THE UNIVERSITY OF READING

IDENTIFYING MISALIGNMENTS IN SEQUENCE ALIGNMENT FOR PROTEIN MODELLING

Danielle Talbot

Thesis submitted to The University of Reading for the

Degree of Doctor of Philosophy

School of Animal and
Microbial Sciences

September 2005

Abstract

Identifying misalignments in sequence alignment for protein modelling

Danielle Talbot — Ph.D. Thesis
September 2005

The difference between the number of known protein sequences and the number of protein structures is vast. Comparative modelling offers a way to bridge this gap. However it is possible to create a number of alternative models for a sequence which differ in the conformation of local regions, or in the sequence alignment used. The challenge, then is to pick the most likely model.

I have studied the empirical RAM potential to see whether it can be used to select the correct model. It was found to be capable of discriminating between models with large differences, but performed less well when differences between protein models were more subtle.

Misalignment between target and parent is the largest cause of error in comparative modelling. MLSAs (Misleading Local Sequence Alignments) are extreme examples where the sequence alignment seems obvious, but the structural alignment is different. Nine MLSAs were studied to try to determine the reasons for their occurrence. Factors such as hydrophobicity, charge interactions and location near the end of a protein chain appeared important, but no single factor was found that could be used to predict the

presence of a MLSA.

SSMAs (Sequence-Structure MisAlignments) are less extreme cases where sequence and structural alignments do not agree. These were also studied and a strong preference for starting and finishing in β-strands was observed. Neural networks were trained to identify regions of sequence likely to be mis-aligned, first using single sequences and then combining predictions for single sequences to predict correct regions in an alignment of two sequences. Single sequence predictions were up to 89.1% correct while the alignment predictions were up to 92.9% correct.

These results have much potential to be used in improving alignment for comparative modelling and software was evaluated to create alternative alignments and score them using the neural networks.

Declaration

I confirm that the work presented within the pages of this thesis, unless otherwise stated, is the work of its author, Danielle Talbot. The use of all material from other sources has been properly and fully acknowledged.

Danielle Talbot

Abbreviations

The following abbreviations are used in this thesis.

Amino Acids

A	Ala	Alanine	M	Met	Methionine
C	Cys	Cysteine	N	Asn	Asparagine
D	Asp	Aspartic acid	P	Pro	Proline
E	Glu	Glutamic acid	Q	Gln	Glutamine
F	Phe	Phenylalanine	R	Arg	Arginine
G	Gly	Glycine	S	Ser	Serine
H	His	Histidine	T	Thr	Threonine
I	Ile	Isoleucine	V	Val	Valine
K	Lys	Lysine	W	Trp	Trytophan
L	Leu	Leucine	Y	Tyr	Tyrosine

Miscellaneous

Å	Ångström
AFP	Aligned Fragment Pairs
ANN	Artificial neural networks
BLAST	Basic local alignment search tool
CE	Combinatorial Extension Algorithm
CSD	Cambridge Structural Database
DALI	Distance matrix Alignment
EM	Energy minimisation
GA	Genetic algorithm
MATRAS	MArkovian TRAnsition of Structure evolution
MCC	Matthews' correlation coefficient
MD	Molecular dynamics
MLSA	Misleading Local Sequence Alignment
MQAP	Model quality assessment program
NMR	Nuclear magnetic resonance
NN	Neural Network
PDB	Protein Databank
PDF	Probability density function
PIMA	Pattern-induced multi-sequence alignment algorithm
PIR	Protein Information Resource
Psi-BLAST	Position specific iterative BLAST
RMSD	Root mean square deviation
ROC plot	Receiver Operating Characteristic plot
Rprop	Resilient back-propagation
SCOP	Structural Classification of Proteins)
SCR	Structurally Conserved Region
SNNS	Stuttgart Neural Network Simulator
SSAP	Sequential Structure Alignment Program
SSEs	Secondary Structure Elements
SSEARCH	Sequence Similarity Search
SSM	Secondary Structure Matching
SSMA	Sequence-Structure Misalignment
SVM	Support Vector Machine
SVR	Structurally Variable Region
Uniprot	Universal Protein Resource
UniprotKB	Universal Protein Resource Knowledgebase
VAST	Vector Alignment Search Tool
WU-BLAST	Washington University BLAST

Acknowledgments

I would like to thank my supervisor Dr. Andrew Martin for his help, advice and infinite patience throughout the course of this work.

Thank you to the MRC for awarding me the studentship that brought me to Reading.

To all of those in the bioinformatics labs of the University of Reading and University College London, thank you for your help and friendship through the years. Especially to Alison who helped me so much at Reading.

I would also like to thank my family for their love, support and most importantly of all for their faith in me. They believed that I could do this when I never thought I could.

Thank you to my friends all over the world who have never doubted that I would make it.

To Fe for the mad conversations and the word "Meep".

And last but most certainly not least I would like to thank Mark, who has had his ears talked off more than once about modelling and how my coding was being stubborn. Without his love and support I don't know what I would have done.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modelling protein structures can offer a great deal of information about function and evolution, as well as addressing the problem of how the three-dimensional structure is embedded in the one-dimensional sequence (the protein folding problem) (Sierk and Pearson, 2004). The difference between the number of protein sequences held in GenBank (Benson *et al.*, 2002) and the number of protein structures held by the PDB (Protein DataBank) (Berman *et al.*, 2000) is vast. Only recently have high throughput methods been put in place to solve protein structure. Comparative modelling offers a way to bridge the gap between the number of structures and sequences. The problem of predicting the structure of a protein from its sequence has been approached in a number of ways which can be broadly split into the following categories of:

- comparative modelling (Sánchez and Šali, 1997; Guex and Peitsch, 1997)
- threading (Skolnick and Kihara, 2001; Panchenko *et al.*, 2000)
- *ab initio* folding (Kolinski and Skolnick, 1998; Ortiz *et al.*, 1998; Simons *et al.*, 2001; Aszódi *et al.*, 1995; Huang *et al.*, 1999)

Comparative modelling is capable of creating a number of different likely protein models and the challenge of being able to pick the most likely of them remains. This

applies to both alternative conformations generated for regions such as loops and to differences in alignments. Misalignment between a target and a parent sequence is the largest cause of error in comparative modelling. One of the most extreme types of misalignment is MLSAs (Misleading Local Sequence Alignments). MLSAs are areas of protein alignment where structural similarity is clear and where the optimal sequence similarity is substantially higher than that seen in the structure alignment (Saqi *et al.*, 1998). A less extreme type of misalignment is SSMAs (Sequence-Structure MisAlignments), where the sequence and structural alignments do not agree. We therefore wish to know whether such regions can be predicted and the alignment in those regions improved.

