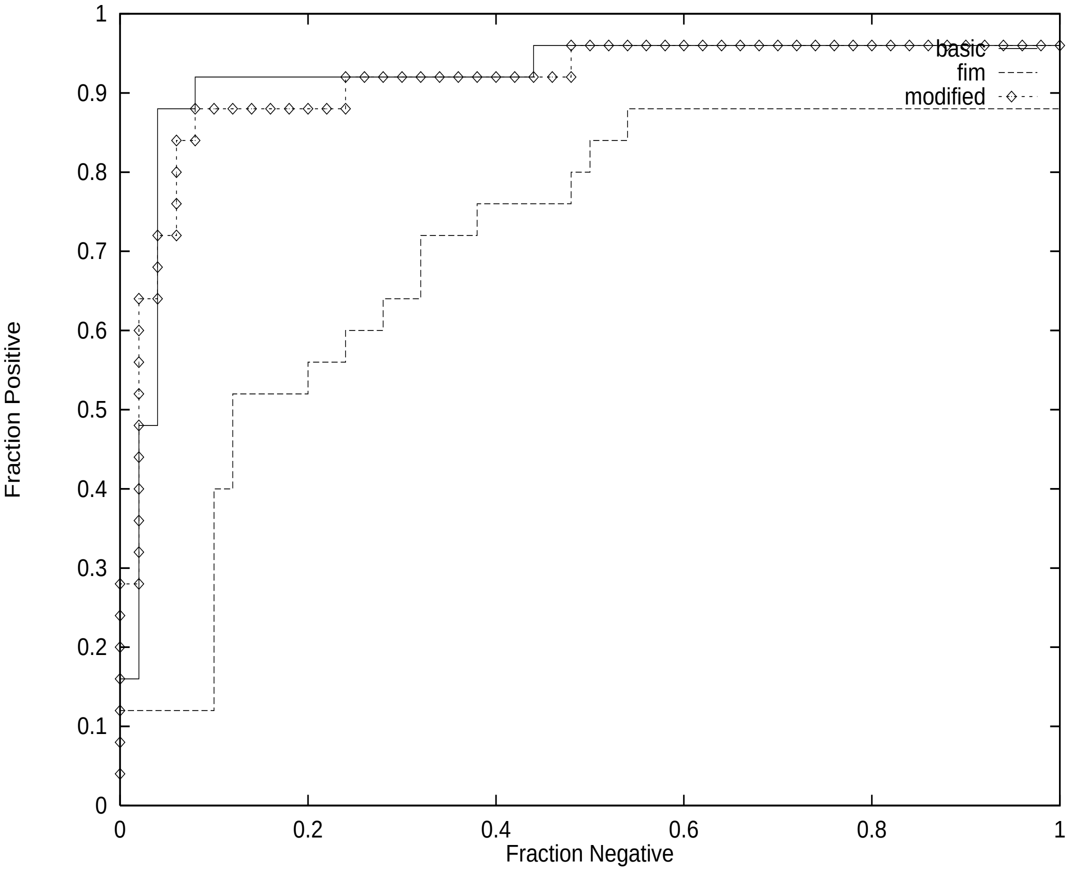


Figure 5.11: The performance in the ferredoxin family with the original model (basic), the model which has deleted nodes (fim) and the model in which the match state distributions were replaced (modified).

The ferredoxin case, which exhibited a major performance loss when the generalized models were modified by deletion, displays a much more neutral behaviour when only the match state distributions were replaced (figure 5.11). The figure represents a situation where the entropy measure was used with the threshold set to two standard deviations below the mean. Comparing the basic and the FIM tests, it is clear that the deletion of that many nodes (42) removes information that is crucial to the HMM. However, just removing the match state distributions (modified) in exactly the same nodes produces a model that is almost as good as the original one (a decrease by 0.6% in the area below the ROC curve).

