Identification and Prediction of Super-secondary Structures

by
Robert Welin

Supervisor: Arne Elofsson, Stockholm Bioinformatics Center

Abstract

Elements of protein secondary structure are connected in ways that are often similar. Therefor, clusters of similar fragments containing a few secondary structure elements can be found. This project has focused on finding such conformations, called super-secondary structures. A database called DBOSS has been set up featuring 112 groups of super-secondary structures, divided in twelve different classes depending on which secondary structures are present. The number of members in each group vary between 2 and 123. Predictions of the super-secondary structures from the sequence of amino acids have also been tested.

Contents

1 Introduction

Each cell in our bodies contains thousands of proteins, and all of them have very different areas of use. For example, they are used as building blocks for cells and tissues, for transportation of small molecules, as enzymes for catalyzing chemical reactions, as antibodies for defense against infection and as hormones for transmitting information between cells. Because of proteins' importance in living tissues, great effort is put into research of the properties and functions of proteins.

The function of a protein depends on its structure, and the structure is hence very valuable information. Protein structure is most often determined using X-ray crystallography or Nuclear Magnetic Resonance (NMR). Such methods can be very accurate with high resolution, but they are time consuming. Hence, methods for predicting structures using computers are highly interesting and a lot of research is being put into the field.

To be able to predict the structure of a protein, methods for predicting local structures have been and are being developed. Previous work, by for example Baker et. al. [4] have lead to quite good methods for predicting secondary structure (see section 2.1.2 on the following page) using the amino acid sequences. Such methods have correctness ratio of about 75% at best. Methods for predicting larger local structure elements, such as super-secondary structures, are wished for. A super-secondary structure can be described as a few secondary structures connected by loops of certain lengths and angles, see section 2.2 on page 5 for further details on those. Such methods have been investigated by for example Sun et. al. [16], Burke et. al. [8] and Baker et. al. [4]. It is obvious that the loops connecting the secondary elements are very important parts of the structures and much of the super-secondary structure research has been focused on prediction of these loops. The results have been promising and show that prediction of super-secondary structures is a difficult but not an impossible task.

This project has focused on the properties of the whole super-secondary structures and their sequences. Most effort has been put into finding usual conformations the protein fragments fall into. The conformations of the super-secondary structures depend on which secondary structures are included, at what angles they attach to each-other, the length of loops between the elements etc. This information has been sought for by setting up a database of super-secondary structures.

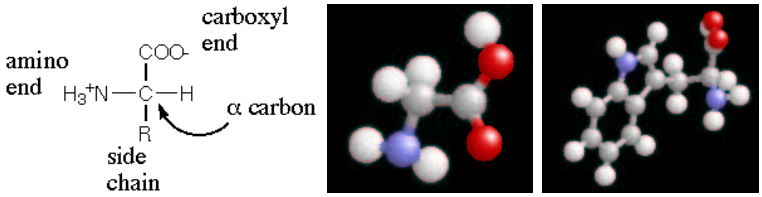
The database, called DBOSS (DataBase of Super-secondary Structures) has also been the base for attempts to predict the super-secondary structure from the amino acid sequence.

2 Background and Theory

2.1 Proteins

2.1.1 Amino Acids and Proteins

Amino acids (fig 1) are the building blocks of proteins. All amino acids are, with the exception of their side-chains, constructed the same way. They consist of a carbon atom (called the α carbon) attached to a carboxyl group (COO^-), an amino group ($NH3^+$), and a side chain. The side chain can be very simple, like the single hydrogen atom in Glycine (Fig 1), or quite complex, featuring cyclic structures in for example tryptophan (Fig 1). The characteristics, such as polarity or non polarity, of the 20 amino acids depend on the properties of their side chains [6].



(a) Amino acid (b) Glycine (c) Tryptophan

Figure 1: Amino Acids

A polypeptide is a linear chain of amino acids. A protein is one or several connected polypeptides. The amino acids in the polypeptides are bonded to each-other by peptide bonds between the α amino group of one amino acid and the α carboxyl group of another. A protein is often hundreds of residues in length, and for an easy over-view, amino acid chains are usually represented as a series of letters. Each letter is an abbreviation of one of the 20 possible amino acids. A short chain could be WSAEDKHKEGVNSHL, which would be a chain of the amino acids Tryptophan (W), Serine (S), Alanine (A) and so on.

2.1.2 Protein Structure

The chains adopt distinct three-dimensional conformations as result of interactions between their amino acids. Hence, the shapes of proteins depend to large extent on the residue sequence [6].

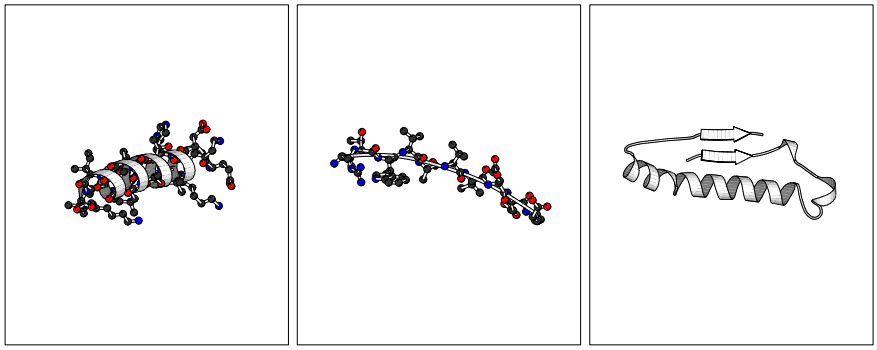
The protein structure is usually divided into four levels, the primary, secondary, tertiary and quaternary structures.

The primary structure is the sequence of amino acids including disulfide bonds. The secondary structures are common three-dimensional structures that proteins consist of. The most common secondary structures are the alpha helices

and beta sheets, but turns and random coils are also quite frequent even though turns are not always secondary structures.

In an alpha helix (fig 2), the polypeptide chain is coiled. The coil is stabilized by hydrogen bonds between carboxyl groups (COO^-) and amino groups ($NH3^+$). The carboxyl group of one residue is H-bonded to the amino group of an residue four positions away. The alpha helix is a coil with 3.4 residues (amino acids) per helical turn.

The beta sheets (fig 2) consists of two chains, or two different parts of the same chain, of amino acids lying side by side. The chains are called beta strands and form angles of 120° at each carbon α atom. The sheet formed therefore looks pleated, and the structure is stabilized by hydrogen bonds between the CO and NH groups of the two chain regions.



(a) Alpha Helix

(b) Beta strand. Note the zig-zag structure.

(c) Protein fragment. α helices are spirals and β sheets are arrows. Loop regions connecting the sheets and helices.

