The use of structural information to improve biological sequence similarity searches

Jeanette Tångrot

Licentiate Thesis
UMINF-03.19

Submitted for the partial fulfillment of the requirements for the degree of Licentiate in Technology.

Department of Computing Science
Umeå Center for Molecular Pathogenesis
Umeå University
SE-901 87 Umeå, Sweden

UMINF-03.19 Print & Media, Umeå Universitet ISSN-0348-0542

Till mina föräldrar,
Eva-Britt och Leif Hargbo

Abstract

Bioinformatics is a fast-developing field that make use of computational methods to analyse and structure biological data. An important branch of bioinformatics is structure and function prediction of proteins. To determine the structure of a protein is a crucial part in the characterisation of the molecule. The structure can also give clues about how the protein functions in the cell. Since the experimental determination of a protein structure can be both difficult and time-consuming, and in some cases is impossible using current techniques, it is desirable to be able to predict the structure. If two protein sequences are very similar, it is known that they share the same structure. However, there are many proteins that share the same fold, but have no clear sequence similarity. To find these relationships, and be able to predict the structure of these proteins, so called "protein fold recognition methods" have been developed.

In this thesis, the field of bioinformatics is briefly surveyed, and two fold recognition methods are presented. Both methods use hidden Markov models (HMMs) to find related proteins, and they both exploit the fact that structure is more conserved than sequence, but in two different ways.

The first paper introduces the reader to the field of molecular biology, and also describes some common tools used for protein sequence comparison. HMMs in general are described in detail, as well as some methods for the construction of multiple structure superposition. Since 3D structure is more conserved than sequence, it is expected that a multiple sequence alignment based on a multiple structure superposition, is more biologically correct than an alignment based on sequence information, especially for proteins with low sequence identities. Our structure anchored HMMs (saHMMs), which are presented in the paper, are constructed from multiple sequence alignments that are based on structural superposition. The paper also describes the selection of representatives for each protein family, that were used for the construction of the saHMMs. In this selection, no protein in a given family have a sequence identity higher than a certain threshold to any other protein in the same family. The threshold is defined as the border to the so-called twilight zone. The saHMMs are shown to be able to find the family relationships for almost 90% of the test cases, even when the saHMMs are based on two proteins only.

The second paper describes the secondary structure HMMs (ssHMMs). These HMMs are based on an ordinary multiple sequence alignment, as well as on the secondary structures of the proteins. When a query sequence is compared to

the ssHMM, a predicted secondary structure is used, and the score based on the sequence is increased or decreased depending on the match of the secondary structures. A rigorous benchmark is also presented, and used to compare automatically generated HMMs with ordinary sequence search methods. The results show that the ordinary sequence search methods tested perform about as well as automatic HMMs built from multiple alignments. The ssHMMs, however, are better at detecting the correct fold of a protein than all the other methods tested.

Preface

The thesis consists of a short survey and the following two papers.

I. J. Tångrot, B. Kågström and U. H. Sauer. Structure anchored HMMs (saHMMs) for sensitive sequence searches. UMINF-03.18

II. J. Hargbo* and A. Elofsson. Hidden Markov Models That Use Predicted Secondary Structures For Fold Recognition. *Proteins: Structure, Function and Genetics* 36:68-76, 1999

In the survey part a short description and introduction into the field of bioinformatics is given.

*Now Tångrot

Acknowledgements

There are many people I would like to thank, and first of all I would like to say THANK YOU to all those deserving it, but who are not mentioned here.

Many thanks to my supervisors, Bo Kågström and Uwe H. Sauer, for guiding me through the research and writing process. I also thank Arne Elofsson, advisor of my Master thesis and co-author of Paper II.

I would also like to thank Gunilla Wikström and Isak Jonsson, both who with their help made the completion of this thesis possible. I also owe Åke Sandgren and Marek Wilczynski many thanks for their help with programming, programs and databases.

Thanks to all my colleagues, both at the Department of Computing Science and at the UCMP, for providing a good working environment.

At the Department of Computing Science, I would like to give a special thank to Lena Kallin Westin, Fredrik Georgsson and Niklas Börlin for advice and discussions. There are several others I would like to mention, but of fear of forgetting someone I just say: 'thank you!' to all those people (especially the "lunch-walkers ") making my lunches and coffee-breaks something to look forward to.

At the UCMP, I would like to especially thank the X-ray group; Stefan Bäckström, Fredrik Ekström, Christin Grundström, Tobias Hainzl, Shenghua Huang, Andreas Hörnberg, Anders Karlsson, Erik Lundberg, Anders Olofsson and Ulrika Wikström for making me feel like home during my irregular visits. Fredrik and Stefan, thanks for encouraging talks and good advice!

And of course many thanks to Olof, for giving me love, care and dinner, and for being who you are. And to Emelie, our daughter, just for being.

This research was conducted using the resources of High Performance Computing Center North (HPC2N).

Financial support has been provided jointly by the Department of Computing Science and Umeå Center for Molecular Pathogenesis (UCMP).

Umeå, November 2003
Jeanette Tångrot

Contents

Chapter 1

Introduction

1.1 Bioinformatics

Bioinformatics is a very broad area of research, with the common denominator that it uses computational methods to analyse and structure biological data, and from this make theoretical predictions about biological processes. Much of the research in bioinformatics is multidisciplinary, including computing science, statistics, and structural and molecular biology.

There are several branches of bioinformatics, some of which are briefly described below. For definitions of terms and acronyms used, we refer to Paper I of this thesis. The list of branches presented here is probably not complete, and some people would probably claim that some important aspect is missed or that some things listed below not at all is bioinformatics in its true sense. This only shows the difficulty in dividing research areas into well-defined categories. Two books covering some aspects of bioinformatics are [1] and [2].

1.2 Phylogenetic trees, genomics

An important part of bioinformatics is the analysis and comparison of genes and genomes. Starting on the genomic scale, phylogenetic trees can be constructed to show the evolutionary relationships between different organisms. These kinds of trees can also be constructed for individual genes, showing how they have evolved and how they are related.

Each amino acid is represented in the DNA by a combination of three bases (of the four A, C, G and T), a so-called codon. One amino acid can be represented by between one and six codons, depending on which amino acid it is, but all codons coding for a specific amino acid are not equally abundant. The patterns of codon usage differ between organisms and between genes in the same organisms, and can be studied to find similarities and differences. The amounts

of the bases C and G in genes also differ between organisms and genes, and give information about the history of a gene, the level of expression, and about evolution.

Having the human and other genome sequenced, it is important to be able to find which parts of this huge amount of DNA that are genes that code for protein. The majority of the human and other large genomes does not code for any gene. Methods have been and are developed to find the start, stop and other patterns characteristic for genes. It is also interesting to locate regulatory regions, for example regulating when and how often a gene is transcribed, or possible cleavage sites, where the final protein is cleaved to for example remove signal sequence after they have been used. During and after translation, some proteins are transported out of the cell, or to the mitochondria (the energy fabrics of the cell). These proteins have a signal peptide telling where they should be located. Neural networks have been used to, based on the sequence, locate this signal and determine where a protein should be located. Some proteins are inserted into the membrane surrounding the cell. Hidden Markov models (HMMs) are used to determine whether a protein is a membrane protein or not, and which parts of the protein are inside the cell, inserted into the membrane, and outside of the cell.

Stochastic grammars have been used to determine evolutionary relationships between biological sequences, and to find a common ancestor. These methods can also be used to determine the secondary structure of RNA molecules, structures that form due to base pairing within the RNA chain.

Statistics in different forms can be used to study genetic diseases. Healthy and sick people are compared on a genetic level, and genetic properties are determined.

1.3 Study of protein function

A field that has grown the latest years is the study of protein function on a larger scale, both experimentally and theoretically. To study chemical modifications, the binding of cofactors, interactions between proteins, etc., is called proteomics (compare to genomics, the study of the genes/genome). Bioinformatics is needed when it comes to the analysis of experimental data. For example, patterns from mass spectrometry experiments can be compared to databases, to find what the sample contains, image analysis of 2D-PAGE pictures can be performed to determine quantities in the individual spots, or the sequences of acetylated mutants can be compared to the wild type to locate differences.

Based on currently available information, metabolic networks, showing which proteins are parts of which pathways and how the pathways are connected, can be constructed. Work is also done to predict metabolic networks, to make models of signal transduction and other important processes in the cell.

A field that has grown rapidly during the last few years, is functional genomics in the form of microarrays. Microarrays are small arrays, where several different DNA strands are attached. These are used to study gene expression in different types of cells and under different conditions. When a gene is expressed, mRNA is produced with DNA as a template, and the RNA in turn is used as a template to build protein molecules. Not all genes are expressed in all cells, and a single gene is only expressed in a given cell when it is needed. The particular mRNA molecules present in a cell at a certain time can be captured using the DNA arrays, and thus capturing information on the genes currently expressed in the cell. The mRNA has the ability to base pair with the DNA, due to the chemical similarity between DNA and RNA. It is this ability that is used in the technique. The RNA molecules bound can be detected and the strength of the signal is a measure of the amount of RNA in the cells.

Bioinformatics enters when processing and analysing the data. Often gene expression is studied under different conditions, for example the expression of genes in starving cells can be compared to that in “normal” cells. In the experiments, it is the difference in expression that is interesting, not the actual expression levels. Scientists are looking for genes that are up- or down-regulated (that is expressed more or less than under "normal" conditions). The task is to find patterns in the expression of different genes. In this way it is, for example, possible to locate genes belonging to a common pathway (cooperating to perform a certain task in the cell), since these proteins should be expressed in a similar way. The very amount of data makes it a difficult task to try and find patterns between genes.

Another, very serious, problem is the often bad quality of the data. This results in unreliable results, since they may differ between experiments. It is very expensive and labour expensive to perform several experiments, why this problem in part has to be solved by clever design of the experiments. Work is currently under way to reduce the impact of such errors.

To find patterns in gene expression, several different approaches have been used, such as graph theory, self-organising maps (SOM) and the singular value decomposition (SVD).

1.4 Databases and information searches

Several databases containing biological data are available via the Internet. These databases might store raw data as well as annotated, or literature references. Much work is done to annotate the raw data and construct cross-links to create new, value added databases. Another area of research is to combine several databases and/or to index web pages, to make it possible to find all data relevant with just one or a few searches, and to quickly find other, related information. To achieve this, work has been done using for example XML.

A field also related to bioinformatics is the visualisation of biological data.

1.5 Biological sequence analysis

The classical branch of bioinformatics is the analysis of biological sequences, such as DNA and protein. This is most often done in the form of sequence alignments, where one sequence is matched as good as possible to another sequence. From a sequence alignment it is possible to determine characteristics common to the two sequences, such as conserved amino acids or conserved properties such as size or charge of the aligned residues.

More information of a whole protein family can be gained from multiple sequence alignments, where many sequences are aligned simultaneously. Several methods have been developed for alignment of multiple sequences. Some examples of approaches used are dynamic programming in different forms (see Paper I), the divide and conquer strategy, where the alignment is divided into small manageable parts, genetic algorithms, which use the analogy of genetic mutations and recombination to find the best alignment. Other approaches are to iteratively improve a rough start alignment, or to progressively align the sequences, i.e. to add one at a time following some schema.

To model sequences and sequence families, Markov chains and other statistical methods are common.

1.6 Structure and function prediction

The ultimate goal of much work in bioinformatics is to be able to predict the structure, and perhaps even the function, of a protein based on its amino acid sequence. The rationale behind this is that it in general is very expensive, difficult and time consuming to determine the structure of a protein experimentally. For certain proteins it is even impossible using current techniques.

The function of a protein is also hard to determine, especially if one has no clues about what the role of the protein could be. If one was able to predict the structure of a protein, this would give clues about possible functions of the molecule, and together with other techniques it would be possible to predict the function of the protein. This prediction can in turn make a base for constructing tailored experiments to determine the true function of the protein. The work of this thesis is done in the area of structure prediction, or rather "sequence similarity searching". The prediction of function based on sequence and structure is still a largely unexplored area, much due to the need for good structure prediction methods to base the work on.

There are many approaches to structure prediction. The most direct, and perhaps most difficult, approach is to make *ab initio* prediction. This means to try to calculate the fold of a protein based on its sequence and knowledge about the amino acids chemical properties, using different energy functions. In a way, this is equivalent to make the protein sequence fold in the computer in a way similar to the way it is done in the cell. Another *ab initio* approach

is the Frankenstein monster model method, which combines small parts from many different known structures, finding the combination of structural parts that seems to fit the sequence best. Here, steric and chemical properties of the amino acids making up the sequence are considered to find the best combination of structural parts. Currently, there has been success only for very chains, in the order of 150 residues [4]. If one would succeed in constructing an efficient *ab initio* method, this would get important insight to the natural folding process and parameters/properties important for defining the particular fold a certain protein adopts.

Other methods use already known structures as templates to construct the fold of a given sequence. A common method is threading, where the sequence is "threaded" through a number of structures, choosing the one that fits the steric and chemical properties of the chain the best. In homology modelling, the sequence is fitted to the sequence of a protein with known structure, and a possible structure is determined based both on the fit of the sequences and on the known structure. For this procedure to be possible, one has to have some way to locate the closest homologue (with known structure) of the sequence, and to be able to fit the sequences in a biologically sensible way. This area, often called fold recognition, is large within bioinformatics. The base for much work in this area is sequence alignments in one way or the other.

Alignment methods have been developed that are able to find the sequence in a database that fits the given sequence best, that is, that gives the best alignment between the two. More sophisticated methods have also been developed, that try to model the sequences of a whole protein family to be able assign the sequence to related proteins, even if they are distant relatives. Two examples of such models are profiles and hidden Markov models (HMMs, see Paper I). To cluster and classify proteins into families, methods based on singular value decomposition and principal vector analysis have also been used.

Every second year, the Critical Assessment of Structure Prediction (CASP, http://predictioncenter.llnl.gov/) takes place, where methods for structure prediction are tested on real targets. Predictors are invited to submit predictions on sequences whose structures are due to be released, but that are not known to the predictors at the time of prediction. This blind test makes it possible to make a fair comparison between the best methods available today. The results from the latest assessments show that expert evaluation and intervention in the prediction procedure still is superior to purely automatic methods, but the gap is decreasing. It seems like the combination of results from several different methods give the best results.

During evolution, the structure of a protein is conserved in a much higher degree than the amino acid sequence. The reason is that some amino acids, or combinations of amino acids, are able to perform about the same task in the protein, and therefore can be substituted for each other without changing the conformation of the protein. For example, at some positions it may be

enough to have a reasonably small amino acid for the chain to fold correctly. For a protein to function correctly, it is the conformation of the chain and a few crucial residues that are important. Another aspect of this is that it is the core of the protein that is important to keep it together in a defined way. Residues located at the surface of the molecule are in most cases not significant at all. All this has the effect that two proteins, very similar in structure and perhaps performing about the same task in the cell, might differ a lot in sequence. This makes it difficult to locate relationships based on the amino acid sequence only.

In the work presented in this thesis, HMMs are used to locate which family a given family most likely belongs to. On the base of the fact that structure is more conserved than sequence during evolution, the work presented here takes two approaches to include structural information into HMMs, to make them more sensitive to distant relationships.

In Paper I, the structure anchored HMM (saHMM) method is presented. The saHMMs are based on alignments derived from the structural superposition of protein structures. These kinds of alignments are supposed to be more biologically correct than those based on sequence and statistics, especially for sequences with very low sequence identity. The saHMMs are also built from a careful selection of proteins, where it is ensured that no protein is more than about 25% identical to any other in the same family. This is to guarantee an even spread among the proteins chosen as representatives for each family. HMMs built from structural alignments of these structurally similar proteins having low sequence identities are designed to be good at finding distant relatives.

In Paper II, secondary structure HMMs (ssHMMs) are described and evaluated. The ssHMM is a combined HMM, taking both the sequence and the secondary structure of the proteins into account. If the secondary structure of the query sequence matches that of the HMM, the score for that match is increased, even if the particular amino acid at that position does not fit well. For a sequence of which one does not know the structure, the secondary structure of course has to be predicted using some secondary prediction method before it can be compared to the HMM.

Chapter 2

Summary of Papers

In this chapter, a summary of each of the two papers included in the thesis is given. Complete references to the papers are found in the Preface.

2.1 Paper I

In Paper I, a general introduction to molecular biology is given, followed by the description of several methods commonly used in bioinformatics and specifically for the analysis of biological sequences. Hidden Markov models (HMMs) are described on a more theoretical level, as wells as some methods for the construction of multiple structural superpositions of proteins.

Based on this background, the structure anchored HMM (saHMM) method is presented. This method makes use of the fact that the structure of a protein is more conserved during evolution than its amino acid sequence. A multiple sequence alignment based on a multiple superposition of the structures of the proteins to align, is therefore likely to be more biologically correct than an alignment based on sequence information. This is especially true for proteins with very low sequence identities. The saHMMs are built from these kinds of structure anchored multiple sequence alignments.

A careful selection of representative sequences to base the saHMMs on is created. For each family of proteins, the representatives are selected such that no representative sequence have a sequence identity to any other in the same family, above a certain threshold. This threshold is defined as the border to the twilight zone, according to the equation derived in [3]. Also, only high quality structures determined using X-ray crystallography are selected.

Parts of this work have previously been presented at several workshops and conferences, including Bioinformatics 2000 (Helsingör, Denmark), Bioinformatics 2001 (Skövde, Sweden) and Bioinformatics 2002 (Bergen, Norge).

2.2 Paper II

In Paper II, a rigorous benchmark to test the performance of different fold recognition methods was developed. This benchmark contains representatives for all proteins with known structure, that are matched against each other. In the study, the benchmark was used to compare the performance of automatically created hidden Markov models (HMMs) with standard sequence search methods such as BLAST and FASTA.

The second part of this work consists of the construction and evaluation of secondary structure HMMs (ssHMMs). In the ssHMMs, predicted secondary structures and multiple alignments are combined into a method that is shown to perform better than the other methods tested. The secondary structures are included in the HMM in a way such that a match between the predicted secondary structure of the query, and the secondary structures represented by the HMM, increases the score for that particular match, even if the amino acid in that query position does not fit well.

In the benchmark test, the abilities of the methods to identify the correct fold of a sequence were measured. The fold was defined according to the SCOP classification. It was shown that the correct fold of a protein was found for 10% of the test cases when HMMs were built from single sequences. Building the HMMs from multiple alignments, and thereby including multiple sequence information, increased that number to 16%. Including secondary structure information further increased the number of correctly assigned folds to 20%. If the true secondary structure of the query was used instead of the predicted, the correct fold was detected for 27% of the test cases. In the study, standard sequence search methods identify the correct fold in 13-17% of the test cases. This is almost as good as the HMMs performs. The reason might be that the alignments used are not diverse enough, and do not contain a large enough number of sequences.

Bibliography

[1] P. Baldi and S. Brunak. *Bioinformatics - the machine learning approach*. The MIT Press, Cambridge, Massachusetts, 1999.

[2] R. Durbin, S. Eddy, A. Krogh, and G Mitchison. *Biological Sequence Snalysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[3] B. Rost. Twilight zone of protein sequence alignments. *Protein Engineering*, 12:85–94, 1999.

[4] K. T. Simons, C Strauss, and D. Baker. Prospects for ab initio protein structural genomics. *Journal of Molecular Biology*, 306:1191–1199, 2001.

I

Paper I

Structure anchored HMMs (saHMMs) for sensitive sequence searches*

Jeanette Tångrot [1,2], Bo Kågström [1] and Uwe H. Sauer [2]

[1]Department of Computing Science
and
[2]Umeå Center for Molecular Pathogenesis
Umeå University
SE-901 87 Umeå, Sweden.
*E-mail:*jeanette@cs.umu.se

Abstract

To be able to predict the structure, and perhaps even the function, of a protein, one needs a way to find relatives to that protein. This is often done by comparing the sequence of the protein of interest to the sequences of others, often in the form of hidden Markov models (HMMs), to find the best match. The structure of a protein is much more conserved during evolution than is its sequence. A multiple sequence alignment based on a multiple superposition of the structures of the proteins to align, is therefore likely to be more biologically correct than an alignment based on sequence information. This is especially true for proteins with very low sequence identities, and this is used in the structure anchored HMM (saHMM) method described here. The saHMMs are constructed from multiple sequence alignments derived from the multiple structure superposition of the corresponding structures.

A careful selection of proteins, to base the saHMMs on, is also developed. Only high quality structures, determined using X-ray crystallography, are selected. For each family of proteins, the representatives are selected such that no representative sequence, have a sequence identity above a certain threshold, to any other in the same family. This threshold is defined as the border to the twilight zone, according to the equation derived in [Rost, Prot. Eng. 1999; 12:85-94].

This work begins with a general introduction to molecular biology, followed by the description of several methods and techniques commonly used in the comparison and analysis of protein sequences. Especially, HMMs are described in some detail, as well as a few methods for the construction of multiple structural superpositions of proteins. Based on this background, the saHMMs and the selection procedure are described.

Keywords: saHMM, protein structure, multiple structure superposition, multiple sequence alignment, hidden Markov model (HMM), SCOP

*From UMINF-03.18, ISSN-0348-0542, 2003.

Contents

1 Introduction

The sequencing of the human and other genomes have generated a lot of biological data to analyse. To fully use the information gathered, all genes have to be located and their roles in the cells have to be determined. For all proteins to be fully characterized, one wants to know their three-dimensional (3D) structures, their molecular and cellular functions, and their interactions with each other and other molecules. With the function of a protein we mean its role in the cell, for example as a building block making up the very walls of the cell or pumps that transport other molecules in and out of the cell, as a helper molecule that makes some chemical reaction go faster, or as a signal sending messages between different cells. Due to the vast amount of proteins, it is not feasible to study each molecule in each genome experimentally. Instead, the characteristics of a newly sequenced protein is usually derived by sequence and/or structure comparison to an already characterized protein. Also, the procedures used to determine the 3D structure of a protein are time-consuming and may be problematic. If possible, it would be better to use computational methods to guide the experimental approaches, and ultimately to predict the structure of a protein based on its sequence only. However, this is still a task for the future.

When comparing proteins it is informative to distinguish between homology and analogy. Analogous proteins are considered as a product of convergent evolution to a similar 3D structure, while remote homologues originate from a common ancestor. A clear distinction is difficult to obtain because functional relatedness is hard to prove. A distinction can in some cases be made based on similarity in side-chain directions [48].

Comparison of newly sequenced proteins to already known proteins gains lot of information. When the yeast genome was sequenced in 1996, a function could be assigned to about 65% of the proteins. About half of these functions were previously determined by experiment, while the other half could be determined by homology to other proteins [16]. In general, about 40-60% of the proteins in a sequenced genome are found to be similar to another protein with known structure (see for example [30]). The specific number depends on the particular genome and the method used to find the similarities. The use of bioinformatics to extract biological information from genome data has been reviewed in [5]. The work is unfortunately not very recent, but still interesting since it describes some common bioinformatics approaches. Here, we focus on the comparison of proteins and protein sequences.

Proteins with sequences that are very similar, are known to also have similar structures (with some exceptions). Sequences more than 20-30% identical are also relatively easy to align to each other. Below this limit the quality of an alignment cannot be guaranteed. The significance of an alignment never gets higher than that of an alignment of two random sequences. This presents a problem when dealing with proteins that are very similar in structure, but whose

sequences are very dissimilar. Similarity in structure says nothing about how similar (or dissimilar) the corresponding sequences are.

Structural divergence as measured in root mean square (RMS, the Euclidean distance from one atom in one structure to the corresponding atom in the other) is exponentially related to sequence divergence measured in sequence identity [17][82]. That is, lower sequence identity means higher RMS, with a sharp increase in RMS in the twilight zone, below 20% sequence identity. In this case, the interesting parts of the proteins are the hydrophobic cores, that keep the structures together and define the fold of the proteins. Outside the core, especially in loop regions, the sequence identity might be very low even in proteins with low RMS in the core. Domains that share the same fold also have the same function down to about 40% sequence identity, and belong to the same functional class down to about 25% sequence identity [82]. Of course these numbers are averages, and there may be pairs where function is better or worse conserved. One problem in comparing functions is the lack of measure of functional similarity, and the lack of a common language and classification of functions.

One common tool for comparing proteins is profile hidden Markov models (HMMs, Section 4). Often a HMM is constructed to model a sequence family of interest. The HMM can then be used to search genomes for previously unannotated members of that family, or sequences can be searched against a data base of HMMs to find which model (and thereby family) that fit the sequences best.

To construct profile hidden Markov models for protein families one needs a multiple sequence alignment for each model to build. This, however, can give rise to problems in the case of very dissimilar sequences, since it is then difficult to generate a good sequence alignment. The fact that "ordinary" models are based purely on sequence alignment means that, even though very similar in structure, proteins too divergent in sequence from the proteins the HMM was built from, will never be found by the model. A striking example is the Runt domain, which was assumed to have a completely new fold since it did not have any significant similarity to any protein with known structure. However, when the structure was solved it turned out that the protein had an S-type Ig-fold, and was very similar to for example STAT-1 and NF-kappa B. This even though it does not have more than about 10% sequence identity to either of these proteins.

This indicates a need for some way to be able to recognise proteins similar in structure, even if the sequences are different. One way to do this would be to construct a HMM based on an alignment from a collection of sequences very similar in structure, but dissimilar in sequence. Since it is impossible to find statistically significant alignments of proteins with a sequence identity below the limit of 20-30% (see Section 7.3.1), and at the same time a sequence alignment is needed to find similarities to other proteins, and in particular to construct a hidden Markov model, some other way of constructing a multiple sequence alignment has to be found.

Our approach is to use superpositions of known structures, and from them

deduce the corresponding sequence alignment. This makes it possible to align protein sequences not similar at all, as long as their structures are similar enough to superimpose. These structure-based sequence alignments can be used to build HMMs, hopefully better at recognising even very distant relatives of the protein family. The goal of this work is to be able to find protein similarities below the twilight zone, far into the midnight zone.

The twilight zone is the border where the percentage sequence identity between two aligned protein sequences no longer tells whether the two proteins are related or not (Figure 1, see also Section 7.3.1). It can be defined by plotting the sequence identity against the alignment length for all possible pairwise alignments of "all" known proteins. It turns out that protein pairs falling above the curve in Figure 1 always are homologous proteins. Around the curve, the number of unrelated pairs start to appear, and increase as one goes below the curve. Below the curve most of the protein pairs are not related at all, but there still exist some pairs that are. These are the pairs we are interested in finding.

Of course, the result of a search for similar structures is only a first step towards the characterization of a new protein. Different members of the same superfamily (see Section 6.6) can have very different functions, even though the structures are similar. Hence, the identification of distant relationships only provides a clue that can be used to guide experiments to focus on the most likely function of the protein.

The rest of the paper is organised as follows. In Section 2 some basic biology important for this work is described. Section 3 describes multiple sequence alignments and methods for constructing them, as well as other methods for comparison of protein sequences. In Section 4, hidden Markov models are treated, and in Section 5 some methods for superposition of protein structures are presented. Section 6 describes some important biological databases. In Sections 7, 9 and 10 our structure anchored HMMs are described, including tests and results derived. Some related work is described in Section 8. Finally, in Section 11 strengths and weaknesses of our method are discussed, as well as plans for future work.

2 Biological background

The information carrier in all living organisms are the strains of deoxyribonucleic acid (DNA). All the genetic material, that is the description of every part of our cells and ourselves (in a biological sense), is stored in the DNA, which is located in the cell nucleus of every cell in the body. This information is then transferred to finally construct the proteins, the molecules that perform most of the work in the cells. This information transfer is called the central dogma of molecular biology, and is illustrated in Figure 2.

DNA consists of four types of bases, commonly called A (for adenine), C (cytosine), G (guanine) and T (thymine), connected into strands. In bioinfor-

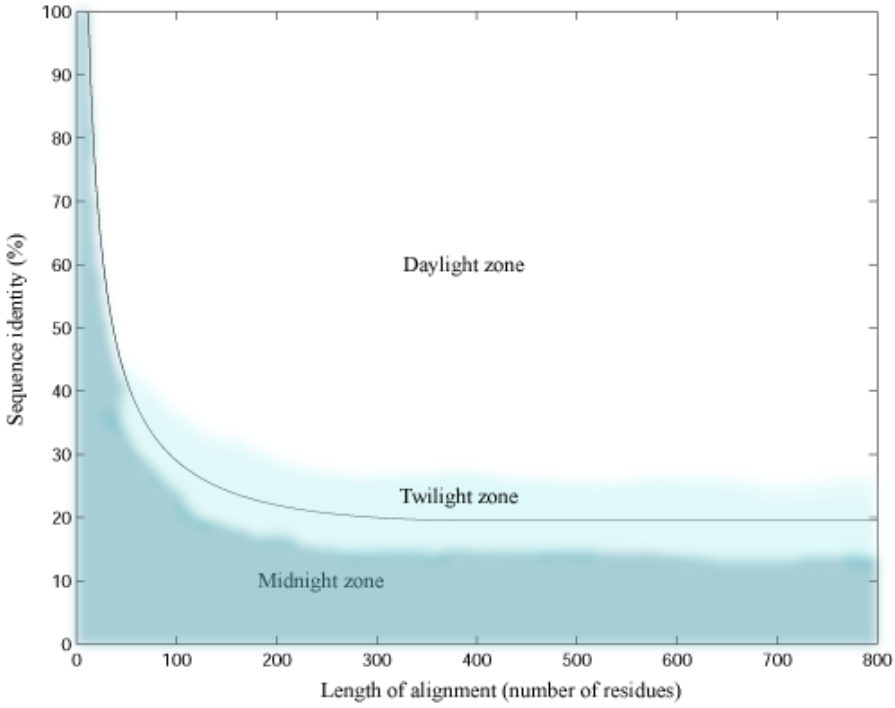


Figure 1: The curve that defines the twilight zone according to Rost [65]. When the percentage sequence identity is plotted to the length of the alignment of two protein sequences, related proteins fall above this curve. The further into the twilight zone one gets, the less likely it is that the two proteins are related.

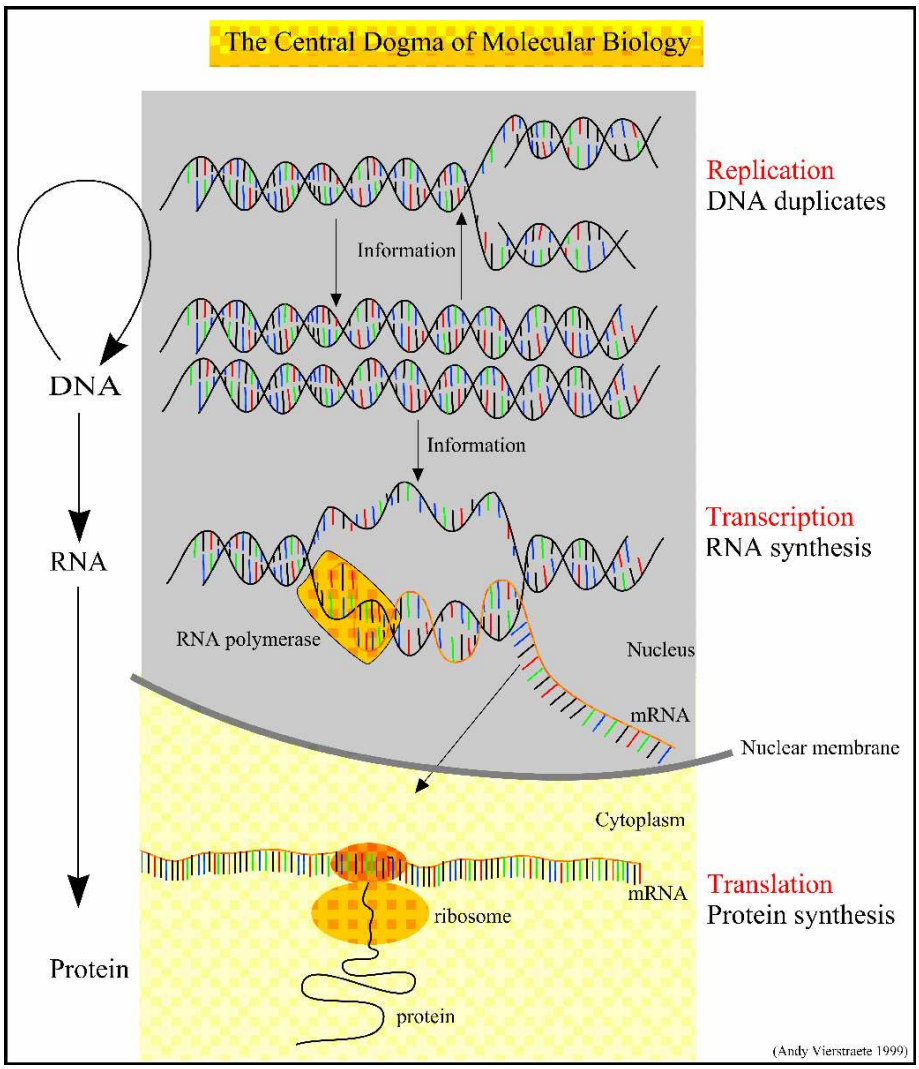


Figure 2: The central dogma of molecular biology. The picture shows the flow of information in the cell, from DNA to protein. The picture is kindly provided by Andy Vierstraete (http://allserv.rug.ac.be/~avierstr/tif.html).

matics, sequences of these letters, corresponding to the genetic information, are investigated and compared. The DNA is stored as double helices, where two strands are twisted around each other. The two strands are connected by base-pairing, such the A binds to T and C binds to G. Whenever an A is seen on one strand, a T appears on the other, meaning that the two strands are each others complement and that each strand contains all the information stored. When cells divide, for example during embryonic development, the DNA is replicated to produce two identical double helices (Figure 2). During replication, each of the two strands act as a template for the new molecule.