1.1 Protein databases

There are a number of databases that record protein data. These databases include:

- Protein Data Bank (Berman *et al.*, 2000)
- Swiss-Prot (Boeckmann *et al.*, 2003)
- trEMBL (Boeckmann *et al.*, 2003)
- Genpept (Benson *et al.*, 2002)

The Protein Data Bank (PDB) is a worldwide archive of structural data of macromolecules (Berman *et al.*, 2000), it contains the structures for not only proteins but also nucleic acids, carbohydrates and protein/nucleic acid complexes. Swiss-Prot is a protein knowledgebase the first release of which came out in mid-1986 (Bairoch *et al.*, 2004). It is cross-referenced with 60 different databases and tries to minimise redundancy caused by different literature reports on the same sequence. Swiss-Prot is also integrated with PIR (Protein Information Resource)

(Wu *et al.*, 2002) to form UniprotKB (Universal Protein Resource Knowledgebase) (Bairoch *et al.*, 2005). UniprotKB is the central access point for extensive curated protein information, including function, classification, and cross-reference (see `http://www.ebi.uniprot.org/`). trEMBL is a protein sequence database supplementing the Swiss-Prot Protein Sequence Data Bank (Boeckmann *et al.*, 2003). It contains the translations of all coding sequences (CDS) present in the EMBL Nucleotide Sequence Database not yet integrated into Swiss-Prot. Genpept is a protein database translated from GenBank (Benson *et al.*, 2002), a genetic sequence database, the annotated collection of all publicly available DNA sequences (Benson *et al.*, 2002) (See `http://www.ncbi.nlm.nih.gov/Genbank/GenbankOverview.html`).

1.2 Why predict protein structure?

It is widely accepted that protein function is related to structure (Bartlett *et al.*, 2003). The functional properties of proteins depend on their three-dimensional structures which, in turn, arise because linear polypeptide chains of particular amino acid sequence can fold to generate compact domains with specific three-dimensional structures (Kyngäs and Valjakka, 1998). Modelling protein structure based solely on their sequence and on the known structures of other proteins could reveal more information about the function of proteins. However, the inability routinely to predict correctly these tertiary structures from their amino acid sequences remains a most challenging problem (Kihara *et al.*, 2001).

At the beginning of 2004 there were over 40,000,000 sequences recorded in the GenBank database (figure 1). However in 2005 there were only 32,104 structures in the Protein Data Bank (figure 2). Taking the numbers of these two databases into consideration, it can be seen why there has been much effort expended in developing methods to predict the structure of a protein based on its amino acid sequence (Chiu

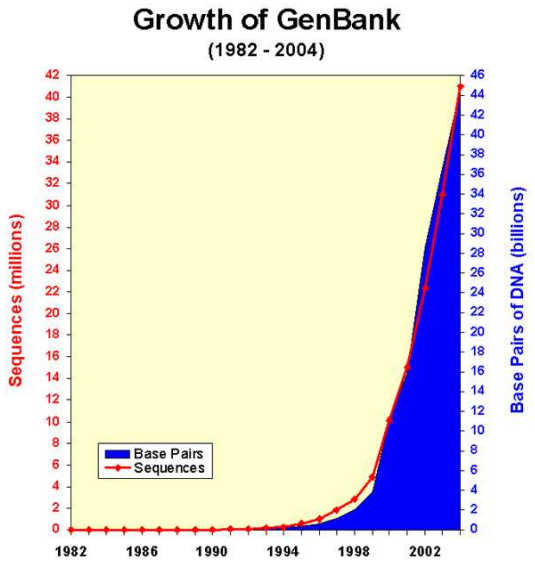


Figure 1: The increase in GenBank between 1982 to 2004, taken from `http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html`.

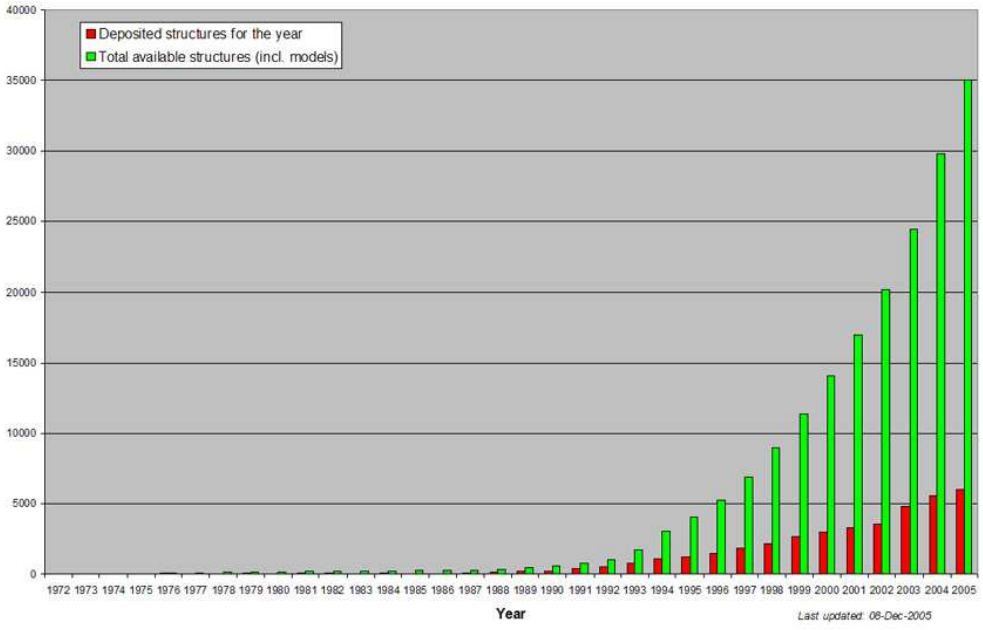


Figure 2: The increase in the Protein Data Bank between 1972 to 2005, taken from `http://www.rcsb.org/pdb/holdings.html`.

and Goldstein, 1998). As the process of obtaining the three-dimensional structure of a protein via X-ray crystallography or NMR is so laborious and expensive compared with computer-based structure predictions, structure prediction will become increasingly important as the predictions become more accurate (Martí-Renom *et al.*, 2000; Vitkup *et al.*, 2001).

High through-put techniques have been developed to extract the sequences of DNA. Indeed numerous genome-sequencing projects have used high through-put sequencing to yield a plethora of protein sequences (Cantor and Little, 1998; Grunenfelder and Winzeler, 2002; Tetko *et al.*, 2005). As genome sequencing projects continue to detect new protein sequences they provide new information for the application of computational methods. This stimulates the field and represents a good alternative to the relatively slow experimental processes of determining protein structure (Rodriguez *et al.*, 1997; Westhead and Thornton, 1998).

Protein structure prediction methods have greatly improved in their ability to provide large sets of structural models for target protein sequence (Pettitt *et al.*, 2005). There has also been a great deal of work in the area of structural genomics. Structural genomics uses a program of high through-put X-ray crystallography and NMR spectroscopy which is aimed at developing a comprehensive view of the protein structure universe (Burley, 2000).

However, while both GenBank and the PDB are expanding at a rapid rate as techniques improve, the growth rate of GenBank is far greater than that of its structural equivalent. Comparative modelling helps to bridge the gap between primary and tertiary structure by allowing the construction of protein models (Saunder *et al.*, 2000).

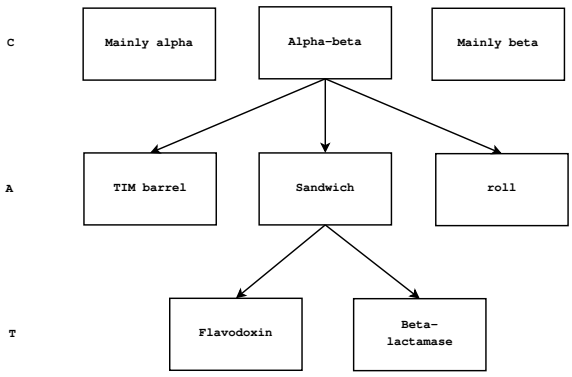


Figure 3: The hierarchical nature of CATH, taken from `http://cathwww.biochem.ucl.ac.uk/`.