Figure 2: Secondary Structures

The tertiary structure in turn is the result of interactions between amino acids of different regions of the polypeptide chain. A tertiary structure is a combination of at least two secondary structures in a certain structural relationship to each-other. Usually, several alpha helices and/or beta sheets are combined into globular structures called domains. Small proteins can consist of only one domain while larger ones usually contain several ones.

A quaternary structure is the arrangement of separate polypeptide chains into the functional protein [2].

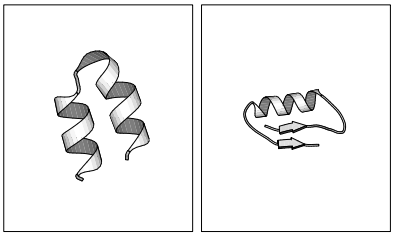
2.2 Super-secondary Structures

There is no exact definition of super-secondary structures accepted by all scientists. In this project, a super-secondary structure is described as following:

A super-secondary structure could be defined as a building block of a tertiary structure, constructed of two or more secondary structures arranged into a specific geometrical arrangement. For a fragment to belong to a certain super-secondary structure, it must contain the same secondary structures in the same order and with the same conformation in space [16].

An example of a quite usual super-secondary structures (fig 3) is two alpha helices connected by a helix hairpin. A helix hairpin is simply a loop connecting two antiparallell (2.1) alpha helices. The longer the loop is, the more different conformations can the motif have. Loops of just two or three residues only have only one possible conformation.

Another example is the beta-alpha-beta motifs (fig 3), which consists of two parallel beta strands connected by loops and an alpha helix. The loops connecting the helix to the strands can vary greatly in length, but the secondary structures' axes are always roughly parallel to each-other [9].



(a) Helix Hairpin

(b) Beta-alpha-beta motif

Figure 3: Super-secondary Structures

A protein loop is a protein region that connects two regular secondary structures, such as a helix and a strand. It can contain short fragments of any of the secondary structures mentioned in section 2.1.2 on page 4.

2.2.1 Previous Work - SLoop

A protein loop is a protein fragment that connects two regular secondary structures, such as a helix and a strand. The SLoop Database[1] contains over 8000 loops of up to 20 residues in length. The loops are clustered into classes according to their length, the type of secondary structures they bind and a distance measure based on the RMS deviation of all atoms in the loops. There are approximately 400 big classes and 3000 small containing only one or two loops.

Each class in the SLoop database contains information about the sequence of the loop, the local structural environment of the loop residues and the angle and distance between secondary structure vectors. Given a sequence, a score

[1] http://www-cryst.bioc.cam.ac.uk/~sloop/

can be calculated reflecting the match of a sequence to each SLoop class. If a tertiary structure is available, a comparison to the mean intervector separation and inter vector angle of the SLoop class can also be made. At each position in a sequence, the probability of finding a specific residue was determined from the probabilities of the residue in the corresponding position of each member loop being substituted to the residue in question. Contributions of a loop to these probabilities are weighted by the inverse of the number of its homologous member loops. The sequence score, S_{seq}, is defined as $S_{seq} = 100 \left(\prod_{i=1}^{i=n} P_i \right)^{1/n}$ where P_i is the probability of matching residue i of the loop [8], [3].

In this project, a database, DBOSS, of super-secondary structures has been set up, quite different from a loop database but with certain similarities anyway. Loops are important parts of super-secondary structures, and the groups of both databases are used as multiple alignments (see section 2.4) for structure predictions etc. DBOSS is much smaller than the SLoop database, containing only 114 groups of super-secondary structures. 80 groups are small, containing only two motifs, and the remaining 34 groups contain between 3 and 123 motifs.

2.2.2 Previous Work - HMMSTR

The I-sites library is a set of short (3-19 residues) sequence segments obtained from a database of known structures that correlate strongly to protein structure.

HMMSTR[2] is a hidden Markov model for e.g. secondary structure prediction based on the I-sites library. The model predicts secondary structures with a 74 % accuracy, which is amongst the best results of todays structure prediction mehods. It can also determine the likelihood of a potential gene coding sequence actually being one, predict backbone torsion angle at a very competitive level and a be used for a few other useful purposes [4].

For the prediction of structures, HMMSTR, as the name implies, uses an HMM based method to analyze the sequences. For details on Hidden Markov Models, see section 2.7 on page 13, but in short an HMM is a statistical model that consists of a set of states, each of which is associated with a probability distribution for generating a symbol and a set of transition probabilities between the states. The symbol mentioned might for example be a secondary structure type or an amino acid residue. The HMM used is different from most other Markov models, profile HMMs, used in protein analysis in that it is not a left-right model (again, see 2.7 for details).

Compared to the Isites database used by HMMSTR, DBOSS consists of protein fragments between 7 and 108 residues in length. Just as HMMSTR, a hidden Markov model was used to predict the structure, even though a profile HMM was used. HMMSTR was also used with some success on super-secondary motifs, but then parts of the motifs were extracted for the predictions.

[2] http://honduras.bio.rpi.edu/ isites/hmmstr/server.html

2.2.3 Previous Work - PSIPRED

PSIPRED[3] is a protein structure prediction server, which predicts secondary structures with an 80% success rate. It is based on the output of PSI-BLAST[4], which is a method for finding related amino acid sequences. PSIPRED incorporates two feed-forward neural networks which perform an analysis on output from PSI-BLAST. [11]

PSI-BLAST (Position Specific Iterated - BLAST) is a BLAST method. BLAST is a set of similarity search programs designed to explore all of the available sequence databases regardless of whether the query is protein or DNA. It uses a heuristic algorithm which seeks local as opposed to global alignments (section 2.4) and is therefore able to detect relationships among sequences which share only isolated regions of similarity.

2.3 Structural and Evolutionary Relationships

Many proteins have structural relationships or share the same evolutionary background with other proteins. Effort has been put into the classification of related proteins. SCOP[5] (Structural Classification of Proteins) is a commonly used protein structure database, based on the Brookhaven Protein Data Bank, PDB (2.6 on page 12). The SCOP database is based on the classification of proteins where proteins, depending on how closely related they are, can be classified into the same family, superfamily, fold or class [13].

family: Two proteins in the same family are believed to have the same evolutionary background. They have a sequence similarity of at least 30% or a smaller sequence identity but very similar functions and structures.

superfamily: Families, whose structures are not very similar, but whose structures, and often functions, share similarities are put into the same superfamily.

fold: Families and superfamilies that have the same major secondary structures in the same structural relationship to each-other are put into the same fold.

class: The different folds have been grouped into classes. The folds are assigned to one of the five structural classes:
1. All-alpha, those whose structure is essentially formed by alpha helices;
2. All-beta, those whose structure is essentially formed by beta sheets;
3. alpha/beta, those with alpha helices and beta strands;
4. alpha+beta, those in which alpha helices and beta strands are largely segregated, and
5. Multi-domain, those with domains of different class and for which no homologues are known at present.