During transcription (Figure 2), parts of the DNA is translated into mRNA (messenger ribonucleic acid), consisting of the bases A, C, G and U (uracil). There is a one-to-one correspondence between the DNA bases and the mRNA bases, usually the mRNA is a simple copy of part of the DNA, with all T's replaced by U's. The mRNA sequences too are interesting from a biological perspective, since they represent molecules that actually perform things, apart from the DNA that mainly stores all information.

The mRNA is then used as a template for proteins, which are produced during translation. Proteins are built from 20 kinds of amino acids, often represented by 20 letters (see Figure 3). There is a three-to-one correspondence between RNA and protein, with three RNA-bases representing one amino acid in the protein. The 20 kinds of amino acids each have different characteristics. They all have a common base (coloured red in Figure 3), where they are linked together to form the protein chain. This chain of amino acids forms the so-called backbone of the protein. Very short stretches of connected amino acids are called peptides. To the common base, each kind of amino acid has a unique side-chain connected, which gives the amino acids their different properties. The 20 kinds of amino acids can be divided into groups with similar properties, for example hydrophilic/hydrophobic (water loving/water avoiding), neutral/charged or small/large.

In the context of a protein chain, the amino acids are called residues. The protein chain folds into a well-defined 3D structure, determined by the actual sequence of amino acids. It is the chemical properties of the amino acids that determine the shape of the protein molecule.

The overall structure of a protein is defined at different levels. The pure sequence of amino acids, represented by a sequence of letters, is called the primary structure of the protein, or simply the sequence. Parts of the chain fold locally to form so called secondary structure elements. There are two major kinds of secondary structures: alpha helices and beta strands, that form beta sheets (see Figure 4). The alpha helices are often shown as spirals in pictures of proteins, while the beta strands are shown as arrows.The secondary structures come together to form super-secondary structures or motifs. Two examples are beta hair-pins, consisting of two anti-parallel beta strands and the short loop connecting them, and the beta-alpha-beta motifs, consisting of two parallel beta

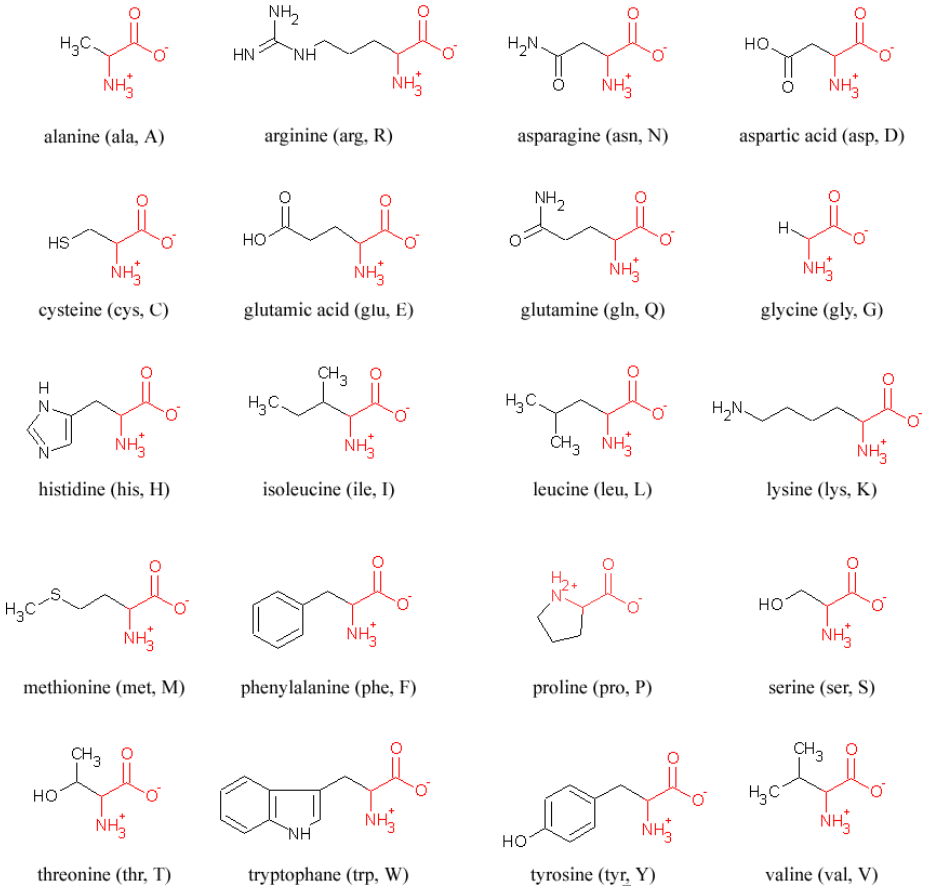
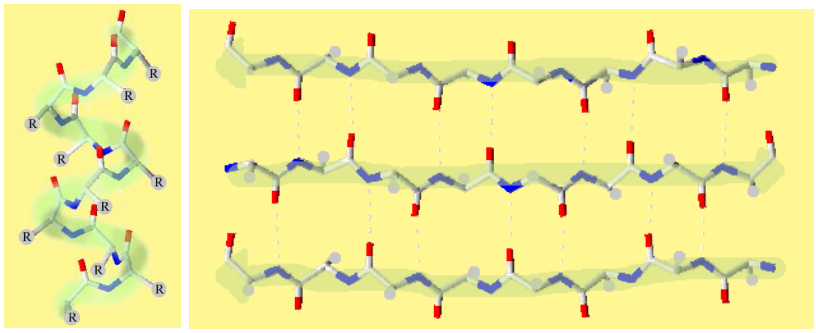


Figure 3: The twenty amino acids. The common parts of all amino acids, that are connected to form the backbone of the protein, are coloured red. In parenthesis are the three-letter and one-letter codes for the amino acids.



(a) (b)

Figure 4: The two most common types of secondary structures. Only the main chain (backbone) is shown, the side chains are indicated by filled grey circles. (a) An alpha helix. (b) A beta sheet consisting of three anti-parallel strands.

strands with an alpha helix in between them. The super-secondary structures are packed together to form domains, that in turn pack to form the tertiary structure of the protein. In general, proteins pack such that amino acids that do not like water are stored in the inside of the protein and form the so called hydrophobic core, while residues that easily interact with water are found on the outside of the protein. A protein domain is a region of the protein that has its own hydrophobic core, and that interact relatively little with the rest of the protein. Domains also can fold independently of other parts of the protein. In Figure 5, an example of a protein with two domains is illustrated.

Sometimes, several protein chains pack together to form complexes, that build up the so called quaternary structure. Very large collections of proteins, sometimes packed together with RNA or DNA, are called macromolecular assemblies. One example of such an assembly is the ribosome, which produces new proteins from an mRNA template.

The particular packing and orientation of the secondary structure elements, and the location of residues important for the structure and/or function of a protein, is called the fold of the protein.

In Figure 6, an example of the different levels of protein folding is shown.

Protein structures can be illustrated in a number of ways. In Figure 7, five different representations of the same protein are shown as an example.

The amino acid sequence of a protein can easily be determined from its corresponding DNA, and the sequence of the DNA is easily found experimentally.

The 3D structure of a protein can be determined by experimental methods such as X-ray crystallography and Nuclear Magnetic Resonance (NMR). As of

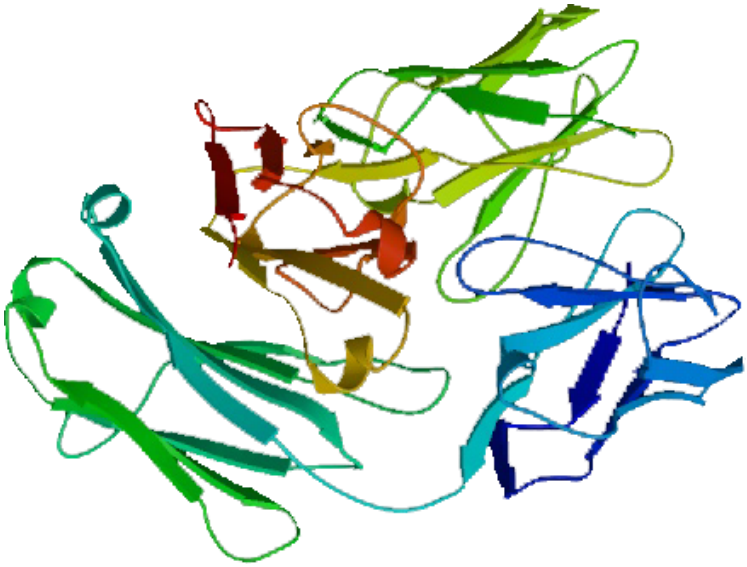


Figure 5: A dimer (two aggregated molecules) of the immunoglobulin light chain. The chain folds into two separate domains, coloured blue and blue-green, respectively, that mainly consist of beta strands. The red and yellow-green domains are the other chain, located in a different direction.

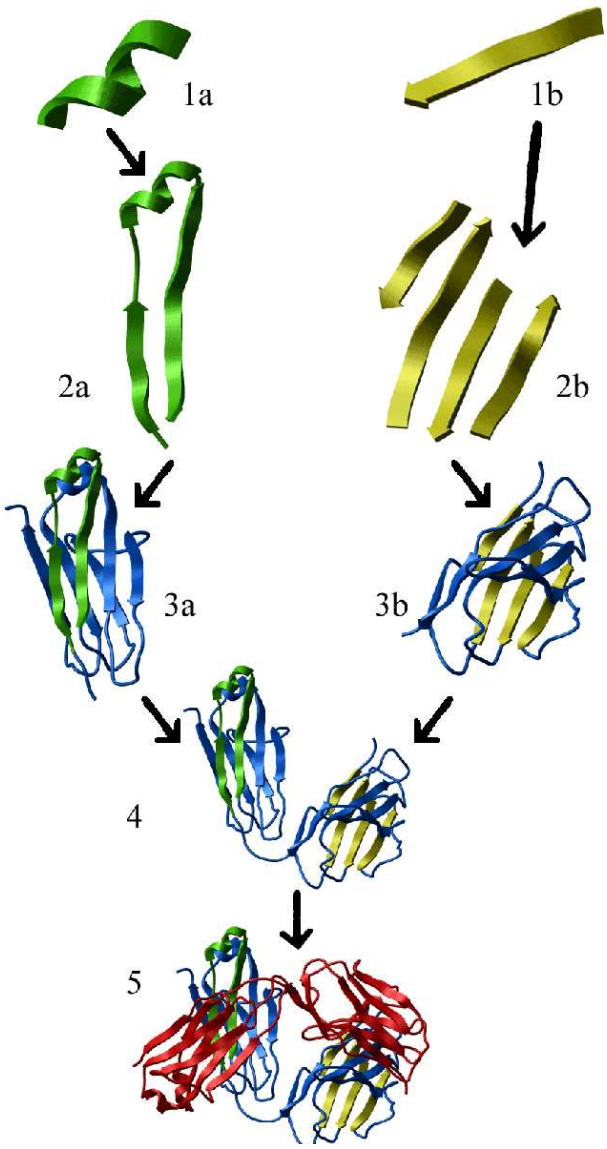


Figure 6: The different levels of protein folding. At the top of the figure, two secondary structures are shown, one alpha helix (1a) and one beta strand (1b). In (2a), the helix is packed with two beta strands to form a beta-alpha-beta motif, that in turn join more strands to form a complete protein domain (3a). The strand in (1b) is packed with more strands, and together they form a beta sheet (2b). In (3b) the sheet together with another sheet form a second domain. The two domains together form the complete folded protein (4), that interact with another identical protein chain to form a dimer (5).

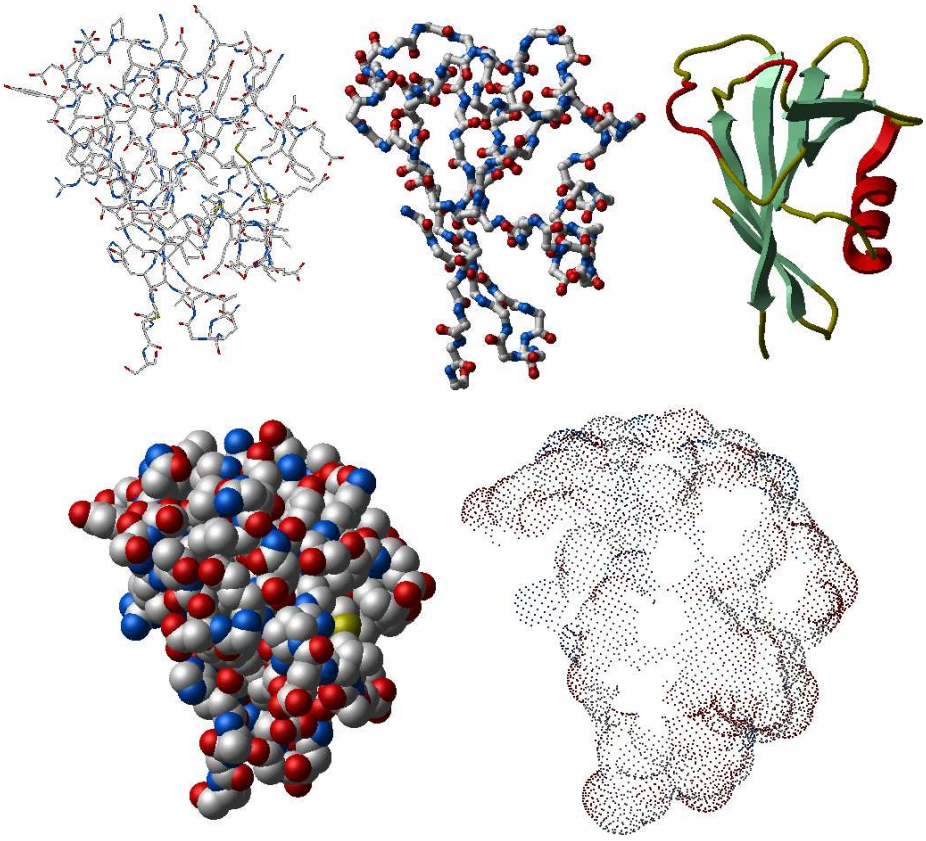


Figure 7: Five different representations of the same protein chain (anti-platelet protein from leech). At the top left the molecule is shown as lines between atoms, represented by the colour of the line. At the middle of the top row, only the backbone of the molecule is shown, now with sticks instead of lines and the atoms represented by balls. At the top right, the backbone of the protein is represented with ribbons, where helices are shown as spirals and beta strands as arrows. The bottom left of the figure shows the protein using a space fill representation, that is each atom in the molecule is represented by a sphere, where the radius corresponds to the Van der Waals distance (the closest any other atom can get without contact). The bottm right shows the area of the protein that is accessible to water molecules, and is perhaps the most true picture of the protein from any other molecules point of view.

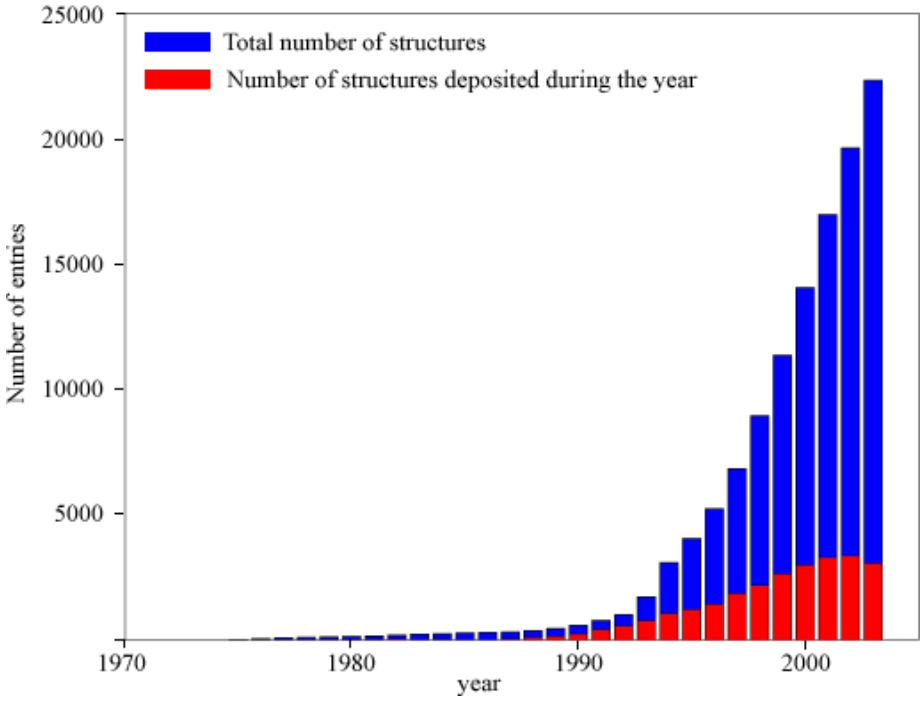


Figure 8: The number of structures deposited in the PDB. The data for 2003 is collected up to 8 September 2003.

September 2003, the structure of 22611 proteins are known and deposited in the Protein Data Bank (PDB, see Section 6.2), and the number is increasing exponentially (see Figure 8).

Currently, there is no way to determine the structure from the sequence only. The most common way to find the structure of a new protein is to compare it to proteins with known structures and predict a conformation based on sequence similarity. The comparison of protein sequences is treated in Section 3. From a sequence comparison of two proteins, similarities can be found between proteins from different organisms or between two proteins in the same organism. These similarities indicate a common evolutionary origin, that the proteins are homologous. Given sequence homology, it is possible to deduce similarity in structure and perhaps even in function. Homology between two proteins from the same organism means that a gene has been duplicated, and that during time, differences have been introduced by mutations and reorganisations. Today, the proteins are tuned for best performance of two different, but probably similar, functions. This is the main strategy for the evolution of new genes and more

complex organisms.

However, not all similarities are indicators for common ancestry. Some similarity may also be introduced by convergent evolution, to create analogous proteins. Analogy appears when two proteins performing the same task in different organisms, have evolved similar properties simply because those properties make the proteins more suitable for the task, without having a common ancestor.

There exist extensive resources for retrieval and comparison of proteins on the Internet. For example there are databases containing protein and DNA sequences, including the complete genomes of several organisms. The Protein Data Bank (PDB, Section 6.2) contains all currently known protein structures. Protein structures are also classified in a number of ways, see Section 6.

To easily access the data, there exist two major "front ends" to the most common databases. EMBL-EBI (European Bioinformatics Institute) offers the SRS (sequence retrieval system), which provides access to data stored in publicly available databases. SRS makes it easy to browse very diverse data, such as literature or biological sequences. The Biology Workbench at San Diego Supercomputer Center (SDSC) offers a similar environment for browsing databases. To the Biology Workbench a number of analysis and modeling tools are connected, eliminating problems with different file formats.

Some useful links are listed in Appendix A.

3 Sequence alignment methods

3.1 Substitution matrices and more

To compare two sequences and find similarities, one of the most direct methods is to construct a dot plot. A dot plot is a plot with the two sequences placed along each axis. A dot is plotted at each positions where the two sequences are identical, that is at the positions where the symbol displayed in the column is identical to the symbol in the row. An example of a dot plot is displayed in Figure 9, where the sequence of human calmodulin is plotted against itself. The dots at the diagonal show the trivial identity of every element in the sequence to itself. Diagonals of dots off the main diagonal show regions where the sequences are identical, in this case of a sequence plotted to itself, stretches of internal repeats.

A more informative way to compare sequences is to align them, that is to position them on top of each other, such that each position in the upper sequence as much as possible matches the symbols in the lower sequence. To make the fit as good as possible, gaps can be inserted in one of the sequences, to enable more positions to match. From an alignment one can gain information on how many and which residues are common between the two sequences. In Figure 10 an example of a pairwise alignment is shown. This information can also be

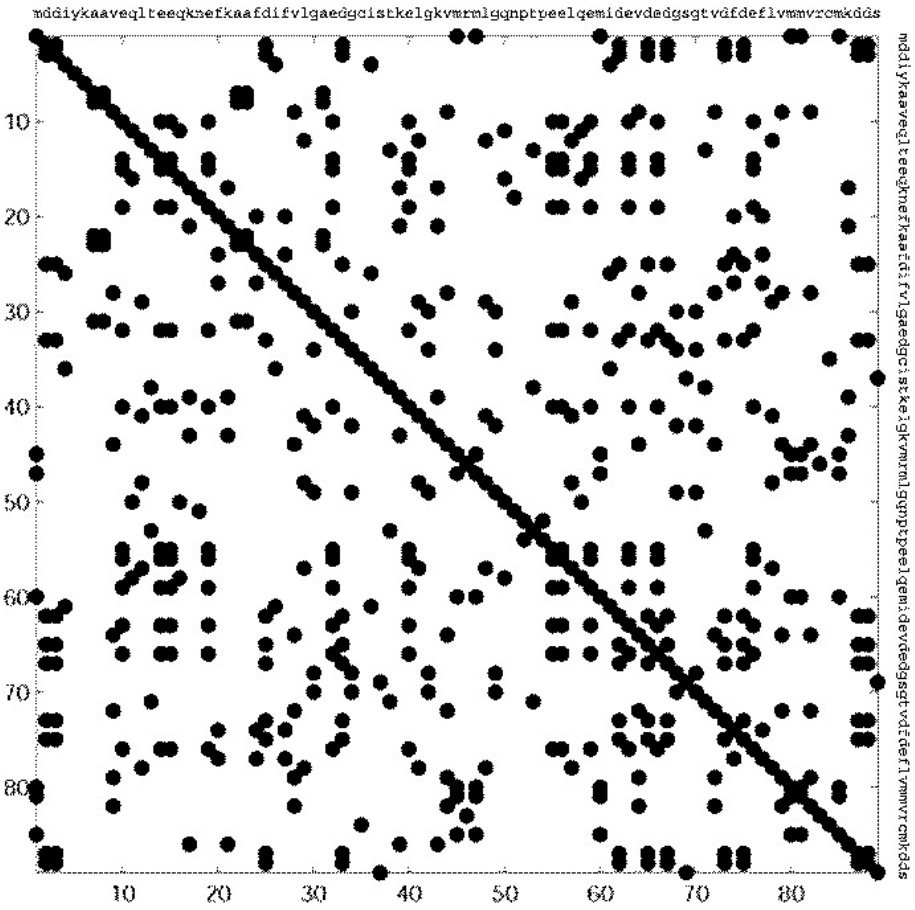


Figure 9: A dot plot of human calmodulin compared to itself. Each dot represents two identical residues. Diagonals of dots indicate stretches of identical residues, in this case of a single sequence compared to itself, diagonals off the main diagonal represent internal repeats. The dots along the main diagonal indicate the fact that the sequence is identical to itself. Along the left and bottom axis of the plot the sequence numbers are shown, and along the upper and right axis the amino acid sequence is shown. Note that the plot is symmetric.

```
trkb_human      EGNP-PTELTQSQMLHIAQQIAAGMVYLASQHFVHRDLATRNCLVGENLLVKIGDFGMSR
q24327          AGSSSPPPLTTSQVLAVAYQIARGMDAIYRARFTHRDLATRNCVISSEFIVKVSYPALCK
consensus       .G...P..LT.SQ.L..A.QIA.GM.......F.HRDLATRNC.......VK........
```

Figure 10: An example of a pairwise alignment with concensus sequence shown in red. trkb_human and q24327 are the names of the sequences. They both are kinases.

captured in a consensus sequence derived from the alignment. In the consensus, only the residues common for the two sequences are kept.

3.2 Dynamic programming

To compare and align two sequences, the most common method used is dynamic programming. Dynamic programming is a general method that guarantees a mathematically optimal alignment of two linear sequences, given a scoring function or a table of scores for matches and mismatches between all amino acids, and penalties for insertions or deletions. Often there are two kinds of penalties for generating an insertion/deletion: a gap opening and a gap extension penalty. The gap opening penalty is used when opening a new gap in a sequence, while the gap extension penalty is used for extending the gap. The gap extension penalty is usually lower than the gap opening penalty, since it is more biologically reasonable to extend an existing gap than to open a new one.

Dynamic programming was first introduced in molecular biology by Needleman and Wunch [58]. The heuristic measure of homology introduced in that paper has since then been developed into a true measure of the distance between sequences, as illustrated in the Smith-Waterman algorithm[73]. The Needleman-Wunch algorithm is designed for constructing global alignments, where the whole of one sequence is aligned to the whole of another. The Smith-Waterman algorithm, on the other hand, is designed for local alignments, where parts of one sequence is aligned to a subsequence of the other. This makes it possible to find alignments between only parts of the sequences, which is the biologically more common situation. The method has also been optimized for time and memory usage [29]. However, the basic idea of dynamic programming is the same in all cases, why the Smith-Waterman algorithm is described in more detail below to illustrate the method. Dynamic programming is today one of the most common methods to optimally align two sequences, whether it is DNA, protein or an abstract "sequence" of structural features. For multiple sequences the method quickly becomes very demanding on computing power.

3.2.1 The Smith-Waterman algorithm

The Smith-Waterman algorithm [73] is used to find similarities between two long sequences, by locating a pair of segments (one from each sequence) such

that the pair has a higher similarity than any other pair of segments. The similarity is calculated using a similarity measure $s(a, b)$ between elements a and b in sequences $A = a_1 a_2 \dots a_n$ and $B = b_1 b_2 \dots b_m$. Introducing gaps in one of the sequences is penalised with a penalty W_k, dependent on the number of gaps, k. An $n + 1 \times m + 1$ similarity matrix H is constructed to find the most similar pair of segments, where the elements H_{ij} can be seen as the similarity of the two segments ending at positions a_i and b_j, respectively. To start, we set the similarity between all sequence positions in A and an empty position b_0 first in B to 0, representing segments where the beginning of sequence B matches internal positions in sequence A (for example if $B = abc$ is matched to $A = xxabc$). The equivalent holds for B matched to an empty position a_0. That is:

$$H_{k0} = H_{0l} = 0 \,\text{for}\, 0 \leq k \leq n \,\text{and}\, 0 \leq l \leq m. \tag{1}$$

Then the other elements are chosen as the maximum similarity which is given by one of four possible combinations of sequence elements:

1. If a_i is matched to b_j, the similarity is calculated as $H_{ij} = H_{i-1,j-1} + s(a_i, b_j)$.
2. If a_{i-k} is matched to b_j, so that a_i is at the end of a deletion of length k (k gaps are inserted after position b_j, and a_i is matched to gap number k), then the similarity is calculated as $H_{ij} = H_{i-k,j} - W_k$.
3. If a_i is matched to b_{j-l}, so that b_j is at the end of a deletion of length l (l gaps are inserted after position a_i, and b_j is matched to gap number l), then the similarity is calculated as $H_{ij} = H_{i,j-l} - W_l$.
4. If $s(a, b)$ can give negative values, 0 is included to avoid negative similarities. 0 means no similarity.

In summary, the similarity up to sequence elements a_i and b_j is determined as:

$$H_{ij} = \max \left\{ \begin{array}{c} H_{i-1,j-1} + s(a_i, b_j) \\ \max_{k \geq 1}\{H_{i-k,j} - W_k\} \\ \max_{l \geq 1}\{H_{i,j-l} - W_l\} \\ 0 \end{array} \right\} \tag{2}$$

This is illustrated in Figure 11.

The segment giving the highest possible similarity is found by locating the largest element H_{ij}, and then backtracking the calculations to find the other matrix elements leading to this value. The backtracking procedure ends when a zero matrix element is found. In this way, the most similar segments from the two sequences and their alignment are found. If one is interested in alternative

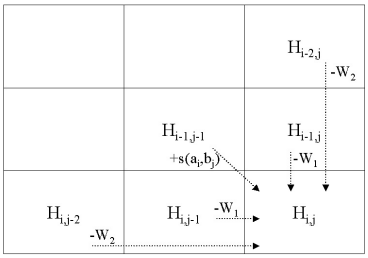


Figure 11: An illustration of how element $H_{i,j}$ of the similarity matrix H is calculated. The arrows symbolise the different alternatives among which the largest result is selected. In this case, k and l are both equal to two. See text for details.

matching segments, the next largest element, not in the same path as the largest element, should be located.

We illustrate the algorithm with the following example. Assume that we have a simple similarity measure

$$s(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and a gap penalty $W_k = 0.1 \cdot k$. Given sequences $A = xxyzxxzy$ and $B = xyzxyzx$, the similarity matrix can be calculated. In this case, $n = 8$ and $m = 7$. First, all elements in the first row and the first column are set to 0 (See Figure 12). Then we continue to calculate element $H_{1,1}$, which represents the similarity of two segments ending at positions $x_{a1}x_{b1}$, where x_{a1} is the x at position 1 in sequence A and x_{b1} is the x at position 1 in sequence B. If we assume that x_{a1} is matched to x_{b1} (alternative 1 above), then the similarity is $H_{0,0} + s(a_1, b_1) = 0 + s(x, x) = 0 + 1 = 1$. If x_{a1} is at the end of a deletion (alternative 2), the similarity is $\max_{k \geq 1}\{H_{1-k,1} - W_k\} = \max_{k \geq 1}\{H_{1-k,1} - 0.1 \cdot k\}$. In this case, k=1 is the only option, since the index $1 - k$ should be equal to or greater than zero (no negative indices!). That is, we get a similarity of $H_{0,1} - 0.1 \cdot 1 = 0 - 0.1 = -0.1$. Correspondingly, if x_{b1} is at the end of a deletion (alternative 3), we get a similarity of -0.1. Alternative 4 above is not relevant in this case, since $s(a, b)$ never yields negative values. To find element $H_{1,1}$, we take the maximum of all these values (cf. Equation (2)):

$$H_{1,1} = \max \left\{ \begin{array}{c} H_{0,0} + s(x_{a1}, x_{b1}) \\ \max_{k \geq 1}\{H_{1-k,1} - W_k\} \\ \max_{l \geq 1}\{H_{1,1-l} - W_l\} \end{array} \right\} = \max\{1, -0.1, -0.1\} = 1.$$

		b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
		–	x	y	z	x	y	z	x
a_0	–	0	0	0	0	0	0	0	0
a_1	x	0							
a_2	x	0							
a_3	y	0							
a_4	z	0							
a_5	x	0							
a_6	x	0							
a_7	z	0							
a_8	y	0							

(a)

		b_0	b_1	b_2	b_3
		–	x	y	z
a_0	–	0	0	0	0
a_1	x	0	1		
a_2	x	0			
a_3	y	0			

		b_0	b_1	b_2	b_3
		–	x	y	z
a_0	–	0	0	0	0
a_1	x	0	1←0.9		
a_2	x	0			
a_3	y	0			

(b)

Figure 12: The construction of the similarity matrix in the example. (a) The result after the first step, when the first row and column are set to 0. (b) Left: The result after calculating element $H_{1,1}$. Right: The result when element $H_{1,2}$ is calculated and added. The arrows show which other element each calculated element is based on.

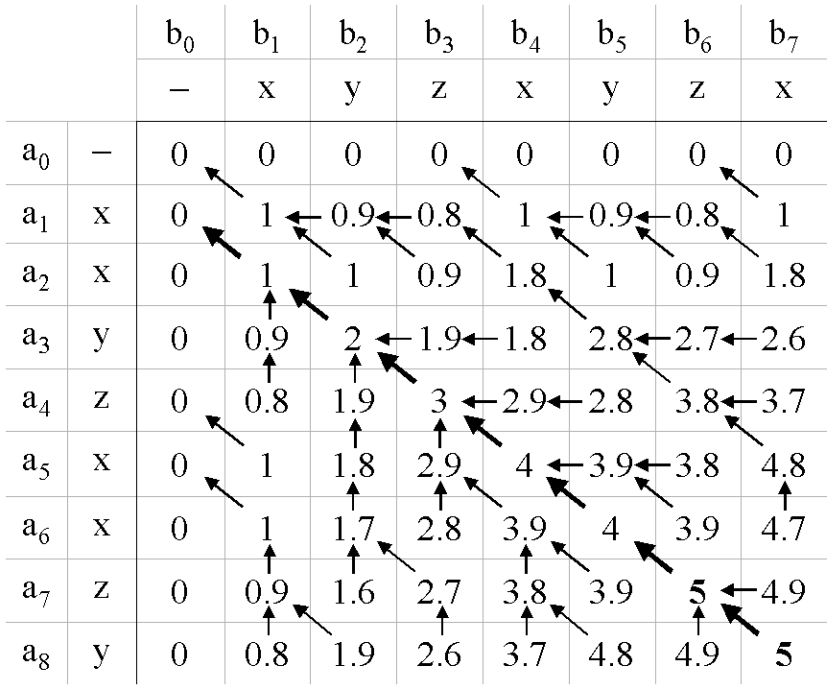


Figure 13: An example of a similarity matrix. Arrows indicate from which element each value is derived, and are used to backtrack the calculations to get the matching sequence segments. Bold arrows represent the optimal path through the matrix, i.e. the two segments having the highest similarity.