1.3 Classification of Protein Structure

It is thought that there are only a limited number of protein folds (Chothia and Lesk, 1986). In 2003 there were approximately 750 distinct folds within the CATH database (Orengo *et al.*, 1997), though this value was increasing by 50 per year (Harrison *et al.*, 2003). The CATH database is made up of protein domains extracted semi-automatically from the PDB (Bernstein *et al.*, 1977). These folds are then classified into a structural hierarchy using a number of automatic and manual techniques (Blake and Cohen, 2001). The classification of proteins occurs at different levels; class, architecture, topology and homologous superfamily. This can be seen in figure 3.

The class level is split into three major groups; mainly-alpha, mainly-beta and alpha-beta, solely on the basis of secondary structure. A fourth class is used for the small number of protein structures with low secondary structure content. 'Architecture' describes the orientations of the secondary structure units, such as bundles, barrels and sandwiches (Orengo *et al.*, 1997) without considering connectivity. The topology level describes the protein fold by adding connectivity information. If structures belonging to the same T-level have suitably high similarities combined with similar functions, the proteins are assumed to be evolutionarily related and put into the same homologous superfamily (Orengo *et al.*, 1997). Criteria used to define homologues, together with

Level	Criteria
C	Mainly-alpha, mainly-beta, alpha-beta or low secondary structure content
A	Based on orientation of secondary structure
T	Automatic assignment, SSAP score ≥ 70
H	Sequence identity $\geq 35\%$, 60% structure equivalent, SSAP score ≥ 80.0 and sequence identity $\geq 20\%$, 60% structure equivalent, SSAP score $\geq$ 80.0, 60% structure equivalent and domains which have related functions. Any other evidence of homology from the literature may also be used.
S (S35)	Sequence identity $\geq 35\%$
N (S95)	First to be classified is the representative

Table 1: The levels of the CATH database used to classify protein domains. The SSAP score is generated by an algorithim and gives a normalised value between 1 and 100 which is independent of the sizes of the proteins or protein domains being scored (Taylor *et al.*, 1994). A score of above 80 indicates very similar folds, a SSAP score of between 70 and 80 are probably related folds with differences in the loop regions and in the orientation of secondary structure (Orengo *et al.*, 1992).

an explanation of the CATH hierarchy are shown in table 1.

Another popular fold classification database is SCOP (Structural Classification of Proteins) (Murzin *et al.*, 1995; Lo Conte *et al.*, 2000). Like CATH, SCOP is a hierarchical classification of protein domains. Although CATH and SCOP classify what a domain is differently, a total of 24,764 domains span the same residues in both databases (Day *et al.*, 2003). Protein domains within SCOP are divided into the following hierarchy:

1. Class - assigned by global characteristics (equivalent to CATH C)
2. Fold - assigned by similar topology (equivalent to CATH T)
3. Superfamily - assigned by clear structural homology (equivalent to CATH H)
4. Family - assigned by clear sequence homology (equivalent to CATH S)
5. Protein - assigned by function
6. Species

There are three key differences between the CATH database and the SCOP database. First, domains in CATH are assigned on a purely structural basis, while in SCOP they are assigned on a functional inheritance. Thus, to be assigned as a domain, the region must be seen as an inherited unit. Second,the assignment of a protein in CATH is mostly automatic while SCOP is mostly manual. Third, the hierarchies of the two databases differ. For example, CATH defines the architecture level which the SCOP database does not.

One of the benefits of using CATH rather than SCOP in the past was the way that entries were numbered. The authors of CATH anticipated the addition of data to each level of their hierarchy when first designing the database (Hadley and Jones, 1999). Each original level of the CATH database was numbered as a multiple of ten, for example the roll architecture is indexed as 3.10 and the barrel architecture is indexed as 3.20. This allowed for new entries to each level to be added to the end of the database or slotted in the middle, e.g. the super-roll architecture is indexed as 3.15 (Hadley and Jones, 1999). As a result of this indexing system once an entry has been given a number that number does not change in future releases of the database. In contrast the entries within the SCOP database were renumbered with each new release of the database (Hadley and Jones, 1999) until release 1.55. Since the 1.55 version of SCOP in March 2001, each database entry has a unique identifier (known as 'sunid'), which remains the same throughout each subsequent release.

1.4 Comparative modelling

The native conformation of a protein depends on its amino acid sequence and the surrounding solvent (Anfinsen, 1973). The amino acid sequence of a protein holds the information required to predict its tertiary structure. Sequence similarity can infer

functional and structural connections to homologous proteins provided that evolutionary distance is not large (Abagyan and Batalov, 1997; Chothia and Lesk, 1986; Sander and Schneider, 1991; Söding *et al.*, 2005). So if a sequence of unknown structure has a homologue it is possible to model its tertiary structure.

If there is discernible similarity between a sequence of unknown structure and a sequence of known structure, an alignment between the two will provide a good starting point from which to initiate comparative modelling (Saqi *et al.*, 1999). Comparative modelling predicts the three-dimensional structure of a given protein sequence (target) based primarily on its alignment to homologous proteins of known structure (templates) within the databases (Martí-Renom *et al.*, 2000; Pillardy *et al.*, 2001), as proteins with similar sequences tend to fold into similar three-dimensional structures even when the sequence relationship between them is very distant (Chothia and Lesk, 1986; Lesk and Chothia, 1980; Hubbard and Blundell, 1987; Flores *et al.*, 1993). There are many different computer programs that can be used for comparative modelling, e.g. MODELLER (Šali and Blundell, 1993), 3D-JIGSAW (Bates and Sternberg, 1999), FAMS (Ogata and Umeyama, 2000) and ESyPred3D (Lambert *et al.*, 2002).

The structure of α-lactalbumin was the first to be predicted using comparative modelling. Browne *et al.* (1969) predicted the structure using hen egg white lysozyme as the template structure. This was done by assembling a small number of rigid bodies obtained from the aligned protein structures (Browne *et al.*, 1969; Blundell *et al.*, 1987; Greer, 1990). These rigid bodies include the conserved core regions, the variable loops that connect them and the side-chains that decorate the backbone (see Fiser and Šálí, *Comparative protein structure modelling*, available online at `http://salilab.org/pdf/086_FiserDekker2000.pdf`).

It has been found that most protein pairs with more than 30% identity are structurally similar (Sander and Schneider, 1991; Rost, 1999). So basing an unknown protein

structure upon a known structure should give a reasonable model. The greater the sequence identity between the target and template structures, the more accurate the model of the target is likely to be. Here sequence identity is defined as:

$$\text{Sequence identity}(\%) = \frac{\text{No. of identical residues}}{\text{No. of residues in the shortest sequence}} \times 100 \qquad (1)$$

Models become less accurate when the sequence identity between target and structure is lower (Martin *et al.*, 1997; Frigerio *et al.*, 1997). The ultimate goal of comparative modelling is to be able to calculate native protein structures using only the information contained in the amino acid sequence (Sippl *et al.*, 1994). This goal is not yet achievable though comparative modelling techniques continue to improve.