6 Discussion

The results show that in most cases the deletion of high variability nodes does not improve the remote homolog detection capabilities of HMMs, it either stays the same or decreases significantly. The notable exception is the trypsine family, which did exhibit a major improvement in one run, but that result was not reproducible. The model building and parameter estimation process of HMMs is an iterative procedure guided by heuristics that cannot guarantee that it escapes local maxima. This is believed to be the source of the difficulties in trying to repeat the trypsine ‘success’. That is, we believe that in the first experiment we were doing modifications on an HMM that had been trapped in a local maxima. Looking at it the other way around, however, our software enabled it to escape the local maxima. Also, by accepting a marginal performance loss of 0.0024 (the decrease in the area under the ROC curve) when removing 92 out of 262 nodes, our method did give a considerably smaller model - which will run faster when doing searches on large protein sequence databases.

The two tests specifically targeted at testing the remote homolog detection performance of this method, the 1try and the 1hurA families, display similar results with ROC_{50} analysis. The trypsine case worked quite well (if you are satisfied with getting a smaller model that performs almost equally well), and while the original model for the 1hurA family did not seem to work at all, our software did improve its performance slightly. The major difference between these two families is the percentage of identical residues in the test sets. The test set of the 1try family (the FSSP-identified homologs) contained sequences with 35% of identical residues or more, while the 1hurA family (also collected from FSSP) contained sequences with a percentage of identical residues at 15 or less. Remembering that the results were *not* averaged over multiple runs, this could be interpreted as a weak indication of the possibilities of modeling high and low variability regions differently.

Comparing the behaviour of the two measures used, the entropy and the encoding cost, suggests that the entropy measure is a little better at selecting high variability nodes. However, the significance of this difference is unclear since the results depend very much on how the user-specified threshold values were set. A different approach, one that does not rely on user-specified thresholds, must be used to evaluate the relative performance of these two measures.

From the results presented it is clear that the performance of the FIM-based approach can be either quite similar to the original model or far from satisfactory. Possible sources of error might include: biased test sets, incorrect behaviour of multiple alignment analysis algorithm, faulty scripts for modification of HMMs and errors in calculating performance metrics.

Clearly the test sets selected for this method (five protein families, of which only two are specifically targeted at remote homolog recognition), might not be an optimal test set for this particular method. However, we believe that the selected families do give some hints on how the FIM-based method performs with respect to the original method. By visual inspection of multiple alignments and by information in protein sequence databases, three families were selected based on: uniform mutations over all of the sequence (globin), well conserved motifs (ferredoxin), fragmented and shorter but multiple conserved regions (adh), remote homology test with fairly good sequence similarity (1try) and a remote homology test with very low sequence similarity (1hurA).

The behaviour of the software developed for analyzing multiple alignments (AMA) was checked by visual inspection of the output's correlation to the columns in the multiple alignments. Information in the form of annotations in protein sequence databases was used to verify that the conserved columns in the multiple alignment was in agreement with the data known about that protein family, and that the variability metrics was in agreement with the multiple alignment. However, we cannot guarantee that the high and low variability regions always will correspond to the known information about where the active sites are and what regions are truly conserved, since the behaviour of the Dirichlet parameters are underlying all of this work. If there are some significant amino acid distributions that the prior has not been trained on (i.e. it does not recognize them as likely combinations of amino acids), the variability estimation of alignments with few sequences could miss important positions/columns.

The scripts produced by AMA were checked by manually verifying that the high variability regions of the multiple alignment were detected, and that the correct nodes of the underlying HMM were deleted. That the deletion did not disrupt the neighbouring nodes and that no anomalies were introduced into the transition probabilities was checked by inspection of the HMM parameters.

The performance metric used for evaluating the FIM-based method (the area under ROC_{50} curves) was introduced by Gribskov and Robinson [GR96] to be an objective, threshold independent measure of how different techniques perform in protein database searches. It has been used successfully in related research for exactly these purposes [GBE+97b].