[3] http://www.psipred.net
[4] http://www.ncbi.nlm.nih.gov/blast/
[5] http://scop.mrc-lmb.cam.ac.uk/scop/

2.4 Sequence Alignment

Nucleotide and protein sequences in organisms are inherited from their ancestors. In this process gene duplications, point mutations, and other events will change the sequences. Related sequences in different organisms will therefore not be identical. Proteins with similar sequences will most likely share structures and functions.

Aligning two polypeptides means finding the optimum match between their sequences and accurate alignments of sequences are needed for many types of analyses. Aligned sequences can be used to identify functions of genes and proteins and alignment methods are used to search for similarities between new sequences and sequences in databases.

The alignment is visualized as a "matrix" where each cell is a residue. The matrix is of dimensions $r \times k$ where r is the number of sequences and k is the number of residues. For example, aligning the sequences GCAA and GAA would give a 2×4 matrix with one gap inserted in the shorter sequence:

```
GCAA
G-AA
```

The sequences are compared for a series of individual characters or character patterns that are in the same order in the sequences. Identical or similar characters are placed in the same column, and non-identical characters can either be placed in the same column as a mismatch or opposite an inserted gap in one of the other sequences.

In an optimal alignment, non-identical characters and gaps are placed to bring as many identical or similar characters as possible in the same columns. To find the optimal pair-wise alignment, scores are given for each residue pair, and penalties are given for each gap inserted in the sequences. The sum of all scores and penalties is then maximized following the formula of $max\,(\sum score(a,b) - \sum gappenalty)$, where a and b are the residues, and *gap* is the penalty for a gap. The gap penalty consists of two elements, the gap opening and the gap extension, so $gappenalty = G_O + MG_E$ where M is the length of the gap, G_O is the gap opening penalty and G_E is the gap extension penalty. The time needed for such a calculation is $O(nm)$, where n and m are the lengths of the sequences.

Two main types of sequence alignment have been recognized, global and local. The global alignment optimizes the alignment over the full length of the sequences. In local alignment, stretches of sequence with the highest density of matches are given the highest priority.

Global alignment:

```
LGPSTKDFGKISESREFDN
       ||||   ||
ENQLERSFGKIN--REDA
```

Local alignment:

```
----FGKI----
----FGKI----
```

To align several proteins, methods for multiple alignment have been developed. One way of measuring how good a multiple alignment is, is to use a method similar to the one presented for pairwise alignments. Introducing scores for each residue pair, the sum of all scores over all residue pairs in the alignment is maximized, $max \sum \sum score(a, b)$. The time needed to use dynamic programming on such a problem with an $r \times k$ matrix, where r is the number of sequences and k is the number of residues, is $O(2^r k^r)$.

Using the sequences

```
QIALIDGSTYEIKTVLD
SAKLWDVREGMCRQTFT
SIEVGIDVTNAYVVAYRA
RQYQFDFKTKRILTLQK
```

the multiple alignment might be like this:

```
-Q-IALIDGST-----YEIKTVLD
--SAKLWDVRE---GMCR-QTFT-
-SIEVGIDVTNAYVVAYRA-----
RQYQ--FDFKT-----KRILTLQK
```

2.4.1 T-Coffee – a multiple alignment algorithm

Algorithms have been developed to do pair-wise as well as multiple alignments. Multi-alignments are quite difficult to optimize, and one of the most successful methods is T-Coffee.

T-Coffee [5] is a heuristics method for multiple sequence alignment, similar to the popular ClustalW [10] method. Both have high degrees of accuracy, although T-coffee claims to be better [5]. T-coffee is faster than the method for multiple alignment described in section 2.4 on the page before and offers a good compromise between speed and accuracy.

In T-Coffee, all sequences are aligned to all to provide a library of alignment information that can be used to guide the progressive alignment. Intermediate alignments are then based not only on the sequences to be aligned, but also on how all of the sequences align with each-other.

So, the T-Coffee algorithm has two main features. The first provides a method for generating multiple alignments using a library of pairwise alignments. The second provides an optimization method, which finds the multiple alignment that best fits the pair-wise alignments in the library.

Two primary libraries are used that consist of all possible pairs of sequences.The libraries contain global alignments respectively local alignments. The libraries

contain several alignments for each pair: two global alignments are used in the global alignment library, and ten local alignments are used in the local alignment library. A weight, showing the relevance, is assigned to each alignment. The two libraries are combined into one primary library by adding the weights for each entry to each-other. This primary library could be used directly to compute a multiple sequence alignment that best matched the weighted pairs of residues. However, to get a better result, the library is extended by comparing each residue pair with residue pairs from all other alignments. To use all the information in the extended library, a progressive alignment method is used. Pair-wise alignments are first made to produce a distance matrix between all the sequences, which in turn is used to produce a phylogenetic tree. The tree is used to direct the grouping of sequences during the multiple alignment process [5], [10].

2.5 Structural comparison

There are several methods for structural comparison between two protein fragments. In this project, the programme LGscore by Cristobal et. al. [7], was used. LGscore is based on an improved algorithm, originally presented by Levitt and Gerstein [12], as described below.

Comparing two structures involves two steps. First, the two objects are aligned optimally through the introduction of gaps in such a way as to maximize their residue-by-residue similarity. This operation generates a similarity score for the number of residues matched. Second, one has to assess the significance of this score in the context of what is known about the proteins currently in the database.

Starting with the two structures in arbitrary orientations, all pairwise distances between every carbon α atom in the first structure and every carbon α atom in the second are computed. All distances are put in a matrix where each element d_{igj} corresponds to the distance between residue i in the first structure and residue j in the second. This distance matrix, D, can be converted into a similarity matrix, S, through the relationship $s_{ij} = M/(1 + (d_{ij}/d_0)^2)$, where $M = 20$ and $d_0 = 5$ Å.

Applying dynamic programming to the similarity matrix, the structures are least-square fitted onto each-other. The steps are repeated until the process converges, and the worst fitting residues are eliminated.

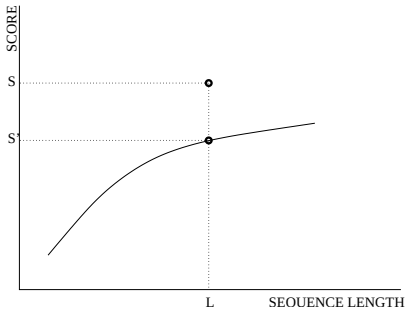
The score optimized by dynamic programming is the sum of the similarity matrix scores S_{ij} minus the total penalty for opening gaps,

$$S_{str} = M\left(\sum \frac{1}{1 + (d_{ij}/d_0)^2} - \frac{N_{gap}}{2}\right) \quad (1)$$

where N_{gap} is the number of gaps in the alignment.