There are nine distinct stages to produce manually a structure from a protein sequence by comparative modelling. Computer programs like Swiss-Model (Peitsch, 1996) and Composer (Sutcliffe *et al.*, 1987a; Sutcliffe *et al.*, 1987b) essentially automate these steps. The nine stages of comparative modelling are:

1. Identify template/parent structure
2. Align targets with parents
3. Identify SCRs (Structurally Conserved Regions)
4. Identify SVRs (Structurally Variable Regions)
5. Inherit SCRs from parent structures
6. Build SVRs
7. Build the side-chains
8. Refine the model

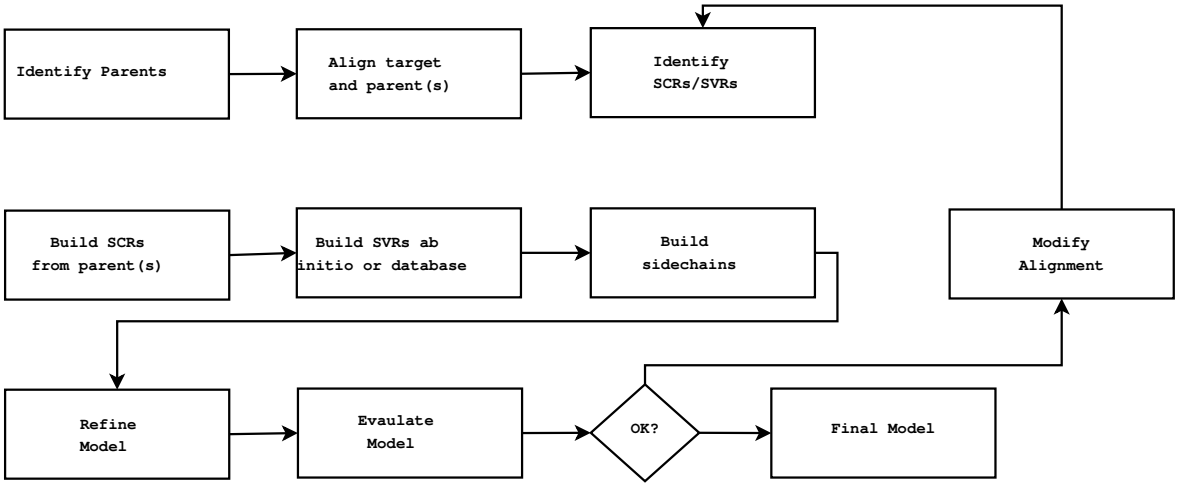


Figure 4: The stages involved in comparative modelling.

9. Evaluate errors within the model

A flowchart of a typical modelling protocol can be seen in Figure 4. Each of these stages will now be discussed.

1.5 Stages of comparative modelling

1.5.1 Finding the Correct Template

The most important feature for choosing the possible template structures is sequence identity. Comparative modelling can produce wrong models if low sequence identity exists between the target protein sequence and the chosen template structure (Facchiano *et al.*, 2001). A high sequence identity (>30%) between two proteins usually implies structural similarity and similar protein function (Shatsky *et al.*, 2006).

Suitable parent structures can be identified through searching the PDB with FASTA (Pearson and Lipman, 1988; Pearson, 1990; Pearson, 1996), BLAST (Basic local alignment search tool) (Altschul *et al.*, 1990), WU-BLAST (Washington University BLAST) (Altschul and Gish, 1996) or SSEARCH (Sequence Similarity Search) (Smith and Waterman, 1981).

Psi-BLAST (Position specific iterative BLAST) (Altschul *et al.*, 1997), which is designed to find remote homologues, can also be used to find a template. However, as the first round of Psi-BLAST is BLAST, if the PDB were used alone, the initial search would give the best hit and further searches would only find more distant homologues. If no hit is found in the PDB using BLAST, then using a larger database (for example, the PDB in conjunction with Genpept) may allow remote homologues in the PDB to be identified. Psi-BLAST creates a position-specific score matrix, a profile of conserved residues between its original sequence and its hits (it treats gap characters as a 21st distinct character (Johnston and Shields, 2005)). It then uses this profile to search against the database. As the matrix used for searching evolves through iterations it can 'move' too far away from the original sequence. This can then lead to sequences, unrelated to the original, being pulled out. So although Psi-BLAST is able to detect more distant relationships than the normal BLAST search, it can also end up pulling out unrelated sequences.

1.5.2 Alignment

The alignment is the most important stage of the comparative modelling process, if it is incorrect then the final model will also be incorrect. In the CASP2 experiment it was shown that the quality of a model was dominated by the correctness of the alignment (Cristobal *et al.*, 2001).

Structural Alignment

Structural alignment is typically based on Euclidean distance between between corresponding residues, rather than the distance between amino acid 'types' in sequence alignment (Kolodny *et al.*, 2005). It seeks to find the optimum set of equivalent residues. There are a number of programs capable of aligning proteins by structure. These include CE (Combinatorial Extension Algorithm) (Shindyalov and Bourne,

1998), SSAP (Sequential Structure Alignment Program) (Taylor and Orengo, 1989), STRUCTAL (Subbiah *et al.*, 1993; Gerstein and Levitt, 1998), DALI (Distance matrix Alignment) (Holm and Sander, 1993; Holm and Park, 2000), LSQMAN (Kleywegt, 1996), MATRAS (MArkovian TRAnsition of Structure evolution) (Kawabata and Nishikawa, 2000), VAST (Vector Alignment Search Tool) (Gibrat *et al.*, 1996) and SSM (Secondary Structure Matching) (Krissinel and Henrick, 2004). There are a number of different ways to superimpose two or more protein structures and, if the proteins are not identical or at least extremely similar in both sequence and structure, then there can be no optimal superposition (Novotny *et al.*, 2004).

Most methods for structural alignment use dynamic programming, such as SSAP, and Monte Carlo optimization, such as DALI. However, when dynamic programming is used it has an inherent ambiguity caused by the problem of the non-uniqueness of optimal structural alignment solutions (Mückstein *et al.*, 2002). This non-uniqueness comes about because different structural alignment programs can lead to different 'optimal' alignments. Although a structural alignment (regardless of which method was used to obtain it) is often referred to in the literature as if it were the one and only true alignment between a pair of proteins this is not the case (Godzik, 1996). As the superimposition of protein folds by dynamic programming is frequently ambiguous (Feng and Sippl, 1996) this means that the reliability of a so-called 'optimal' alignment has the potential to vary along its length. For proteins that are close in homology the differences between alignments created by different programs are only minor and confined to residues outside the hydrophobic core (Godzik, 1996). As the structural alignments used in this research were all between domains that were at least within the same homologous family the issue of the non-uniqueness of the structural alignment should not prove to be a problem.

CE does not use dynamic programming, instead it constructs a structural alignment by joining well aligned fragment pairs known as AFPs (Kolodny *et al.*, 2005). AFPs

are pairs of fragments, one from each protein, which confer structural similarity and are based on local geometry rather than global (Shindyalov and Bourne, 1998). By connecting these fragments, which are locally aligned, an overall structural alignment is achieved. VAST aligns SSEs (Secondary Structure Elements) using a graph theory-based approach to form a structural alignment (Madej *et al.*, 1995; Novotny *et al.*, 2004).