That is $H_{1,1} = 1$, a value derived from element $H_{0,0}$ (see Figure 12b, left).

If we move on to element $H_{1,2}$, alternative 1 gives the similarity $H_{0,1} + s(x_{a1}, y_{b2}) = 0+0 = 0$ and alternative 2 gives the similarity -0.1 as derived above for $H_{1,1}$. Alternative 3 can give two values for the similarity, one for $l = 1$ and one for $l = 2$, of which we want to choose the largest: $\max\{H_{1,2-1} - W_1, H_{1,2-2} - W_2\} = \max\{H_{1,1} - 0.1 \cdot 1, H_{1,0} - 0.1 \cdot 2\} = \max\{1 - 0.1, 0 - 0.2\} = 0.9$, derived from element $H_{1,1}$. Element $H_{1,2}$ is the maximum of all three alternatives:

$$H_{1,2} = \max \left\{ \begin{array}{c} H_{0,1} + s(x_{a1}, y_{b2}) \\ \max_{k \geq 1}\{H_{1-k,2} - W_k\} \\ \max_{l \geq 1}\{H_{1,2-l} - W_l\} \end{array} \right\} = \max\{0, -0.1, 0.9\} = 0.9,$$

derived from element $H_{1,1}$ (see Figure 12b, right).

In this way, all the elements in the matrix can be calculated (see Figure 13). In Figure 13, arrows indicate which previous elements the values are based on. After the matrix H is filled, the largest element is located, representing

the pair of fragments that have the largest similarity. In this case, elements $H_{7,6} = H_{8,7} = 5$ is the largest value. By following the arrows we can backtrack the calculations from $H_{8,7}$ to $H_{7,6}$ to $H_{6,5}$ ... , all the way to element $H_{1,0}$ which has the value 0. This yields the aligned segments

$$\begin{array}{cccccccc} x & x & y & z & x & x & z & y \\ - & x & y & z & x & y & z & x \end{array}$$

containing no gaps (except for the initial one) and two mismatches (at positions six and eight in the alignment).

3.3 Scoring matrices

The similarity measure used in dynamic programming is often displayed as scoring matrices, where the amino acids are listed along the margins of the matrix and each element contains the score for substituting a particular amino acid for another. The simplest matrix is the unit matrix, with ones along the diagonal and zeroes otherwise. This is the matrix representation of the similarity score in Equation (3).

A more commonly used series of scoring matrices are the PAM (percent accepted mutation) matrices[19]. These are constructed by first counting the number of accepted point mutations, i.e. the number of observed mutations, in a collection of phylogenetic trees. To get a sharper picture of the actual mutations, the sequences in the trees are compared with their inferred ancestors instead of each other. These accepted point mutations are combined with the observed mutability of each amino acid (the probability that a given amino acid will change during a small evolutionary interval) to construct a mutation probability matrix of one PAM. Matrices for a greater evolutionary interval N can be generated by multiplying the PAM-1 matrix with itself N times. The PAM-250 matrix, which is the most commonly used for distantly related proteins, is generated in this way. This matrix corresponds to the expected observed mutations in sequences that are 80% different.

Another common series of matrices is the BLOSUM (blocks substitution matrix) series[37]. The BLOSUM matrices are constructed from ungapped blocks, that is ungapped multiple alignments of short stretches of related proteins. The substitutions observed in each column of the alignments are recorded, resulting in a frequency table of the number of times each of the 210 $(20+19+...+1)$ possible amino acid pairs occur. The frequency table is then used to calculate the odds ratio between the observed number of times a pair occur and the frequencies expected by chance. The resulting matrix is the scoring matrix. A number of matrices are constructed from blocks with different levels of sequence identities between the aligned sequences. For example, BLOSUM80 is constructed from blocks where the aligned sequence segments are at least 80% identical to some other segment in the alignment.

3.4 BLAST and other pairwise methods

BLAST (Basic Local Alignment Search Tool) [3] is the most commonly used method to find similarities between a query sequence and sequences collected in a database. The sequences may be protein or DNA, in any combination. BLAST searches for pairs of segments, one from the query and one from the database, with the best possible similarity score. The score is calculated using dynamic programming and some substitution matrix, for example one of the PAM matrices (see Section 3.1). To decrease the search time, the strategy of BLAST is to locate short segment pairs with a fixed length, called words, that have a similarity score above some threshold T. Any such word pair is then extended in each direction to determine if it is part of a segment pair with a significant similarity score.

To report the significance of the final result, BLAST uses expectation values (E-values) instead of raw similarity scores. The E-value describes the number of hits one can expect to have a given score just by chance, when searching a data base of a certain size. An E-value of 1 for a hit, would mean that one can expect to find one match in the given data base having the same score just by chance. The lower E-value, that is the closer the E-value is to 0, the less likely it is to find a match just by chance, and the more significant the hit is. Since the E-value is dependent on the size of the data base and the length of the query sequence, E-values in general cannot be compared between different programs or between searches in different data bases.

FASTA [63] is a heuristic method that finds local alignments. In contrast to BLAST, it uses a substitution matrix only for the extension step, when matched sequence fragments are extended.

SSearch (Sequence Similarity Search [62]) does a rigorous Smith-Waterman search for similarity between a query sequence and a group of sequences, which makes it a very sensitive, but also very slow, method.

3.5 Multiple sequence alignments

A common tool when comparing biological sequences is to construct alignments, in particular multiple sequence alignments. A multiple sequence alignment tries to align more than two sequences, so that the symbols in each column are as similar as possible. An example of a multiple sequence alignment is shown in Figure 14. Here, four sequences belonging to the DEATH domain family are aligned. The first column contains the names of the sequences, then comes the actual alignment. Gaps are symbolised by '-' in the alignment.

A consensus sequence, capturing the "essence" of the multiple alignment, can be constructed by taking the most common letter in each column. The last row in Figure 14 contains the consensus sequence for the alignment, calculated using the assumption that at least 60% of all letters in a column must be conserved to generate the consensus. The consensus in Figure 14 includes conserved

```
d1d2zb_          --------LSSKYSRNTELRRVEDNDIYRLAKILDENSCWRKLMSII--PKGMDVQACSG
d1d2za_          LDNTMAIRLLPLPVRAQLCAHLDALDVWQ-----QLATAVKLYPDQVEQISSQK------
d1cy5a_          --MDAKARNCLLQHREALEKDIKTSYIMD-----HMISDGFLTISEEEKVRNE-------
d3ygsp_          -SMDEADRRLLRRCRLRLVEELQVDQLWD-----VLLSRELFRPHMIEDIQRAG------
consensus/60%    ...s.t.Rhh.h.sRtpLhcclcssplac.....phhSshhhh.s.lEplpstt......
```

Figure 14: An example of a multiple sequence alignment of four sequences belonging to the DEATH domain family. The last row is the consensus sequence with a 60% cutoff. See text for details.

properties as well as conserved residues. Conserved residues are indicated by red upper-case symbols. The blue lower-case symbols represent properties common to the residues in the column, such as size (s means 'small' for example), hydrophobicity (p means 'polar') or charge.

From a multiple sequence alignment, it is possible to find which residues and which parts of the sequences that are conserved in that group of sequences. If residues or sequence fragments are conserved, they are most likely important for that family of proteins, either for the fold of the proteins or for their functional role. A multiple sequence alignment is therefore an important tool to characterize protein families.

3.6 Automatic multiple sequence alignment

During the years several methods have been developed for automatic multiple sequence alignment. Most use dynamic programming in one form or the other. Below, some common methods are described in more detail.

3.6.1 MSA

MSA (Multiple Sequence Alignment, [34]) uses dynamic programming in several dimensions to construct an optimal multiple sequence alignment. Each sequence to align makes up a dimension in the dynamic programming matrix, and the optimal path (corresponding to the optimal alignment) through this multidimensional matrix is computed using a variant of Dijkstra's algorithm. The optimal path in this case, is the path that minimises the cost of the multiple alignment, which is the same as minimising the sum of the costs for the pairwise alignments induced by the multiple alignment. The procedure to minimise a cost is completely analogous to maximising a similarity as described in Section 3.2.1. To save memory and time, only alignments with a cost below some threshold are searched. Also, only alignments where the induced pairwise alignments each get a (pairwise) cost below some threshold are considered. Despite these considerations, the space and time requirements of MSA are very demanding. The search space and memory requirements get multiplied by the length of every additional sequence to align, meaning that only a moderate

number of sequences can be aligned simultaneously. A maximum of eight to ten sequences, with sequence lengths up to 1000 residues, seems to be the limit.

3.6.2 ClustalW

ClustalW has become one of the most commonly used methods for multiple sequence alignment.

In ClustalW [79], first all pairs of sequences are aligned separately to calculate pairwise distances, either by a fast approximate method or by full dynamic programming. The distances calculated from the alignments are used to construct a guide tree, using the Neighbour-Joining method [67]. This tree is used in the actual alignment process to guide the addition of sequences to the multiple alignment. The tree is followed from the tips of the branches to the root, starting with the sequences that are most similar and hence closest on the tree. At each branching point, a pairwise alignment is produced by a full dynamic programming algorithm, either between two sequences, a sequence and an alignment from a previous step, or between two alignments. In the case of aligning an alignment, the score at each position is calculated as an average of all possible combinations. For example, when aligning an alignment with three sequences with one with two, the score at each position is the average of $3 \cdot 2 = 6$ comparisons. The procedure of progressively adding sequences to the alignment, based on how they are placed in an initial phylogenetic tree of all the sequences, is sometimes called the "progressive approach".

The sequences are weighted such that very similar sequences get lower weights, while sequences without any close relatives get higher weights. This is to better collect the information in the multiple alignment.

The specific weight matrix to use to determine the similarity between residues in the alignment is chosen depending on the sequence similarity between the two sequences to align in each step (see Section 3.1). This means that different weight matrices can be used at different stages of the alignment procedure.

The gap opening and gap extension penalties are set to some initial values that are modified depending on the current sequence alignment. The gap opening penalty is scaled depending on the weight matrix used and the similarity of the sequences to align, and is increased with the sequence length of the shortest sequence. The gap extension penalty is increased as a function of the difference in sequence lengths between the two sequences to align. The penalties are also modified in a position specific way. If a gap is present at a certain position, the penalties are decreased at that position to promote new gaps at the same place. The penalties are increased at positions near already existing gaps, since it is unlikely to have two gaps very close to each other. In hydrophilic stretches, which most likely are loop regions, the penalties are also lowered. If no gaps occur at or near to a certain position, and it is not in a hydrophobic stretch, the gap opening penalty is multiplied with a residue-specific gap propensity. This

propensity is previously determined by counting the frequency of each residue on either side of a gap, in alignments of proteins with known structure.

One drawback with the progressive approach is that errors introduced in the beginning cannot be corrected later, as more information becomes available. The method is heuristic, and there is no guarantee that the optimal solution will be found [59][24].

3.6.3 T-Coffee

T-Coffee (Tree-based Consistency Objective Function For alignmEnt Evaluation) [59] is a relatively new method for multiple sequence alignment that uses a progressive approach, but avoids getting stuck in local minima by including information from all pairwise alignments in each step of the procedure. The two main features of the method is the ability to combine many different sources of data, presented in the form of pair-wise alignments, and the optimization method that finds the multiple sequence alignment that best fits all the data.

First, a primary library of pair-wise sequence alignments is produced. There is no need for the alignments to be consistent, so it is possible to include two or more different alignments of the same two sequences, possibly generated by different methods. In the default implementation of T-Coffee, pairwise alignments are generated in two ways. A library of global alignments, one for each pair of sequences, is produced using ClustalW (Section 3.6.2,[79]). A collection of local alignments is produced by Lalign [42] to create a local library. The ten non-overlapping alignments that score highest for each pair of sequences are used. Lalign is a variant of the Smith-Waterman algorithm (Section 3.2.1, [73]), and comes from the FASTA package [63]. To give higher priority to more reliable alignments, a weight equal to the percent sequence identity within the alignment is associated to each alignment. In the libraries, the alignments are represented as lists of pairs of aligned residues. The local Lalign library and the global ClustalW library are combined by creating a new entry for each pair of aligned residues. If a pair occurs in both libraries, the entries are merged to a single one, with a weight equal to the sum of the two weights.

To further make use of the information in the library, the consistency of each pair of residues is examined with respect to all other alignments, and a weight reflecting this consistency is assigned to the pair. In this way, some of the information contained in the whole library is reflected in the individual weights for each pair, and the alignment procedure is guided towards consistency with all alignments in the library. This heuristic makes it less likely to get stuck in a local minimum during the progressive alignment procedure. To calculate the new "consistent" weights, each aligned pair of residues is checked with all other alignments including one of these residues. For example, if residue a in sequence A is aligned to residue c in sequence C, and residue b in sequence B also is aligned to residue c, then the score for aligned residues a and b is

increased.

The calculated scores can then be used in a progressive alignment approach. First, pair-wise alignments are constructed to calculate the distances between all the sequences, which in turn are used to construct a phylogenetic guide tree. In the alignment process, the two sequences closest on the tree are aligned first. The sequences are aligned using dynamic programming (Section 3.2.1) and the weights in the library. The pair of sequences are then fixed with respect to each-other according to the alignment. Then the second closest sequences are aligned, or a sequence is added to the previous alignment, all depending on the guide tree. The process continues until all sequences are aligned. To align a group of sequences aligned in a previous step to another sequence or group, the average scores in each column are used.

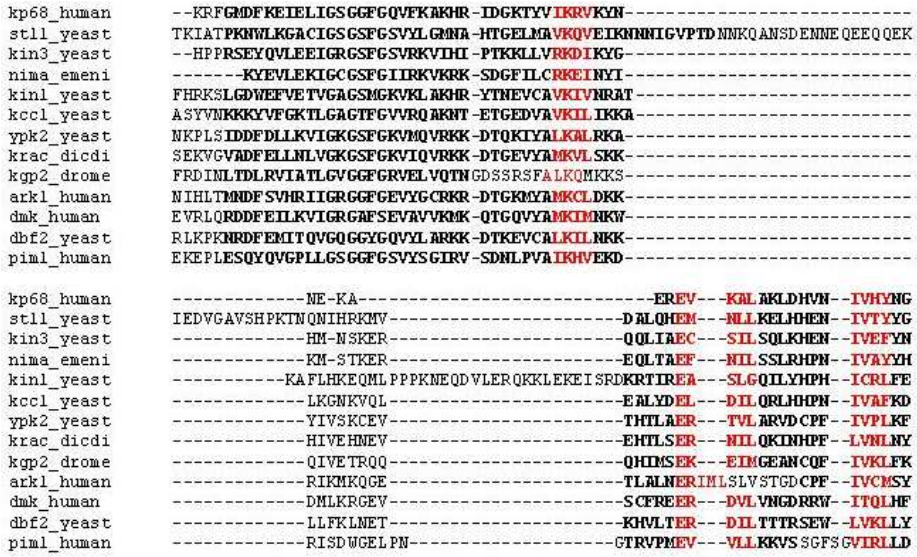
3.6.4 An example of a multiple sequence alignment

In Figure 15, two alignments of the same sequences, but constructed using two different methods, are shown. In Figure 15a, the alignment is produced by T-Coffee, and in Figure 15b it is produced by ClustalW. The sequences are chosen from a reference alignment in BaliBASE (Benchmark Alignment dataBASE, [80]), which is a collection of reliable multiple sequence alignments. BaliBASE is constructed to be used for evaluation and comparison of multiple sequence alignment methods. The alignments in BaliBASE are refined manually to ensure that conserved residues as well as secondary structure elements are aligned. The parts of the alignments in Figure 15 that are shown in bold are aligned identical to the reference alignment in BaliBASE. Sections marked in red are indicated as core blocks in BaliBASE. The alignerd sequences are a collection of kinases.

The alignment produced by T-Coffee (Figure 15a) follows the BaliBASE alignment quite well, and most of the core elements are correctly aligned. Only the sequences of kgp2_drome and that of ark1_human are slightly misaligned in one core element each. The alignment from ClustalW (Figure 15b) also follows the BaliBASE alignment quite well for most of the sequences. But two of them, kp68_human and st11_yeast, are completely misaligned with respect to the reference alignment. These kind of mistakes might be caused by early misalignments, that cannot be updated as more information becomes available from the rest of the sequences.

3.7 Profiles

A profile [32] represents a protein family as a position-dependent scoring matrix. The profile is most often constructed from a multiple sequence alignment of the family members, resulting in a matrix M with 21 columns and L rows, where L is the length of the alignment. Each row represents a position in the alignment, and the corresponding columns contain the scores for that position. The first 20 columns store the scores for each of the 20 amino acids. These scores are



(a)

(b)

Figure 15: An example of two multiple sequence alignments of the same sequences, produced using two different methods. The alignment in a) is made using T-Coffee, while that in b) is the result from ClustalW.

calculated as a function of the number of occurrences of each amino acid at each position in the alignment. The score of amino acid a at position p becomes

$$M(p,a) = \sum_{b=1}^{20} W(p,b) \cdot Y(a,b),$$

where *W(p,b)* is a weight based on the number of occurrences of amino acid b at position p and *Y(a,b)* is a substitution score from some matrix (for example the Dayhoff matrix containing mutational distances, see Section 3.1). The last column contains a penalty for insertions and deletions at that position in the alignment. This makes it possible to punish insertions and deletions more inside regions of secondary structure elements than between them, where gaps occur more frequently.

A profile can be searched against a database using dynamic programming. Instead of scoring similarities between a query sequence and a database sequence using an ordinary scoring matrix, the position specific matrix is used. The similarity score for a residue in the database sequence compared to a position in the profile simply becomes the score in the column corresponding to the residue, at the row representing that position in the profile.

3.7.1 PSI-BLAST

PSI-BLAST (Position-Specific Iterated BLAST) [4] is an extension of BLAST (Section 3.4) that has proven to be sensitive to weak sequence similarities [50]. It uses a position-specific score matrix similar to the profiles described above, but without the column for gap penalties. In PSI-BLAST, a position specific score matrix is automatically constructed from the alignments resulting from a BLAST run. The BLAST search is repeated using the matrix instead of the query sequence, and the procedure is iterated using the new results acquired in each run.

4 Hidden Markov Models

The methods and techniques described in the previous section can be seen as different representations of sequence alignments and, in the end, protein families. Sometimes, however, the sensitivity of these approaches are not high enough, or one likes a more detailed "description" of a certain protein family. In these cases, one solution could be to try to model the family, and use this model to get deeper insights into the characteristics of the family and/or to find more members of the same family. In this section, hidden Markov models of protein families are described. For a more complete description of hidden Markov models and their use in molecular biology, see for example [7] or [21].

In a Markov model, a probability is assigned to symbols in a sequence, based on which symbols are seen in the preceding positions in the sequence. A sequence in this case can be any sequence of symbols or events. The order of the Markov model is the number of preceding symbols the probabilities are based on. A simple first order Markov model of a protein sequence would be a set of arrays a_k, one for each amino acid, with the probabilities $P(ik)$ of seeing amino acid k after amino acid i. The probability that an observed sequence belongs to the model would then be the product of the probabilities for each amino acid in the sequence, with some special treatment of the first amino acid, since that one is not preceded by any other. These kind of models work well in some occasions, but they do not give much information about the sequences they model.

A more robust approach is to assign a probability to each residue in each position. For example, at position 3 in the alignment in Figure 19, the probability of having an N would be 33%, and that of an M would be 67%. All other residues would have a probability 0 in this position. In this way, something similar to a profile of the aligned sequences can be constructed. A slightly more complicated, but more statistically correct, way to model the group of sequences is to use hidden Markov models (HMMs). In short, a profile HMM is a statistical model of a multiple sequence alignment, where probabilities are assigned to each position in the alignment, and to the transitions between positions. The analogy to multiple sequence alignments makes it possible to draw conclusions about the group of sequences that are modeled, making hidden Markov models more appealing than the simple Markov model described above. The HMM can, as can multiple alignments and profiles, be used to locate structurally or functionally important residues, since they are conserved in the sequences. The HMMs are also useful for finding other sequences, similar to the ones modeled.

Some of the advantages with HMMs, compared to for example simple profiles, are position specific scores for amino acids and for insertions/deletions. With many other methods, a single gap penalty is chosen regardless of where in the sequence a gap is inserted. This does not model true sequences very well, since the probability for insertions or gaps is much higher in loop areas than in an alpha helix, for example. Another advantage is that the HMMs are built on a formal probabilistic basis, and that less skill and manual interventions is required for using HMMs than for profiles. A limitation with profile HMMs is that they do not allow consideration of any higher order correlations, such as interactions between residues in different parts of the chain, or base pairing of RNA-bases in a model of RNA.

A profile HMM consists of a collection of states of three kinds (Figure 16): match states corresponding to the positions in the consensus sequence, insert states that model insertions with respect to the consensus, and delete states representing deletions with respect to the consensus. The match and insert states emit symbols, in this case amino acids, with a probability $e_i(x)$ that symbol x is emitted from state i. The delete states are silent, not emitting

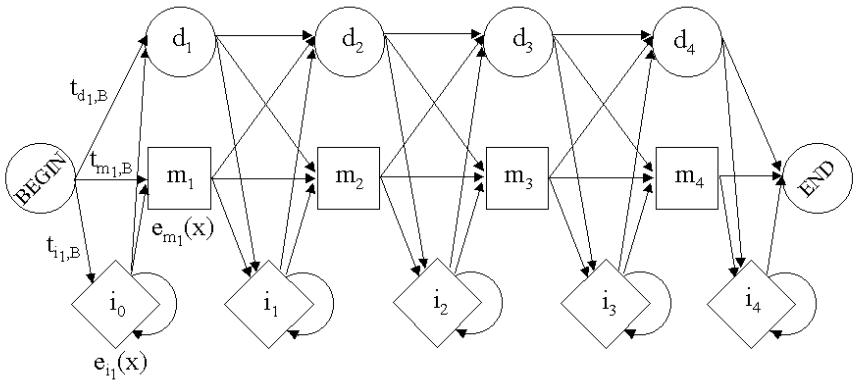


Figure 16: A schematic picture of a HMM. Circles symbolize delete states, squares are match states, and insert states are pictured as diamonds. Arrows between the states indicate the possible transitions. To each transition a transition probability t is associated. Match and insert states are associated with an emission probability e. These are only shown for the first few states in the figure.

any symbols. There are also transitions between the different states (arrows in Figure 16), and a probability t_{ij} to move from state i to state j is associated with the transition. The insert states have self transitions, a transition back to itself, to allow for arbitrary length insertions relative to the consensus sequence. To model the beginning and end of a sequence belonging to the alignment, two special states not emitting any symbols are added first and last in the HMM. An HMM of length N has N match states with corresponding delete states, and $N + 1$ insert states in between the match states.

Using the probabilities, a sequence can be emitted by the HMM. Assume that the sequence "TLVSM" is observed. This sequence can be emitted by the HMM in Figure 16 in a number of ways. One possible state sequence resulting in the observed sequence is $m_1 \rightarrow m_2 \rightarrow m_3 \rightarrow m_4 \rightarrow i_4$. That is to go from the begin state to match state m_1 emitting symbol "T", then move on to state m_2 emitting symbol "L", to state m_3 emitting an "V", to state m_4 emitting an "V" and finally go to state i_4 emitting symbol "M" before going to the end state. Another possibility is the state sequence $d_1 \rightarrow d_2 \rightarrow d_3 \rightarrow d_4 \rightarrow i_3 \rightarrow i_3 \rightarrow i_3 \rightarrow i_3 \rightarrow i_3$, skipping all match states and emitting all symbols from state i_3 by using the transition back to itself. Yet another possibility is $i_0 \rightarrow m_1 \rightarrow d_2 \rightarrow m_3 \rightarrow i_3 \rightarrow m_4$, as is illustrated in Figure 17. All these possible state sequences have different probabilities, but there is no way to tell which state sequence emitted the observed sequence - the state sequence is hidden for us. That is why hidden Markov models are called *hidden*.

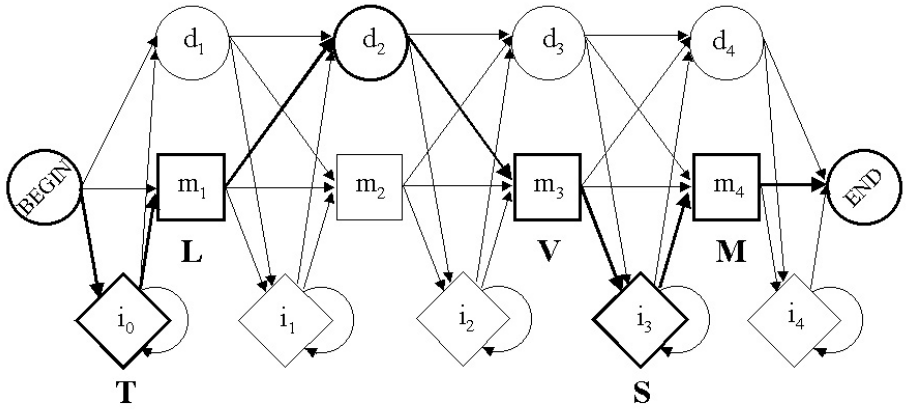


Figure 17: An example of how the sequence "TLVSM" can be generated by the HMM illustrated in Figure 16. The bold arrows and states show the path followed to generate the sequence. The bold letters are the symbols emitted at each match and insert state the path passes through. Together, these symbols form the sequence.

4.1 The Plan7 architecture for HMMs

The HMMs in HMMER2.0 (http://hmmer.wustl.edu/), which is the HMM implementation we used, do not look exactly as described above. Instead, the Plan7 architecture, illustrated in Figure 18, is used. The basics are the same as described earlier, with a number of match states corresponding to consensus positions, associated insert and delete states, and transitions between the states. Unlike the previously described architecture, Plan7 does not have any transitions, in any direction, between insert and delete states. This reduction of transitions from 9 to 7 per node is one of the reasons for the name Plan7. The B and E states are, as above, states used to enter and exit the main model. The special states S, N, J, C and T control which kind of alignment the model is most likely to generate. The S and T states are start and termination states, respectively. None of them emit any symbols. The N state is used to model unaligned N-terminal sequence. Every time it makes a transition to itself, a symbol is emitted. The same holds for the C state, which models C-terminal sequence not aligned to the actual model. These two states make it possible to model local alignments with respect to the sequence (for example a single domain in a multidomain protein) - the parts of the sequence not aligned to the main model are "captured" by the N and C states. The J state is used to model regions in between two matching domains in a sequence. The dotted arrows in the figure illustrate transitions between the B state and match states, and between match states and the E state. These makes it possible to model local

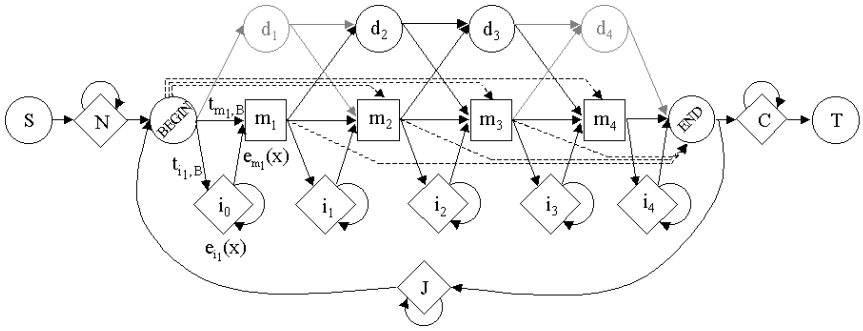


Figure 18: The Plan7 architecture used in HMMER2.0. See text for details.

alignment with respect to the main model. Using one of the dotted transitions, it is possible to skip some of the match states in the beginning and/or end of the model without having to pass through a number of delete states. The alignment mode is determined by the actual values for the transitions between the special states, and is decided when building the model. If one likes more than one type of alignment mode, several HMMs have to be constructed for the same sequences.

There are three interesting problems, answering different questions, to solve when working with profile HMMs: the scoring problem, the alignment problem and the training problem. These are discussed in the following sections.

4.2 The scoring problem

The scoring problem is to find the probability that a certain HMM generated an observed sequence. It tries to answer the question: Is sequence s related to the sequences modeled by the HMM? If the probability is high, then it is also very likely that the observed sequence is related to the group of sequences that are modeled by the HMM.

A sequence $s = x_1 \dots x_L$ with length L, following the state path $q = q_0 \dots q_{N+1}$ through an HMM μ with N states, has the probability

$$P(s \mid (q, \mu)) = \prod_{i=1}^{N+1} t_{q_i, q_{i-1}} \prod_{j=1}^{N} e_j(x_{l(j)}) \tag{4}$$

where $l(j)$ is the index in the sequence for symbol x at state q_j. This is simply the product of the probabilities of going from one state to the other (the transitions t) and the probabilities e of emitting the symbols in the sequence at the given states. To get the probability of the HMM emitting the sequence, we have to choose a suitable path for the sequence. The most common approaches

are to sum over all possible paths or to take the path which has the highest probability. To sum over all possible paths can be expressed as:

$$P(s \mid \mu) = \sum_q P(s \mid (q, \mu)) \tag{5}$$

However, to compute the probabilities for all possible paths often is too computationally exhausting, especially when there are many models to compare the sequence to. The path with the highest probability is called the Viterbi path [7]:

$$P(s \mid \mu) = \max_q P(s \mid (q, \mu)) \tag{6}$$

Strictly, it is not the probability $P(s \mid \mu) = P(s = x \mid x \text{ is generated by model } \mu)$ that is interesting, since it describes the probability of seeing sequence s in a collection of sequences generated by the given model. The question of interest is to find the probability, given a sequence s, that this sequence is generated by the model: $P(x \text{ is generated by model } \mu \mid x = s) = P(\mu \mid s)$. To calculate this probability, Bayes' rule can be used:

$$P(\mu \mid s) = \frac{P(s \mid \mu)P(\mu)}{P(s)} \tag{7}$$

To avoid computing the unknown probabilities $P(\mu)$ and $P(s)$, the question is slightly twisted - instead of calculating the probability that the model generated the sequence, the odds that the sequence was generated by model μ rather than model η is calculated:

$$\frac{P(\mu \mid s)}{P(\eta \mid s)} = \frac{P(s \mid \mu)}{P(s \mid \eta)} \frac{P(\mu)}{P(\eta)} \tag{8}$$

Here, η is generated as a null model that tries to fit all sequences in the universe of sequences (for example a sequence database). The relative probability $\frac{P(\mu)}{P(\eta)}$ of the two models can be estimated as the expected number of hits divided by the number of sequences scored.

4.2.1 Scoring in HMMER2.0

In HMMER2.0 the two models μ and η are considered equiprobable, so the relative probability is set to 1.

The null model is in HMMER2.0 a single "insert" state that can make transitions back to itself and a dummy end state equal to the END state in the actual model. The null model insert state emits symbols according to a distribution equal to the average amino acid composition in SWISS-PROT 34. The score reported by the program is the logarithm of the right hand side in equation (8) - a log-odds score. To correct for bias in sequence composition, HMMER2.0

actually uses a second null model in addition to the simple one described above. This model is useful for HMMs modeling sequences with unusual sequence compositions, preventing unrelated sequences with the same unusual composition from getting unreasonably high scores.

E-values

In addition to the raw scores, an E-value is reported from HMMER2.0. The E-value is an expectation value; it is the expected number of sequences in the database *not* related to the model, that score higher than or equal to the reported score $S = y$. This is the number of hits with a score greater than or equal to y, that one can expect just by chance in a database of size N. See also Section 3.4. By default, the E-value in HMMER is calculated as an analytic upper bound roughly equal to $\epsilon = Nz^{-y}$, where z is the base of the logarithm, in this case 2 [8]. More correct values can be obtained by calibrating the HMM before using it for sequence searches. When calibrating the model, an extreme value distribution $P(S < y) = \exp(-e^{-\lambda(y-\mu)})$ is fitted to the scores generated by the model, and the E-value can then be calculated as $\epsilon = NP(S \geq y)$. The scores to fit the distribution are generated from a Monte Carlo simulation of a sequence database. To find the parameters λ and μ, the log likelihood is maximized. That is, the maximum of the logarithm of the likelihood of getting the simulated scores from a distribution defined by λ and μ is determined:

$$\max_{\lambda,\mu}\{\log P(y_1, \ldots, y_n \mid \lambda, \mu)\} \tag{9}$$

To find the maximum, the zeroes of the partial derivatives with respect to λ and μ are found with the help of a Newton-Raphson algorithm. In practise, only the right tail of the histogram of scores is fitted, because the left tail (the low scoring sequences) does not obey the extreme value distribution. The right tail, around $\epsilon = 1$, empirically fits the distribution quite well, and since this is the region of interest it is recommended to calibrate the models.