To calculate the significance of this score they used a set of structural alignments of unrelated proteins to calculate a distribution of S_{str} dependent on the

alignment length, l. From this distribution they then calculated a P value dependent on S_{str} and l. The P value is the probability that a better score would occur by chance.



The score distribution was integrated to determine a distribution function P_{str}, defined as the probability that matching two random structures will give a z value greater than or equal to Z with $P_{str}(z > Z) = 1 - exp[-exp(-Z)]$.

Z is a significance measure of the score, as it shows the distance between the estimated and the true score. Figure 2.5 shows an example of how a score distribution could appear, and Z would be determined by

$$Z = \frac{S - S'}{sd}$$

where sd is a standard deviation for the score distribution and $S'(L)$ and $sd(L)$ are calculated from data from the earlier optimization.

2.6 The Protein Data Bank

The two most common ways of determining the structure of proteins is using X-ray crystallography or NMR. These two methods give an accurate model of the protein in three dimensions. Thousands of proteins have been structurally determined using these or similar methods, and the results are collected in databases such as the Protein Data Bank (PDB)[6].

The Protein Data Bank is an international data bank for structure data of biological macromolecules. 98% of the data is derived from using the two methods mentioned above [1]. The PDB files give a lot of information regarding the proteins. For example, they contain information about the sequence of amino acids, information about all the atoms that belong to each amino acid and the coordinates of all atoms.

To get information about which residue belongs to which secondary structure, a programme called Stride can be used. Using a pdb file for input, Stride identifies the secondary structures of a protein from the coordinates of its atoms. As output, it gives the amino acid sequence with information about which secondary structure each residue belongs to.

[6] http://www.rcsb.org/pdb/

2.7 Hidden Markov Models

The result of a stochastic process is determined by chance. The result is not known before the event, but depends on the event itself. A simple example of a stochastic event is the throwing of a die where the result an be any integer between 1 and 6, but is not known before the throw.

A Markov model is a stochastic process that is useful when the system being studied can be modeled as a random walk between a discrete set of states. If a stochastic process does not depend on the past states, it fulfills the Markov condition and can be represented as a Markov model. Again, in the example of the die, it is obvious that the result of the throw only depends on the trow and not on any of the previous throws. Therefore, the throw of a die can be represented by a Markov model.

For probabilistic modeling in pattern analysis of macromolecules, a special type of stochastic processes are often used, Hidden Markov Models or HMMs. An HMM is a statistical model that consists of a set of states, each of which is associated with a probability distribution for generating a symbol and a set of transition probabilities between the states. The symbol mentioned might be a (super-)secondary structure type or an amino acid residue [4]. The HMM is a generalization of the regular Markov model, in the sense that the result of the HMM is no longer directly retrievable from the output of the system. That means that the output of an HMM can be seen as the result of a double stochastic process, where the output of a stochastic process depends on the output of another stochastic process.

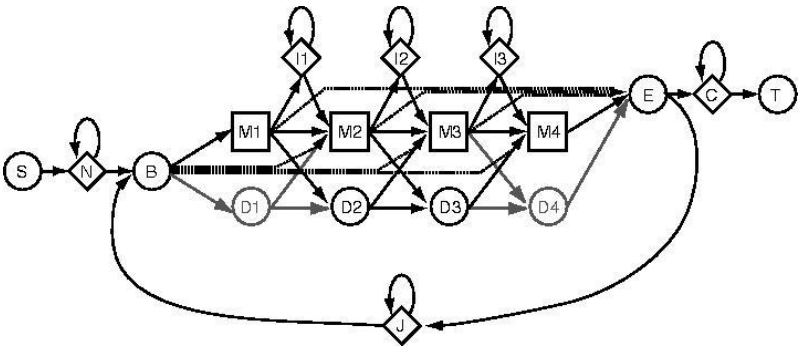


Figure 4: An example of a profile HMM. This HMM is used in the programme HMMER. B=Begin state for entering main model, M=Match state, I=Insert state, D=Delete state, E=End state for exiting main model

The special class of HMMs used to model similarities between proteins are called profile HMMs (figure 4). The HMM in the figure is used in HMMER[7], used in this project. In profile HMMs, only transitions to states with the same or higher indexes are allowed, which is sometimes called left-to-right order. Three different kind of states are introduced, match-, insert and delete-states (abbreviated M-, I and D-states). The sequence of match states should correspond to the residue

[7] http://hmmer.wustl.edu/

sequence of the model, the insertion states should model insertions of residues in a sequence in comparison to the model and the delete states should model deletions of residues in the sequence [15].

3 Methods

3.1 Test and Validation Data

Starting with a set of 10361 protein PDB files, a set of proteins of unique superfamilies were chosen. Hence, the set of proteins used shared no more than the same fold. The final set of proteins used for testing and training contained 385 proteins, all of unique superfamilies. All fragments were of one of twelve super-secondary structure classes, depending on which secondary structures the motifs consisted of, and which order they were in. Hence, the classes where $\alpha\alpha$, $\alpha\beta$, $\beta\alpha$, $\beta\beta$, $\beta\beta\alpha$, $\alpha\beta\beta$ etc.

3.2 The Problem Approach

The problem was attacked the following way:

1. The set of protein data files was used to find all possible super-secondary structures. These were grouped after their class of super-secondary structure. Within each group, all fragments were compared to all other fragments for structural similarities and similar fragments were clustered. The clustering resulted in a database divided into twelve groups, one for each super-secondary structure class, with families of different motifs.

2. To see how well the super-secondary structures could be predicted from the amino acid sequence of a protein fragment, a couple of different methods were tried. A programme based on Hidden Markov Models was decided upon. Trying one super-secondary structure class at a time, one protein fragment was removed from its cluster and tested on all the clusters in the same set.

3.2.1 Simplifications

As described earlier, a super-secondary structure is a combination of a few connected secondary structures. For classification of the different kinds of super-secondary structures, the following assumptions were made.

Super-secondary structures were considered to be two or three alpha helices or beta sheets connected by loops, as ss-loop-ss or ss-loop-ss-loop-ss, where ss=secondary structure.

Only alpha helices and beta strands of certain lengths were considered secondary structures. All structures that did not fit those criteria were considered loops. The alpha helices had to be at least 7 residues long, and the beta strands had to be at least three residues long.