The differences between the standard nodes and the FIMs are: lack of specific amino acid distribution information, no probability distribution on the insertion lengths and the many-to-one mapping of deleted match states into one FIM. We have not made exhaustive tests, but the two probing experiments made in section 5.3 clearly points in one direction: *the high variability regions do contain some valuable information*. If it is only the length information between motifs that is important, then these results tie in very well with the results reported for Meta-MEME, which concentrates on modeling the conserved regions, i.e. the motifs, but also includes information about the length between these motifs [GBE+97b].

The explicit goal of the work done by Grundy et al. was to reduce the number of parameters used in HMMs, i.e. slightly different from the present work. This was needed for efficient modeling of protein sequence families where the training sets available are small. They report that their Meta-MEME exhibits both increased sensitivity and selectivity (analogous to the definitions of sensitivity and specificity used in this report), and especially when the training sets used are small. Since their motif-based model contains fewer parameters than the standard HMM, it seems quite natural that they will work better with small amounts of data. The main differences between our approach and Meta-MEME is threefold: the identification of motifs, the topology of the models and the length information.

In Meta-MEME motifs are identified by extracting them from the training sequences alone [GBE+97b], while in our approach we have used multiple alignments to focus the underlying HMMs on regions with low variability. The motif identification process should not be the source of the difference in performance between these two approaches, since it was visually verified (with information available in protein sequence annotations and in Prosite documents for the 4Fe-4S family, see section 5.1) that the algorithm in our approach did mark the active sites as more conserved than the rest of the protein sequence. That means that they were always retained to the end, where the model only consisted of the residues in the vicinity of the active sites. Lowering the threshold even further would have concentrated the model on the second active site that, according to the entropy measures of the multiple alignment (which were calculated using the Dirichlet mixture prior), is marginally more conserved than the first one (see figure 5.2). Again, since the results are not averaged over multiple runs, it is difficult to estimate the significance of these findings.

The topology of the two models differ in that the motif-modeling subset of Meta-MEME consists of a gapless sequence of match states [GBE+97b], i.e. no exceptions in the form of insertions or deletions are allowed, whereas in our approach the full set of nodes (match, insert and delete) is retained in the motif-modeling regions. This should give our approach more flexibility since, in theory, it would be more tolerable to cases where the motifs vary slightly in length. However, we do not believe that this is an important factor in the overall performance of the FIM-based approach, since examination of the relative weights on the transitions in the created model indicates that it is heavily biased to a particular sequence of match states.

The only remaining difference between our approach and Meta-MEME is the length information. In our approach the length information between motifs is discarded and all insertion length are equally likely. The insert states are essentially the same as in the Meta-MEME approach (with an exponential length distribution [GBE+97b]). We believe that all of the above are evidence in support of the conclusion that the length information between motifs is crucial for the performance of the HMM, and especially for its capability to reject unrelated sequences.

Although the tests only represent single runs, it could be that the results indicate that a model which concentrates on the truly conserved regions, and includes explicit modeling of insertion lengths between these regions, could be very useful in modeling protein sequence families and detecting remote homologs. Since the two experiments done with replacement of the amino acid distributions in the match states gives a performance decrease, they indicate that even though they are identified as having high variability, the inserts between the conserved regions do have a non-uniform amino acid distribution. However, more work is needed before definitive conclusions can be drawn on these issues.

7 Conclusion

In the presented work we have investigated the effects of decreasing the size of an HMM by removing the nodes that are modeling high variability regions. The high variability regions were located with information theoretic analysis of multiple alignments and the underlying HMMs were modified by deletion of nodes and addition of FIMs. This work was based on the assumption that high variability regions could be unnecessary, or even misleading, when trying to detect remote protein homologs.

7.1 Contribution

The results do not support a definitive conclusion, since a few cases exhibit a performance increase, while some cases indicate that the performance either stays the same or decreases when ignoring high variability regions. Two additional tests indicate that when there is a significant performance loss due to deletion of high variability nodes, a much smaller decrease occurs when the nodes are preserved but the position-specific amino acid distributions are removed. This indicates that the length information between the conserved regions, and possibly some amino acid distributions that encode for structurally important information, is valuable for the model. Taken together, these results support the hypothesis that *there is some valuable information present in the high variability regions and, in general, for a model to be successful in discriminating between true and false remote homologs, this has to be taken into account*. However, since the number of runs on each test set was so small, more research is needed in this area before a definitive conclusion can be stated.