Several studies have been made that compare some of the various structural alignment methods, such as those by Novotny*et al.* (2004) and Kolodny*et al.* (2005). In the earlier study by Novotny, CE, DALI, MATRAS and VAST proved to be the most accurate through a variety of tests that involved modelling the Cα backbone of the protein or modelling multi-domain proteins. None of the methods achieved a 100% success rate (Novotny *et al.*, 2004). In the Kolodny study six methods were compared: SSAP, STRUCTAL, DALI, LSQMAN, CE and SSM. They concluded that STRUCTAL and SSM performed the best, followed by LSQMAN and CE (Kolodny *et al.*, 2005).

In this work SSAP was used to produce the structural alignments, as it was the method that we were most familiar with. SSAP is residue based and gives an accurate determination of the structural similarities between a pair of protein sequences (Harrison *et al.*, 2003). It uses double dynamic programming and therefore requires a similarity measure for each pair of equivalent residues in order to find the optimal alignment (Kolodny *et al.*, 2005). For SSAP this similarity measure is created from the overlap of a list of distances between the residue and every other residue in the structure for each pair of residues. This overlap of distance is optimised through further dynamic programming. As well as an alignment, SSAP also gives a score as part of its output. This score is a measure which combines the similarity of the aligned residues (accounting for the length of the alignment) and the number of residues in the smaller protein (Kolodny *et al.*, 2005).

SSAP was compared with other methods in the study by Kolodny and did not do

well. However this was because the Kolodny study looked at local alignments while SSAP aligns globally.

In comparative modelling, with an unknown target structure, it is not possible to align this way. Instead we have to rely on guessing the structural alignment from the sequence alignment. However a structural alignment can be used as a way in which to compare the validity of the modelling method. It is also useful for the classification and organization of known structures (Orengo *et al.*, 1997; Shindyalov and Bourne, 2000).

Sequence Alignment

An alignment between two sequences of residues is usually calculated by optimizing an alignment scoring function, the two common ingredients of which are a gap penalty function and a matrix of substitution scores for replacing one residue with another (Marti-Renom *et al.*, 2004). The simplest way to align two sequences is to use the dynamic programming Needleman and Wunsch algorithm (Needleman and Wunsch, 1970) which finds the best alignment by the amino acid sequence alone. Given two sequences A and B of lengths l_A and l_B respectively and a similarity matrix s such that residue A_i with B_j scores $s_{i,j}$, with a length-dependent gap penalty $W(k)$ (generally of the form $W(k) = W_o + W_e(k)$ i.e. a fixed opening penalty and a second (smaller) length dependent element):

$$S_{i,j} = s_{i,j} + \max \begin{cases} S_{i+1,j+1} \\ \max(S_{i+1,j+k} - W(k)) & \text{for } 2 \leq k \leq l_B - j \\ \max(S_{i+k,j+1} - W(k)) & \text{for } 2 \leq k \leq l_A - i \end{cases}$$

This algorithm applies penalties to its alignment 'score' dependent on any insertions and/or deletions that are found. Needleman and Wunsch calculates a global alignment rather than a local alignment as the Smith-Waterman algorithm (Smith and Waterman,

1981) does. The Smith-Waterman algorithm, using the same definitions as for the Needleman and Wunsch algorithm, is:

$$S_{i,j} = \max \begin{cases} S_{i+1,j+1} + s_{i,j} \\ \max(S_{i+1,j+k} - W(k)) & \text{for } 2 \leq k \leq l_B - j \\ \max(S_{i+k,j+1} - W(k)) & \text{for } 2 \leq k \leq l_A - i \\ 0 \end{cases}$$

Local alignment algorithms are better at finding more distant patches of homology. However, as global alignment algorithms try to align all the amino acid residues within a sequence they are best used if the two sequences are known to be homologous.

One attempt to include structural information is PIMA (Pattern-Induced Multi-sequence Alignment) algorithm (Smith and Smith, 1992) which encourages gaps outside regions of secondary structure by modifying the gap penalty. This means that gaps are more frequently found in the structurally variable regions of the protein domain.

1.5.3 Identifying the Structurally Conserved and Structurally Variable Regions, and Building the SVRs

The structurally conserved regions (SCRs) of a protein often play a major part in its function and/or stability. Typically these consist of elements of secondary structure (Li *et al.*, 1999). Since these are conserved, they are generally easy to model. The structurally variable regions (SVRs) of a protein typically include the loop regions. These may vary greatly in structure even between proteins with a high sequence identity.

The SCRs of the template can be identified by overlaying several parents and observing where they most frequently overlap in structure. When there is only one parent structure it becomes more difficult to identify the SCRs. SCRs tend to be the regions of secondary structure and also ligand- and substrate-binding areas. So if only one

parent structure can be found then these areas of the protein will be used to form the SCRs. Once identified the structure is simply inherited from the parent to the target sequence.

Structurally variable regions pose more of a problem as they cannot just be inherited from parent to target due to their variability. SVRs often contribute to binding sites and determine the functional specificity of a given protein framework (Fiser and Šali, 2003); the complementarity determining region loops of antibodies are a classic example. In these proteins the SVRs make up the antigen binding regions which vary in order to give them their specificity for binding antigen. Structurally variable regions can be built in a number of ways:

- by hand using molecular graphics (e.g. de la Paz *et al.* (1986))
- by knowledge-based methods (e.g. Jones and Thirup (1986), Greer (1981), Fidelis *et al.* (1994))
- by *ab initio* methods (e.g. Fine *et al.* (1986), Bruccoleri and Karplus (1987))
- by a combination of the above methods (e.g. Martin *et al.* (1989))

Building a loop by hand involves closing gaps in the 3-D structure of the loop. This is not very effective for longer variable regions. It is normally followed by cycles of energy minimization.

Knowledge-based methods rely on databases of loops that are searched using constraints in order to find the loop that best fits the core region to which it is to be added and the sequence being modelled. These constraints include measurements of distances between the secondary structures that hold the loops. The loop that best fits these measurements and the target sequence for the loop can then be transplanted onto the SCRs. One such database is SLoop (Donate *et al.*, 1996). In 2001 the SLoop

database contained over 10 000 loops up to 20 residues in length, which cluster into over 560 well populated classes (Burke and Deane, 2001).

Knowledge-based methods depend on how well saturated the conformational space is about the SVR region. In general longer loops have more available conformational space while the conformational space for shorter loops is more restricted. Thus the conformational space for shorter loops tends to be better saturated in the PDB while, for longer loops, only a small sample of the available conformations are observed (Fidelis *et al.*, 1994).

Ab initio methods involve conformational searches which programs such as CONGEN (Bruccoleri and Karplus, 1987) can perform. Molecular dynamics can be used as an alternative to this or in conjunction with it.

These methods can be used separately or in combination to create a model of a SVR that can be used within the protein model.

1.5.4 Building the Side-chains

The determination of side chain orientation presents a difficult problem due to the combinatorial nature of searching the large conformational space accessible to most amino acid residues (Feig *et al.*, 2000). To search and analyze all the possible conformations of a amino acid side chain would take an extremely long time. However, side chains have been observed usually only to be found a limited number of low energy rotameric states (Chandrasekaran and Ramachandran, 1970; Ponder and Richards, 1987). Instead of considering the full geometrically possible conformational space, only a small number of rotamers can be used to describe most naturally occurring conformers of a side chain (Liang and Grishin, 2001). Therefore side chain modelling is best done through the use of a rotamer library such as the backbone-dependent SCWRL (Bower *et al.*, 1997).