According to the HMMER User's Guide (http://hmmer.wustl.edu/), E-values of 0.1 or less in general are significant hits.

4.3 The alignment problem

The alignment problem is to find the state sequence (path) through a given HMM that generated an observed sequence. If one finds that path, one also has the optimal alignment of the sequence to the model and to other sequences generated by / related to the HMM. The solution is to find the path that gives the highest probability for the sequence, as given by equation (4).

```
d1d2zb_          --------LSSKYSRNTELRRVEDNDIYRLAKILDENSCWRKLMSII--PKGMDVQACSG
d1d2za_          LDNTMAIRLLPLPVRAQLCAHLDALDVWQ-----QLATAVKLYPDQVEQISSQK------
d1cy5a_          --MDAKARNCLLQHREALEKDIKTSYIMD-----HMISDGFLTISEEEKVRNE-------
d3ygsp_          -SMDEADRRLLRRCRLRLVEELQVDQLWD-----VLLSRELFRPHMIEDIQRAG------
consensus/60%    ...s.t.Rhh.h.sRtpLhcclcssplac.....phhSshhhh.s.lEplpstt......
```

Figure 19: The multiple sequence alignment shown in Figure 14, displayed again for convenience.

4.4 The training problem

The training problem is to find the parameters of the HMM, i.e. the transition and emission probabilities. If an alignment of the sequences to model is given, it is rather a question of building a HMM, not training. In this case, the consensus positions, where most positions in the column are filled with symbols, are set to match states. All gaps with respect to the consensus are counted as delete states, and all insertions correspond to symbols emitted by insert states. The transition probabilities can be calculated by simply counting the number T_{ij} of observed transitions between one state, i, and another state, j, divided by the total number of transitions from that state:

$$t_{ij} = \frac{T_{ij}}{\sum_{j'} T_{ij'}} \tag{10}$$

As an example we consider the alignment in Figure 19, and the column before the one with an conserved 'E' at the end of the alignment. If this column is a match state, there are three transitions from it to the next match state (the next column with the conserved 'E'). There is also one transition from this state to a delete state, since the first sequence has a gap instead of the conserved 'E'. There are no transitions to insert states at this position. This means that the transition probabilities from this match state become $t_{m,m} = 3/(3+1+0) = 0.75$, $t_{m,d} = 1/(3+1+0) = 0.25$ and $t_{m,i} = 0$.

The emission probabilities are calculated as

$$e_j(x) = \frac{E_j(x)}{\sum_{x'} E_j(x')} \tag{11}$$

where $E_j(x)$ is the number of occurrences of symbol x at position j.

As an example we consider the alignment in Figure 19. At position three in the multiple alignment there are two symbols 'M' and one symbol 'N'. This means that the emission probabilities at this position would be $e(\mathrm{M}) = 2/(1+2) = 0.667$ and $e(\mathrm{N}) = 1/(1+2) = 0.333$. All other emission probabilities would be equal to zero.

For the insert states, background frequencies are often used for the emission probabilities. It is assumed that the symbols in insertions are more or less random, so that the probability of emitting an "A" should be the same as the

frequency of an "A" in the universe of protein sequences. The reason for this assumption is that the number of observations often is too small to determine all the parameters, especially in inserts.

A serious problem with this "raw" calculation of probabilities is the risk of overfitting the model to the data. If the HMM fits the data too well, it will only recognise the sequences used in the training of the model, and no related sequences. In the worst case, a sequence differing from the observed training sequences in just one single position can get a probability of zero, since this very amino acid has not been observed at that position. To handle this problem, pseudocounts can be added to the raw counts. In this way all possible symbols will get a probability greater than zero at all positions, even if they are not observed, making it possible to generate / recognise sequences that differ slightly from the training sequences. Also, in the case of proteins, one knows from alignment of homologous proteins that some substitutions of amino acids are more likely than others. For example, tyrosine and phenylalanine often occur in the same place in an alignment, while they both rarely substitute for proline. Knowing that phenylalanine and tyrosine often substitute for each other, a small count can be added to one of them each time the other is observed, increasing the probability for both amino acids.

A problem with a limited amount of training sequences is the occurrence of biased data. Often there are many similar sequences belonging to the same family, and a few more unique ones. To get a good model of all sequences in the family, not just the majority of very similar ones, the few "unique" ones should get a higher weight. This can be achieved by using tree-based weighting, where sequences with few neighbours on the same branch get higher weights.

If no alignment is given, the model has to be trained from the raw data. First a random alignment is produced, most simply by aligning the first residue of each sequence and then aligning all the others without gaps until the end of the sequences. From this "random" alignment the parameters can be calculated to create an initial model. All sequences are then aligned to the model, resulting in a new alignment which can be used to calculate new parameters. The procedure is then iterated until the alignment and parameters converge. To avoid getting stuck in a local minima, with a suboptimal alignment, a few variations in this procedure are implemented. However, in HMMER2.0 the training of HMMs is not implemented at all, since sequence alignment programs such as ClustalW give much better alignments, resulting in better HMMs, than the HMM training.

In this work, we use alignments based on structural superimposition as the base for building structure anchored HMMs (saHMMs).

4.4.1 Dirichlet mixtures

In the default settings of HMMER2.0, Dirichlet mixtures are used to define the pseudocounts to add at each position, in order to avoid overfitting.

Let $\mathbf{p}$ be a probability vector, containing a possible distribution over the twenty amino acids. That is, element p_i is the probability of amino acid i, $p_i \geq 0$ and $\sum_i p_i = 1$. A Dirichlet density ρ is a statistical density over all probability vectors, meaning that it gives high probability to some distributions (probability vectors) of amino acids, and low to others. For example, a certain Dirichlet density may give high probability to distributions where one single amino acid dominates, that is to conserved distributions. Other densities might give high probability to distributions where amino acids share a common feature, such as hydrophobicity or size, dominate, while even others favour distributions where no particular kind of amino acid dominates.

For a particular $\mathbf{p}$, the value of the density is

$$\rho(\mathbf{p}) = \frac{\prod_{i=1}^{20} p_i^{\alpha_i - 1}}{Z}, \tag{12}$$

where Z is a constant that makes ρ sum to unity, and α_i are the parameters of the density.

A Dirichlet mixture is a mixture of Dirichlet densities. The individual densities ρ_j are called components of the mixture, and each component is associated with a mixture coefficient q_j, that functions as a weight for the component. The mixture coefficients sum to 1. A Dirichlet mixture ρ with l components has the form

$$\rho = q_1\rho_1 + \ldots + q_l\rho_l. \tag{13}$$

At each position of the alignment, the probability of each amino acid is calculated based on the observed number of occurencies in that column. Pseudocounts are added from each component ρ_j of the Dirichlet mixture, each contributing with different number of counts depending on the particular density. The pseudocounts from each component are scaled according to how likely it is that the individual component has produced the observed data.

The mixture used in HMMER2.0 is a nine-component mixture, where the parameters (q_j, α_j) are estimated based on the multiple sequence alignments in the Blocks database[36].

5 Structural superposition of protein structures

In Figure 20, a superposition of four protein structures is shown. Superposition is the structural equivalent of an alignment, to try to fit two or more structures as good as possible "on top" of each other. To construct a structural alignment means to find equivalences between residues in two proteins based on their coordinates in 3D space. As with sequence alignments, the alignment and/or superposition of multiple structures is more complicated but also more informative than pairwise alignments.

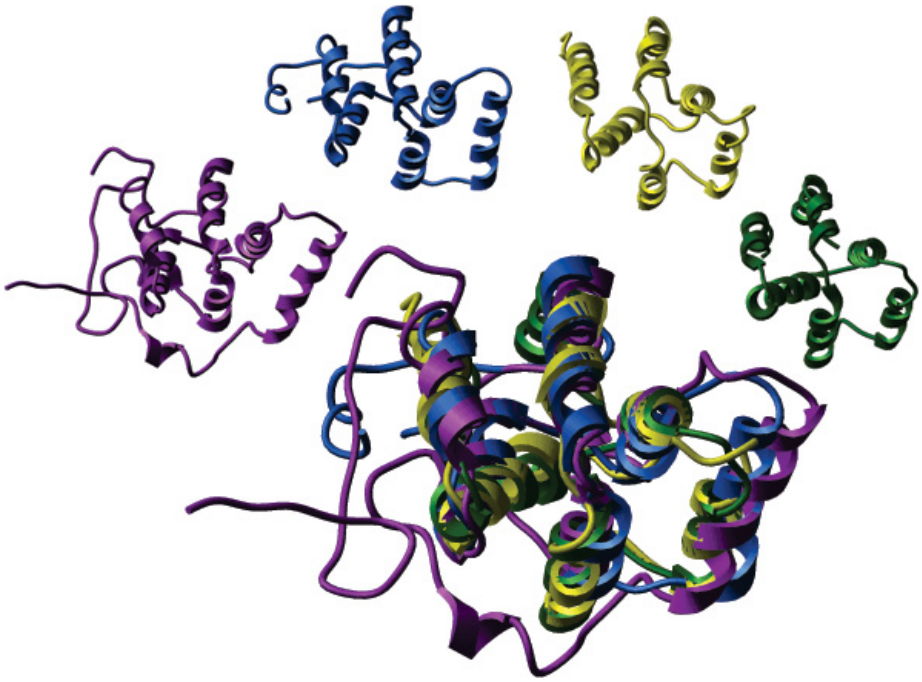


Figure 20: An example of four superimposed structures belonging to the DEATH domain family. Above the superposition smaller images are shown for the individual structures. The four protein domains are: d1d2zb_ (magenta), d1d2za_ (blue), d3ygsp_ (yellow) and d1cy5a_ (green), following the SCOP nomenclature.

For a useful structure superposition method there are a few requirements to be met. Preferably all proteins representing a family should be superimposed simultaneously, or at least the order of adding the structures to the superposition should not affect the final result. Secondly, there must be some way to find the residue equivalences from the structural superposition.

To find a superposition is not a trivial task, since the proteins can differ in size or have slightly different angles between their secondary structure elements, and still have the same overall fold. Even if the similarity is obvious by eye, it is difficult to parameterize it and make a computer do it automatically.

To determine a sequence alignment based on residue equivalences from the superposition of structures is an equally difficult task. How can one determine which of two residues in one protein, on roughly the same distance from a residue in another protein, should be considered equivalent to the other residue? The different sizes of the proteins are also an issue here. If the structures are superimposed as rigid bodies, the centre of the superposition might be quite well defined, while the further away you get from the centre, the further apart are the structures, even though the basic structure is the same. For example, two helices that are situated at the same position with respect to the other elements of the protein in two structures, might be parallel but still some distance apart in the superposition, since one protein could be more loosely connected or have longer loops than the other. This kind of situation makes it very difficult to determine which residue in one protein corresponds to which in the others, especially if one wants to do it automatically. An example of such a situation is the rightmost helix in Figure 20. Here the blue and the magenta helices are clearly equivalent, but since they are tilted in slightly different directions, they only overlap perfectly on a few residue in the middle. The ends of the helices probably are too distant for a computer program to consider them equivalent.

Several methods have been developed to compare protein structures. Comparison is done with two main purposes, to determine whether two structures are related through evolution, and to find exactly how similar two proteins are by pairing residues that are located at similar positions in space. Most methods developed are designed to compare just two proteins at a time, and almost all multiple methods use pairwise alignments as a starting point. Methods for structural alignments are reviewed in ([39], [27], [47])

A very common approach to find matching residues in the proteins to align, is to use dynamic programming ([38], [66],[26],[23],[68],[83],[43],[78]). Dynamic programming finds the optimal solution for the superposition of two structures, but this is given the scoring function that is optimized during the process. This is also the main difference between these methods. Some alternatives are to minimize the difference in distance between aligned residues, to compare intra-protein distances [38], to combine and compare features such as surface accessible area, secondary structure and sequence information [43], to minimize the "soap area" between the backbones of the two structures [23] and to compare

the discrete curvature of the backbones [83]. Genetic algorithms have also been used to find initial equivalences [75].

Lately, several methods have been developed that represent the secondary structure elements as vectors, and find the best matching between those as a first step in the alignment procedure ([53],[75],[70],[2],[72],[84],[54]). The reason for this choice is to reduce search space for the initial alignment, and to ensure biologically relevant alignments since the secondary structures are the building blocks of the structures.

For the actual rotation and translation to superpose the structures, most methods use some kind of iterative least squares procedure, that minimizes the RMSD between equivalenced residues ([53],[66],[83],[43],[75],[70],[2],[72]). Often equivalenced residues are found using nearest neighbours or dynamic programming. These equivalences are then superimposed, and the procedure is iterated until either the RMSD, the equivalenced residues, or both have converged.

Other methods to find the optimal superposition and/or equivalences are Monte Carlo optimization [38], [54] and dynamic programming [78].

There are a few methods with more "unique" approaches, that uses hashing to find common submotives[49], searches all possible combinations of rotations and translations to find the maximum number of matched C_α [20], or assembles structurally similar fragment pairs using combinatorial extension [71].

To construct multiple structural alignments, a common approach is to perform pairwise alignments and include proteins in the alignment directed by a guide tree [66],[68] or pairwise similarity scores [85]. Other methods find the transformations that minimize the RMSD of all proteins simultaneously [52],[83] or align all proteins to the structure that is closest to all others [26].

In the following, four methods are described in more detail. SSAP is a method for pairwise structural alignments, used to construct the protein classification in CATH. DALI is also a pairwise method that uses distance matrices for comparison. The method is used to construct the DaliDD and FSSP. MAPS is a multiple method that attempts to find the rotations and translations that best superposes all proteins at the same time. This program was used in addition to STAMP for difficult cases in our method. Finally, STAMP is the multiple structure alignment program we chose to use in our implementation.

5.1 SSAP

SSAP (Structure and Sequence Alignment Program) [78] compares protein structures using a dynamic programming approach (see Section 3.2), and is used in the construction of CATH (Section 6.5, [60]). The dynamic programming algorithm finds the optimal alignment of two sequences, given a similarity score for matching symbols in one sequence with symbols in the other. In SSAP, this score is based on comparing the distances from the given residues to other residues in the same protein. These distances represent a structural environ-

ment for the residue, and is assumed to be the same at corresponding positions in similar structures.

The distance based similarity score could be calculated in a number of ways. To simply sum the distances to say 5 residues forward and 5 backward in the chain, and compare these sums, will work well as long as no gaps or insertions are involved. However, these kinds of "discontinuities" are quite common. SSAP handles this by applying the dynamic programming twice, first to find the best equivalence score, and then to find the optimal alignment given these scores. To base the equivalence scores solely on the distances between residues gives high scores to all similar distances, even if the distances are measured between residues that are in completely different relative positions. Therefore, the comparison is done using vectors between residues, instead of plain distances. These vectors are defined with respect to a local coordinate system defined for each residue, to make it possible to compare residues in different directions. The coordinate systems are based on the geometry of the bonds to the C_α atom. The score for matching residue i in one protein to residue k in the other becomes:

$$S_{ik} = \max\left\{a/(\mathbf{V}_{ij}^{A} - \mathbf{V}_{kl}^{B})^2 + b\right\} \tag{14}$$

Here, $\mathbf{V}_{ij}^{X}$ is the vector between residues i and j in a protein X, a is a constant limiting the maximum possible score and b is a constant preventing division with very small numbers. $\max\{\ldots\}$ is used to represent the dynamic programming procedure, so that S_{ik} is the maximum score obtained by dynamic programming over all possible j in protein A and all possible l in protein B. Having defined this score between matching symbols/positions, it is straightforward to construct a scoring matrix and find the optimal alignment. Every calculation of equation (14) gives an alignment of matched inter atomic vectors, which represents an alignment of the structures. This information is included in the second run of dynamic programming, in addition to the maximum score S_{ik}. To include the alignment information, the values along the trace-backs of the equivalence matrices are added to the corresponding elements in the scoring matrix.

In an extension of SSAP [77], other information than directional is included in the score, information such as hydrophobicity, phi/psi angles, solvent-accessible surface area, etc. SSAP is also combined with the multiple sequence alignment program MULTAL to enable multiple structure alignments [76]. In the multiple structure alignment, the columns of amino acids are replaced by sets, or bundles, of vectors. These bundles are reduced to average vectors and an error term, measuring the divergence inside the set. To construct a multiple alignment, first the most similar protein pairs are aligned separately, and a consensus structure is derived. Then all similarities are recalculated, in this case between remaining structures and the consensus structures, before again the most similar are aligned. In this way the most similar structures are aligned at each stage, until the alignment is complete. We have not been able to test

this combination of programs.

5.2 DALI

DALI [38] is an algorithm for pairwise structural alignment of proteins. Instead of comparing the actual coordinates of the proteins, the method finds similarities between distance matrices computed from the structures. The distance matrix used is one containing all pairwise distances between C_α atoms in a protein. This matrix contains all information needed to reconstruct the protein structure, except for the chirality[1] of the molecule. In Figure 21, an example of a distance matrix, or actually two, is shown. Due to the symmetry of the distance matrices, data can be shown above the diagonal (or actually the anti-diagonal) for one protein, and below for the other.

In DALI, the distance matrices are first systematically compared to find all matching hexapeptide-hexapeptide contact patterns. If contact pattern ($i_A \dots i_A+5$, $j_A \dots j_A+5$) in protein A is similar to pattern ($i_B \dots i_B+5$, $j_B \dots j_B+5$) in protein B, then the hexapeptide $i_A \dots i_A+5$ is equivalenced to $i_B \dots i_B+5$, and hexapeptide $j_A \dots j_A+5$ is equivalenced to $j_B \dots j_B+5$ in the alignment. The patterns of pairs of hexapeptides in the matrix of protein A are compared to the patterns of hexapeptide pairs in the matrix of protein B. The similarity is calculated as

$$\phi_E(i,j) = \left\{ \begin{array}{cc} \left(\theta_E - \frac{|d_{ij}^A - d_{ij}^B|}{d_{ij}^*}\right) w(d_{ij}^*), & i \neq j \\ \theta_E, & i = j \end{array} \right\} \tag{15}$$

where d_{ij}^* is the average of d_{ij}^A and d_{ij}^B, $\theta_E = 0.2$ is a similarity threshold, and w is a function giving less weight to the common, but not so informative, pairs in the long distance range. The pairs are stored in a list, that then is sorted according to score. The 40000 highest scoring pairs are kept to use in the next step; the actual alignment, which is produced by Monte Carlo optimization. The procedure starts with producing a number of seed alignments. These are constructed from all triplets of non-overlapping hexapeptides in the pair list. For instance, the pairs (a, b)-(a', b'), (a, c)-(a', c') and (b, c)-(b', c') could form the triplet (a, b, c)-(a', b', c'), if a, b, c are segments from sequence A and a', b', c' are segments from sequence B. Each singlet (x, x') in the triplets is used to generate a seed alignment. The seed alignments are extended using overlapping contact pairs - for example, if the alignment contains the residue pair (i_A, i_B), all pairs including this residue pair can be used for extending the alignment.

[1] The chirality of a molecule is its "handedness". Compare to a left and a right hand - they are identical with respect to internal distances between for example fingers, but are each others mirror images.

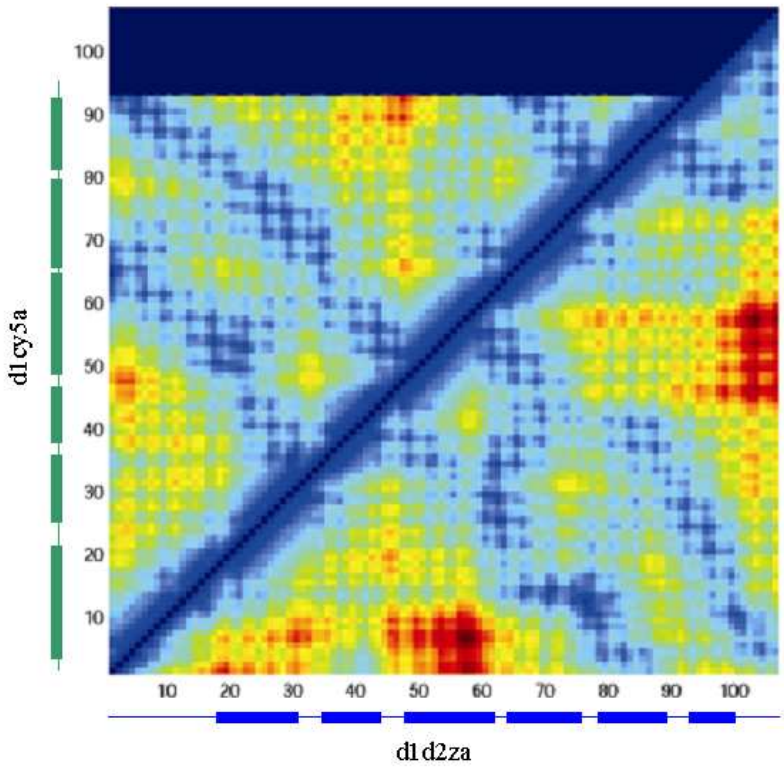


Figure 21: An example of two distance matrices. The range of colours goes from blue through green and yellow to red, where blue indicate very short distances and red very long. The section above the diagonal (actually the anti-diagonal) are the distances between atoms in the protein domain d1cy5a, the section below show distances between atoms in d1d2za. Both protein domains are members of the DEATH domain family. Along the axis the secondary structures are shown schematically. Both domains consist of only alpha helices (shown as rectangles). The blue sections along the diagonal show short range contacts inside the helices. Matches between the two domains along the diagonal correspond to similarity in backbone conformation, that is in secondary structures. Matches off the diagonal reveal similarity in contacts between secondary structure elements, that is in tertiary structure.
The 3D structures of the two proteins can be seen in Figure 20.

The score of the alignment is calculated as the sum of all pairwise similarities:

$$S = \sum_{i=1}^{L} \sum_{j=1}^{L} \phi(i,j) \tag{16}$$

Here, $i = (i_A, i_B)$ and $j = (j_A, j_B)$ are pairs of equivalenced residues, L is the length of the alignment (the number of equivalenced pairs), and ϕ is a similarity measure, for example the one in equation (15). The highest scoring alignments are selected and optimized in parallel. The optimization consists in extending the alignment based on overlapping contact pairs, keeping the new residue pairs according to a probability function $p = \exp(\beta(S' - S))$. S' and S are the old and new scores, respectively, and β is a parameter governing how probable it is to keep "bad" alignments. After a few rounds of extension, the alignments are trimmed by removing negatively scoring matches. The procedure continues with expansion and trimming until the score has not improved for 20 cycles. Finally, the best alignment is refined by optimizing 10 variations of that alignment, each having 30% of the aligned blocks randomly removed.

5.3 MAPS

MAPS (Multiple Alignment of Protein Structures)[52] is based on the (protein TOPological comparison) program TOP[53], which compares two protein structures and is used as the starting point for MAPS. In TOP, a first fit is done based on the secondary structure elements (SSEs), which are represented as vectors from the N-terminal to the C-terminal end of the SSE. Each possible pair of SSEs from one protein is compared to each possible pair from the other by trying to find a rotation and translation that superimposes the endpoints of the two vector pairs. If a similarity is found (angles and distances between superimposed vectors are similar), the superposition is refined to minimize the RMSDs (root mean square deviations) of the angles between all SSEs by a least squares method. The procedure of finding more similarities and refining the superposition is iterated until convergence. Then matching atoms are identified, so that at least three consecutive residues are aligned and the aligned residues are each others nearest neighbours. The distance, as well as the difference in direction of the $C_\alpha - C_\beta$ bond, between two aligned residues should be below a certain value. The transformation is refined using these residue equivalences, and the process of identifying equivalences and refining the superposition is iterated until convergence of equivalenced residues. The whole procedure is repeated for all pairs of SSEs, and the best match is selected.

In MAPS, all proteins are first pairwise aligned as above to give equivalent residues and a rotation and translation to best superimpose each pair of proteins. These values are used as a starting point to minimize the total distance between all proteins:

$$\Delta = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{k=1}^{M_{ij}} \frac{1}{w_{ijk}} \left| \mathbf{x}'_{ik}(\mathbf{r}_i, \mathbf{t}_i) - \mathbf{x}'_{jk}(\mathbf{r}_j, \mathbf{t}_j) \right|^2 \quad (17)$$

where N is the number of proteins to align, M_{ij} is the number of equivalent residues in proteins i and j, w_{ijk} is a weight, and $\mathbf{x}'_{ik}$ and $\mathbf{x}'_{jk}$ are the transformed coordinates for equivalent residues k in protein i and j, respectively. The transformed coordinates $\mathbf{x}'$ are dependent on three angles $\mathbf{r}$ to define the rotation and three parameters $\mathbf{t}$ to define the translation. Δ is minimized using a non-linear least squares method.

Three criteria are used to detect equivalent segments: the length of an aligned segment should be equal to or above a given number (default 3), the difference in direction of the $C_\alpha - C_\beta$ bond between two aligned residues should not be too large, and the distance between aligned residues should be less than a given value (default 3.8Å). This new set of equivalenced atoms is used to again find the rotations and translations that minimizes Δ in equation (17), and the procedure is iterated until convergence of equivalenced residues and RMSD (which is correlated to Δ).

MAPS reports the pairwise TOP alignments as well as the final multiple sequence alignment, and has an option to produce transformed coordinate files for the superimposed proteins. Only segments with equivalenced residues are reported in the sequence alignment.

5.4 STAMP

STAMP (Structural Alignment of Protein Sequences) [66] aligns several sequences based on their structural similarity. A tree based on pairwise comparisons is used to determine the order in which the structures are aligned.

An overview of the procedure STAMP uses is shown in Figure 22. To start, STAMP needs the structures to be reasonably superimposed, a superimposition which is refined in the procedure, and which is used to construct the guide tree. The structural domains are superimposed in the order indicated by the precalculated tree. First, a matrix containing the distances between each residue in one domain to each residue in the other is calculated. The optimal way through the matrix is found, resulting in a list of equivalent residues with corresponding C_α positions. These positions are used to calculate the transformation (translation and rotation) of one structural domain that gives the lowest RMSD (root mean square deviation) towards the other. The domain is transformed, resulting in a new set of coordinates, and the calculations are repeated until convergence. STAMP then moves on to the next pair to be superimposed.

The initial multiple superimposition or multiple sequence alignment needed by STAMP can be produced by (i) constructing a multiple sequence alignment of the domains to be superimposed, (ii) constructing a simple "alignment" where

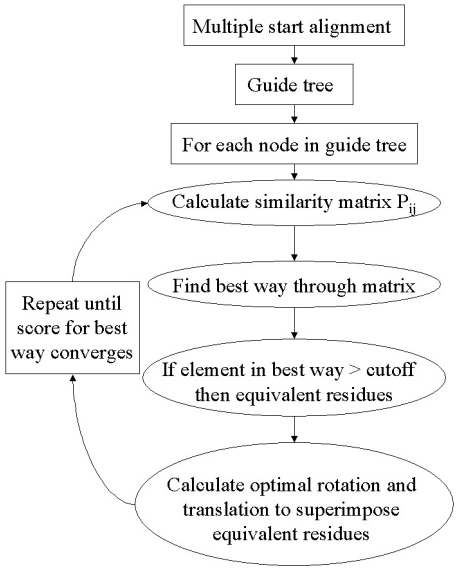


Figure 22: The STAMP procedure for alignment and superposition of protein structures. See text for details.

	A	B	C	D
A	-	1	2	3
B	-	-	3	4
C	-	-	-	1.5
D	-	-	-	-

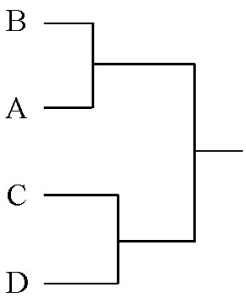


Figure 23: To the left is a table containing the RMSD (in Å) for four hypothetical proteins A, B, C and D. To the right is the guide tree constructed based on these values. Using this tree, proteins A and B would be superimposed first, then proteins C and D, and last the two alignments would be superimposed on each other.

the sequences are stacked on top of each other with no gaps, starting from the N-terminus, or (iii) pairwise superimposition of all structural domains against one of them, after which the superpositions are converted to a multiple sequence alignment. The initial alignment is used to construct a tree that guides the order of superimposition. For each pair of domains to be superimposed, the k positions aligned for this pair in the initial alignment, with no gaps, are compared and fitted (rotated and translated) to minimize the RMSD (root mean square deviation) for the pair. The RMSD is calculated as:

$$RMSD = \sqrt{\sum_{i=1}^{k}(dx_i^2 + dy_i^2 + dz_i^2)/k} \tag{18}$$

where $dx_i = x_{A_i} - x_{B_i}$, $dy_i = y_{A_i} - y_{B_i}$, $dz_i = z_{A_i} - z_{B_i}$ are the distances between x-, y- and z-coordinates, respectively, for equivalent atoms i in molecules A and B. This is a measure of the distance between equivalent atoms in the two molecules, and consequently measures how well the two molecules are superimposed, and how similar they are in structure.

If N proteins are to be aligned, the RMSD values for each of the possible $N(N-1)/2$ pairs are used to construct the guide tree, such that pairs with low RMSD are close to each other in the tree, while pairs with high RMSD are further away from each other. In Figure 23, the construction of a tree from a distance matrix containing RMSD values is illustrated. To construct the tree, each molecule is assigned to its own subset. Then the two subsets that have the lowest RMSD are joined together, and the lengths of the branches are set as the

distance. In Figure 23, molecule A and B have the lowest RMSD, and are joined first. These two subsets are then treated as a single subset when the process is iterated, until all subsets have been joined to a single set, the tree. When two subsets containing more than one molecule are compared to determine the distance, the average RMSD values from all possible pairings of molecules from the two subsets are used.

The structural domains are then superimposed starting from the leaves in the tree, superimposing a pair of domains at each node until reaching the root of the tree. In this way, the most similar domains are compared first, leaving the comparison and alignment of more distantly related domains until later in the procedure. At internal nodes, where more than two domains are to be superimposed, average values are used for domains belonging to the same branch of the tree.

The actual superimposition in each node starts by calculating a distance matrix for the two domains to be superimposed. For each residue i in domain A, the probability P_{ij} of structural equivalence to residue j in domain B is calculated as:

$$P_{ij} = \exp(-\frac{d_{ij}^2}{2E_1^2}) \exp(-\frac{s_{ij}^2}{2E_2^2}) \tag{19}$$

where d_{ij} is the distance between C_α atoms for residues i and j, s_{ij} is a measure of the chain configuration, and $E_1$and E_2 are constants. This results in a $m \times n$ matrix, if A contains m residues and B contains n. If more than two domains are to be superimposed in a node, domains on the same branch are kept fixed to each other, and the average P_{ij} for all possible combinations is computed for each position ij. Say domains A and B superimposed on one branch, are to be compared to domains C and D from the other branch, then all possible combinations are A-C, A-D, B-C, and B-D. If a comparison is made to a gap, a neutral value of 0 is used.

The best way through the matrix, i.e. the path that yields the highest sum of P_{ij} values, is calculated using a modified Smith-Waterman algorithm ([73], Section 3.2.1). The path corresponds to the best possible set of equivalent residues. From this set, the pairs having a P_{ij} larger than a threshold T are used to obtain two sets of equivalenced C_α positions. The sum of P_{ij} values can be seen as the score S of the set of C_α positions.

The two sets of equivalent C_α positions can be seen as two sets A and B of k vectors $\mathbf{a}_i$ and $\mathbf{b}_i$ $(i = 1, \ldots, k)$, if k is the number of equivalent positions. Each vector $\mathbf{a}_i$ and $\mathbf{b}_i$ contains three elements representing the x-, y-, and z-coordinates of the residue at position i. Given these two sets, the problem is to find a rotation matrix **R,** and a translation **t** which when applied to set A yields a transformed set of coordinates $\tilde{\mathbf{a}}_i$ which minimizes the RMSD (equation (18)) to set B. In the case of several domains being compared, the average C_α coordinates for domains belonging to the same branch are used.