The programme Stride classifies a protein into seven different types of secondary structures: coils, alpha helices(H), beta strands(E), turns, 310 helices, bridges and pihelices. Because of similarities between the structures, all helices were considered the same type. In this study, helices and strands shorter than 7

respectively 3 residues, and bridges, coils and turns were all considered to be part of loops.

The maximum length of the loops was set to 10 residues to get fairly compact super-secondary structures.

3.3 Structural Comparison

As mentioned in section 2.5 on page 11, the programme LGscore by Cristobal et. al. was used for structural comparison between all fragments.

The fragment pairs to be compared were of varying lengths, of the same super-secondary structure classes and should be similar over the entire length of the shortest of the fragments. Therefore, 90% of the length of the shortest fragment was set as the minimum requirement for similarity. No requirements were used to force the fragments to be of similar lengths, simply because the angles between the secondary structures are more interesting than the lengths. The programme resulted in a score reflecting the similarity between the two fragments compared.

To find super-secondary structures that occur more than once, all super-secondary structure motifs in the 385 proteins in the data set were compared to all other within the same class. All $\alpha\alpha$ motifs were compared to each-other, all $\alpha\beta$ were compared etc. More than three million comparisons were made, and the compared pairs resulting in a score better than a certain cut-off limit (i.e. very similar) were kept. The result was 1000 different motifs combined into 2000 pairs.

The mentioned cut-off limit was determined by trial and error, and the chosen limit was a compromise between having many pairs and having extremely similar pairs.

3.4 Clustering

The reason for clustering the data is to find families of motifs. Ideally, a number of well-defined clusters without any other similar clusters would be the result. To get well-defined clusters where all the members are very similar, the members have to be closely related, and the clusters will be very small. The problem with that is that many clusters are going to be extremely similar to each-other. The problem of finding a good way of clustering the data is the problem of finding the golden mean between having well defined clusters and unique clusters.

A few methods for clustering were considered. The data to be clustered consisted of twelve groups of protein fragments, each group consisting of up to 700 pairs of fragments, and up to 300 unique fragments. Two methods were considered good for the purpose and tested.

1. One way of obtaining few large clusters is to cluster all connected points, regardless of how vaguely related they are:

The procedure:
Choose a point. Cluster the point's neighbours. Find the neighbours' neighbours and put them into the cluster. Keep going until all points with any connection to the points in the cluster are connected. Choose a new point outside of the existing clusters and start over.

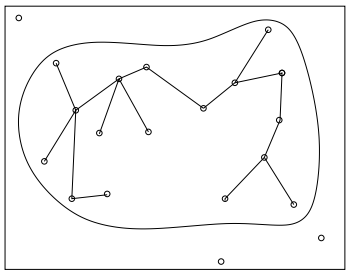


Figure 5: The figure shows putting everything that is related into the same cluster. The tiny circles are protein fragments and the areas circled are the clusters.

2. Another tested way of clustering is to force the members to be more or less closely related to each-other. This can be done by choosing a center point and allowing a radius that all the cluster members have to be inside of. For simplicity, the radius was the number of steps away from the center point.

The procedure:
Choose a radius (number of steps away from the center). Choose a center point. Find the neighbours and cluster them (step 1). Find the neighbour's neighbours and, if the radius is greater than one step, put them into the cluster (step2). Continue until the number of steps equals the radius or no more neighbours can be found. Make sure the new cluster does not overlap with any existing clusters. Choose a new center point outside of the existing clusters and start over.

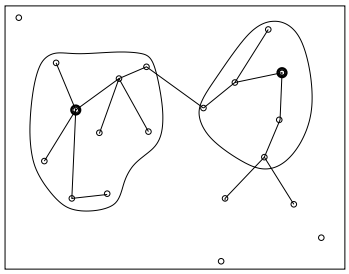


Figure 6: The figure shows method 2 with a radius of two steps. The marked dots are the center points and the areas circled are the clusters.

One drawback of method 2 over method 1 is that quite closely related fragments can end up in different clusters and some can even be left out, whereas the corresponding drawback of method 1 is that the fragments in the same cluster may be quite different, especially in a large cluster. Another advantage for method 1 is that all possible fragments will be in the clusters. Not allowing

any overlap between the clusters, some fragments will inevitably be left out. A rough visualisation of this can be made by imagining covering an area with circles without overlap, where the circles represent the clusters and the area between the circles represents the fragments left out.

The results from clustering method 1 and 2 with different radiuses can be seen in table 1.

Method	Av. # of clusters per class	Av # of fragments per cluster	# of fragments included	# of small	# of 3-10	# of big
Method 2 radius 2	18	3.8	843 of 1033	119	94	11
Method 2 radius 3	16	4.2	861 of 1033	110	79	14
Method 2 radius 4	15	4.9	910 of 1033	106	66	12
Method 2 radius 5	13	6.0	955 of 1033	97	51	11
Method 1	10	8.1	1033 of 1033	78	41	9

Table 1: The radius was the maximum number of "steps" away from the center point. Small clusters contained two members and big ones more than 10.

The data to be clustered was already filtered before the clustering, so only protein fragments similar to at least one more fragment were in the data. Hence, there is no reason to leave fragments out of the clusters. Using a method that puts many fragments into clusters should be better than a method that puts few fragments into clusters. Using short radiuses, quite few of the elements were in the clusters and the clusters were fairly small. The disadvantages of method 2 should also be quite apparent. Using a greater radius, those problems were solved, but the difference to method 1 was also reduced to a minimum. The difference was quite small between method 2 using a long radius and method 1 and method 1 captured all protein fragments. Hence, clustering method 1 was chosen for the construction of the database.

There is a risk of obvious sequence similarities between two structurally similar fragments of the same protein. Therefore, only one fragment from each protein was allowed to be clustered. During the clustering, a fragment of a protein already represented in the cluster of interest was disgarded in the clustering. Due to this weeding, only 72% of the fragments were put into clusters, 660 out of 1033 fragments. On average, 9 clusters were found in each super-secondary structure class, each cluster containing 5.7 fragments. 81 clusters contained two elements, 29 clusters contained between 3 and 10 elements and 5 clusters were larger than 10.

3.5 Structure Prediction

Hidden Markov Models (HMM, 2.7)were used for predicting the super-secondary structures. HMMs are common in structure predictions, and often show good results. The programme used was HMMER.

HMM was used on one group of super-secondary structures at a time to see if there is any relationship between sequence and structure under the given conditions.

Due to the chosen way of clustering, there were no fragments were in two different clusters, all fragments were unique. Neither did any clusters contain more than one fragment from the same protein. In each round of the testing, one fragment was removed from the clusters to be used as a test fragment. The clusters, none of them containing the test fragment, were multiply aligned (2.4) using T-Coffee and compared to the test fragment with HMMER. A score was the result of each comparison, and the cluster with the best score was considered the the most likely one.