Rotamer libraries work on the principle that the presence of steric forces mean that there are certain rotational positions of bonds within an individual amino acid side-chain that are preferred over others. The steric forces cause certain rotameric states to be lower in energy than others which will cause the side-chain to twist out of the way of neighboring atoms, inflicting a high dihedral energy on the residue (Dunbrack and Karplus, 1993). Obviously an amino acid will adopt one of these preferred positions in the native structure of the protein if that position doesn't involve a clash with another part of the protein structure. Figure 5 (Shetty *et al.*, 2003) shows an example of a rotamer library for five amino acids.

There are two types of rotamer library, backbone-dependent (e.g. SCWRL, which is both an algorithm and a library and the work of Dunbrack and Karplus (1993)) and backbone-independent(e.g. the work of Ponder and Richards (1987), Summers and Karplus (1991), Tuffery *et al.* (1991), De Maeyer *et al.* (1997), Lovell *et al.* (2000)). Backbone-dependent rotamer libraries reference the local backbone conformation angles (phi and psi) (Dunbrack and Karplus, 1993) unlike their independent counterparts. Because a backbone-dependent rotamer library takes its surrounding environment into account it is preferable.

SCWRL, a backbone-dependent rotamer library, selects the most appropriate rotamer by first building all the possible rotamers. The method to find the optimal rotamer begins by first freezing all possible disulphide bonds. The most favourable rotamer is then checked for steric clashes with the backbone of the protein. If a clash is found SCWRL moves to the next most preferable rotamer and again checked for steric clashes. This continues until a rotamer is found which does not clash with the backbone. Although clashes with the backbone will have been eliminated, clashes with other side-chains will still occur. These clashes will have their side-chains placed in a group and then rotated through less favoured rotamers (which do not clash with the backbone). As these side-chains are rotated they will sometimes cause clashes with

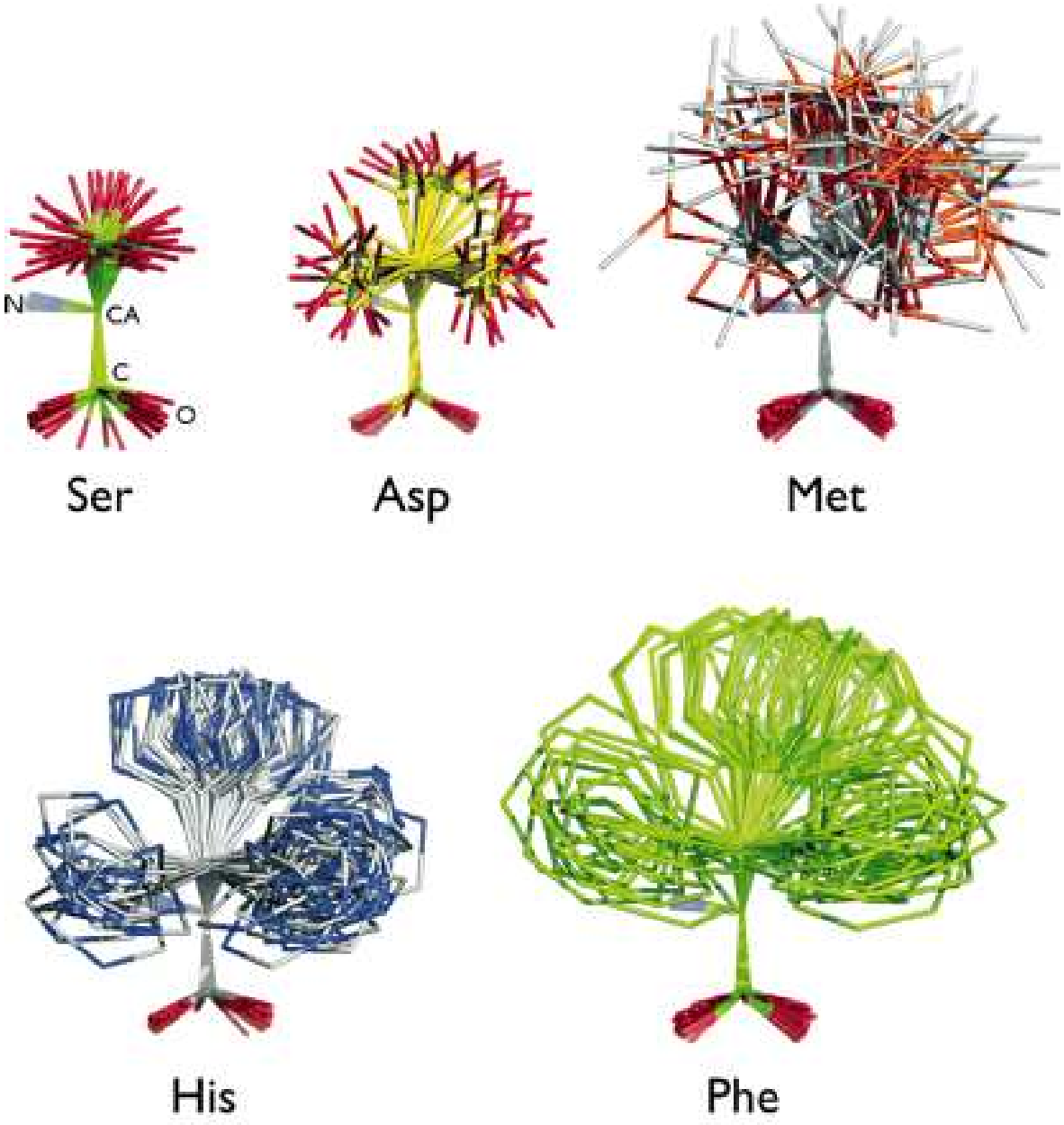


Figure 5: Examples of rotamer libraries for different amino acid residues. For each amino acid residue the side chain is shown in each of its possible conformations. Figure taken from Shetty *et al.* (2003).

side-chains outside of the group. These are added to the group and rotated through their possible rotatmer positions. Eventually the groups will stop adding members and at this time the rotamers with the least steric clashes are searched for and chosen.

1.5.5 Refining the Model

Protein models can be refined using energy minimization or molecular dynamics. Models are refined in order to alleviate clashes that would not be found in the native fold of a protein. Molecular modelling packages such as such as CHARMM (Brooks *et al.*, 1983) and NAMD (Kale *et al.*, 1999) can perform both of these refinement procedures.

In energy minimization (EM), the force field is used to calculate forces on the atoms and standard minimization techniques such as steepest descents or conjugate gradients (Fletcher and Reeves, 1964) are used to minimise the energy. Steepest descents is a simple 'downhill' algorithm. The algorithm iterates taking steps starting at a point $\mathbf{P}_i$ and moving in the direction of the local downhill gradient $\bigtriangledown f(\mathbf{P}_i)$ to a point $\mathbf{P}_{i+1}$. Steepest descents can be rather inefficient as each step is orthogonal to the previous step. Thus the algorithm takes many small steps in descending a long narrow energy well even when this adopts a perfect quadratic form (See Press *et al.* (1986)). In conjugate gradients, each step is taken in a direction which is a 'conjugate' of the previous direction and the steepest downhill step, leading to a more efficient descent of the energy surface. All minimization methods are sensitive to local minima.