7.2 Future work

We suggest future work with the aim of examining the structural and functional importance of the high variability regions, and on the possibilities of using different constructs for modeling high and low variability regions of protein sequences. For example, in the present work we have used rather large training sets and it could be interesting to see how the method would perform on average when the amount of data available is small. One way of testing that would be to have a set of training sequences and then perform random sampling and test the performance over several runs, very much like Grundy et al. did in [GBE+97b]. The 'simulation' of small data sets by generalization of models used in this project only makes the model more 'fuzzy'. It does not change the topology of the created HMM. A better way would be to create training sets of increasing size by random sampling from all of the available family members, and estimate how the performance changes on average over ten or more repeats on the same data.

Deleting nodes and replacing them with a FIM is not the only possible approach to modeling high variability regions. Meta-MEME essentially uses the insert state just after a motif and modifies the probability on the transitions so that the expected number of insertions matches the average number of insertions observed in the data. The method of locating high variability nodes is independent of the way that it is implemented. If it is enough to include the length information between motifs, further work could just as well modify the insert state immediately before the deleted region, similar to Meta-MEME. That work would then be a comparison between the value of using different topologies for the conserved regions. That is, in Meta-MEME a gapless profile is used, while in HMM[15] the

[15] We are referring to the topology used by SAM.

‘profile’ would be more tolerant of small insertions and deletions in the conserved region. That would perhaps give it a performance advantage.

The threshold values used for the entropy and the encoding cost measures were set by subjective evaluation of their effect on the model. A more reliable approach would be to measure the performance changes with respect to the number of FIMs that each threshold level inserts into the model. Alternatively, the number of deleted nodes could be used as a guideline for setting the threshold values.

Also, the biological significance of the positions and regions of the protein sequences that AMA marks as having very low variability could be worth further research. That work would be focused on analyzing the ‘anomalous signals’ that AMA finds, i.e. positions of the multiple alignment that have variability estimates that are very far from the mean, and investigating how they are correlated to the active sites and other structurally and/or functionally important regions of the protein(s).

In general we can conclude that there is scope for future work in a number of directions. The topology and the underlying assumptions of the HMMs used might not be optimal for all tasks, it could be that different architectures (or new combinations of them) could be more efficient in sequence comparisons and in creating multiple alignments. In this work we also have identified a number of questions with regard to the structural and functional importance of the high variability regions, and their contribution to the discrimination capabilities of HMMs.

Acknowledgements

This project would not have been possible without the helpful discussions held with several people.

Kimmen Voronov Sjölander gave an excellent introduction to the field of computational molecular biology, and she also held a tutorial that included practical usage of molecular biology databases and tools developed for manipulating the data. She is also the one that suggested that this work could be suitable for an M.Sc. dissertation, and was helpful in answering many of the questions during the initial stages of the work.

Melissa Cline has been very helpful in installing software from UCSC and answering many of the questions that have come up during this project.

Björn Olsson has given advice on the content and structure of this report and without his patience with late drafts and results, it would never have been completed. He also suggested some alternatives to the testing procedures used in this work.

The discussions with Dan Lundh on alternatives to Dirichlet density mixtures for variability estimation have been interesting and informative.

For mathematical background and working details of HMMs, the insights by Kevin Karplus proved to be very valuable. He is also the one that provided an alternative view on the usefulness of FIMs, i.e. that they might not be optimal for the purpose at hand.

Finally, the author wishes to thank Ajit Narayanan for helpful comments on the dissertation topic in general, and on the connections to computational linguistics in particular.

References

[ABB87] Abola, E. E., F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng. (1987). Protein Data Bank, in Crystallographic Databases-Information Content, Software Systems, Scientific Applications, F. H. Allen, G. Bergerhoff, and R. Sievers, eds., Data Commission of the International Union of Crystallography, Bonn/Cambridge/Chester, pp. 107-132.