The translation $\mathbf{t}$ is calculated as the difference between the centres of masses for the two sets of coordinates. The orthogonal matrix $\mathbf{R}$ is found as the matrix that minimizes the energy function

$$W = \frac{1}{2}\sum_i \left(\mathbf{R}\mathbf{a}_i - \mathbf{b}_i\right)^2 = \frac{1}{2}\sum_i \left(\tilde{\mathbf{a}}_i - \mathbf{b}_i\right)^2. \tag{20}$$

This is actually (a variant of) the Procrustes problem. It can be shown [44][45] that finding the eigenvalues μ_p and eigenvectors $\mathbf{e}_p$ of the matrix $\mathbf{U}^{\mathrm{T}}\mathbf{U}$ solves the problem. The elements of matrix $\mathbf{U} = (u_{kl})$ are calculated as

$$u_{kl} = \sum_i b_{ik}a_{il} \tag{21}$$

and $\mathbf{U}^{\mathrm{T}}$ is the transposed matrix. If vectors $\mathbf{f}_p$ are defined as

$$\mathbf{f}_p = \frac{1}{\sqrt{\mu_p}}\mathbf{U}\mathbf{e}_p \tag{22}$$

the rotation matrix $\mathbf{R}$ can be constructed as

$$r_{kl} = \sum_p f_{pk}e_{pl} \tag{23}$$

The domain is transformed using the calculated $\mathbf{R}$ and $\mathbf{t}$, resulting in a new set of coordinates that can be used to calculate a new distance matrix according to equation (19). The calculations are repeated until convergence, meaning that the score S does not change more than 0.1% compared to the previous iteration. STAMP then moves on to the next node and pair of domains/averaged domains to be superimposed, until reaching the root, meaning that all structural domains are superimposed.

The method is not designed to align different topologies or connectivities (where the secondary structure elements are connected in different ways even though their relative positions in space are the same), but since we use structural and functional families, such cases are not relevant for us. The heuristic tree-based addition of structures to the alignment is not guaranteed to give the optimal solution to the problem, but in most cases the resulting solution will be close enough to the optimal one.

5.5 Comparison between an alignment based on structure and one based on sequence

In Figure 24 two alignments are shown, one based on structural superposition and constructed using STAMP (Figure 24a), and one based on sequence anlysis using T-Coffee (Figure 24b). Residues forming beta strands are marked in red,

while residues forming alpha helical structures are marked in blue. The figure shows that the secondary structure elements are much better aligned in the alignment produced by STAMP than in the sequence-based alignment. In this case, the alignment based on structure is obviously biologically more correct.

6 Databases - protein classifications

A listing of useful biological databases can be found in Baxevanis [11], including a short description of each database. A longer description of each database is provided through the Nucleic Acids Research web site. Some links and short descriptions are provided here in Appendix A. In this section, we describe some of the most common databases that are relevant for this work, with focus on databases containing classifications of proteins. We start by mentioning some important sequence data bases.

6.1 Sequence databases

There are three important databases storing genetic information, that is nucleotide databases containing DNA sequences. GenBank[2] is the NIH (National Institute of Health, USA) genetic sequence database. GenBank is an annotated collection of all publicly available DNA sequences and is maintained at the National Center for Biotechnology Information (NCBI). The EMBL Nucleotide Sequence Database (sometimes called EMBL-Bank)[3] is the main resource of nucleotide sequences in Europe, and is maintained at the European Bioinformatics Institute (EBI), which is a part of the European Molecular Biology Laboratory (EMBL). The third collection of nucleotide sequences can be found in the DNA database of Japan (DDBJ)[4]. The three databases cooperate, and exchange new and updated database records on a daily basis. Each database entry gets a unique accession number, making it possible to refer to a specific gene sequence. The main sources of DNA, and also RNA sequences, are submissions from individual researchers, genome sequencing projects and patent applications.

One of the main sources of protein sequence information is the Swiss-Prot Protein Knowledgebase (Swiss-Prot)[5]. It is a curated protein sequence database, that is aimed to provide a high level of annotation, as little redundancy as possible and a high level of integration with other databases. Swiss-Prot is maintained by the Swiss Institute for Bioinformatics (SIB) together with the EBI. The Swiss-Prot release of October 2003 contains 136356 entries.

[2] http://www.psc.edu/general/software/packages/genbank/genbank.html
[3] http://www.ebi.ac.uk/embl/index.html
[4] http://www.ddbj.nig.ac.jp/
[5] http://www.ebi.ac.uk/swissprot/

```
d1dfca2  -----QVNIYSVTRKRYAHLSARPADEIAVDRDVPWGVDSLITLAF--QDQRYSVQTADH
d1dfca1  EAVQIQFGLIN-CGNKYLTAE-AFGFKVNASASSL-KKKQIWTL-----EAAVCLRSHLG
d1dfcb3  ---CAQVVLQA-ANERNVS------TDLSANQDEE-TDQETFQLEIDRDTKKCAFRTHTG

d1dfca2  RFLR----HDGR-LVARPEPATGYTLE-FRSGKVAF-RDCEGRYLAPSGPSGTLKAGKAT
d1dfca1  RYLAADKDGNVTCEREVPGPDCRFLIVAHDDGRWSLQSEAHRRYFG-GTE-DRLSC-FAQ
d1dfcb3  KYWILTATGGVQSTASSKNASCYFDIE-WRDRRITL-RASNGKFVTSKKN-GQLAA-SVE

d1dfca2  KVGKDELFALEQS---
d1dfca1  TVSPAEKWSVHIAMHP
d1dfcb3  TAGDSELFLMKLIN--
```

(a)

```
d1dfca2      .......... .......... ....QVNIYS VTRKRY.... ..........
d1dfca1      EAVQIQFGLI NCGNKYLTAE AFGFKVNASA SSLKKKQIWT LEAAVCLRSH
d1dfcb3      .......... .......... .......... .....C.... ..........

d1dfca2      ..AHLSARPA DEIAVDRDVP WGVDSLITLA FQDQRYSVQT ADHRFLRHDG
d1dfca1      LGRYLAADKD GNVTCEREVP GPDCRFLIVA HDDGRWSLQS EAHR......
d1dfcb3      ..AQVVLQAA NERNVSTDLS ANQDEETDQE TFQLEIDRDT KKCAFRTHTG

d1dfca2      RLVA...... ......RPEP ATGYTLEFRS GKVAFRDCEG RYLAPSGPSG
d1dfca1      .......... .......... ......RYFG GTEDRLSCFA QTVSPAEKWS
d1dfcb3      KYWILTATGG VQSTASSKNA SCYFDIEWRD RRITLRASNG KFVT.SKKNG

d1dfca2      TLKAGKATKV GKDELFALEQ S.
d1dfca1      VHIAMHP... .......... ..
d1dfcb3      QLAASVET.A GDSELFLMKL IN
```

(b)

Figure 24: An example of an alignment based on a) structural superposition (using STAMP) and b) a sequence alignment method (T-Coffee). Regions marked with red are beta strands, regions marked with blue are of alpha type. The three protein sequences are all domains in fascins; actin-crosslinking proteins.

The TrEMBL database (Translated EMBL)[6] contains the translations of all coding sequences present in the EMBL Nucleotide Sequence Database. That is, the DNA sequences in EMBL-Bank that code for a protein are translated into the corresponding protein sequence. Only sequences which are not yet integrated into Swiss-Prot are stored in TrEMBL. A subset of TrEMBL, called SP-TrEMBL, contains sequences that eventually will be incorporated into Swiss-Prot.

PIR (Protein Information Resource)[7] produces the Protein Sequence Database (PSD), which contains protein sequences that are functionally annotated.

To collect the information in these three databases, the United Protein Databases (UniProt)[8] project was formed in 2002. UniProt aims to create a central database of protein sequence and function by joining the forces of the Swiss-Prot, TrEMBL and PIR protein database activities.

6.2 PDB

The Protein Data Bank (PDB, http://www.rcsb.org/pdb/, [12]) is the single world-wide collection of structural data of proteins and other biological macromolecules. In the PDB, all protein structures are stored in an organised way, and all entries get a unique PDB accession code. The data in the individual structure files is ordered according to the PDB format, making it easy to parse and extract specific information. However, due to changes in the format during the years since the data bank was established in 1971, not all files follow the format completely. Also, the file format is adapted to structures determined by X-ray crystallography, why some parameters may not be relevant to structures determined using NMR and other techniques, and other parameters would be needed. Work is under way to solve these problems.

6.3 Pfam

Pfam ([74], http://pfam.cgb.ki.se) is a semi-automatically created database of multiple sequence alignments of protein domain families. The families are defined based on clear common ancestry and sequence similarity. The database is purely sequence based, but is mentioned here since it uses HMMs to define families and construct alignments. Pfam consists of two sets of alignments with corresponding HMMs; PfamA and PfamB.

The base of Pfam is a collection of high quality seed alignments. The initial members of a seed alignment are collected from a number of sources, including structural alignments, Swiss-Prot (see Section 6.1) and published alignments. The sequences are aligned by an automatic alignment method, most

[6] http://www.ebi.ac.uk/trembl/index.html
[7] http://pir.georgetown.edu/
[8] http://pir.georgetown.edu/uniprot/

often ClustalW, and checked manually. From each seed alignment a HMM is built, which in turn is used to search a non-redundant collection of sequences from SWISS-PROT and SP-TrEMBL (see Section 6.1), called Pfamseq, for additional members. The seed is updated with selected sequences until all known members are found. These are aligned to the HMM to construct a full alignment of the family. Where available, structural information is used to ensure that each Pfam family corresponds to just one structural domain. The seed alignment, the HMM built from it, the full alignment and some annotation and cross-references to other families make up Pfam-A. Pfam-B is a less reliable collection of multiple sequence alignments, initially constructed by automatically clustering the rest of pfamseq (all sequences not included in Pfam-A). In later releases [10][9], Pfam-B has been constructed from all protein domain families in the ProDom database, not included in Pfam-A. ProDom is an automatically generated database of protein domain families [18].

As of 1999, 70% of the SCOP (Section 6.6) families are found in Pfam, and 57% of the Pfam families exist in SCOP [22].

6.4 DALI databases

The Dali Domain Dictionary[40] clusters protein domains by so called fold space attractors. Each domain is regarded as a point in a high-dimensional fold space, and a multivariate scaling method, similar to principal component analysis, is used to find the groups of proteins sharing common features. At the next level, the domains are clustered into fold types, where members of a fold have a mutual DALI [38] Z-score above 2. Then DALI and a neural network approach is used to cluster protein domains into groups of homologous proteins constituting functional families, largely consistent with SCOP superfamilies (Section 6.6). The domains are automatically defined based on compactness and recurrence. The FSSP (Fold classification based on Structure-Structure alignment of Proteins) database is a similar classification based on whole protein chains, instead of domains [41].

6.5 CATH

In CATH[60], protein domain structures are classified into five levels: protein class (C), architecture (A), topology (T), homologous superfamily (H), and sequence family (S). The classification is, as far as possible using current techniques, done automatically, with the goal of completely automatic classification in the future. The database classifies single domains, so multidomain proteins are divided into separate domains using an automatic procedure. In those cases where the procedure fails, the domain borders are determined manually.

The class (C-level) describes the content of α helices and β sheets in the structures. There are four classes: mainly α, mainly β, $\alpha - \beta$ and a special class

grouping all domains with low secondary structure content. The class is determined by an automatic procedure, which examines the secondary structure composition of one representative for each sequence family. The architecture (A-level) describes the general arrangement of secondary structures and is determined manually, while the topology (T-level) further groups the structural domains based on the overall fold. Fold in this case means that the number and arrangement of secondary structures are similar, and that the connectivity between secondary structures are the same. The homologous superfamilies (H-level) group domains by high structural similarity and similar functions. The T- and H-levels are determined by structural comparison of representative proteins using the SSAP program (Section 5.1, [78]), with different cutoffs for the two levels. For a protein to belong to a certain homologous superfamily, it must also have a common function to the other members in the superfamily. Function is determined from Swiss-Prot, the PDB file or literature.

At the lowest level (S-level, sequence family), protein domains with high sequence similarity (>35% identical) are clustered. These domains are assumed to have very similar structures and functions. The sequence similarity is determined by pairwise comparisons using the Needleman-Wunsch algorithm [58], and the sequences are clustered into families by single linkage cluster analysis.

From the PDB [12], only crystal structures with a 3.0 Å resolution or better and NMR structures are selected. These are sorted so that low resolution, native X-ray structures are first and mutant NMR-structures become last. The domain listed highest is chosen as representative for the sequence family in the classification.

In addition to the actual classification, the database contains derived data such as structural alignments and family templates. Also, for each structure in CATH, a number of graphical representations are provided, together with a report containing information from the PDB file, domain boundary data and functional data.

The CATH database has recently been extended into the CATH-protein family database (CATH-PFDB), which includes sequences found by searching the non-redundant GenBank database with CATH domain sequences using profile based search methods.

6.6 SCOP

We chose to use the SCOP (Structural Classification of Proteins) [57] database (version 1.61) as the gold standard. In SCOP, all proteins with known structures are divided into groups based on different levels of similarity. The classification is done at the domain level (see Section 2), meaning that different parts of a single protein may appear in multiple families in the classification, even in different classes. The aim is to capture evolutionary relationships between protein domains. In SCOP, a domain is defined as an evolutionary unit, either observed in

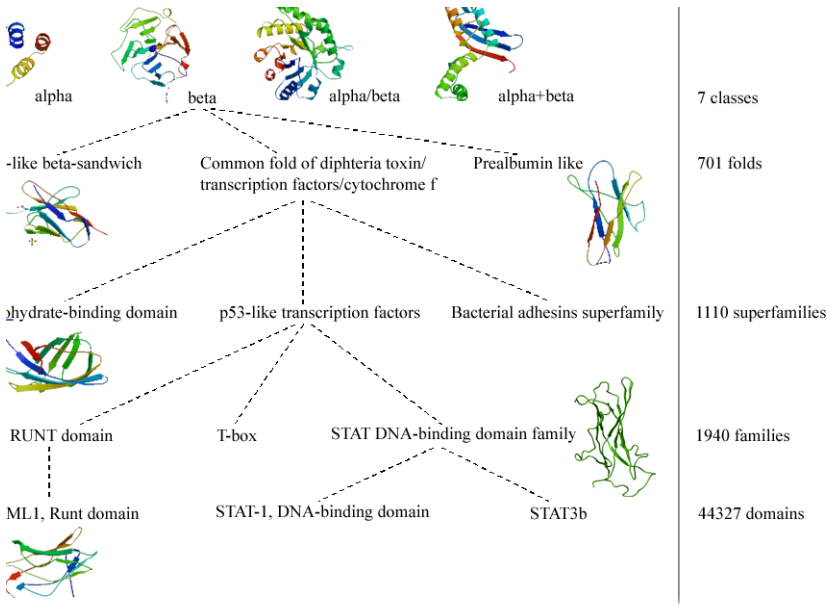


Figure 25: A schematic picture of the SCOP classification, together with the number of entries at each level, as of version 1.61 (to the right). Pictures of protein structures are taken from the PDB.

isolation in nature or together with different domains in different multidomain proteins.

In Figure 25, a schematic picture of the SCOP classification is shown. At the lowest level in the classification are the actual protein domains (bottom in Figure 25), sorted by species. Protein domains very similar in structure, and with experimentally determined similarities in function, are put into the same family, the next higher level. Especially, domains having a sequence identity of 30% or more are assigned to the same family. Families of proteins with similar structures, but uncertain similarity in function, are part of the same superfamily. One level higher is the fold, where superfamilies with roughly the same arrangement of secondary structures and the same topology are grouped together. The highest level in the SCOP hierarchy is the classes, where folds consisting of the same kinds of secondary structure elements are grouped. Apart from the four main classes shown in Figure 25, all alpha-helices, all beta-sheets and the two kinds of mixtures of alpha and beta, there exist three more true classes - multidomain proteins, membrane proteins and small proteins. There

are also four classes containing peptides, low resolution structures and other groups of proteins that could not be included in the actual classification. These are not considered as true classes.

SCOP includes all proteins in the PDB until the date they started working on the current release of SCOP, and most of the proteins whose structures have been published but not included in the PDB. The last few years a new version of SCOP has been published every 4–6 months. The database is curated, meaning that the similarity of the proteins is determined manually by a group of experts. The investigation is done using both visual inspection and comparison of structures. Automatic tools are used to speed up the classifications. Sequence comparison can be used to group domains with high sequence similarity to the same family, while structural alignments are used to suggest a fold for a protein of interest, even though manual inspection must be used to verify the result and choose an appropriate superfamily and family for the domain. The manual check of the classification is the reason why the SCOP database often is used as the gold standard for grouping of similar protein domains.

The ASTRAL Compendium [15] is a collection of sequences for the domains classified in SCOP. The sequences can be retrieved filtered according to different criteria such as sequence similarity.

6.6.1 PALI

PALI (Phylogeny and ALIgnment of homologous protein structures)[6] is a database of structure-based sequence alignments and phylogenetic trees for each SCOP family. For each family the database provides a multiple structural alignment, all possible pairwise alignments and two phylogenetic trees, one based on structure similarity and the other on similarity of aligned residues. The structural alignments are constructed using the program STAMP ([66], Section 5.4).

Also, in the latest version [31], sequences homologous to the members are aligned to the family, and Position Specific Scoring Matrices (PSSMs) and HMMs are constructed based on these enriched alignments. The alignments, PSSMs and HMMs are available in the database.

6.7 Homstrad

HOMSTRAD[56] is a database of aligned protein structures. The classification is based on SCOP, Pfam, PROSITE, SMART and sequence comparisons by PSI-BLAST and FUGUE. The information of all these methods/databases are combined and the family definitions are defined manually to group proteins that share sequence/structure similarity. For each family, a structure-based alignment constructed using COMPARER[68], manually checked and edited, is provided. The family alignment is composed of representative members only.

In the HOMSTRAD families, the sequences on the average have 30% sequence identity, and even if the sequence identity between a pair of sequences

from the same family should be below 20%, they are often bridged by another sequence having more than 20% identity to each of the sequences.

7 The structure anchored HMM method (saHMM)

In the previous sections, we have discussed methods to construct multiple sequence alignments based on statistics (Section 3) and methods to represent these alignments and model protein families, in particular hidden Markov models (HMMs, Section 4). HMMs have proven to be very powerful at recognising new members of protein families (see for example [55], [61]). However, as any statistical method, HMMs based on conventional sequence alignments have difficulties in detecting very distant relationships, between protein sequences with very low sequence identities. Therefore, several attempts have been made to use structural information, both together with HMMs and with other methods, to be able to detect these relationships (see Section 8).

The observation this work is based on, is that sequence alignments based on statistical methods might differ significantly from those constructed based on structural superposition of the corresponding protein structures (see Section 5.5). Since sequence alignments based on structural superposition align those residues that are close in space, and really can be regarded as each others equivalents, these alignments should be biologically more significant than alignment based on statistics and comparison of the symbols representing the individual residues. Especially for sequences with very low sequence identity, the structure based sequence alignments should be more reliable in a biological sense. The idea behind this work is to use multiple sequence alignments constructed from multiple structural superposition of protein structures, to build HMMs that might be better at finding distant relationships between proteins, far below the twilight zone (see Sections 1 and 7.3.1). Our method and some preliminary results have previously been reported in [81] and presented at several workshops.

7.1 Outline of the method

In Figure 26, a "flowchart" showing the main steps of our method is shown. In the first step, only those sequences in a family having very low sequence identity with respect to each other are selected as representatives for that particular family of proteins. The structures of these proteins are then multiply superimposed, so that one superposition of structures is made for each family.

From the superposition a multiple sequence alignment is deduced, based on which residues are close to each other in space (or preferably on top of each other) when the structures are superimposed. This step often is performed simultaneously as the structural superposition, depending on which method (program) is used for the structural comparison. The resulting multiple sequence alignment is (hopefully) much better than what could be achieved from aligning

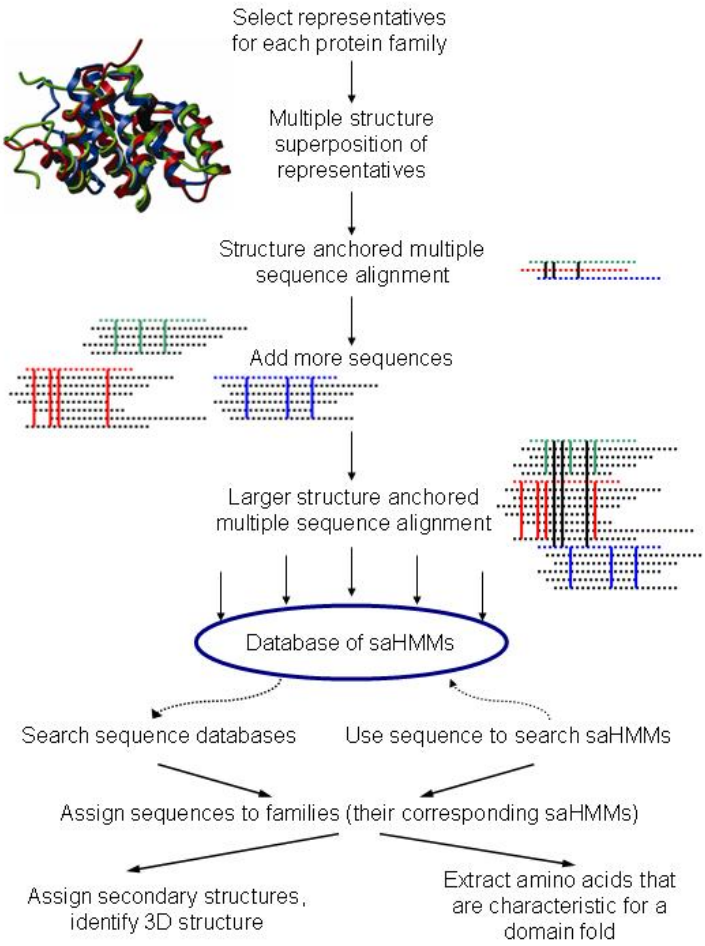


Figure 26: A schematic illustration of our method. We start by constructing a structural superposition of all proteins in the family. This is then used to extract a structure-based multiple sequence alignment, that in turn is used to build a structure anchored hidden Markov model (saHMM) for the family. From the resulting database of saHMMs useful information can be extracted. See text for details.

the sequences based on sequence information. Especially in the case of low sequence identity, one would expect such a structure-based alignment to be more biologically correct than one based on sequence information only.

To get more sequences in the alignment, and thus be able to build better hidden Markov models (HMMs) in a later step, each of the representatives can be searched independently against some sequence database using BLAST (see Section 3.4) or some other sequence alignment tool. The so found sequences (similar to the query sequence) are then aligned by sequence to the member used as query, and consequently to the other members through the structure-based alignment. In this way, several new sequences (with no structure information) can be added to the structure-based alignment, creating a large multiple sequence alignment based on structure. This, however, has not been implemented yet (see Section 11).

The structure-based multiple sequence alignment is used to build a structure anchored HMM (saHMM) representing the family. The construction of one model for each family yields a whole database of saHMMs, which in turn can be searched for similarities to new sequences. If one has a particularly interesting sequence, this can be searched against the database to find which (if any) saHMM fits the sequence best, and thus which family the sequence most likely belongs to. If, on the other hand, one is particularly interested in a certain protein family, the corresponding saHMM can be used to search sequence databases or newly sequenced genomes for more members of the family. In both cases, the fact that the saHMMs are built from structure anchored alignments, means that matching a sequence to a saHMM also matches the sequence to the corresponding structure. Also, by aligning the new sequence to the proteins the saHMM is built from, makes it possible to assign secondary structures to the sequence.

7.2 Output

From a users point of view, the method should be a black box, into which a sequence can be sent, and out comes some (hopefully useful) results. These results are the name of the family (or a list of families) the sequence resembles most, a measure of how good the match is, and links to structures representative for the family. It could also be possible to produce an alignment of the query sequence to the sequences the corresponding saHMM is built from. Based on such an alignment of a query sequence to proteins for which the structures are known, secondary structures could be assigned to the query sequence.

What happens inside the black box is that the sequence is searched against all saHMMs using hmmpfam in the HMMER2.0-package. The result of this search is a list of names of saHMMs that match the sequence, sorted by the E-value of the match. An example of the output is shown in Figure 27.

The E-value corresponds to the probability that the similarity found is ran-

dom. The lower (closer to zero) the E-value is, the less probable it is that the match is random, i.e., the more likely it is that the query sequence really is related to the sequences the saHMM is built from (see also Section 4.2.1). From this list the best hit (or the best hits) is chosen, and the name and a description of the family corresponding to the successful saHMM is reported, as well as the names of and links to the protein domains the saHMM represents. All this information can be precomputed.

Another possibility is to search a whole genome with a single saHMM, looking for sequences belonging to a certain family. The output of such a search would be similar to the previous, except that instead of a list of families that most likely fits a query sequence, one will get a list of those protein sequences from the genome that most likely belong to the family modeled by the saHMM.

7.3 Implementation

7.3.1 Selection of sequences to use

To build the models we needed to define groups of proteins with similar structures, and select representatives from each group. We chose to use the family level in the SCOP classification (Section 6.6) as groups of protein domains to superimpose.

In the PDB (Section 6.2), and consequently in SCOP, there is a high degree of redundancy [14]. Some proteins have a huge number of entries, only differing in single positions, while the majority of proteins only have one entry. Consequently, in SCOP some families contain lots of domains, while other only have one or two members. The number of families in superfamilies and superfamilies in folds are also skewed, but not correlated. Both the PDB and SCOP are biased towards proteins that crystallize or that are suitable for NMR experiments.

To avoid getting an alignment biased towards sequences very common in the family, and to get maximum spread in the representatives from each family, we decided to use only sequences with mutual sequence identities below a certain limit. The limit was defined as the border to the so called twilight zone as described in [65]. The actual curve that defines the border to the twilight zone differs depending on the data it is based on, and on slightly different definitions between authors, but the basic idea is the same. If all known proteins are pairwise aligned, the resulting sequence identity can be plotted against the alignment length. In such a plot, pairs of non-related proteins will have low sequence identity over mostly short alignment lengths, while related proteins often have higher sequence identities and longer alignments. It turns out that protein pairs falling above the curve in Figure 28 always are homologous proteins. Around the curve, the number of unrelated pairs start to appear, and increase as one goes below the curve. Below the curve most of the protein pairs are not related at all, but there still exist some pairs that are. That two proteins are related does not imply that they have a high sequence identity. The twilight

```
hmmpfam - search one or more sequences against HMM database
HMMER 2.2g (August 2001)
Copyright (C) 1992-2001 HHMI/Washington University School of Medicine
Freely distributed under the GNU General Public License (GPL)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
HMM file:                 all.hmm
Sequence file:            scop2.fa
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Query sequence: d1d2zb_
Accession:      [none]
Description:    Very interesting protein domain

Scores for sequence family classification (score includes all domains):
Model    Description                                       Score    E-value  N
-------- -----------                                       -----    ------- ---
a.77.1.1 DEATH domain, DD                                  320.8    1.4e-94   1
g.7.1.1  Snake venom toxins                                -28.6        9.8   1

Parsed for domains:
Model    Domain  seq-f seq-t    hmm-f hmm-t      score  E-value
-------- ------- ----- -----    ----- -----      -----  -------
a.77.1.1   1/1       1   143 [.     1   163 []   320.8  1.4e-94
g.7.1.1    1/1      20    55 ..     1    67 []   -28.6      9.8
//
```

Figure 27: Example of output from HMMER. The first rows are some general information. Then the name of the database of HMMs is presented. In this case the search is done against the database 'all.hmm', containing HMMs for a number of protein families. The sequences of the query proteins are in this case stored in the sequence file 'scop2.fa'. Then the actual result are presented. The first (and only shown) query is the protein domain d1d2zb_. Some general information derived from the sequence file is presented along with the query name. This search generated two hits. The best hit is to the HMM named 'a.77.1.1', modeling the same family, which in other words is the DEATH domain family. The score of this hit is 320.8 and the E-value is 1.4e-94. This is a very significant hit, so the query most likely belongs to the DEATH domain family. The other hit, g.7.1.1, has an E-value of 9.8, which in some cases might give interesting clues, but is not significant. The last rows show the actual residues matched to the HMM. It is also possible to get a sequence alignment of the query to the HMMs (not shown).

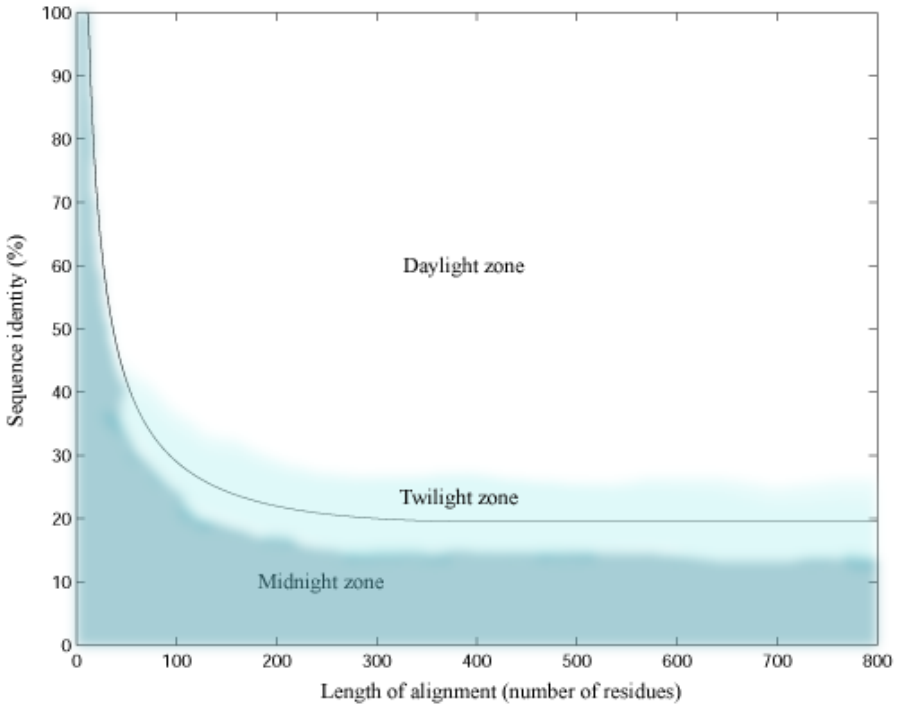


Figure 28: The curve that defines the twilight zone according to Rost [65]. When the percentage sequence identity is plotted to the length of the alignment of two protein sequences, related proteins fall above this curve. The further into the twilight zone one gets, the less likely it is that the two proteins are related.

zone is the border where the percentage sequence identity between two aligned protein sequences no longer tells whether the two proteins are related or not (Figure 28).

In practice, this "border" is fuzzy, hence it is called the *twilight zone*. The equation derived by Rost to model this "border" is

$$pI = \begin{cases} 100 & for\ L \leq 11 \\ 480 \cdot L^{-0.32 \cdot \left(1+e^{-L/1000}\right)} & for\ 11 < L \leq 450 \\ 19.5 & for\ L > 450 \end{cases} \tag{24}$$

where pI is the cut-off percentage of residues identical over an alignment length of L residues. This means that for short alignments the sequence identity has to be very high for the two sequences to be considered related, while for longer alignments even quite low percentage identities are significant. The equation is fitted to the data such that it excludes most false positives, i.e. most pairs

falling above this curve really are related. Pairs of proteins falling under this curve might be related, but most probably are not. One goal of this work is to be able to detect those pairs that fall under the curve even though they are related.

The procedure to select representatives for each family is illustrated in Figure 29. The representatives were chosen by taking all proteins belonging to the family (as defined in SCOP) and running STAMP (Section 5.4, [66]) on all pairs of proteins. In the cases where STAMP failed, MAPS was used to produce a better start alignment for STAMP to work on. If STAMP failed again, the MAPS alignment was used in cases where it was long enough, otherwise the two sequences were considered "troublesome" and treated as very similar. If the STAMP alignment of two sequences revealed a sequence identity above the limit for that alignment length, as defined in equation (24), one of the sequences was discarded. If one of them had higher resolution than the other, the one with the best resolution was kept. If the resolutions were within 10% of the mean value of the two resolutions, the protein with the best R-value was chosen, and if these were equal, one was chosen randomly. To guarantee high quality structures to make the superimpositions on, only X-ray structures with a better (lower) resolution than 3.6Å were chosen, and all structures with worse resolutions, or determined using NMR or any other technique, were discarded. After going through the first round of selection, all removed proteins were checked against all left to assure that only sequences with too high sequence identities were discarded. The rationale behind this is that in the process of removing proteins, it may happen that sequence A is removed due to high similarity to sequence B. If B later is removed due to similarity to sequence C, it may well happen that A and C have a similarity below the threshold, and thus A now lacks a representative and has to be reintroduced.

An alternative to the SCOP classification would be to use CATH, a similar database that is built on automatic clustering of the proteins (see Section 6.5). This would make the method less dependent on A. Murzin (and co-workers), who runs SCOP, but on the other hand CATH as usually seen as a less reliable classification of proteins. This far, CATH is less straightforward to use, and SCOP is the commonly used database in similar studies.