3.5.1 Measuring the performance of the prediction method

In the case of a prediction problem with K classes, a contingency matrix Z of dimensions $K \times K$ can be obtained. In the case of this project, each class is one class of super-secondary structure. The number x_{ij} represents the number of times the input is predicted to be in class j while belonging to class i. Thus, when $i = j$, the prediction is correct. The number of inputs associated with class i is $x_i = \sum_j (z_{ij})$, and the number of inputs predicted to be in class j is $y_j = \sum_i (z_{ij})$. Simply put, x_i is the sum of row i, and y_j is the sum of column j.

The percentage $Q_i = 100\frac{z_{ii}}{x_i}$ captures the percentage of inputs correctly predicted to be in class i relative to the total number of inputs in class i. The percentage $Q_i^M = 100\frac{z_{ii}}{y_i}$ of course captures the percentage of inputs correctly predicted to be in class i relative to the total number of inputs predicted to be in class i. This provides an estimate of the conditional probability of correct prediction of class i.

The over-all percentage $Q_{total} = 100\frac{\sum_i z_{ii}}{\sum_{ij} z_{ij}}$ which is simply the relation between the number of correct predictions and the total number of inputs.

It is possible to collapse the contingency matrix into a single number. Even though the method is not perfect, it contains more information and should therefore be better than Q_{total} as a measure of performance. This "number" is called the generalized squared correlation, GC^2:

$$GC^2 = \frac{\sum_{ij} \frac{(z_{ij} - e_{ij})^2}{e_{ij}}}{N(K-1)} \tag{2}$$

where $N = \sum_{ij} z_{ij}$ is the total number of inputs, $e_{ij} = x_i y_j / N$ is the expected number of data in cell i, j of Z and K is the number of classes.

4 Results

4.1 Intermediate results

As already mentioned in 3.1 on page 15, a database of approximately 10 000 proteins was used to begin with. Only allowing proteins of unique super-families, 385 proteins were used for the actual test. These 385 proteins were investigated, and divided into fragments of different super-secondary structures.

As described in section 3.2.1 on page 15, alpha helices and beta strands connected by loops of different lengths were used in the division of the proteins into super-secondary motifs. The 385 proteins were divided into all possible super-secondary structures of lenghts of two and three secondary structures. 7438 super-secondary structures were found when there was no limit on loop lengths. 5284 of these had loop lengths shorter than the limit.

4.1.1 Structural Comparison

Class	Min length	Max length	Av. length	# before SC	# after SC
EE	7	46	17	1243	54
EEE	14	67	29	729	128
EEH	17	72	33	194	4
EH	7	58	21	675	119
EHE	17	70	33	391	103
EHH	23	82	40	99	0
HE	10	43	23	673	221
HEE	20	62	34	183	11
HEH	21	70	38	315	68
HHE	26	69	41	82	0
HH	13	103	32	490	266
HHH	28	108	49	223	59

Table 2: Found Super-secondary structures, the average lengths are for those with loop lengths shorter than the used limit (H-alpha helix, E-beta strand). "# before Structural Comparison" is the number of found motifs (with short loop lengths) and "# after Structural Comparison" is the number of motifs which have at least one relative.

As seen in table 2, the lengths of motifs only containing strands are noticeably shorter than the helix motifs. LGscore requires a length of at least six residues and is then very sequence dependent, which might have lead to the loss of many EE motifs, simply because beta strands are often quite short (79 of the EE motifs are shorter than 10 residues and 218 are shorter than 12). Regardless of the reason for the few EE motifs, the low number of EE might explain the low number of HEE and EEH as these are obviously combinations of HE or EH and EE.

4.1.2 Clustering

Using the clustering method described in section 3.4 on page 16, the result was 115 clusters in the twelve super-secondary structure classes giving an average of 9.6 clusters per group (see table 3). The diversity was big with $\alpha\alpha\beta$ and $\beta\alpha\alpha$ not getting any clusters and $\alpha\alpha$ and $\beta\beta\beta$ getting 21 clusters each. Also, $\alpha\beta\beta$ and $\beta\beta\alpha$ only got small clusters with two members, and $\alpha\alpha\alpha$ only had one big cluster.

Of the 1033 motifs showing great enough similarities with other motifs, 660 were in the clusters. The rest were not in the clusters because only one fragment per protein was allowed in one cluster. Of the 660 fragments in clusters, 498 were in clusters containing at least three fragments. Each cluster contained on average 5.7 fragments, but the diversity was great here too. 81 clusters contained only two fragments while four contained more than 40, the biggest having 123 members ($\beta\alpha$).

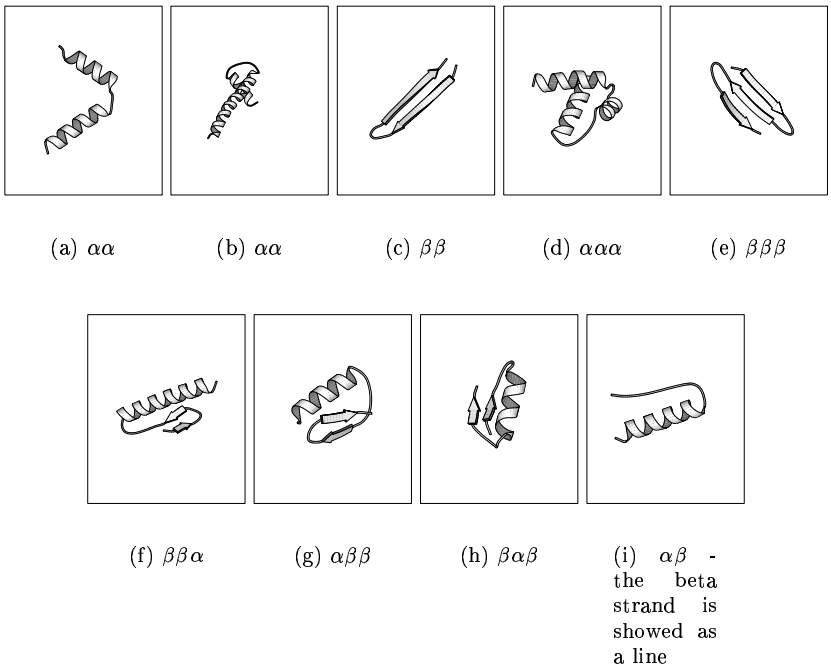
Class	Quantity	Average size	Max size	# of big ($\geq$3)	# of big ($\geq$10)	Av. dist.	Av. std dev
EE	16	2.9	5	6	0	0.039	0.039
EEE	21	3.2	10	5	1	0.029	0.0066
EEH	2	2	2	0	0	0.046	0.0003
EH	10	9.2	65	3	2	0.033	0.0054
EHE	13	5.8	46	5	1	0.028	0.0072
EHH	0	-	-	-	-	-	-
HE	13	11.6	123	4	1	0.032	0.0066
HEE	4	2	2	0	0	0.030	0.0053
HEH	9	4.8	24	3	1	0.031	0.0072
HHE	0	-	-	-	-	-	-
HH	21	7.7	110	7	1	0.026	0.0097
HHH	6	2.8	7	1	0	0.026	0.012
Total	115	5.7	-	34	7	0.032	0.0099

Table 3: Cluster statistics (H-alpha helix, E-beta strand). The last columns are the average distance between two fragments in a cluster and the standard deviation for the distance. 0.06 is the maximum possible distance and low is good.