[ABG+94] Allison, Stephen F., Mark S. Boguski, Warren Gish and John C. Wootton. Issues in searching molecular sequence databases. *Nature Genetics*, vol. 6, pp. 119-129. February.

[AW94] Allison, L. and C.S. Wallace. (1994). The Posterior Probability Distribution of Alignments and Its Application to Parameter Estimation of Evolutionary Trees and to Optimization of Multiple Alignments. *Journal of Molecular Evolution*, 39:418-430. Springer-Verlag New York Inc.

[Alt91] Altschul, Stephen F. (1991). Amino Acid Substitution Matrices from an Information Theoretic Perspective. *J. Mol. Biol.*, vol. 219, pp. 555-565.

[AGM+90] S.F. Altschul, S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman, *J. Mol. Biol*. 215, 403-10 (1990).

[ABH94] Appel R.D., Bairoch A., Hochstrasser D.F. (1994). A new generation of information retrieval tools for biologists: the example of the ExPASy WWW server. *Trends Biochem. Sci.* 19:258-260.

[BG96] Bailey, Timothy L. and Michael Gribskov. (1996). The megaprior heuristic for discovering protein sequence patterns. *Proceeding of the Fourth International Conference on Intelligent Systems for Molecular Biology*, June, AAAI Press.

[BB94] Bairoch, Amos and Brigitte Boeckmann. (1994). The SWISS-PROT sequence data bank: current status. *Nucleic Acids Research*, 22:3578-3580.

[Bai93] Bairoch A.; (1993). Nucleic Acids Res. 21:3097-3103.

[BWW+95]Berger, Bonnie, David B. Wilson, Ethan Wolf, Theodore Tonchev, Maria Milla and Peter S. Kim. (1995). Predicting coiled coils by use of pairwise residue correlations. *Proc. Natl. Acad. Sci. USA*, vol. 92, pp. 8259-8263, August.

[BT91] Branden, Carl and John Tooze. (1991). *Introduction to Protein Structure*. Garland Publishing, Inc.

[BHK+93] Brown, Michael, Richard Hughey, Anders Krogh, I. Saira Mian, Kimmen Sjölander and David Haussler. (1993). Using Dirichlet mixture priors to derive hidden Markov models for protein families, In Hunter, L., D. Searls and J. Shavlik, (eds), *ISMB-93*, Menlo Park, CA. AAAI/MIT Press, pp. 47-55.

[BKM+96] Bucher, Philipp., Kevin Karplus, Nicolas Moeri and Kay Hoffman. (1996). A Flexible Motif Search Technique Based on Generalized Profiles. *Computers and Chemistry*, January, 20(1):3-24.

[Coh96] Cohen Jon. (1996). Protease Inhibitors: A Tale of Two Companies. *Science*, Vol. 272, 28 June, pp. 1882-1883.

[CT91] Cover, Thomas M. and Joy A. Thomas. (1991). *Elements of Information Theory.* John Wiley & Sons, Inc.

[Edd96] Eddy, Sean. (1996). Hidden Markov models. *Current Opinion in Structural Biology,* 6, pp. 361-365.

[Edd95] Eddy, Sean. (1995). Multiple alignment using hidden Markov models. *Proc. Third Int. Conf. Intelligent Systems for Molecular Biology,* C. Rawlings et al., eds. AAAI Press, Menlo Park, pp. 114-120.

[GHK+96] Grate, Leslie, Richard Hughey, Kevin Karplus and Kimmen Sjölander. (1996). Tutorial in Stochastic Modeling Techniques: Understanding and using Hidden Markov models. University of California, Santa Cruz, CA.

[GR96] Gribskov, Michael and Nina L. Robinson. (1996). The Use of Receiver Operating Characteristic (ROC) Analysis to Evaluate Sequence Matching. *Computers Chem*. 20, pp. 25-34.