PALI (Section 6.6.1) is a collection of STAMP-generated alignments of SCOP families. The alignments, and PSSMs and HMMs constructed from them, are available in the database. However, since the alignments are based on whole SCOP families, and not only those members having a very low sequence identity, we had to make our own alignments. Additionally, the alignments are not easily available other than one by one, and much of the work was done before we knew about the database.

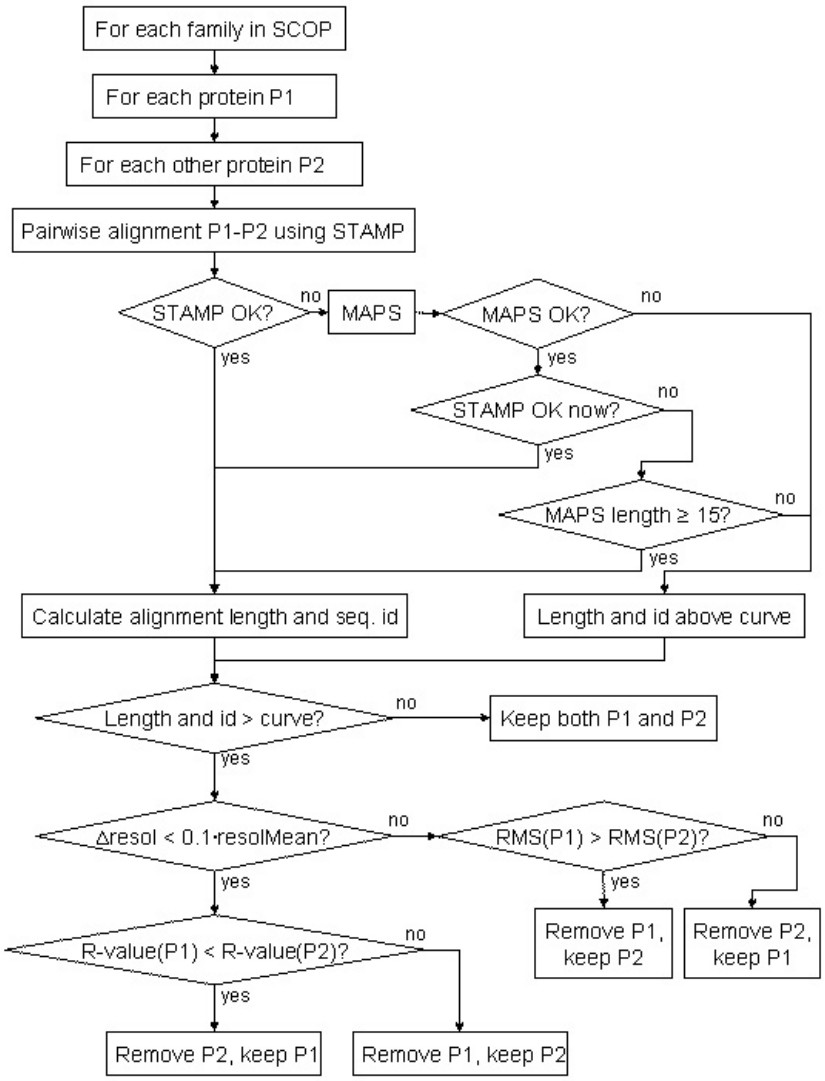


Figure 29: A flowchart showing the procedure used to select representative proteins from each SCOP family. Each member in a family was compared to each other member, by constructing a pairwise structural alignment using STAMP. The length of the structure based alignment and the resulting pairwise sequence identity were calculated, and if the numbers fell above the curve in Equation (24), that is the two proteins are too similar, one of them was removed. The protein to remove was chosen as the one with the highest (worst) resolution, or if the resolutions were similar, as the protein with highest R-value.

7.3.2 Construction of superpositions and multiple alignments

The STAMP method and program (Section 5.4) was chosen for the construction of structural superpositions. The program also generates the structure-based multiple sequence alignment based on structural equivalences. The reason for choosing STAMP was that it produces the best sequence alignments among the programs tested. The first choice was MAPS (Section 5.3), since this method (program) superimposes multiple structures simultaneously. The program produces very nice superimpositions when looking at the structures in 3D, but can only find very short stretches of aligned residues, those that are really close in space, and therefore the program was abandoned. STAMP produces longer sequence alignments, and also has the benefit that the output can be easily parsed to a format suitable for input to HMMER. Important to note is that there does not exist many publically available programs for multiple superposition of protein structures. The only real alternative to STAMP and MAPS is the multiple version of SSAP, which is not readily available to us.

STAMP needs an initial alignment to start from. We use the ROUGHFIT option, which generates an initial alignment where the sequences are aligned from their N-terminal ends. This works well in cases where the sequences are roughly of the same lengths, and where there is high structural similarity. However, in general it often fails to generate a good starting point. In those cases where STAMP completely fails to align the domains in the family, we chose to use MAPS to generate a better initial alignment in the form of superimposed structures.

7.3.3 Construction of HMMs

In Figure 30 the procedure followed to construct one saHMM for each SCOP family is illustrated. The representatives of each SCOP family with at least two members left after the selection procedure described in Section 7.3.1 were multiply superimposed as described in the previous section. The multiple sequence alignments produced by STAMP were fed into HMMER version 2.0 to construct HMMs, using default parameters. The type of HMM was chosen to be optimal to find alignments and/or hits local with respect both to the HMM and with respect to the query sequence. All HMMs were calibrated to get fitted E-values.

8 Related work

The idea to use protein structures and structure-based alignments to improve recognition of related proteins is not new. The incorporation of structural information has in many cases proved to improve the ability to find remote relationships. Perhaps the most straight forward approach is to use structural alignments to generate substitution matrices for use in sequence comparisons. Blake

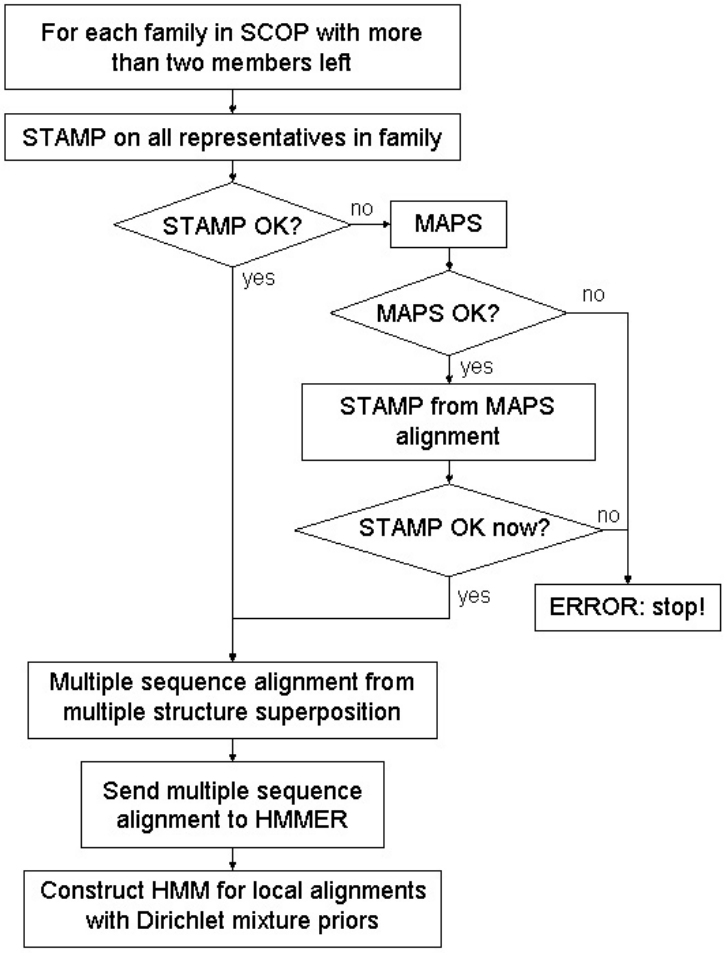


Figure 30: How the family saHMMs were constructed. Only SCOP families with two or more representatives left after the selection described in Section 7.3.1 were used, since at least two proteins are needed to construct an alignment. The representatives in each family were structurally aligned as described in Section 7.3.2, using MAPS when needed. The resulting structure anchored multiple sequence alignments were fed into HMMER2.0, using standard parameters, to produce the final saHMMs.

and Cohen [13] use the CATH classification and the structure alignment program AFFC[23] to construct a series of substitution matrices based on structural alignments. Their results indicate that the addition of structural information improves the alignments produced for sequences with low sequence identity.

Rice and Eisenberg[64] used pairwise structural alignments to construct a 3D-1D substitution matrix, H3P2, that is used to match a sequence and its predicted secondary structure to representative proteins with known structures. Each position in the query sequence is represented by one of seven residue classes and one of three secondary structure classes. Each position in the representative structures is described by one of seven residue classes, one of three secondary structure classes and two classes indicating whether the residue is buried or not. This makes the substitution matrix five-dimensional. It is shown that the H3P2 matrix detects more distant homologies, where the sequence identity is very low, than common matrices such as PAM250 and BLOSUM62.

Kelley et al.[46] use structural alignments to produce so called 3D-PSSMs (three-dimensional Position Specific Scoring Matrices), which are used in a dynamic programming procedure together with secondary structure matching and solvation potentials. The addition of 3D information is shown to increase the methods ability to recognise homologous sequences.

FUGUE ([69], http://www-cryst.bio.cam.ac.uk/fugue/) is a program for homology recognition that uses environment-specific substitution tables derived from structural alignments and structure-dependent gap penalties, to construct profiles based on structural information. Query sequences are searched against a sequence database to construct a PSI-BLAST alignment, which is converted to a profile that is compared to the structure-based profiles. The structural alignments are collected from HOMSTRAD ([56], Section 6.7). They find that FUGUE is better than common sequence comparison and fold recognition programs such as PSI-BLAST and THREADER.

Structural information has also been included in HMMs. Hargbo (now Tångrot) and Elofsson [35] constructed HMMs based on both sequence and secondary structure simultaneously, so called ssHMMs. The ssHMMs match and insert states have, in addition to the emission probabilities for amino acids (Section 4), also associated an emission probability for secondary structures. This means that even though the actual sequence symbol does not match the HMM, a position can get higher probability if the secondary structure matches that of the model. The secondary structures for query sequences, which of course are not known, are predicted using some secondary structure prediction method before the search. The ssHMMs were shown to perform better than "ordinary" HMMs in recognizing proteins having the same fold as the modeled proteins.

Secondary structure information have been included in profiles as well. Lüthy et al. [51] use secondary structure specific substitution tables to construct secondary structure-based profiles. They show that the secondary structure-based profiles are better than profiles based on an ordinary Dayhoff table at recognis-

ing distant homologies.

Gnanasekaran et al. [28] used structure-based multiple sequence alignments to construct profiles of conserved transmembrane β strand regions in porins. These profiles were successfully used to search for β stranded membrane proteins in a sequence database of mostly membrane proteins.

Al-Lazikani et al. [1] combine trusted multiple sequence alignments, derived from structural alignments, with sequence alignments of close relatives, and use the resulting alignment as a base for a hidden Markov model, in a way similar to our idea. However, they use manual intervention where required, and implement the method on a single family only. Since the goal of their work was to locate the SH2 domain in Janus kinases, which they successfully did, the method itself was not evaluated or compared to other methods. Their positive results were encouraging for further testing of our approach.

Griffiths-Jones and Bateman[33] have come up with a strategy very similar to ours. In their work, they build HMMs from structural alignments derived from the HOMSTRAD database ([56], Section 6.7). HMMs were built from the HOMSTRAD alignments and from alignments of the same sequences produced by ClustalW and T-Coffee. The HMMs were used to search Pfamseq (see Section 6.3) for sequences belonging to the same family as those included in the model. Only those HOMSTRAD alignments that correspond to a single Pfam family were used, so that the Pfam database could be used as a list of family memberships. The specificity and sensitivity were calculated for the HMMs' ability to place sequences in the correct family. In their study, they found that the sensitivity of HMMs built from T-Coffee alignments did not differ significantly from HMMs from HOMSTRAD alignments, and that ClustalW-based HMMs performed slightly worse, bordering on significance.

Their method differs from ours in that they use HOMSTRAD families instead of SCOP. We have also taken great care only to include sequences with very low sequence identity in our alignments, to ensure maximum possible spread in training data. In the HOMSTRAD families, the sequences have higher sequence identities (Section 6.7).

9 Testing the saHMM method

To test the performance of the method a number of tests were constructed.

9.1 Going into the midnight zone

To start with, two selections were made according to the procedure described in Section 7.3.1, one filtered according to the curve described in Equation (24), and one using a sequence identity threshold lying 10 % below this curve. This

second selection would show the effect of going even deeper into the "midnight zone".

An saHMM was built for each family based on the sequences left after filtering. Then all sequences in SCOP (Section 6.6) were searched against all HMMs to find which model(s) that fitted each sequence. The results of these searches were summarized in a sensitivity-specificity plot. The sensitivity is defined as the fraction of sequences that get a hit to their corresponding family HMM. For a given E-value e in the search results, the sensitivity is calculated as

$$sens = \frac{tp}{tp + fn}. \tag{25}$$

Here, *tp* denotes the number of true positives, that is the number of sequences that find their corresponding family HMM with an E-value less than or equal to *e,* and *fn* is the number of false negatives, that is the number of sequences whose family HMM get an E-value greater than e and therefore are not found.

The specificity measures how specific the results are, that is the fraction of of all hits that are correct relationships. For a given E-value *e,* the specificity is calculated as

$$spec = \frac{tp}{tp + fp}, \tag{26}$$

where *tp* is defined as above and *fp* is the number of false positives, that is the number of HMMs that get an E-value less than or equal to e to some sequence not belonging to the same family as the HMM.

In the ideal case, the specificity would be one for all sensitivity values from zero to one, and then drop rapidly, meaning that no false hits are found until all true relationships are identified.

9.2 A worst case scenario

Secondly, a worst case scenario was studied, namely the case of having only two sequences to base the saHMMs on. This is a worst case since the family specific information increases with more proteins in the family. An HMM built from just two sequences risks to either be too specific for those sequences only, or to include too much prior information and match "anything". Too few examples to build the HMM from makes it very difficult to distinguish between important characteristics, such as conserved residues, and properties that just happen to be the same for those particular sequences, without being a characteristic for the whole family. The performance of HMMs is expected to increase with an increasing number of sequences to base the HMM on, since more sequences makes it possible to capture properties common for the whole family. To investigate the worst possible performance, only those families that had just two sequences left after the filtering procedure were collected. The investigation was done for the selection 10% below the twilight zone curve.

9.3 The effect of few proteins in the HMM

Since most families only have just a few members left after the filtering procedure, most have just two members left, we decided to investigate the effect of the number of sequences in the HMM on its ability to recognise family members. For this purpose, we chose to take a closer look at the Ig-Vset domain family, since it is one of the largest families available. The family contains 953 members, and after selection (based on the twilight zone curve) 25 of these are left. Of these 25, all possible combinations of two proteins were chosen and structurally aligned, and HMMs were constructed from the resulting pairwise alignments. The procedure was repeated for all possible combinations of three, four, five, etc. members, up to the complete set of 25 members. Since the number of possible combinations rapidly become huge, 1000 of the possible combinations were randomly chosen for each of the groups of three to 22 members. Each HMM was used to search SCOP for family members, and the number of members found was counted. The results were summarized in a figure (see Figure 37). Along the x-axis, the number of proteins included in the HMMs are indicated, and along the y-axis the number of family members found are shown. This means that the first "column" in the plot shows the ability of HMMs built from combinations of two proteins to find family members, the second shows the performance of HMMs built from three sequences, etc. Each mark (stretch) in the figure represents one or more HMMs, since several HMMs can find the same number of family members. Each column is normalised, such that the mark representing most HMMs gets a weight of one, and the others lower weights corresponding to the number of HMMs represented by each mark. The colours of the marks denote these normalised values, such that red marks represent many HMMs, while dark blue ones represent only a few.

9.4 The effect of structure anchoring

Finally, the effect of basing the HMMs on structural superimposition, instead of sequence based alignments, was assessed. For each family of protein domains where it was possible to construct an saHMM, the domain sequences were also aligned using the sequence-based alignment program T-Coffee. More information on T-Coffee can be found in Section 3.6.3. The sequence-based alignments were then used to build HMMs, resulting in a second database of HMMs corresponding to SCOP families. In the following those HMMs will be called tcHMMs since they are built from T-Coffee alignments.

We use the default parameters in T-Coffee for the test. One of the reasons for selecting T-Coffee as the reference sequence alignment program is that it is superior[59] to ClustalW, one of the most commonly used sequence alignment programs.

All the sequences in SCOP were searched against the two databases of HMMs. For each domain sequence there is only one correct saHMM and one

tcHMM that should be found, since one domain in SCOP only belongs to one SCOP family. However, hits to HMMs representing families in the same superfamily as the query sequence might also be acceptable. The reason SCOP was chosen as the collection of sequences to test on, is that we know which family each domain sequence belongs to, and therefore we also know which HMM to expect a hit to.

The way the sequences to build HMMs from were chosen, ensures that all sequences belonging to a family with a corresponding HMM have a significant sequence similarity to sequences used to build the HMM (see Section 7.3.1). Because of this similarity, it should be expected that all sequences quite easily can find their "family HMM". To really test the HMMs capability to recognise sequences with very low sequence similarity to other sequences in the family, but where the structure is conserved, another set of saHMMs (and corresponding tcHMMs) was constructed. In this collection, a number of HMMs were constructed for each family, leaving out one representative sequence at a time. Assuming we have a family with representative domains A, B and C, one structure-based alignment was constructed from each of the groups AB, AC and BC, and saHMMs were constructed from these alignments. The question is then if the saHMM built from AB is found when searching with sequence C (or sequences similar to C in sequence), and similarly if AC is found by sequence B and relatives, and BC by sequence A and sequences similar to it. Corresponding HMMs were constructed from sequence-based alignments of the subgroups.

10 Results and evaluation of the saHMMs

10.1 Number of representatives left after selection

From the SCOP classification, only true classes were used in this work (see Section 6.6), that is, for example peptides and low resolution structures were removed prior to the selection, and are not included in the numbers shown below.

After going through the selection procedure described in Section 7.3.1, 2794 of the 42434 domains in SCOP1.61 were left. These are distributed over 748 families, each having between 2 and 35 members. About half of the families are left with only two members. From these, we were able to construct saHMMs for 746 families.

The second selection made, selecting only those proteins that have a sequence identity 10% below the twilight-zone curve, to any other selected protein from the same family, was slightly smaller. In this set, 2149 domains was left after selection, distributed over 653 families with between 2 and 32 members each. Also here, about half of the families are left with only two members. 653 saHMMs could be constructed from this set of sequences.

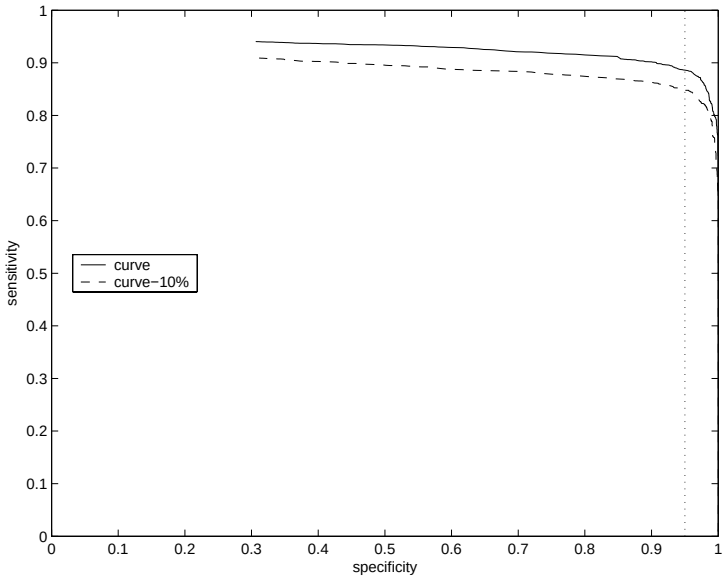


Figure 31: A sensitivity-specificity plot over the result from the twilight curve selection and the selection based on a threshold 10% below the curve. The sensitivity is the fraction of all true relationships recognised, and is plotted along the x-axis. The specificity is the fraction of all relationships found that are correct, and is plotted along the y-axis.

10.2 The effect of going deeper into the midnight zone

First, the performance of the two selections made was compared.

In Figure 31, the specificity is plotted versus sensitivity for a number of E-values ranging from zero to ten. The line shows the results for the selection based on the twilight zone curve, while the broken line shows the results for the selection 10 % below the curve. For the selection defined by the twilight zone curve, the performance is very good, with almost 90% of the sequences finding their corresponding family HMM before the number of false hits exceeds 5% (vertical dotted line in the figure). The selection based on a threshold 10% below the twilight zone curve performs slightly worse, with about 85% of the true relationships found at a false hit rate of 5%. But this is still quite high, showing that even though we are well below the twilight zone, it is possible to identify a large fraction of the members of a protein family.

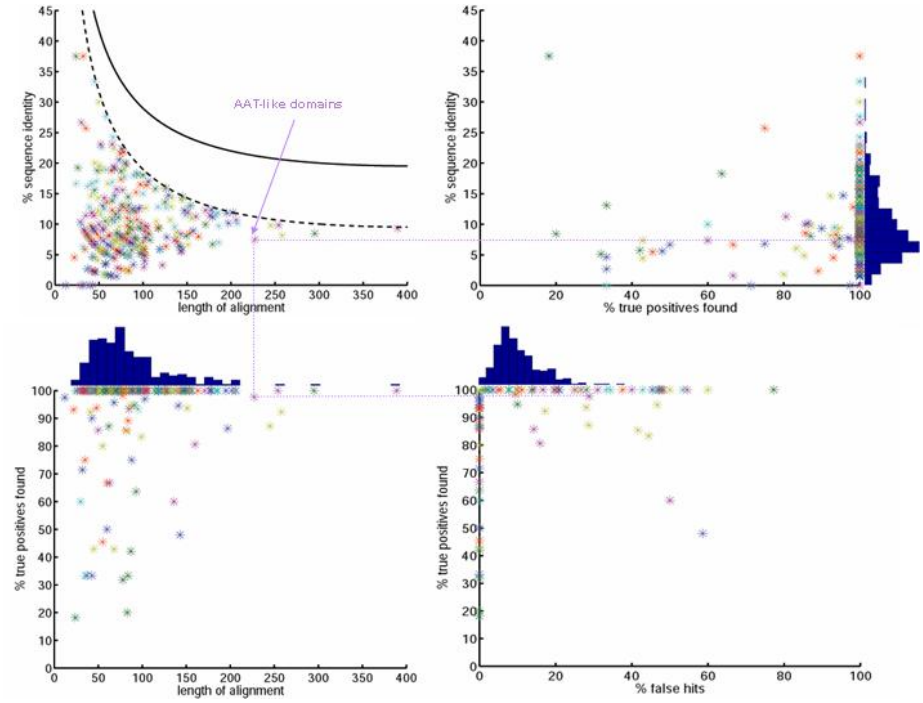


Figure 32: The four figures 33, 34, 35 and 36 collected in the same figure. Top left: Percentage sequence identity as a function of alignment length. The threshold 10% below the twilight zone curve (continuous line) is shown with dashes. Top right: Percentage sequence identity as a function of percentage true positives found. Bottom left: Percentage true positives found as a function of number of residues aligned. Bottom right: Percentage true positives found as a function of percentage false hits.

10.3 A worst case scenario

To investigate the worst possible performance, only those 349 families that had just two sequences left after the filtering procedure were collected. In Figures 33, 34, 35 and 36, each star represents one of these families. A star in one figure has a corresponding star in the other figures, coloured the same way and representing the same family. The four figures are collected in Figure 32, to illustrate how they are related. As an example, the family of the AAT-like domains is marked with an arrow, and the four stars representing this family are connected by dotted lines.

When the sequence identity between the two proteins in each family is plotted as a function of alignment length (Figure 33), it turns out that most pairs have quite low sequence identities (well below the threshold) distributed over

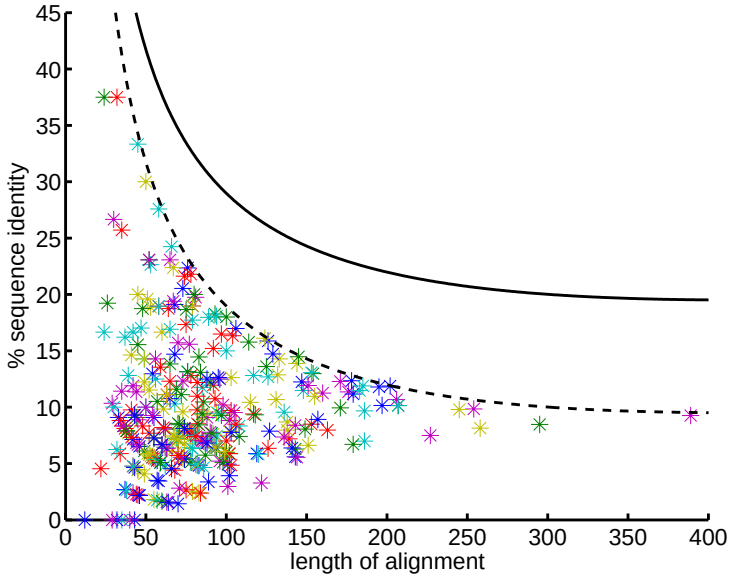


Figure 33: Percentage sequence identity as a function of alignment length. The threshold 10% below the twilight zone curve (continuous line) is shown with dashes.

relatively short alignment lengths (less than 100 aligned residues).

The percentage positives found was calculated as the number of family members that had an E-value below 0.1 (those are members to which the HMM had a significant hit), divided by all family members. Most families found 100% of their members. This is very positive, but should not be too surprising since all family members have a relatively high sequence identity (over the threshold of 10% below the twilight zone curve) to at least one protein included in the HMM. It seems like percentage sequence identity within the HMM does not affect the performance noticeably, as can be seen in Figure 34. Here, percentage sequence identity is plotted as a function of percentage true family members found. The distribution displayed along the right axis shows the number of families at each sequence identity, finding 100% of their members. Considering that most families have low sequence identity, no correlation can be found between sequence identity and ability to find family members.

When plotting the percentage family members found as a function of alignment length (Figure 35), longer alignments are found to give slightly better results.

The percentage false hits was calculated as the fraction of the proteins found by a HMM (with an E-value below 0.1), that do not belong to the family. In

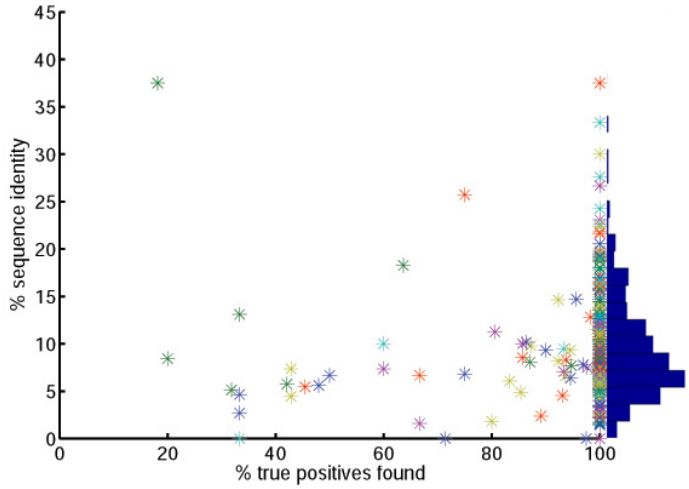


Figure 34: Percentage sequence identity as a function of percentage true positives found.

Figure 36, it can be seen that most HMMs either find all their family members, or they do not get any false hits at all. In addition, the major part of all family HMMs find more than 80% of their family members, with less than 20% false hits, which is promising. The histogram at the top shows the distribution of families that find 100% of their family members.

10.4 The effect of few proteins in the HMM

The results from the searches with saHMMs built from combinations of different number of proteins are shown in Figure 37. Along the x-axis, the number of proteins included in the HMMs are indicated, and along the y-axis the number of family members found are shown. For HMMs built from two sequences only, the results are very diverse. Some combinations of two proteins make it possible to find about 900 of the 953 family members, while other combinations find less than 50. Two categories of HMMs are slightly more represented; those who find very few members (about 20) and those who find almost all (about 900). These results indicate that it is impossible to know whether an HMM representing a family with only two members left after filtering makes a good job at recognising all possible members (known and unknown) or not. The trend when adding more representatives to the HMMs, is that the majority of the HMMs find more and more family members. At about ten proteins in the HMM, the performance

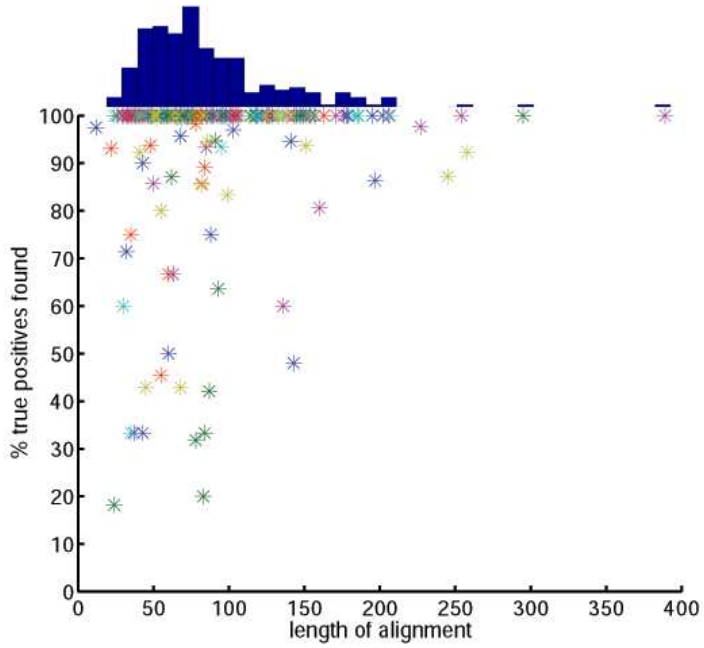


Figure 35: Percentage true positives found as a function of number of residues aligned.

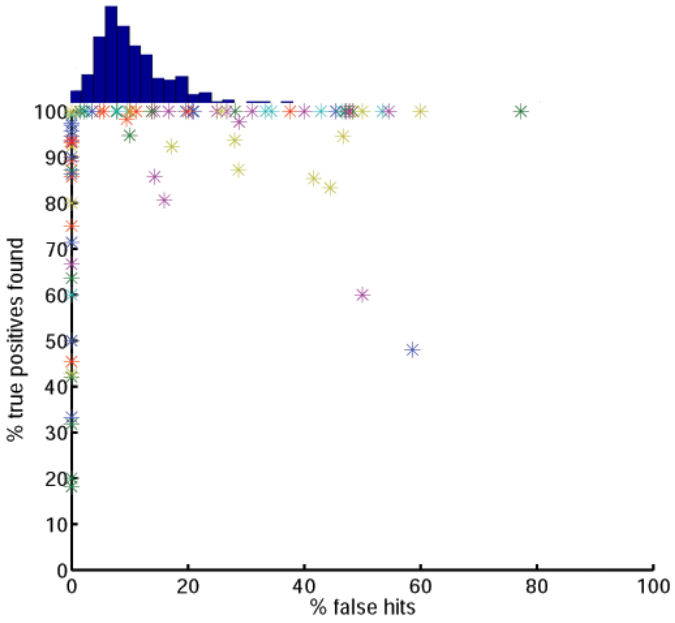


Figure 36: Percentage true positives found as a function of percentage false hits.