In figure 7, representatives from a few of the clusters are shown. 7 (a) shows a typical alpha-alpha corner, which is two roughly perpendicular helices connected by a short loop region. A beta hair-pin can be seen in figure 7 (c), and the beta-alpha-beta motif described in section 2.2 on page 5 can be seen in 7 (c).

Of the 81 clusters, seven contained at least 10 fragments. The classes containing such large clusters were $\alpha\beta$, $\beta\alpha$, $\alpha\beta\alpha$, $\beta\alpha\beta$, $\alpha\alpha$ and $\beta\beta\beta$. $\beta\alpha$ contained two big clusters. Figure 8 on page 23 shows fragment examples from the big clusters.

The $\alpha\beta\alpha$ and $\beta\alpha\beta$ motifs consist of $\alpha\beta$ and $\beta\alpha$ motifs. This is quite obvious when looking at the motifs represented in for example figure 8. One of the $\beta\alpha$ motifs and the $\alpha\beta$ could easily be combined into something very similar to the $\alpha\beta\alpha$ motifs.



(a) $\alpha\alpha$ (b) $\alpha\alpha$ (c) $\beta\beta$ (d) $\alpha\alpha\alpha$ (e) $\beta\beta\beta$

(f) $\beta\beta\alpha$ (g) $\alpha\beta\beta$ (h) $\beta\alpha\beta$ (i) $\alpha\beta$ - the beta strand is showed as a line

Figure 7: Members of some of the motif clusters.

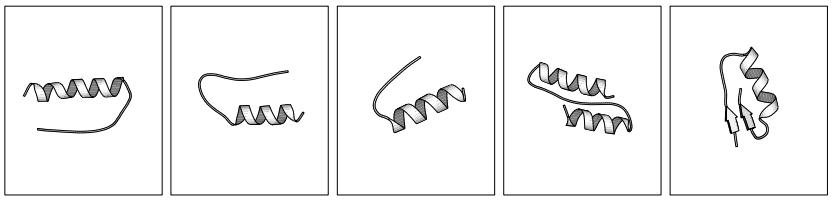
The two $\beta\alpha$ motifs are very similar. In fact, the two clusters they represent are very similar. It is hard to tell whether this is due to differences the structural comparison programme has found or due to mistakes in the algorithm.

4.1.3 Structure prediction

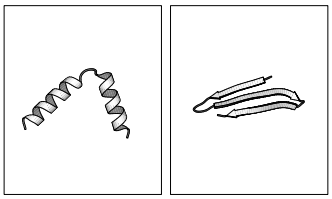
The method used for each test was the same, only with different clusters. First of all, only clusters containing at least three fragments were allowed. One protein fragment at a time was removed from its cluster to be used as a test fragment, and for the alignment to be possible, at least two fragments had to be left. The clusters (none containing the test fragment) were multiply aligned and combined into a temporary database. The residue sequence of the test fragment was compared to all clusters using the programme HMMER[8]. Each comparison resulted in an E-value between 0 and 1. The E-value shows the significance of the match. An E-value of around 1 is what can be expected just by chance and the cluster giving the lowest E-value was considered the best match. This procedure was run over and over again until most fragments had been used as test fragments. If all E-values were 1, that is if no sequence similarities could be found, no best match was chosen.

The clusters from each super-secondary structure class ($\alpha\beta$, $\alpha\alpha$ etc.) were separately tested. That is, each database contained only clusters from the same

[8]http://hmmer.wustl.edu/



(a) $\alpha\beta$ (b) $\beta\alpha$ (c) $\beta\alpha$ (d) $\alpha\beta\alpha$ (e) $\beta\alpha\beta$



(f) $\alpha\alpha$ (g) $\beta\beta\beta$

Figure 8: Members of the big clusters.

class of super-secondary structures. All classes that contained at least two clusters containing at least three elements were tested. The seven tested classes were $\beta\beta, \beta\beta\beta, \beta\alpha, \beta\alpha\beta, \alpha\beta, \alpha\beta\alpha$ and $\alpha\alpha$.

The contingency matrix described in section 3.5.1 on page 19 is supposed to be used with methods that only give yes/no output. The method described in this report also gives an output for sequences that could not be determined to belong to a certain cluster. For an easy over-view of the "unknown" output, the numbers of unknown were put in an extra column of the matrices.

In the matrices, the numbers in row i are the number of inputs associated with class i, the numbers in column j are the number of predictions associated with class j, and the numbers in the last column is the number of non-predicted ones.

To calculate the GC^2 value, the numbers of non-prediction were ignored. GC^2 ranges between 0 and 1, where GC^2=1 reveals a fully perfect prediction and GC^2=0 a fully imperfect one.

The results for the structure predictions can be viewed in figure 4.1.3 on the next page and table 4 on page 25.

In the test of $\beta\beta$ no elements at all were correctly predicted. The low number of inputs (20) suggest that this is not statistically liable, but it is still a remarkably bad result. GC^2 was not calculated due to division by zero, but can be considered bad.

Overall, in 31 % of the cases the structure was correctly predicted. Using a random choosing procedure would give a ratio of 21%. This indicates that the method is at least better than random.

$$Z_{\alpha\alpha}=\left(\begin{array}{ccccccc|c} 0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 24 & 2 & 9 & 44 & 2 & 10 & 8 & 8 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 3 & 0 & 0 & 0 & 2 \end{array}\right) \quad Z_{\beta\alpha\beta}=\left(\begin{array}{ccccc|c} 0 & 0 & 0 & 0 & 1 & 2 \\ 10 & 10 & 1 & 0 & 13 & 12 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 3 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right)$$

$$Z_{\alpha\beta\alpha}=\left(\begin{array}{ccc|c} 1 & 1 & 0 & 2 \\ 4 & 5 & 3 & 12 \\ 1 & 0 & 1 & 1 \end{array}\right) \quad Z_{\alpha\beta}=\left(\begin{array}{cccc|c} 34 & 23 & 7 & 1 & 25 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 2 & 2 & 0 & 0 & 0 \end{array}\right)$$

$$Z_{\beta\alpha}=\left(\begin{array}{ccc|c} 0 & 4 & 0 & 6 \\ 4 & 42 & 2 & 17 \\ 0 & 1 & 0 & 2 \end{array}\right) \quad Z_{\beta\beta\beta}=\left(\begin{array}{ccccc|c} 0 & 1 & 0 & 0 & 1 & 6 \\ 2 & 0 & 0 & 2 & 0 & 2 \\ 1 & 0 & 4 & 3 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 5 & 0 & 4 \end{array}\right)$$

$$Z_{\beta\beta}=\left(\begin{array}{cccccc|c} 0 & 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 4 \\ 0 & 2 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right)$$

Figure 9: Tables

With the exception of $\beta\beta$, the two-secondary structure motifs show better results than the three-secondary structure motifs. This could suggest that the tested method works better for less complex structures.