[GLE90] Gribskov, Michael, Roland Lüthy, David Eisenberg. (1990). Profile analysis. *Methods in Enzymology,* 183:146-159.

[GME87] Gribskov, Michael, Andrew D. McLachlan and David Eisenberg. (1987). Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA*, vol. 84, pp. 4355-4358.

[GBE+97a]Grundy, William N., Timothy L. Bailey, Charles P. Elkan and Michael E. Baker. (1997). Hidden Markov Model Analysis of Motifs in Steroid Dehydrogenases and their Homologs. To appear in *Biochemical and Biophysical Research Communications*. Corresponding author: Michael E. Baker, Department of Medicine, University of California, San Diego, La Jolla, CA 92093-0623.

[GBE+97b]Grundy, William N., Timothy L. Bailey, Charles P. Elkan and Michael E. Baker. (1997). Meta-MEME: Motif-based Hidden Markov Models of Protein Families. To appear in *CABIOS*. Corresponding author: William N. Grundy, Department of Computer Science and Engineering, University of California, San Diego, La Jolla, California 92093-0114.

[Has97] Haseltine, William A. (1997). Discovering Genes for New Medicines. *Scientific American,* March, pp. 78-83.

[HH96] Henikoff, Jorja G. and Steven Henikoff. (1996). Using substitution probabilities to improve position-specific scoring matrices. *CABIOS*, vol. 12 no. 2, pp.135-143.

[HS96a] Holm, Liisa and Chris Sander. (1996). The FSSP database: fold classification based on structure-structure alignment of proteins. *Nucleic Acids Research*, vol. 24, no. 1, pp. 206-209.

[HS96b] Holm, Liisa and Chris Sander. (1996). Mapping the Protein Universe. *Science*, vol. 273, 2 August, pp. 595-602.

[HR97] Huelsenbeck, John P. and Bruce Rannala. (1997). Phylogenetic Methods Come of Age: Testing Hypotheses in an Evolutionary Context. *Science,* vol. 276, 11 April.

[HK96] Hughey, Richard and Anders Krogh. (1996). Hidden Markov models for sequence analysis: extension and analysis of the basic method. *CABIOS*, vol 12 no. 2, pp. 95-107.

[HBK96] Hughey, Richard, Christian Barret and Kevin Karplus. (1996). Scoring Hidden Markov Models. Computer Engineering, University of California, Santa Cruz, CA 95064, USA.

[HKB+96] Hughey, Richard, Anders Krogh, Christian Barret and Leslie Grate. (1996). *SAM: Sequence Alignment and Modeling Software System*. Baskin Center for Computer Engineering and Information Sciences, University of California, Santa Cruz, CA 95064, Technical report UCSC-CRL-95-7, January 1995, Updated for version 1.3.1 (May 10, 1996).

[Kar95] Karplus, Kevin. (1995). *Regularizers for Estimating Distributions of Amino Acids from Small Samples*. Baskin Center for Computer Engineering and Information Sciences, University of California, Santa Cruz, CA 95064, Technical report UCSC-CRL-95-11.

[KSB+97] Karplus, Kevin, Kimmen Sjölander, Christian Barret Melissa Cline, David Haussler, Richard Hughey, Liisa Holm and Chris Sander. (1997). Predicting protein structure using hidden Markov models. Computer Engineering, UCSC, Santa Cruz, CA 95064, USA.

[KBM+94] Krogh, Anders, Michael Brown, I. Saira Mian, Kimmen Sjölander and David Haussler. (1994). Hidden Markov Models in Computational Biology. J. Mol. Biol., 235, pp. 1501-1531.

[LNC93] Lehninger, Albert L., David L. Nelson and Michael M. Cox. (1993). *Principles of Biochemistry.* Second edition. Worth Publishers, Inc.

[LBB+95] Lodish H., D. Baltimore, A. Brek, S. L. Zipursky, P. Matsudaira and J. Darnell. (1995). *Molecular cell biology.* Third edition. Scientific American Books Inc.