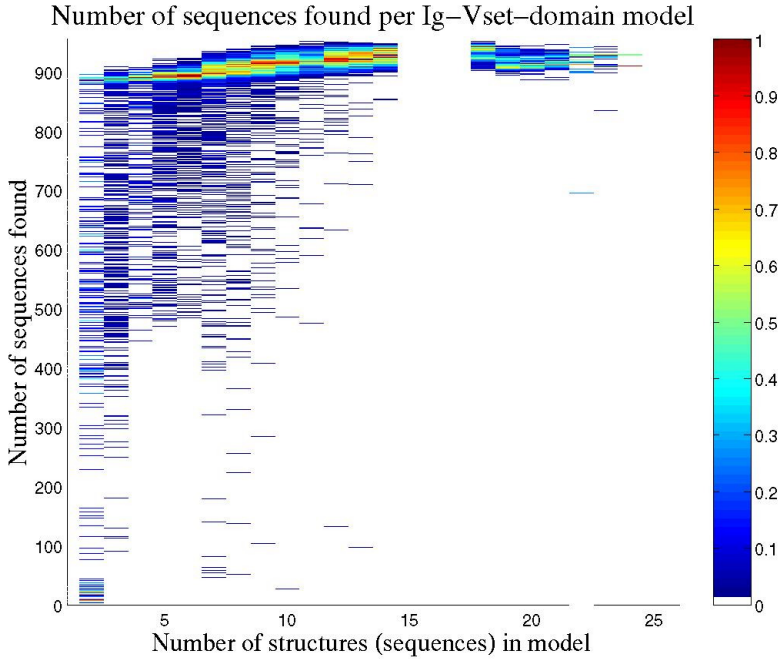


Figure 37: The number of family members found for different combinations of representatives from the Ig-Vset domain family. Each "column" in the plot represent the result for HMMs built from combinations of the corresponding number of representatives. See Section 9.3 for an explanation of the plot.

seems to approach a "steady-state", and thereafter, not much is gained by adding more members. This is contrary to what one might expect - adding more family members should make the HMM better at recognising diverse family members and less specific for the few examples used in the construction. The "steady-state" might be due to increasing difficulties at superimposing many structures simultaneously. Another reason for this behaviour might be that trying to merge too much information, in the form of many different members, makes the HMM too "broad", so that it is not able to match anything really good.

10.5 Comparison to sequence-based alignments

For each family with at least two members left after filtering with the twilight curve as cutoff, two HMMs were built. The first, the saHMM, is based on structural alignments, while the other, the tcHMM, is based on a sequence alignment of the same sequences, constructed with T-Coffee (Section 3.6.3). The

	Family members found, %		False hits, %	
saHMMall	82.09	(24,811)	1.78	(450)
tcHMMall	93.86	(28,367)	4.11	(1,215)
saHMMabo	26.11	(533)	15.13	(95)
tcHMMabo	37.04	(756)	23.87	(237)

Table 1: The percentage of all family members found and the percentage of all hits that are false for each kind of HMM. Numbers in parenthesis are absolute numbers. saHMMall and tcHMMall are HMMs based on structure superpositions and T-Coffee alignments respectively, made on all representatives in the family. saHMMabo and tcHMMabo are corresponding HMMs made from all representatives but one at a time in a family.

two sets of HMMs were used to search all SCOP sequences for family members. The results are shown in the first two rows of Table 1. The saHMMs on average find 82.09% of the family members, and only 1.78% of sequences found by the HMMs are false hits, that is, does not belong to the same family as the HMM. The tcHMMs on the other hand find 93.86% of all family members, but 4.11% of the hits are false. Surprisingly, the tcHMMs are better at recognising family members than the saHMMs are. But this ability comes at a cost, the tcHMMs have more than the double amount of false hits compared to the saHMMs. To get reliable results, the saHMMs seem superior.

In the, perhaps more realistic, case where one sequence at a time was removed from the selection prior to building the HMMs, the results are similar. Here, only the sequences removed were counted as true hits. That is, when searching the HMM lacking sequence x against SCOP, a true hit was counted if x is found, ignoring all other family members. All hits to non-family members were counted as false hits. The results are displayed in the last two rows of Table 1. 26.11% of the saHMMs lacking one representative, are able to find that sequence among all others, while 15.13% of the hits are false. The tcHMMs find 37.04% of the "missing" sequences, but 23.87% of the hits are false. Here too, the tcHMMs find more family members, but the relative increase in the number of false hits, compared to the saHMMs, is higher than the relative increase in true hits.

It is worth noting that the tcHMMs are not an established method, but was invented by us for testing purposes. These might be used as a complement to the saHMMs, when a higher coverage is required. Since the tcHMMs too are based on the selection of sequence below the twilight zone, these too would be less biased than many other methods.

11 Discussion and future work

The main strength of the saHMM method is that it uses accurate multiple sequence alignments as the base for HMMs. With accurate we mean the correct alignment from a structural point of view. Aligning sequences based on where residues are placed in space is intuitively more correct than basing an alignment on the raw sequence and algorithms for symbol recognition and comparison. This is especially true when the goal is to find similarity in structure between protein domains. An additional strength gained by using alignments based on structural superimpositions is that even domains where the sequences differ a lot can be aligned, cases where sequence-based approaches may fail to find similarities to base the alignment on.

The main weakness of this approach is the lack of good multiple structure superimposition methods/programs, meaning that the alignments we use as a base for the saHMMs still not are optimal. Most programs for structural comparison only work with two structures at a time to do pair-wise superimpositions. And even though some programs do excellent structural superimpositions using rigid bodies, they often fail to deduce the sequence alignments. In our implementation we trust on STAMP to give a good multiple sequence alignment, based on the structures. Even though STAMP is nor perfect, the results we get are quite good, with relatively low false positive rate.

Some other program, existing or not yet implemented, could produce more accurate alignments. An alternative could be to use different programs for different families, depending on the characteristics of the family and/or program. Another alternative would be to construct the structure anchored alignments manually or semi-manually, starting with an automatically derived superimposition and manually determining which residues to align. But such an approach would be VERY time consuming and require expert knowledge of the proteins in question, and therefore is not a realistic alternative.

There are reports that also superposition of structures may give ambiguous alignments [25], that distinct alignments can be generated that are identical in the number of residues aligned and in the RMSD of the superposition. This indicates that great care has to be taken in the choice of target function to optimize, and that perhaps also other parameters than purely geometric ones should be considered.

To be able to construct good sequence alignments based on structural superposition is an area where more work has to be done.

In our implementation we use the SCOP classification (see Section 6.6) to divide protein domains into families. SCOP is a highly reliable classification, since it is manually curated by experts, and therefore is a very good base for the HMMs. However, this is also the drawback of using this database. The existence of the database relies on a few people, and the inclusion of new protein structures in the database cannot be done immediately. There exist some au-

tomatically created databases, but the exact classification of domains depends on the method used.

We chose to use the family level in the SCOP classification as the base for our HMMs. The results might be better if one works on the superfamily level, especially by pooling the result of all families in the same superfamily. In the work of Gough et al. [30], they show that constructing many HMMs representing the same superfamily, and combining their results, give higher coverage than an HMM constructed from all members of the superfamily.

The structure base sequence alignment can be extended to include sequences without known structures, that are similar in sequence to one or more of the individual proteins the alignment is built from. To construct a larger structure anchored multiple sequence alignment in this way would give the HMM more family specific information, and decrease the need for general prior information. This was discussed in Section 7.1, but has not been implemented yet. To implement the addition of more sequences there are some questions to be answered, and some decisions to be made. For example, if one sequence is aligned differently to two members of the family, which alignment should be used? How many extra sequences should be included in the alignment? How much can the number of extra sequences differ between members in a family without introducing too much bias? How similar to a family member or to each other can the extra sequences be, to be useful? These questions remain to be answered.

Of the all the new protein structures released in the PDB (Section 6.2) in 1998, 92.6% were structurally similar to known folds and two thirds of these had related functions as well. Of those proteins that had no clear sequence similarity to any other protein with known structure, 75% had high structural similarity to previously known folds, and almost 50% of these had related functions [48]. Since the number of new structures grow all the time, these numbers can be expected to have increased since the study was done, and continue to rise as the structural genomics projects proceed. Only 26% of the new structures had no clear sequence similarity to already known structures. We were able to construct saHMMs for almost half of the families in SCOP, but since the number of structures in each protein family are likely to increase rapidly, this number is expected to increase as the number of structures known for each protein family grows. This will make the saHMMs cover more of the protein space, and as more structures become known, more members can be included in the models and make them even better.

To increase the performance of the HMMs the parameters in HMMER2.0 might have to be optimized. For example, in these kind of experiments, an HMM designed to find a global match to the HMM locally in the sequence might be better suited. Or more or less or different prior information should be used to best take care of the structural information in the structure anchored sequence alignments and the careful selection of only very distant representatives.

The main work to be done now is first to study the alignments the saH-

MMs are based on, and relate these to the results of the individual saHMMs. Then the saHMMs should be compared to other currently available methods for recognition of distant relationships, and benchmark the performances.

To increase the sensitivity of the saHMMs, the inclusion of more sequences using for example BLAST as discussed above, should be implemented.

Within a near future, the method should be available as a server, where it is possible to submit a sequence and get a family assignment back, if a relationship can be found. The performance and reliability of such a server is expected to increase as more structures become known, and as the saHMMs are optimized. Also, the database of saHMMs the server relies upon should be automatically updated as new releases of SCOP become available.

Since the saHMMs are based on structure anchored sequence alignments, it could be possible to assign a secondary structure to a query sequence. If a match is found between a query and an saHMM modeling a certain family, the query can be aligned to the family members (representatives) the HMM is based upon using the saHMM. The structures of all representatives are known, meaning that such an alignment would give a clue about the secondary structures of the query sequence. If the representatives all have an alpha helix at a certain position, then the corresponding residues in the query should form an alpha helix too. This kind of secondary structure predictions remain to be implemented and evaluated.

12 Acknowledgements

This research was conducted using the resources of High Performance Computing Center North (HPC2N).

References

[1] B. Al-Lazikani, F. B. Sheinerman, and B. Honig. Combining multiple structure and sequence alignments to improve sequence detection and alignment: Application to the SH2 domains of Janus kinases. *PNAS*, 98(26):14796–14801, 2001.

[2] N. N. Alexandrov and D. Fischer. Analysis of topological and nontopological structural similarities in the PDB: New examples with old structures. *Proteins: Structure, Fiunction and Genetics*, 25:354–365, 1996.

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

[4] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[5] M. A. Andrade and C. Sander. Bioinformatics: from genome data to biological knowledge. *Current Opinion in Biotechnology*, 8:675–683, 1997.

[6] S. Balaji, S. Sujatha, S. Sai Chetan Kumar, and N. Srinivasan. PALI - a database of Phylogeny and ALIgnment of homologous protein structures. *Nucleic Acids Research*, 29:61–65, 2001.

[7] P. Baldi and S. Brunak. *Bioinformatics - the machine learning approach*. The MIT Press, Cambridge, Massachusetts, 1999.

[8] C. Barrett, R. Hughey, and K. Karplus. Scoring Hidden Markov Models. *CABIOS*, 13(2):191–199, 1997.

[9] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, S. Griffiths-Jones, K. L. Howe, M. Marshall, and E. L. L. Sonnhammer. The pfam protein families database. *Nucleic Acids Research*, 30:276–280, 2002.

[10] A. Bateman, E. Birney, R. Durbin, S. R. Eddy, K. L. Howe, and E. L. L. Sonnhammer. The Pfam Protein Families Database. *Nucleic Acids Research*, 28:263–266, 2000.

[11] A. D. Baxevanis. The molecular biology database collection: 2003 update. *Nucleic Acids Research*, 31(1):1–12, 2003.

[12] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[13] J. D. Blake and F. E. Cohen. Pairwise sequence alignment below the twilight zone. *Journal of Molecular Biology*, 307:721–735, 2001.

[14] S. E. Brenner, C. Chothia, and T. J. P. Hubbard. Population statistics of protein structures: lessons from structural classifications. *Current Opinion in Structural Biology*, 7:369–376, 1997.

[15] S. E. Brenner, P. Koehl, and M. Levitt. The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Research*, 28(1):254–256, 2000.

[16] G. Casari, A. De Daruvar, C. Sander, and R. Schneider. Bioinformatics and the discovery of gene function. *Trends in Genetics*, 12(7):244–245, 1996.

[17] C. Chothia and A. M. Lesk. The relation between the divergence of sequence and structure in proteins. *The EMBO Journal*, 5(4):823–826, 1986.

[18] F. Corpet, F. Servant, J. Gouzy, and D. Kahn. ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons. *Nucleic Acids Research*, 28(1):267–269, 2000.

[19] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. *Atlas of Protein Sequence and Structure*, volume 5, chapter 22, pages 345–352. Natl. Biomed. Res. Found., Washington, 1978.

[20] K. Diederichs. Structural superposition of proteins with unknown alignment and detection of topological similarity using a six-dimensional search algorithm. *Proteins: Structure, Function and Genetics*, 23:187–195, 1995.

[21] R. Durbin, S. Eddy, A. Krogh, and G Mitchison. *Biological Sequence Snalysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[22] A. Elofsson and E. L. L. Sonnhammer. A comparison of sequence and structure protein domain families as a basis for structural genomics. *Bioinformatics*, 15(6):480–500, 1999.

[23] A. Falicov and F. E. Cohen. A surface of minimum area metric for the structural comparison of proteins. *Journal of Molecular Biology*, 258:871–892, 1996.

[24] D.-F. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987.

[25] Z.-K. Feng and M. J. Sippl. Optimum superimposition of protein structures: ambiguities and implications. *Folding and Design*, 1:123–132, 1996.

[26] M. Gerstein and M. Levitt. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science*, 7:445–456, 1998.

[27] J.-F. Gibrat, T. Madej, and S. H. Bryant. Surprising similarities in structure comparison. *Current Opinion in Structural Biology*, 6:377–385, 1996.

[28] T. V. Gnanasekaran, S. Peri, A. Arockiasamy, and S. Krishnaswamy. Profiles from structure based sequence alignment of porins can identify ß stranded integral membrane proteins. *Bioinformatics*, 16(9):839–842, 2000.

[29] O. Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162:705–708, 1982.

[30] J. Gough, K. Karplus, R. Hughey, and C. Chothia. Assignment of Homology to Genome Sequences using a Library of Hidden Markov Models that Represent all Proteins of Known Structures. *Journal of Molecular Biology*, 313:903–919, 2001.

[31] V. S. Gowri, S. B. Pandit, P. S. Karthik, N. Srinivasan, and S. Balaji. Integration of related sequences with protein three-dimensional structural families in an updated version of PALI database. *Nucleic Acids Research*, 31:486–488, 2003.

[32] M. Gribskov, A. D. McLachlan, and D. Eisenberg. Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci.USA*, 84:4355–4358, 1987.

[33] S. Griffiths-Jones and A. Bateman. The use of structure information to increase alignment accuracy does not aid homologue detection with profile HMMs. *Bioinformatics*, 18(9):1243–1249, 2002.

[34] S. K. Gupta, J. D. Kececioglu, and A. A. Shäffer. Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *Journal of Computational Biology*, 2:459–472, 1995.

[35] J. Hargbo and A. Elofsson. Hidden markov models that use predicted secondary structures for fold recognition. *Proteins: Structure, Function, and Genetics*, 36:68–76, 1999.

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

[37] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89:10915–10919, 1992.

[38] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233:123–138, 1993.

[39] L. Holm and C. Sander. Searching protein structure databases has come of age. *Proteins: Structure, Function and Genetics*, 19:165–173, 1994.

[40] L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595–602, 1996.

[41] L. Holm and C. Sander. Dali/fssp classification of three-dimensional protein folds. *Nucleic Acids Research*, 25(1):231–234, 1997.

[42] X. Huang and W. Miller. A time-efficient, linear-space local similarity algorithm. *Advan. Appl. Math.*, 12:337–357, 1991.

[43] J. Jung and B. Lee. Protein structure alignment using environmental profiles. *Protein Engineering*, 13(8):535–543, 2000.

[44] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, A32:922–923, 1976.

[45] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, A34:827–828, 1978.

[46] L. A. Kelley, R. M. MacCallum, and M. J. E. Sternberg. Enhanced Genome Annotation Using Structural Profiles in the Program 3D-PSSM. *Journal of Molecular Biology*, 299(2):499–520, 2000.

[47] P. Koehl. Protein structure similarities. *Current Opinion in Structural Biology*, 11:348–353, 2001.

[48] W. A. Koppensteiner, P. Lackner, M. Wiederstein, and M. J. Sippl. Characterization of novel proteins based on known protein structures. *Journal of Molecular Biology*, 296:1139–1152, 2000.

[49] N. Leibowitz, Z. Y. Fligelman, R. Nussinov, and H. J. Wolfson. Automated multiple structure alignment and detection of a common substructural motif. *Proteins: Structure, Function and Genetics*, 43:235–245, 2001.

[50] E. Lindahl and A. Elofsson. Identification of related proteins on family, superfamily and fold level. *Journal of Molecular Biology*, 295:613–625, 2000.

[51] R. Lüthy, A. D. McLachlan, and D. Eisenberg. Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities. *Proteins: Structure, Function and Genetics*, 10:229–239, 1991.

[52] G. Lu. An automated approach for multiple alignment of protein structures. Manuscript, 1998.

[53] G. Lu. TOP: a new method for protein structure comparisons and similarity searches. *Journal of Applied Crystallography*, 33:176–183, 2000.

[54] T. Madej, J.-F. Gibrat, and S. H. Bryant. Threading a database of protein cores. *Proteins: Structure, Function and Genetics*, 23:356–369, 1995.

[55] M. Madera and J. Gough. A comparison of profile hidden Markov model procedures for remote homology detection. *Nucleic Acids Research*, 30(19):4321–4328, 2002.

[56] K. Mizuguchi, C. M. Deane, T. L. Blundell, and J. P. Overington. HOMSTRAD: A database of protein structure alignments for homologous families. *Protein Science*, 7:2469–2471, 1998.

[57] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

[58] S. B. Needleman and C. D Wunch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[59] C. Notredame, D. G. Higgins, and J. Heringa. T-COFFEE: A Novel Method for Fast and Accurate Multiple Sequence Alignment. *Journal of Molecular Biology*, 302:205–217, 2000.

[60] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH - a hierarchic classification of protein domain structures. *Structure*, 5:1093–1108, 1997.

[61] J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *Journal of Molecular Biology*, 284:1201–1210, 1998.

[62] W. R. Pearson. Searching protein sequence libraries: comparison of the sensitivity and selectivity of the smith-waterman and fasta algorithms. *Genomics*, 11:635–650, 1991.

[63] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, 85:2444–2448, 1988.

[64] D. W. Rice and D. Eisenberg. A 3D-1D Substitution Matrix for Protein Fold Recognition that Includes Predicted Secondary Structure of the Sequence. *Journal of Molecular Biology*, 267:1026–1038, 1997.

[65] B. Rost. Twilight zone of protein sequence alignments. *Protein Engineering*, 12:85–94, 1999.

[66] R. B. Russel and G. J. Barton. Multiple protein sequence alignment from tertiary structure comparison: Assignment of global and residue confidence levels. *Proteins: Structure, Function and Genetics*, 14:309–323, 1992.

[67] N. Saitou and M. Nei. *Mol. Biol. Evol.*, 4:406–425, 1987.

[68] A. Sali and T. L. Blundell. Definition of general topological equivalence in protein structures: A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. *Journal of Molecular Biology*, 212:403–428, 1990.

[69] J. Shi, T. L. Blundell, and K. Mizuguchi. FUGUE: Sequence-structure Homology Recognition Using Environment-specific Substitution Tables and Structure-dependent Gap Penalties. *Journal of Molecular Biology*, 310:243–257, 2001.

[70] E. S. C. Shih and M.-J. Hwang. Protein structure comparison by probability-based matching of secondary structure elements. *Bioinformatics*, 19(6):735–741, 2003.

[71] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11(9):739–747, 1998.

[72] A. P. Singh and D. L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology, Halkidiki, Greece, June 21-26, 1997*, volume 5, pages 284–293. ISMB, AAAI, 1997.

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

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

[75] J. D. Szustakowski and Z. Weng. Protein structure alignment using a genetic algorithm. *Proteins: Structure, Function and Genetics*, 38:428–440, 2000.

[76] W. R. Taylor, T. P. Flores, and C. A. Orengo. Multiple protein structure alignment. *Protein Science*, 3:1858–1870, 1994.

[77] W. R. Taylor and C. A. Orengo. A holistic approach to protein structure alignment. *Protein Engineering*, 2(7):505–519, 1989.

[78] W. R. Taylor and C. A. Orengo. Protein structure alignment. *Journal of Molecular Biology*, 208:1–22, 1989.

[79] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix chioce. *Nucleic Acids Research*, 22(22):4673–46800, 1994.

[80] J. D. Thompson, F. Plewniak, and O. Poch. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15:87–88, 1999.

[81] J. Tångrot, B. Kågström, and U. H. Sauer. Fishing for protein sequences in the midnight zone. *High Performance Computing Center North (HPC2N) progress report - 2001*, pages 6–7, 2002.

[82] C. A. Wilson, J. Kreychman, and M. Gerstein. Assessing annotation transfer for genomics: Quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. *Journal of Molecular Biology*, 297:233–249, 2000.

[83] T. D. Wu, S. C. Schmidler, T. Hastie, and D. L. Brutlag. Regression analysis of multiple protein structures. *Journal of Computational Biology*, 5(3):585–595, 1998.

[84] A.-S. Yang and B. Honig. An Integrated Approach to the Analysis and Modeling of Protein Sequences and Structures. I. Protein Structural Alignment and a Quantitative Measure for Protein Structural Distance. *Journal of Molecular Biology*, 301:665–678, 2000.

[85] A.-S. Yang and B. Honig. An Integrated Approach to the Analysis and Modeling of Protein Sequences and Structures. III. A Comparative Study of Sequence Conservation in Protein Structural Families using Multiple Structural Alignments. *Journal of Molecular Biology*, 301:691–711, 2000.

A Appendix

URL's related to

- Databases:
  - Biology WorkBench - http://workbench.sdsc.edu/
    A web-based tool to search many protein and nucleic acid sequence databases, integrated with access to analysis and modeling tools.
  - SRS - http://srs6.ebi.ac.uk
    An interface that provides access to data stored in publicly available databases. SRS makes it easy to browse very diverse data, such as literature or biological sequences.
  - GenBank -[9]
    All known nucleotide and protein sequences.
  - EMBL Nucleotide Sequence Database - http://www.ebi.ac.uk/embl/index.html
    All known nucleotide and protein sequences.

[9] http://www.psc.edu/general/software/packages/genbank/genbank.html

  - DNA Data Bank of Japan (DDBJ) - http://www.ddbj.nig.ac.jp/
    All known nucleotide and protein sequences.
  - TrEMBL - http://www.ebi.ac.uk/trembl/index.html
    Translated EMBL. All DNA sequences stored in the EMBL data base are translated into their corresponding protein sequences.
  - Swiss-Prot - http://www.ebi.ac.uk/swissprot/
    Annotated protein sequence information.
  - PIR - http://pir.georgetown.edu/
    Functionally annotated protein sequences.
  - UniProt - http://pir.georgetown.edu/uniprot/
    A project aimed at creating a central data base of protein sequence and function.

- Classifications of proteins
  - CATH - http://www.biochem.ucl.ac.uk/bsm/cath_new/index.html
    Hierarchical classification of protein domain structures.
  - SCOP - http://scop.mrc-lmb.cam.ac.uk/scop/
    Structural Classification of Proteins. Familial and structural protein relationships.
  - PDB - http://www.rcsb.org/pdb/
    A data base containing the 3D-coordinates of all currently known macromolecular structures (predominantly protein).
  - Pfam - http://pfam.cgb.ki.se
    Multiple sequence alignments and hidden Markov models of protein domains.

- Programs and servers
  - BLAST - http://www.ncbi.nlm.nih.gov/blast/
    Server for sequence searches using BLAST, PSI-BLAST or other variants.
  - ClustalW - http://www.ebi.ac.uk/clustalw/
    Server for multiple sequence alignments.
  - T-Coffee[10] -
    The T-Coffee program itself and information related to it, links to servers that construct multiple sequence alignments using T-Coffee.
  - HMMER - http://hmmer.wustl.edu/
    The HMMER program package and information related to it.

[10] http://igs-server.cnrs-mrs.fr/~cnotred/Projects_home_page/t_coffee_home_page.html

- CE Home Page - http://cl.sdsc.edu/ce.html
  Databases and tools for comparison of protein structures using combinatorial extension.
- CL Home Page - http://cl.sdsc.edu/cl1.html
  Using the compound likeness method for finding similarities in protein structures.
- MAPS - http://bioinfo1.mbfys.lu.se/TOP/maps.html
  Description of the program MAPS (Multiple Alignment of Protein Structures) and links to download it.
- TOP - http://bioinfo1.mbfys.lu.se/TOP/top.html
  The TOP manual - a program for TOPological comparison of protein structures. The foundation for MAPS.
- MATCH3D -
  http://omega.omrf.ouhsc.edu/zhangc/programs/match3d.html
  Program to do three dimensional structure homology searches by representing a protein as a set of vectors (the secondary structures).
- DALI - http://www.ebi.ac.uk/dali/
  Comparison of protein structures.
- STAMP -
  http://www.hgmp.mrc.ac.uk/Registered/Option/stamp.html
  A server to make multiple structure alignments.

II

Paper II

Hidden Markov Models That Use Predicted Secondary Structures For Fold Recognition[†]

Jeanette Hargbo[‡] and Arne Elofsson

Department of Biochemistry, Stockholm University
Stockholm, Sweden.

Abstract

There are many proteins that share the same fold but have no clear sequence similarity. To predict the structure of these proteins, so called protein fold recognition methods have been developed. During the last few years, improvements of protein fold recognition methods have been achieved through the use of predicted secondary structures (Rice and Eisenberg, J Mol Biol 1997;267:1026 1038), as well as by using multiple sequence alignments in the form of hidden Markov models (HMM) (Karplus et al., Proteins Suppl 1997;1:134 139). To test the performance of different fold recognition methods, we have developed a rigorous benchmark where representatives for all proteins of known structure are matched against each other. Using this benchmark, we have compared the performance of automatically-created hidden Markov models with standard sequence-search methods. Further, we combine the use of predicted secondary structures and multiple sequence alignments into a combined method that performs better than methods that do not use this combination of information. Using only single sequences, the correct fold of a protein was detected for 10% of the test cases in our benchmark. Including multiple sequence information increased this number to 16%, and when predicted secondary structure information was included as well, the fold was correctly identified in 20% of the cases. Moreover, if the correct secondary structure was used, 27% of the proteins could be correctly matched to a fold. For comparison, BLAST2, FASTA, and SSEARCH identifies the fold correctly in 13 17% of the cases. Thus, standard pairwise sequence search methods perform almost as well as hidden Markov models in our benchmark. This is probably because the automatically created multiple sequence alignments used in this study do not contain enough diversity and because the current generation of hidden Markov models do not perform very well when built from a few sequences.

Keywords: protein structure; HMM; SCOP; HSSP; threading; BLAST; FASTA; SSEARCH; protein fold recognition

---

[†]With permission from *Proteins: Structure, Function and Genetics, 36 (1999), pp. 68–76.*

[‡]Now Tångrot. Present adress: Department of Computing Science *and* Umeå Center for Molecular Pathogenesis, Umeå University, 901 87 Umeå

PROTEINS: Structure, Function, and Genetics 36:68–76 (1999)

Hidden Markov Models That Use Predicted Secondary Structures For Fold Recognition

Jeanette Hargbo and Arne Elofsson*
Department of Biochemistry, Stockholm University, Stockholm, Sweden

ABSTRACT **There are many proteins that share the same fold but have no clear sequence similarity. To predict the structure of these proteins, so called "protein fold recognition methods" have been developed. During the last few years, improvements of protein fold recognition methods have been achieved through the use of predicted secondary structures (Rice and Eisenberg, J Mol Biol 1997;267:1026–1038), as well as by using multiple sequence alignments in the form of hidden Markov models (HMM) (Karplus et al., Proteins Suppl 1997;1:134–139). To test the performance of different fold recognition methods, we have developed a rigorous benchmark where representatives for all proteins of known structure are matched against each other. Using this benchmark, we have compared the performance of automatically-created hidden Markov models with standard-sequence-search methods. Further, we combine the use of predicted secondary structures and multiple sequence alignments into a combined method that performs better than methods that do not use this combination of information. Using only single sequences, the correct fold of a protein was detected for 10% of the test cases in our benchmark. Including multiple sequence information increased this number to 16%, and when predicted secondary structure information was included as well, the fold was correctly identified in 20% of the cases. Moreover, if the correct secondary structure was used, 27% of the proteins could be correctly matched to a fold. For comparison, blast2, fasta, and ssearch identifies the fold correctly in 13–17% of the cases. Thus, standard pairwise sequence search methods perform almost as well as hidden Markov models in our benchmark. This is probably because the automatically-created multiple sequence alignments used in this study do not contain enough diversity and because the current generation of hidden Markov models do not perform very well when built from a few sequences.** **Proteins 1999;36:68–76.** **1999 Wiley-Liss, Inc.**

Key words: protein structure; HMM; Scop; HSSP; threading; blast; fasta; ssearch; protein fold recognition

INTRODUCTION

The most promising method for predicting the structure of a protein is to identify a protein with a known structure that shares the same fold. Traditionally, this has been done by identifying proteins that have similar sequences. However, of late, many examples of structures that have similar folds but no detectable sequence similarity have been found,. This has led to the development of methods to detect the fold of a probe sequence from a library of known target folds. These methods are often referred to as fold recognition methods.

Fold recognition methods can roughly be divided into three different types, based on the type of information that they use. Within each category there are many different implementations. The three types of methods are sequence-based methods,[1,2] structure-based methods,[3,4] and prediction-based methods.[5–9,10] In this study, we introduce a new method that combines multiple-sequence-alignment methods with predicted secondary structure information. We also compare the performance of hidden Markov models with standard sequence-based methods. All these comparisons are made with a more rigorous benchmark than those used in most earlier studies.

Sequence-based methods are the oldest methods for fold recognition.[11] It seems a bit surprising that sequence-based methods are able to detect a similar fold of proteins that show no sequence similarity, but the amino-acid sequence contains much information about the physical environment at each position in the sequence. Thus, even if there is no detectable sequence similarity between two proteins that have the same fold, the corresponding positions in the proteins will have similar properties. Moreover, there are many examples where there is no obvious sequence similarity, but where two proteins clearly are homologous. Of course, these targets might be detected with improved sequence-based methods. One way to increase the performance of sequence-based methods is to use information from a family of sequences, instead of from just one sequence. With the inclusion of multiple sequence alignment information and modern computational methods, such as hidden Markov models, sequence-based methods have proven to be successful in fold recognition.[2]

Abbreviations: Scop, a structural classification of proteins database; HMM, Hidden Markov Model; ssHMM, hidden Markov models that use secondary structure information; predHMM, hidden Markov models that use secondary structure information with predicted secondary structures.

*Correspondence to: Arne Elofsson, Department of Biochemistry, Stockholm University, 106 91 Stockholm, Sweden. E-mail: arne@biokemi.su.se

Received 16 September 1998; Accepted 1 March 1999

A hidden Markov model (HMM), or more correctly a profile-HMM, is a generalized version of a profile that is mathematically more consistent. A general description of HMMs (applied in speech recognition, where they were originally used) has been written by Rabiner and Juang.[12] In biology, HMMs have been used in many different areas, such as gene prediction,[13] membrane protein prediction,[14] and protein sequence comparisons.[1,2] One major difference between profile-HMMs and a profile is that in a profile the penalty for gaps or insertions are the same in every position of the alignment, even though some regions are more variable than others. Ideally, these regions should have a smaller penalty for gaps than more conserved areas. In the HMM, the penalties are position-dependent, and are learned from the training data.

An alternative type of information has been used in the structure-based fold recognition methods. These methods do not use sequence information to determine if two proteins have the same fold or not. Instead, they use an energy function that describes how well a probe sequence matches a target fold. The energy function is often obtained from a database of known protein structures, and can be used, for instance, to describe the environment of each residue[15] or the probability of finding two residues at a certain distance from each other.[3,4]

Proteins having a similar fold also have similar secondary structures, so that even though the amino acid sequences may have changed a great deal during evolution, the secondary structure will still be the same for related proteins belonging to the same fold. Today, the secondary structure can be predicted from the amino acid sequence with an accuracy of more than 70%.[16] Several approaches attempt to use this information, in addition to the amino acid sequence, to recognize the correct fold.[5,6,9] Fischer and Eisenberg[5] align a probe sequence to known folds and then calculate the probability of the protein having a certain fold. The score for an aligned amino acid normally depends on how likely it is to have that particular amino acid in that position in the fold, but Fischer and Eisenberg also take the predicted secondary structure into account, increasing the score if it fits the secondary structure of the fold and decreasing the score otherwise. The addition of the secondary structure information seems to help significantly in recognizing the correct fold, indicating that, even though the predicted secondary structure is not completely correct, it still contains a lot of useful information that could complement other information.

Usually, a HMM only uses the amino acid sequence when modeling a protein family, making very distant homologues difficult to recognize. The aim of this work is to create a HMM that uses the predicted secondary structure in addition to the primary sequence. By combining the information from both sequence and secondary structure, it should be possible to recognize even distant or non-homologous proteins that share a similar fold. The idea of using secondary structure predictions and multiple sequence information HMMs has been proposed earlier but not tested in this type of benchmark.[8,17] In addition, our implementation of this approach differs from earlier attempts.

MATERIALS AND METHODS

An Implementation of HMMs Using Secondary Structure Information

The program package HMMER, version 1.8.4,[18] was modified to include secondary structure information when building a hidden Markov model (HMM) of a protein family, as well as when matching an amino acid sequence to an HMM. The secondary structure HMMs (ssHMMs) are models of protein families based both on amino acid sequence and on secondary structure information.