In molecular dynamics (MD), atoms are initially assigned random velocities for a given temperature from a Boltzmann distribution. The force on each atom is then calculated from the force field and using the equation $F = ma$, the acceleration on each atom can be calculated. From the accelerations, the velocities are updated and a time-step is taken to recalculate the atomic coordinates (McCammon and Harvey, 1987). The procedure then iterates. Typically, the Verlet Leapfrog algorithm is employed in

which velocities at time $t + \frac{1}{2}$ are calculated from the velocities at time $t - \frac{1}{2}$ using the accelerations calculated at time t:

$$\underline{v}(t + \frac{1}{2}\Delta t) \leftarrow \underline{v}(t - \frac{1}{2}\Delta t) + \Delta t \frac{\underline{f}(t)}{\underline{m}}$$

where $\underline{v}(t)$ are the velocities of the atoms at time t, $\underline{f}(t)$ are the forces on the atoms at time t and $\underline{m}$ are the masses of the atoms. The coordinates are then updated based on the new velocities:

$$\underline{r}(t + \Delta t) \leftarrow \underline{r}(t) + \Delta t \underline{v}(t - \frac{1}{2}\Delta t)$$

where $\underline{r}(t)$ are the coordinates of the atoms at time t. Minimizing snapshots from an MD simulation can help to overcome the local minima problem of EM.

Molecular dynamics is more capable of reducing major clashes than energy minimization. Energy minimization programs simply seek to find a local energy minimum, and therefore can fail to find the global energy minimum. Molecular dynamics seeks to provide a dynamic simulation that mimics the dynamics of the protein either *in vacuo* or in the environment of a solvent. However classical molecular dynamics is limited by the amount of real time that can be simulated with current methods and computers, and most of that time is usually spent computing the interactions among water atoms (Nymeyer and García, 2003).

1.5.6 Evaluating the Model

If the structure of a protein has been solved by experimental methods, then the quality of a model can be assessed directly. In these cases, this evaluation also allows the quality of the modelling methods to be assessed (Siew *et al.*, 2000; Cristobal *et al.*, 2001).

The overall accuracy of useful comparative models spans a wide range, ranging from

models with only the correct fold to more accurate models that are comparable to structures determined by low resolution X-ray crystallography or medium resolution NMR spectroscopy (Baker and Šali, 2001). Once the structure of a protein has been solved experimentally then the model and structure can be compared and an RMSD (Root Mean Square Deviation) can be calculated. The RMSD is calculated by superimposing the structures over one another and minimising the difference between equivalent points such as the coordinates of the backbone of the protein. For proteins that are approximately 90% identical in sequence, the RMSD for their backbones, excluding loops, will be expected to be below 0.5Å whereas if the sequence identity drops to 30% then, the expected RMSD increases to approximately 4Å or higher (Contreras-Moreira *et al.*, 2003). The equation to calculate the RMSD is:

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} d_i^2}{N}} \tag{2}$$

where N is the number of equivalent atom pairs and d is the distance between the atoms in these pairs.

The benefit of using RMSD as a measure of how accurate a modelled structure is that it is a single simple number. This allows for simple comparison of the global accuracy of different models. However the problem with RMSD is that it is sensitive to a few bad outliers or a single bad torsion which can skew the value of an otherwise good model. The RMSD for a model which is mostly correct, but has one bad region can be very high (Cristobal *et al.*, 2001).

An alternative to the RMSD value is RMS/coverage graphs (Hubbard, 1999). The method is sequence-length dependent and samples the lowest ranked residues in terms of RMSD from a large number of structural superpositions, each having a different number of equivalent residues (Siew *et al.*, 2000). Figure 6 shows the RMS/coverage graph for a number of predictions of the same protein sequence. The coverage refers to

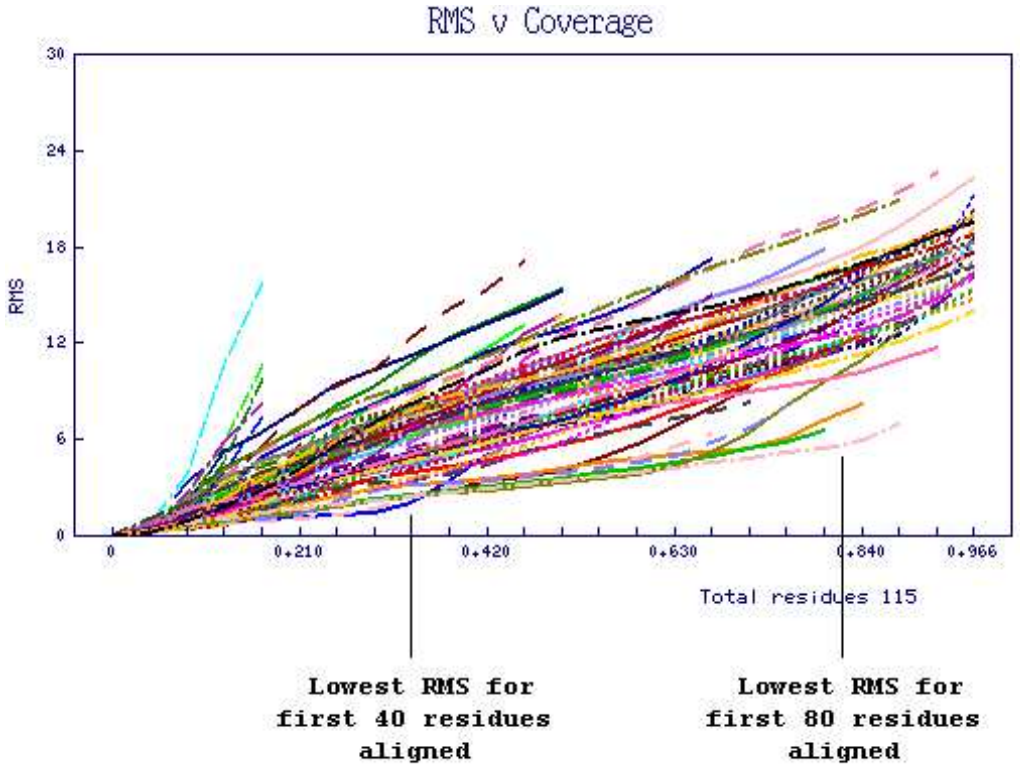


Figure 6: Example of a RMS/coverage graph for the CASP3 target T0046. This is created by sampling the lowest ranked residues in terms of RMSD from a large number of structural superpositions. Taken from (Hubbard, 1999).

the fraction of the target being predicted for the number of residues being considered (Hubbard, 1999). The residues being considered in the coverage do not have to be consecutive. Therefore residues 1,3,10,88 can be considered as a coverage of 0.04 just as residues 1-4 are (Hubbard, 1999).

However in a real modelling situation the structure has not been solved so there is no known structure with which to compare the model.

Therefore in order to assess the model, other factors must be examined. In general, errors in comparative models include errors in side chain packing, distortions and shifts of core segments of the fold, errors in modelling of insertions (e.g. loops) and errors resulting from an incorrect alignment and fold assignment (John and Šali, 2003). Obviously the sequence identity between the parent and target sequence is an important

indicator of model quality. With a sequence identity of greater than 70-90%, then the modelled structure is likely to be highly accurate though there may be some differences between it and the correct structure in the loop regions. However if the sequence identity is below 30%, then the model is extremely likely to contain errors.