[MSE96] McClure, Marcella A., Chris Smith and Pete Elton. (1996). Parameterization studies for the SAM and HMMER methods of hidden Markov model generation. *Proc. Fourth Int. Conf. Intelligent Systems for Molecular Biology,* D. States et al., eds. AAAI Press, Menlo Park pp. 155-164.

[PL88] Pearson, W.R. and D.J. Lipman. (1988). Improved tools for biological sequence comparison. *Proc. Natn. Acad. Sci. USA,* 85, pp. 2444-2448.

[Pen96] Pennisi, Elizabeth. (1996). Teams Tackle Protein Prediction. *Science*, vol. 273, 26 July.

[Rab89] Rabiner, Lawrence R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE.*, 77 (2), pp.257-286.

[Rob92] Robbins, Robert J. (1992). Challenges in the human genome project. I*EEE Engineering in Biology and Medicine*, March, pp.25-34.

[SS91] Sander, Chris and Reinhard Schneider. (1991). Database of homology derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9:56-68.

[SK83] Sankoff, David and Joseph B. Kruskal, eds. (1983). *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley Publishing Company, Inc.

[Sea92] Searls, David B. (1992). The Linguistics of DNA. *American Scientist*, Vol. 80, November-December, pp. 579-591.

[Sjö97] Sjölander, Kimmen Voronov. Personal communication.

[SKB+96] Sjölander, Kimmen, Kevin Karplus, Michael Brown, Richard Hughey, Anders Krogh, I. Saira Mian and David Haussler. (1996). Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *CABIOS*, vol. 12 no. 4, pp.327-345.

[SED97] Sonnhammer, Erik L. L., Sean R. Eddy and Richard Durbin. (1997). Pfam: a Comprehensive Database of Protein Domain Families Based on Seed Alignments. *Proteins* 28:405-420.

[SWS93] Stultz, Collin M., James V. White and Temple F. Smith. (1993). Structural analysis based on state-based modeling. *Protein Science*, vol. 2, pp. 304-31, Cambridge University Press.

[Tau96] Taubes, Gary. (1996). Software Matchmakers Help Make Sense of Sequences. *Science*, vol. 273, 2 August.

[THT94] Thompson, Julie D., Desmond G. Higgins and Toby J. Gibson. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, Vol. 22, No. 22, pp. 4673-4680.

[Wel97] Wells, James A. (1997). Hormone Mimicry. *Science*, vol. 273, 26 July. 23

Appendix A

Tables 1 to 5 give the areas below the ROC curves for all of the 24 experiments done with removal of nodes and insertion of FIMs for each of the selected protein families. Each column represents a certain level of generalization (0.3, 0.5, 0.7 and 1.0 bits of encoding saving) and each row is the performance of either the basic model or the modified ones. For each modified model the measure and the threshold are given. The threshold is always with respect to the mean and in units of standard deviations.

Globin	**0.3 bits**	**0.5 bits**	**0.7 bits**	**1.0 bits**	**Model length**
basic	0.783135	0.792216	0.791892	0.793189	141
entropy -0.5	0.776216	0.791892	0.795351	0.798054	140
entropy -1.0	0.687892	0.756973	0.770919	0.782378	135
entropy -2.0	0.550703	0.604865	0.606379	0.624216	115
encoding +0.0	0.614270	0.601838	0.614270	0.614595	76
encoding +0.5	0.631892	0.699027	0.631892	0.649297	104
encoding +1.0	0.734594	0.792216	0.734594	0.756000	119

Table A1: Results for a single run on the globin family.

Ferredoxin					**Model**
1	**0.3 bits**	**0.5 bits**	**0.7 bits**	**1.0 bits**	**length**
basic	0.671200	0.756000	0.808800	0.877600	106
entropy -0.0	0.610400	0.732800	0.779200	0.867200	104
entropy -1.0	0.089600	0.112800	0.222400	0.484000	56
entropy -2.0	0.097600	0.111200	0.335200	0.513600	47
encoding +1.0	0.441600	0.572000	0.717600	0.782400	92
encoding +0.0	0.223200	0.528800	0.632000	0.747200	81
encoding -1.0	0.000000	0.000000	0.000000	0.000000	42

Table A2: The results for the first run on the ferredoxin family.