Ordinary profile HMMs consist of a sequence of match states, analogous to positions in a multiple sequence alignment, and corresponding insert and delete states. To each insert and match state a probability distribution over all amino acids is associated, these distributions giving the probability of a certain amino acid, given that particular state. The parameters of the model are the probabilities for transitions between states and the amino acid probability distributions, and these are optimized so that all sequences belonging to the modeled family obtain high probabilities and all other sequences low. Thus a sequence $s = x_1 \ldots x_L$ following the path $q = q_0 \ldots q_{N+1}$ through model μ has the probability

$$P(s|q, \mu) = \prod_{i=1}^{N+1} T(q_i|q_{i-1}) \prod_{i=1}^{N} P(x_{l(i)}|q_i) \quad (1)$$

where $T(q_i|q_{I-1})$ is the probability for a transition from state q_{I-1} to q_i and $l(I)$ is the index for amino acid x in the sequence in state q_i, $P(x_{l(I)}|q_i)$ is the probability of having amino acid $x_{l(I)}$ in state q_i, and N is the number of states in the path. The lower indexes represent the position in the path. The theory behind HMMs has been described in more detail in earlier work.[1,18,19] In comparison with sequence profiles, one of the major differences is that for each position there is a correct transition probability for each gap and insertion parameter.

The ssHMM has an extra distribution of probabilities for the secondary structures E, H, and L associated with each insert and match state. In each state, the model emits a probability for the amino acid, as before, but in addition to this it emits another probability for the secondary structure assigned to that position. In this way, the probability for the sequence is higher if the secondary structure is the same as in the modeled family. The total probability for a sequence $s = x_1 \ldots x_L$ having the secondary structure $ss = y_1 \ldots y_L$ given the path $q = q_0 \ldots q_{N+1}$ and model μ is now:

$$P(s,ss|q, \mu) = \prod_{i=1}^{N+1} T(q_i|q_{i-1}) \prod_{i=1}^{N} P(x_{l(i)}|q_i) \prod_{i=1}^{N} P(y_{m(i)}|q_i) \quad (2)$$

where $y_{m(I)}$ is the secondary structure emitted in state q_i. The emission probabilities of the secondary structures are found in the same way as the amino acid emission probabilities when training the model. The combined HMM will be referred to as a secondary structure HMM (ssHMM). The

modified HMMER program is available from http://www.biokemi.su.se/~arne/sshmm/

As the number of parameters in the model increases, additional information is needed to produce a useful model. To decrease the number of free parameters, the emission probabilities $P(x|i_k)$ for the insert states are set equal or to some background frequency. The problem with having too little information, i.e., too few training sequences, concerns fitting, i.e., a HMM created from this data will be able to recognize only proteins that are very closely related to the proteins used to create the HMM. In this situation, a prior distribution can be used, and the model is not allowed to specialize too much. However, a prior distribution assumes that any change from one amino acid to another is equally probable, which is not the case. [19] A standard HMM could be seen as building a sequence profile using an identity matrix, which certainly is not the most efficient matrix to use. The inclusion of substitution parameters into HMMER can be made through the use of a special prior distribution using a substitution matrix. The inclusion of substitution matrices are made when building the HMM by adding a partial count to all amino acid types when a certain amino acid is found in a position. This partial count is related to the probability of an amino acid having been replaced by another particular amino acid. In this study, we have used the Pam250 substitution matrix, which was included in the HMMER package. For the secondary structure counts, we were not able to create a prior distribution that significantly improved the performance. Therefore we chose not to use any. At the beginning of the training, all secondary structures are assumed to occur at equal probabilities. Thus, even if a position is found in only one secondary structure type, the other secondary structure types will also have a small probability of occurrence.

A library of ssHMMs was built from the sequences and secondary structures of a representative set of all proteins with a known structure. For a given protein, all related proteins in Swissprot were found through the HSSP database,[20] and the secondary structure was assumed to be the same for all proteins in a family. The multiple sequence alignment from HSSP, together with the secondary structure, was used to build a ssHMM, as described above. For comparison with the original HMM method, HMMs not using the secondary structure were also created, as were HMMs (and ssHMMs) using substitution matrices. These last will be referred to as HMM-pam and ssHMM-pam. Finally, another set of HMMs, ignoring multiple sequence alignments, were created. These will be referred to as HMM-single, ssHMM-single, etc. For a complete description of all HMMs built see Table I.

To match a protein against a library of HMMs, a query sequence is matched against all HMMs. We examined the four different alignment algorithms included in HMMER local, global, endsfree, and fragmentary matches. However, in all cases, the hmms program that uses a global alignment algorithm performed best, and only results using this algorithm were evaluated in this study. When a protein is matched against a ssHMM it is necessary to assume the secondary structure of the protein; this was done in two different ways. First, the correct secondary structure was used. Second, the secondary structure predicted by predator[21] was used. The tests using the predicted secondary structures are referred to as predHMM etc. (see Table I). The rather mediocre performance of 68% was probably due to the fact that 45% of the sequences in our database had 10 or fewer homologous sequences in HSSP. For comparison with the standard sequence search methods we have used blast2,[22] fasta,[23] and ssearch[23] on our benchmark. These methods were used with default parameters, and the scoring has been done by using the expectation-values.

TABLE I. Description of Information Used in Methods Studied†

Name	SS in HMM	Query True SS	Query Pred SS	Substitution matrix	MSA
HMM					X
predHMM	X		X		X
ssHMM	X	X			X
HMM-single					
predHMM-single	X		X		
ssHMM-single	X	X			
HMM-pam				X	X
predHMM-pam	X		X	X	X
ssHMM-pam	X	X		X	X
HMM-pam-single				X	
predHMM-pam-single	X		X	X	
ssHMM-pam-single	X	X		X	
blast2				X	
fasta				X	
ssearch				X	

†SS in HMM, secondary structure in the HMM; Query True SS, correct secondary structure in query sequence; Query Pred SS, predicted secondary structure in query; MSA, multiple sequence alignment.

Measuring the Performance

To compare the performance of different fold recognition methods, it is of great importance to use a large and well-crafted benchmark. Several recent studies[6,24,25] have shown that a useful benchmark can be created using Scop[26] as a standard for classifying proteins into families of similar fold or of evolutionary relationship. Scop is a database in which all known protein structures are classified into a hierarchical classification: class, fold, superfamily, and family. In this study we have focused on proteins that have the same fold but belong to different families, according to Scop. Two proteins that are classified into to the same fold have the same secondary structure elements in a similar topological arrangement, while two proteins that belong to the same family have a clear common evolutionary origin. Two proteins classified into the same fold but to different families might belong to the same superfamily or they might not.

We created a benchmark from the pdb40 dataset of Scop version 1.37. This dataset contains a subset of Scop where

no proteins have more than 40% sequence identity to any other member of the dataset.[25] However, this dataset did not completely match the latest release of HSSP in that (1) HSSP was created from another subset of pdb and (2) the proteins in Scop are divided into domains, whereas proteins in HSSP are not. To overcome this problem, we matched each sequence in pdb40 to the HSSP database and replaced the sequence with the HSSP sequence if the match had a significance better than 1.e-5 using fasta, and if the alignment produced was of the same length as the original sequence. Using this procedure, 1,130 out of 1,272 sequences in pdb40 were retained. This procedure removed all Scop entries of the "non-proteins" class and many of the peptides, as they were not present in HSSP. For each of the 1,130 sequences, the multiple sequence alignment and the secondary structure were read from HSSP. On average, 26 sequences were included in a sequence family. However, many of these sequences were identical or almost identical to the original sequence. This dataset of sequences and multiple sequence alignments is available from http://www.biokemi.su.se/~arne/sshmm/

In our benchmark, see Figure 1, all proteins were matched to the HMMs of all other proteins, and for each pair the folds and families (according to Scop) were recorded. As the family classification in Scop is a subclassification of a fold, two proteins can belong either to the same family, to two different families but to the same fold, or to two different folds. If the two proteins belong to the same family, we have eliminated them from further consideration, because this indicates that they are homologous and thereby not a good test of fold recognition methods. If the fold, but not the family, of the two entries is the same, the match was considered to be a *true* match, while if the two entries belong to different folds they were considered to be a *false* match. To create a good benchmark it is necessary to have a large and complete set of proteins; in our benchmark set there are 730 proteins that have at least one true match, i.e., there are 400 proteins in the database that do not have any true match. These 400 entries were retained, because they provided potentially important information about false matches. The total number of true hits is 8,312, and there are more than 1.2 million false hits in the benchmark (see Table II). The benchmark includes proteins from 359 different folds and 666 different families in Scop. We believe that this benchmark contains a significant fraction of all possible targets for fold-recognition.

TABLE II. Description of the Benchmark

Data	Number of data points
Protein domains in pdb40	1,272
Protein domains both in HSSP and in pdb40	1,130
Protein domains with at least one true match (another domain from the same fold but from another family)	730
Number of pairwise comparisons	1,273,618
True matches (protein domains from the same fold but different families)	8,312
False matches (protein domains from different folds and families)	1,265,306
Number of different protein families	666
Number of different protein folds	359

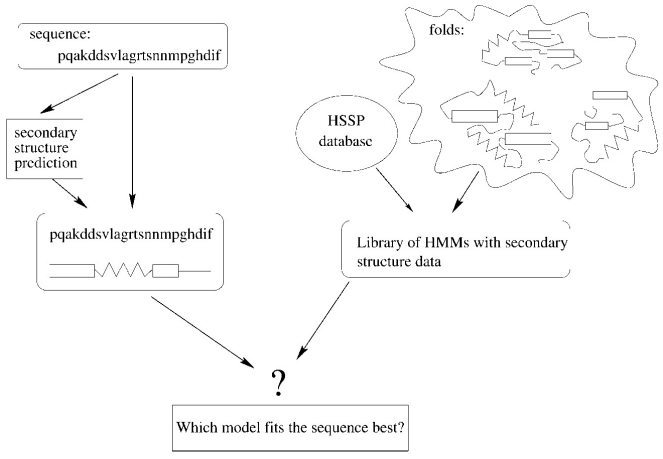


Fig. 1. A schematic description of the ssHMMs. First a library of representative folds is created, second, all homologous sequences of these proteins are found. These multiple sequence alignments, together with the secondary structures of the representative proteins, are used to construct the library. For the probe sequence, a secondary structure prediction is performed. Finally, the sequence with the predicted secondary structure is probed against all folds in the fold library.

We have used two different criteria to analyze the performance of a fold recognition method on our benchmark. First, we simply examined at what rank the first true hit was found. This is a very intuitive measure, however, and it does not measure the reliablity of a match of a certain score. For some proteins there are several possible correct hits and with this measure the first match could be to any one of these proteins, while for others there is only a single match. Second, as a complementary measure, we have used specificity-sensitivity plots, or spec-sens plots, as in Rice and Eisenberg.[6] The main advantage of this method is that it describes the ability of a method to find all pairwise matches in the benchmark. The sensitivity is based on the model's ability to find all members of the same fold. In other words:

$$\mathrm{SENS}(score) = \mathrm{TP}(score)/(\mathrm{TP}(score) + \mathrm{TN}(score)) \quad (3)$$

where TP(*score*) is the number of true hits that have a score above *score*, and TN(*score*) is the number of true hits with a score less than *score*. The specificity measures the probability that a pair of sequences with a score greater than a certain threshold really belong to the same fold. The specificity is defined as:

$$\mathrm{SPEC}(score) = \mathrm{TP}(score)/(\mathrm{TP}(score) + \mathrm{FP}(score)) \quad (4)$$

where FP(*score*) is the number of false hits that have a score above *score* and TP is defined as above. The sensitivity is plotted as a function of specificity, each point in the plot corresponding to a certain score. One difference between our two measures is that the spec-sens curves represent a method's ability to recognize all proteins from the same fold (but from different families), while the simple counting method measures the ability of a method to identify any member of the same fold (but from another family).

RESULTS AND DISCUSSION

Every two years there is a community-wide effort, CASP, to analyze protein structure prediction methods by blind predictions, allowing predictors to "guess" the structure of soon-to-be solved protein structures.[27] At the second CASP process in 1996, five groups were selected for the best performance in the threading category. One of these groups used predicted secondary structures,[7] another group used hidden Markov models (HMM),[2] a third group used a hidden Markov model that only used secondary structure and matched a predicted secondary structure against this model.[8] The last two groups[4,28] used either human expert knowledge or a physical energy function in their threading studies. The success of using HMMs and the idea of using predicted secondary structures makes it a natural step to try to combine these two methods, as we have done in this study.

This study is based on matching all proteins in our test set against all other proteins of the test set. Each protein is classified as belonging to a protein family and as having a certain fold, according to Scop.[26] The Scop classification is hierarchical, i.e., a fold is a superset of one or several families, and thus two proteins might belong to the same fold but to different families. Two proteins from the same fold, but from different families, are not assumed to be homologous but still have a similar structure. A match between two proteins is ignored if the two proteins belong to the same family, it is considered as a true match if the proteins belong to different families but to the same fold, and it is considered to be a false match if the proteins belong to different folds. Using this benchmark, we have compared the performance of the newly developed ssHMMs, standard HMMs, and pairwise sequence comparisons methods.

Secondary Structure Increases the Performance of HMMs

Earlier studies showed that including predicted secondary structure sequence into single sequence-based search methods increased the performance significantly.[5,6,9] Therefore, we believed that the same would be true for hidden Markov models. In Figure 2 it can be seen that our assumption are apparently correct, as the sensitivity of a hidden Markov model is increased when the secondary structure is included. For instance, at a specificity of 5%, the sensitivity increases from 2% to 30% if the true

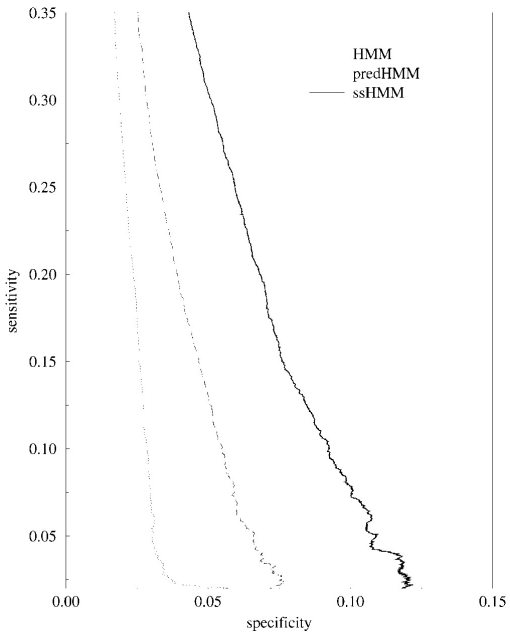


Fig. 2. A specificity versus sensitivity plot of HMM, predHMM, and ssHMM. It can be seen that the sensitivity increases when predicted or true secondary structure information is included.

TABLE III. Sensitivity of Methods at Specificity = 5% and 10%

Name	Spec = 5%	Spec = 10%
HMM	2%	1%
predHMM	13%	1%
ssHMM	30%	8%
HMM-single	0%	0%
predHMM-single	6%	0%
ssHMM-single	17%	0%
HMM-pam	11%	6%
predHMM-pam	11%	2%
ssHMM-pam	26%	7%
HMM-pam-single	8%	2%
predHMM-pam-single	17%	5%
ssHMM-pam-single	24%	11%
Blast2	3%	2%
Fasta	5%	3%
ssearch	13%	6%

TABLE IV. Fraction of Possible True Hits Placed at Ranks 1, 5, 10, and 25

Name	#1	#5	#10	#25
HMM	12%	24%	32%	45%
predHMM	19%	38%	47%	59%
ssHMM	30%	49%	59%	69%
HMM-single	4%	15%	24%	38%
predHMM-single	10%	29%	38%	51%
ssHMM-single	14%	34%	44%	56%
HMM-pam	16%	30%	40%	51%
predHMM-pam	20%	36%	45%	57%
ssHMM-pam	27%	48%	56%	67%
HMM-pam-single	10%	22%	31%	44%
predHMM-pam-single	17%	35%	44%	55%
ssHMM-pam-single	21%	39%	48%	60%
Blast2	17%	30%	37%	48%
Fasta	13%	25%	37%	43%
ssearch	17%	25%	30%	40%

secondary structure is used and to 13% if the predicted secondary structure is used (Table III). The fraction of the possible hits that were ranked in first place is increased as well, from 12% to 30% when using the secondary structure, and to 19% if the predicted secondary structure is used (Table IV). The increase in performance is similar to that reported for single sequence-based methods; for instance, Fischer and Eisenberg increased the fraction of hits found in first rank from 54% to 65% by using predicted secondary structures and the BLOSUM62 matrix.[29] In the study by Rice and Eisenberg, the sensitivity increased from approximately 15% to 30% when predicted secondary structures were used at 5% specificity.

It should also be noted that our benchmark seems significantly more difficult than the benchmark used by Fisher and Eisenberg, as they were able to detect 54% of the proteins in first place using sequence alignment methods, while we were able to detect only 17%. The difficulty of the benchmark used by Rice and Eisenberg seems to be similar to the difficulty of ours.

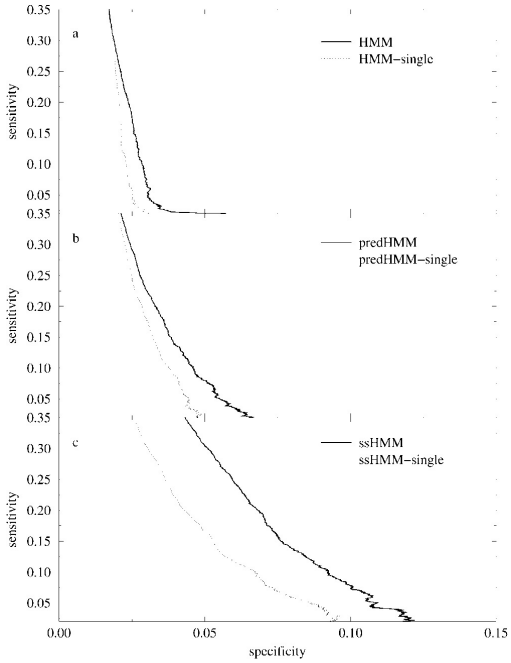


Fig. 3. When multiple sequence alignment is used (bold lines) the sensitivity of the hidden Markov models is increased, compared to using only single sequence alignments. In (**a**) standard HMMs are used, in (**b**) predHMMs, and in (**c**) ssHMMs.

Using Multiple Sequence Information Increases the Performance of HMMs

It has been assumed that using multiple sequences improves the performance of sequence-based search methods. However to our knowledge, there has been no studies showing that this is in fact true, using as complete benchmark as the one we have used here. Figure 3 shows that the sensitivity at a given specificity is increased for models built from multiple sequences compared to models built from just one sequence. This is most obvious for the ssHMMs, where at a specificity of 5%, the sensitivity increases from 17% to 30% when using multiple sequences to build the ssHMMs, compared to single sequences. A clear increase can also be seen for ordinary HMMs, and when using predicted secondary structures. The number of sequences placed at rank one is more than doubled when building models from multiple sequence alignments. They increase from 14% to 30% for the ssHMMs, from 10% to 19% using predHMMs, and from 4% to 12% for the ordinary HMMs (Table IV). It should be remembered that when using multiple sequence alignments we have used only automatically-created alignments from HSSP, and for many proteins these alignments do not contain enough

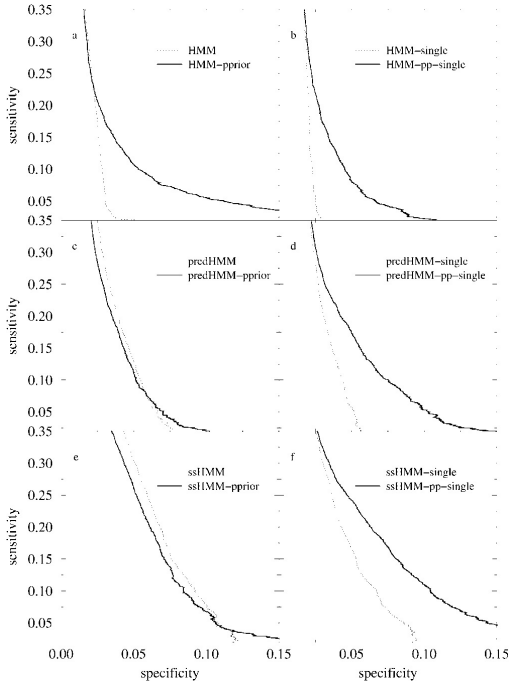


Fig. 4. The specificity is increased when a substitution matrix is used (bold lines). In (**b,d,f**) HMMs created from single sequences are used, while in (**a,c,e**) multiple sequence HMMs are used. In (a,b) standard HMMs are used, in (c,d) predHMMs, and in (e,f) ssHMMs.

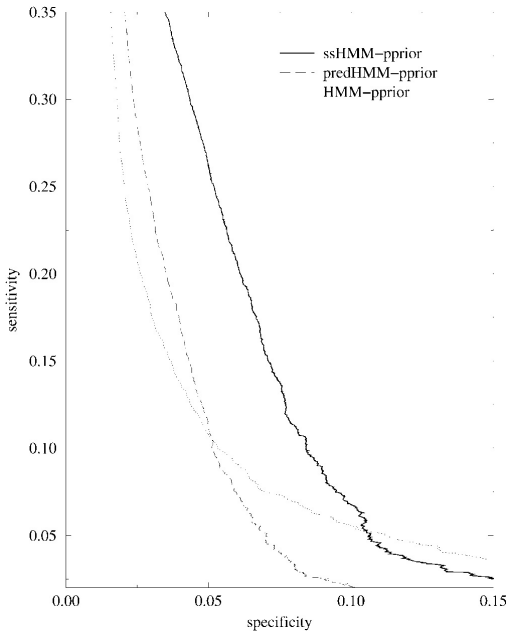


Fig. 5. A final comparison of HMMs that use multiple sequence information as well as substitution matrices. It can be noted that at higher specificities the sensitivity is lower for the ssHMMs and predHMMs than for the HMMs that do not use secondary structure information.

diversity to perform as well as HMMs created from a more diverse set of sequences.

Using a Substitution Matrix Increases the Performance of HMMs

A standard hidden Markov model does not include any information about which substitutions are most likely, i.e., a substitution matrix is not used. If the protein family is large enough and diverse enough this should not be a problem. However, in our benchmark, we have many small families with low diversity. By including a substitution matrix we attempted to overcome this problem. As can be seen in Figure 4a,b, the use of a substitution matrix when building the models increased the sensitivity significantly. For hidden Markov models built from multiple sequence alignments, the sensitivity increases from 2% to 11%, at a specificity of 5%, when using the substitution matrix. When comparing Figures 4a and 3a, and Figures 4a and 4c, it can be seen that the use of a substitution matrix helps more than the use of multiple sequence alignments.

In Figure 4d,f, it can be seen that the ssHMMs and predHMMs built from single sequences have higher sensitivities when using a substitution matrix than when not. However, for the ssHMMs built from multiple sequence alignments, using a substitution matrix does not seem to improve the performance. On the contrary, the sensitivity decreases from 30% to 26% when the substitution matrix is added to the ssHMMs (Fig. 4e, Table III). This indicates that the prior distribution might not be optimized for the secondary structure HMMs. For these, another prior, where the secondary structure is included, could be used.

When Creating a Hidden Markov Model It Is Best to Use Multiple Sequence Alignments and Substitution Matrices

From the previous results it was concluded that the use of multiple sequence alignments and substitution matrices give the best results. A comparison between the HMM-pam methods with or without secondary structure information can be seen in Figure 5. At a low specificity ($<5\%$ for predHMM and $<10\%$ for ssHMM), the secondary structure HMMs have a higher sensitivity than the ordinary HMMs. For higher specificities, however, the ordinary HMMs have a higher sensitivity. One possible explanation of this is that the ssHMMs give very high scores to some false matches. When studying false matches with high scores for predHMM-pams, we found that there were a few families that caused a very large part of these false positives. The majority of these matches were between

different families that all consisted of a various number of alpha helices. From this data, it seems plausible that the contribution from the secondary structure was ranked too high in comparison with the contribution from the sequence. The secondary-structure-based HMMs still place more correct sequences at high ranks than the ordinary HMMs (Table IV). For example, the number of sequences correctly ranked as number one is increased from 16% to 27% when adding the secondary structure, and to 20 % when using predicted structures.

In Tables III and IV and in Figure 5, a summary of all methods is shown. The ranks clearly support the conclusions that using multiple sequence alignment, predicted secondary structures, and a substitution matrix improves the performance of HMMs. For instance, when using a single sequence HMM, only 4% of the probe sequences recognize a correct target. This figure increases to 12% when using a multiple sequence alignment, and to 10% when using either a predicted secondary structure or a substitution matrix. When using a combination of all three methods, the number of probe sequences that recognize a correct target is increased further to 20%. The number of probes that recognize a correct target among the top 10 hits is increased from 24% to 31–38% when using multiple sequence alignments, predicted secondary structures, or substitution matrix, and to 45% when using all three.

The sensitivity shows a pattern similar to that of the ranks, although there are also some notable differences. First, it can be seen that predHMM-pam-single performs better than the HMMs that use multiple sequence alignments. This might indicate that the use of substitution matrices is not the optimal choice with the ssHMMs, as discussed above. Second, the standard HMMs that use substitution matrices perform better at higher specificity than the predHMMs. This might be due to the occurrence of a few false positives that have very high scores, as described above.

HMMs Perform as Well as but not Better Than Single-Sequence-Based Methods

The performances of all these methods were compared with the performance of single-sequence-based methods—fasta, blast, and ssearch. It could be assumed that the performance of ssearch should be similar to the performance of single sequence HMMs using a substitution matrix. However, ssearch performs better than HMM-single, as can be seen in Tables III and IV. Actually, all the single-sequence-based methods perform significantly better than HMM-pam-single and when it comes to ranks, they actually perform as well as standard HMMs. When studying the spec-sens curves it can be seen that the performance of blast and fasta are not superior to HMM-pam-single. However, ssearch still performs as well as standard (multiple sequence) HMM methods.

The reason why the multiple sequence information does not improve the performance further is probably due to the following. (1) In our benchmark, 45% of the HMMs are built from sequences with less than 10 sequences and HMMER is not optimized for small families. Furthermore, even in the case where there are several sequences they are often very similar, and thus still fail to provide the necessary diversity. (2) The gap penalties in a HMM are calculated individually for each position in the model. However, when an HMM is created from a family with low diversity, and thus few gaps, the gap penalties will not be optimal for recognizing a distant member of the family. (3) Blast, fasta, and ssearch use an extreme value distribution to fit the scores. This method has been included in HMMER-2.0, and consequently the performance has improved (data not shown). (4) When a hidden Markov model is created, it includes a process of optimizing the transition probabilities. Ideally, one should make several tries and create several hidden Markov models for a given sequence family and then use the one that performs best. However, this was not possible in this study, due to computational limitations. All these points show some of the limitations of the current generation of HMMs, but also indicate some easy methods to improve the performance of HMMs.

In fold recognition it is not enough to identify the correct fold of a protein, it is also necessary to make the correct alignment between the two proteins to obtain three-dimensional studies. In the alignments obtained for ssHMM and the other methods from our benchmark, however, most pairs in our benchmark contained proteins that were very distantly related, or not homologous at all, and these proteins are extremely difficult to align correctly. We were, unfortunately, not able to detect any significant improvement of the alignments using ssHMM (data not shown). In a future study we plan to create an alignment benchmark using a set of less difficult proteins to align and examine whether ssHMMs, or standard HMMs, are able to align proteins better than standard pairwise sequence methods.

Use of ssHMM in CASP3

The ssHMM method, together with other methods and manual judgment, were used for blind predictions in the CASP3 process.[27] Three successful fold predictions were made of CASP3 targets T0046, T0053, and T0071a. T0046 (gamma-adaptin, ear domain) is an IG-like fold, and several methods (ssHMM, standard HMMs, and threader[3]) consistently scored high for IG-like domains. For T0053 (CbiK protein), we mainly focussed on the threader results. Our best prediction was T0071 (Alpha adaptin ear domain), in which, using ssHMM, we were able to identify the first 125 residues as an IG-like fold. We were also able to produce a rather good alignment, with 21 out of 125 residues correctly aligned.

Summary

The program package HMMER was modified to allow the construction of hidden Markov models (HMMs) that use the secondary structure, in addition to the amino acid sequence, to model protein families. This was accomplished by adding a distribution over emission probabilities for secondary structures to each match and insert state in the model. It was shown that the resulting secondary structure HMMs perform better than the ordi-

nary HMMs, with both the true and the predicted secondary structures used to recognize proteins having the same fold as the modeled sequences. We have also analyzed the performance of automatically-created HMMs, using a rigorous benchmark. It was shown that using a substitution matrix improved the performance of HMMs. Finally, it was shown that the automatically-created HMMs did not perform significantly better than single sequence based methods.

REFERENCES

1. Krogh A, Brown M, Mian I, Sjölander K, Haussler D. Hidden Markov models in computational biology: application to protein modeling. J Mol Biol 1994;235:1501–1531.
2. Karplus K, Sjölander K, Barrett C, et al. Predicting structures using hidden Markov models. Proteins Suppl 1997;1:134–139.
3. Jones D, Taylor W, Thornton J. A new approach to protein fold recognition. Nature 1992;358:86–89.
4. Flöckner H, Domingues F, Sippl M. Proteins folds from pair interactions: a blind test in fold recognition. Proteins Suppl 1997;1:129–133.
5. Fischer D, Eisenberg D. Protein fold recognition using sequence-derived predictions. Protein Sci 1996;5:947–955.
6. Rice D, Eisenberg D. A 3D–1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. J Mol Biol 1997;267:1026–1038.
7. Rice D, Fischer D, Weiss R, Eisenberg D. Fold assignments for amino acid sequences of the CASP2 experiment. Proteins Suppl 1997;1:113–122.
8. Di Francesco V, Geetha V, Garnier J, Munson P. Fold recognition using predicted secondary structure sequences and hidden Markov models of proteins folds. Proteins Suppl 1997;1:123–128.
9. Rost B, Schneider R, Sander C. Protein fold recognition by prediction-based threading. J Mol Biol 1997;270:471–480.
10. Elofsson A, Fischer D, Rice D, Le Grand SDE. A study of combined structure/sequence profiles. Fold Des 1996;1:451–461.
11. Dayhoff M, Barker W, Hunt L. Establishing homologies in protein sequences. Methods Enzymol 1983;91:254.
12. Rabiner L, Juang B. An introduction to hidden Markov models. Los Alamitos CA: IEEEE ASSP Magazine. Jan 4–15, 1986.
13. Krogh A. Two methods for improving performance of an HMM and their application for gene finding. ismb 1997;5:179–186.
14. Sonnhammer E, von Heijne G, Krogh A. A hidden Markov model for predicting transmembrane helices in protein sequences. ismb 1998;6:175–182.
15. Bowie JU, Lüthy R, Eisenberg D. A method to identify protein sequences that fold into a know three-dimensional structure. Science 1991;253:164–170.
16. Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol 1993;232:584–599.
17. Hubbard JT, Park J. Fold recognition and ab initio structure predictions using hidden Markov models and β-strand pair potentials. Proteins 1995;23:398–402.
18. Eddy SR. HMMER—hidden Markov model software. http://www.genome.wustl.edu/eddy/hmmer.html
19. Durbin R, Eddy S, Krogh A, Mitchison G. Biological sequence analysis. Cambridge, UK: Cambridge University Press; 1998.
20. Sander C, Schneider R. Database of homology derived protein structures and the structural meaning of sequence alignment. Proteins 1991;9:56–68.
21. Frishman D, Argos P. Seventy-five percent accuracy in protein secondary structure prediction. Proteins 1997;27:329–335.
22. Altschul S, Madden T, Schaffer A, et al. Gapped blast and ψ-blast: a new generation of protein database search programs. Nucleic Acids Res 1997;25:3389–3402.
23. Pearson W. Comparison of methods for searching protein sequence databases. Protein Sci 1995;4:1145–1160.
24. Abagyan R, Batalov S. Do aligned sequences share the same fold? J Mol Biol 1997;273:355–368.
25. Brenner S, Chothia C, Hubbard T. Assessing sequence comparison methods with reliable structurally identified evolutionary relationships. Proc Natl Acad Sci USA 1998;95:6073–6078.
26. Murzin AG, Breener SE, Hubbard T, Chothia C. Scop: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 1995;247:536–540.
27. Moult J, Hubbard T, Bryant S, Fidelis K, Pedersen J. Critical assessment of methods of proteins structure predictions (CASP): round II. Proteins Suppl 1997;1:2–6.
28. Murzin A, Bateman A. Disant homology recognition using structural classification of proteins. Proteins Suppl 1997;1:105–112.
29. Henikoff S, Henikoff J. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA 1992;89:101915–10919.