Running the prediction method on just the seven clusters containing at least 10 fragments, the result of the prediction was a ratio of 17%, with 73% falsely predicted and 10% not predicted. This gives a GC^2 value of 0.15. A random method would give a ratio of 14%, so it is a little better than random.

Class	# of inputs	% correctly	%falsly	% not predicted	GC^2
HH	129	36	56	8	0.06
EHE	58	19	53	29	0.19
HEH	30	23	29	48	0.48
HE	99	35	38	27	0.06
EH	77	54	14	32	0.54
EEE	35	17	42	42	0.41
EE	20	0	38	62	-

Table 4:

$$\mathrm{Z}_{bigclusters}=\left(\begin{array}{cccccc|c} 1 & 0 & 4 & 1 & 0 & 4 & 0 \\ 0 & 4 & 1 & 0 & 0 & 2 & 1 \\ 2 & 1 & 0 & 3 & 7 & 7 & 2 \\ 1 & 0 & 2 & 4 & 6 & 3 & 0 \\ 2 & 0 & 3 & 0 & 3 & 13 & 1 \\ 2 & 0 & 2 & 5 & 2 & 5 & 6 \end{array}\right)$$

5 Conclusions

5.1 DBOSS

Various $\alpha\alpha$, $\beta\alpha\beta$ and $\beta\beta$ motifs are recognised to be quite usual super-secondary structures [14]. The structures $\alpha\alpha$ and $\beta\alpha\beta$ are both quite frequent in DBOSS. Also, $\beta\alpha$ and $\alpha\beta$ are quite common in DBOSS, which is reasonable as they combine into $\beta\alpha\beta$. $\beta\beta$ motifs are however not very frequent in DBOSS, which might be the result of LGscore being very sensitive for short sequences. $\beta\beta\alpha$ and $\alpha\beta\beta$ not being represented at all could seem a little bit odd, even though they were the least common motifs in the data set. The low result indicates that the motifs do not have many "easy conformations" to fall into and are more likely parts of two different motifs. For example, the α in the $\beta\beta\alpha$ could be a part of another motif close to the $\beta\beta$ motif.

5.2 Structure prediction

With a 31% ratio, the prediction method is better than random, which indicates that there certainly is a connection between amino acid sequence of entire motifs and super-secondary structure. Even though 31% is not an overwhelming result, it indicates reaching decent results using a similar method could be possible. Improving the method should improve the results too, so using the amino acid sequence of the entire motifs might be a way of getting reasonably good predictions of super-secondary structure.

To get better results, the data-base can probably be improved. As seen in figure 8 on page 23, two $\beta\alpha$ clusters were very similar and should perhaps have been combined. Such possible overlaps between clusters can be investigated to gain a better data-base. Also, having a database where the classes are put together focusing on the loops connecting the secondary structures should give better results.

The table shows a comparison between previously done structure prediction work and this project. Comparing the different methods directly is not possible because of the different conditions, but the table gives an idea of their performance.

Method	Average Length	# of classes	Performance
PSIPRED	10–15	3	76%
SLoop	5–8	appr. 400	–
HMMSTR/ Rosetta	10–15	many	20%
This project	30–35	7	31%

Table 5: The average lengths are estimates. The number of classes in "This project" is the average number of big groups in each class.

Acknowledgments

I would like to thank my supervisor Arne Elofsson for his valuable advice throughout the project. I would also like to thank everyone else at SBC for advice and help when needed and for making the coffee breaks all too nice.

References

[1] HM. Berman, J Westbrook, Z Feng, G Gilliland, T. N. Bhat, H Weissig, IN. Shindyalov, and PE. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–42, 2000.

[2] K.D. Berndt. http://broccoli.mfn.ki.se/pps_course_96/ss_960723_1.html.

[3] D.F. Burke. http://www-cryst.bioc.cam.ac.uk/~sloop/info.html.

[4] Bystroff C, Thorsson V, and Baker D. Hmmstr: a hidden markov model for local sequence-structure correlations in proteins. *J. Mol. Biol.*, 301:173–90, 2000.

[5] Notredame C, Higgins DG, and Heringa J. T-coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, 302:205–17, 2000.

[6] G.M. Cooper. *The cell: a molecular approach*, chapter 3, pages 48–54. ASM Press, 1997.

[7] S. Cristobal, A. Zemla, D. Fischer, L. Rychlewski, and A. Elofsson. How can the accuracy of a protein model be measured? *BMC Bioinformatics*, 2:5, 2001.

[8] Burke D.F., Deane C.M., and Blundell T.L. A browsable and searchable web interface to the database of structurally based classification of loops - sloop. *Bioinformatics*, 16:513–9, 2000.

[9] Cooper J. http://www.cryst.bbk.ac.uk/pps2/course/section9/sss/.

[10] Thompson JD, Higgins DG, and Gibson TJ. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22:4673–80, 1994.

[11] McGuffin LJ, Bryson K, and Jones DT. The psipred protein structure prediction server. *Bioinformatics*, 16:404–5, 2000.

[12] Levitt M. and Gerstein M. A unified statistical framework for sequence comparison and structure comparison. *Proc Natl Acad Sci U S A*, 95:5913–20, 1998.

[13] A. Murzin, S. Brenner, T. Hubbard, and C. Chothia. Scop: a structural classification of proteins database. *J Mol Biol.*, 247:536–40, 1995.

[14] ExPASy Molecular Biology Server. http://www.expasy.ch/swissmod/-course/text/chapter2.htm.

[15] Eddy SR. Profile hidden markov models. *Bioinformatics Review*, 14:755–63, 1998.

[16] Z. Sun, X. Rao, L. Peng, and D. Xu. Prediction of protein supersecondary structures based on the artificial neural network method. *Protein Engineering*, 10:763–769, 1997.