Ferredoxin					**Model**
2	**0.3 bits**	**0.5 bits**	**0.7 bits**	**1.0 bits**	**length**
basic	0.848800	0.886400	0.905600	0.916800	78
entropy -1.0	0.268800	0.564000	0.703200	0.769600	56
entropy -2.0	0.098400	0.434800	0.532000	0.707200	43

Table A3: The results for the second run on the ferredoxin family.

					Model
ADH	*0.3 bits*	*0.5 bits*	*0.7 bits*	*1.0 bits*	**length**
basic	0662222	0.668148	0.671852	0.673333	267
entropy -0.2	0.654074	0.663333	0.666296	0.669629	260
entropy -0.6	0.634074	0.112800	0.222400	0.484000	232
entropy -1.2	0.073704	0.442593	0.581482	0.609259	178
encoding +0.0	0.035185	0.046519	0.169630	0.337778	153
encoding +0.5	0.307407	0.485185	0.554074	0.618148	178
encoding +1.0	0.649630	0.658518	0.658148	0.655185	213

Table A4: Results for a single run on the ADH short-chain family.

Trypsine 1	**0.3 bits**	**0.5 bits**	**0.7 bits**	**1.0 bits**	**Model length**
basic	0.878200	0.878100	0.883100	0.883000	225
entropy -0.0	0.878100	0.901200	0.904300	0.906300	224
entropy -1.0	0.879700	0.896500	0.902500	0.899400	203
entropy -2.0	0.898900	0.900400	0.903300	0.903600	153
encoding +1.0	0.879700	0.889100	0.889000	0.898000	213
encoding +0.0	0.879300	0.879600	0.884500	0.896400	172
encoding -1.0	0.000000	0.000000	0.000000	0.000000	42

Table A5: The results of the first run on the trypsine family.

Trypsine 2	**0.3 bits**	**0.5 bits**	**0.7 bits**	**1.0 bits**	**Model length**
basic	0.890100	0.903700	0.903400	0.905100	262
entropy -0.0	0.890100	0.901200	0.903200	0.903700	260
entropy -1.0	0.896200	0.899200	0.901900	0.902200	236
entropy -2.0	0.887300	0.894300	0.894600	0.902700	187

Table A6: The results of the second run on the trypsine family.

1hurA	**0.3 bits**	**0.5 bits**	**0.7 bits**	**1.0 bits**	**Model length**
basic	0.046711	0.046316	0.046974	0.045000	180
entropy -0.1	0.048421	0.046579	0.046316	0.045921	179
entropy -0.6	0.047500	0.047105	0.046316	0.045790	169
entropy -1.2	0.046053	0.045921	0.045658	0.045263	155
encoding +0.0	0.047106	0.050921	0.049737	0.049737	94
encoding +0.5	0.049605	0.047632	0.047895	0.047500	126
encoding +1.0	0.045790	0.046184	0.046316	0.046842	152

Table A7: Results for a single run on the 1hurA family.

Appendix B

Two additional tests were done where only the position-specific amino acid distributions were changed. These were done on the trypsine and the ferredoxine families. No generalization was done in these tests.

Trypsine	**Area under ROC$_{50}$ curve**	**Number of nodes left with position-specific amino acid distributions**
basic	0.909400	262
entropy -0.0	0.911700	259
entropy -1.0	0.900100	223
entropy -2.0	0.895400	166

Table B1: Results from a single run on the trypsine family.

Ferredoxine	**Area under ROC$_{50}$ curve**	**Number of nodes left with position-specific amino acid distributions**
basic	0.916800	78
entropy -0.0	0.918400	76
entropy -1.0	0.906400	50
entropy -2.0	0.910400	36

Table B2: Results from a single run on the ferredoxin family.