Insertions and deletions and their placement within the structure will also affect the accuracy of the model. If large insertions or deletions are required error will probably have been introduced into the model. This will be especially true if the insertions/deletions are within regions of secondary structure. Insertions are generally worse than deletions as there is nothing to guide their structure.

There are programs, such as ProCheck (Laskowski *et al.*, 1993), which are capable of evaluating a modelled structure. ProCheck assesses a modelled structure by looking at the geometry of its amino acid residues. It compares the bond angles and lengths against to the 'ideal' values found by the analysis of the Cambridge Structural Database (CSD) done by Engh and Huber (1991). However, while it can be useful for finding errors such as D-amino acids, it is not very effective in evaluating model quality as the parameters it evaluates are those that have been optimized while building the model (for example during energy minimization).

Programs such as PROSA II (Sippl, 1993) are used specifically for predicting the quality of a model. PROSA II is based on the inverse folding approach and evaluates the environment of each residue in a model with respect to the expected environment as found in the high resolution X-ray structures (Sasin and Bujnicki, 2004).

If the evaluation of the model suggests that it contains a great deal of error then we return to the sequence alignment and try to adjust it. The cycle of comparative modelling can continue until the evaluation suggests that the model may be accurate.

There are a number of comparative modelling servers such as Swiss-Model which are capable of evaluating the models that they create. For example Swiss-Model calculates a confidence value for each atom in a protein model. This confidence factor is calculated

using:

1. The number of template structures used
2. The deviation of the model from the template(s)
3. The distance trap value used for framework building (Schwede *et al.*, 2003)

The confidence factor is calculated as:

$$C = \begin{cases} 85\left(\frac{1}{N}\right)\left(\frac{D}{2.5}\right) & \text{for non-fully constructed atoms} \\ 99 & \text{for fully constructed atoms} \end{cases} \quad (3)$$

where C is the confidence factor, N is the number of selected template structures and D is the distance trap value.

The distance trap value is also used during the model building process. The templates are fitted, then each residue is examined to see if one template differs from the others by more than the distance trap. If it does, this template is not used for that residue. More specific information is not available in the literature or on the Swiss-Model web site.

1.6 MODELLER

The difference between MODELLER and the nine stages set out earlier is that it does not treat the SCRs and SVRs separately. Instead it produces the structures of both at the same time. The program models a structure by satisfying spatial constraints (Šali and Blundell, 1993; Fiser *et al.*, 2000). The spatial constraints that MODELLER works with can come from a variety of sources. These sources include:

- protein structures
- NMR experiments

```
>P1;1hstA1awcAseq0
sequence:pdb1hst::A::A::::
---------------------------SHPTYSEMIAAAIRAEKSRGGSSRQSIQKYIK-
SHYKVGHNADLQIKLSIRRLLAAGV--LKQTKGVGASGSFRLAK----------*

>P1;1awc
structure:pdb1awc::A::A::::
IQLWQFLLELLTDKDARDCISWVGDEGEFKLNQPELVAQKWGQRKNKPTMNYEKLSRALRY
YY----DGDMICKVQGKRFVYKFVCDLKTLIG-YSAAELNRLVIECEQKKLARM*
```

Figure 7: Example of a MODELLER alignment file. 1hst (a chicken histone protein) is the target sequence and 1awc (a mouse GA-binding protein) is the parent.

- rules of secondary structure packing
- cross-linking experiments
- fluorescence spectroscopy
- residue-residue and atom-atom potentials of mean force

It can also perform many additional tasks, including de novo modelling of loops in protein structures, optimization of various models of protein structure with respect to a flexibly defined objective function, and multiple alignment of protein sequences or structures (Webb, 2005). Examples of the files that MODELLER requires can be found in figures 7 and 8.

After MODELLER has been given a sequence alignment and one or more parent structures as input, it calculates Cα distance and dihedral angle restraints for the target. These restraints are encoded in the form of a probability density function (PDF). The PDF is derived from an analysis of a database of 105 family alignments that include 416 proteins with known 3D structure (Šali and Overington, 1994). If there is more than one parent used then the PDF is calculated so that the model will be more restrained in those regions where the parent structures are more conserved. The spatial restraints include the distance between carbon alphas, hydrogen bonds and

```
INCLUDE
SET ATOM_FILES_DIRECTORY = './:/data/pdb/'
SET PDB_EXT = '.ent'
SET STARTING_MODEL = '1'
SET ENDING_MODEL = '1'
SET DEVIATION = '0'
SET KNOWNS = '1awc'
SET HETATM_IO = off
SET WATER_IO = off
SET HYDROGEN_IO = off

SET ALIGNMENT_FORMAT = 'PIR'
SET SEQUENCE = '1hstA1awcAseq0'
SET ALNFILE = '/home/1hstA1awcAseq0.alignment'
CALL ROUTINE = 'model'
```

Figure 8: Example of a MODELLER control file. 1hst (a chicken histone protein) is the target sequence and 1awc (a mouse GA-binding protein) is the parent. 1hstA1awcAseq0.alignment is the alignment file shown in figure 7.

main-chain and side-chain dihedral angles. These restraints are then transferred to the target sequence. The final model tries to satisfy the restraints to give a final model.

When modelling loops, MODELLER first concentrates on the non-hydrogen atoms, by optimizing their positions within a fixed environment. This optimization relies on a protocol consisting of the conjugate gradient minimization and molecular dynamics with simulated annealing (Fiser and Šali, 2003). The CHARMM-22 potential function (MacKerell *et al.*, 1998) then restrains:

- bonds
- angles
- some dihedral angles
- improper dihedral angles (Fiser and Šali, 2003)

Comparative modelling is not without problems. The greatest source of error in protein modelling is the alignment of the parent and target sequences. The final modelled structure from a mis-aligned parent/target pair is going to be different from the correct structure. The fold will still be right but the threading of the residues onto the structure will be wrong and the RMSD will be high.

However no modelling method can be more accurate than the alignment it is given. If the alignment between parent and target is wrong then this will limit the accuracy of the model.

Another source of error in comparative modelling is the modelling of SVRs, the loop regions of proteins. Loop regions are capable of varying greatly even between proteins of similar structure and function, which is why they cannot be directly inherited from the parent structure. Attempts have been made to categorise loops in the hope that it would extend the applicability of using a database search approach (Ring *et al.*, 1992; Oliva *et al.*, 1998; Rufino *et al.*, 1996). However, the database methods are limited because the number of possible conformations increases exponentially with the length of a loop (Jacobson and Šali, 2004). The general rule is that the longer the loop is, the more likely it is to be incorrect as the different possible conformations of its structure increase.

Creating a protein model by hand is far more time-consuming and liable to error than creating the model via automated methods such as Swiss-Model or MODELLER. With a computer modelling program the main error is in the original alignment that is fed into it. When modelling by hand the possibility of human error at each stage must also be considered when evaluating the final result.

MODELLER was used in this research rather than Swiss-Model as Swiss-Model makes no attempt to build difficult loops (i.e. those with large insertions/deletions) (Martin *et al.*, 1997). Not being able to compare the loop regions would have made some portions of this study meaningless. The ability to run the MODELLER program

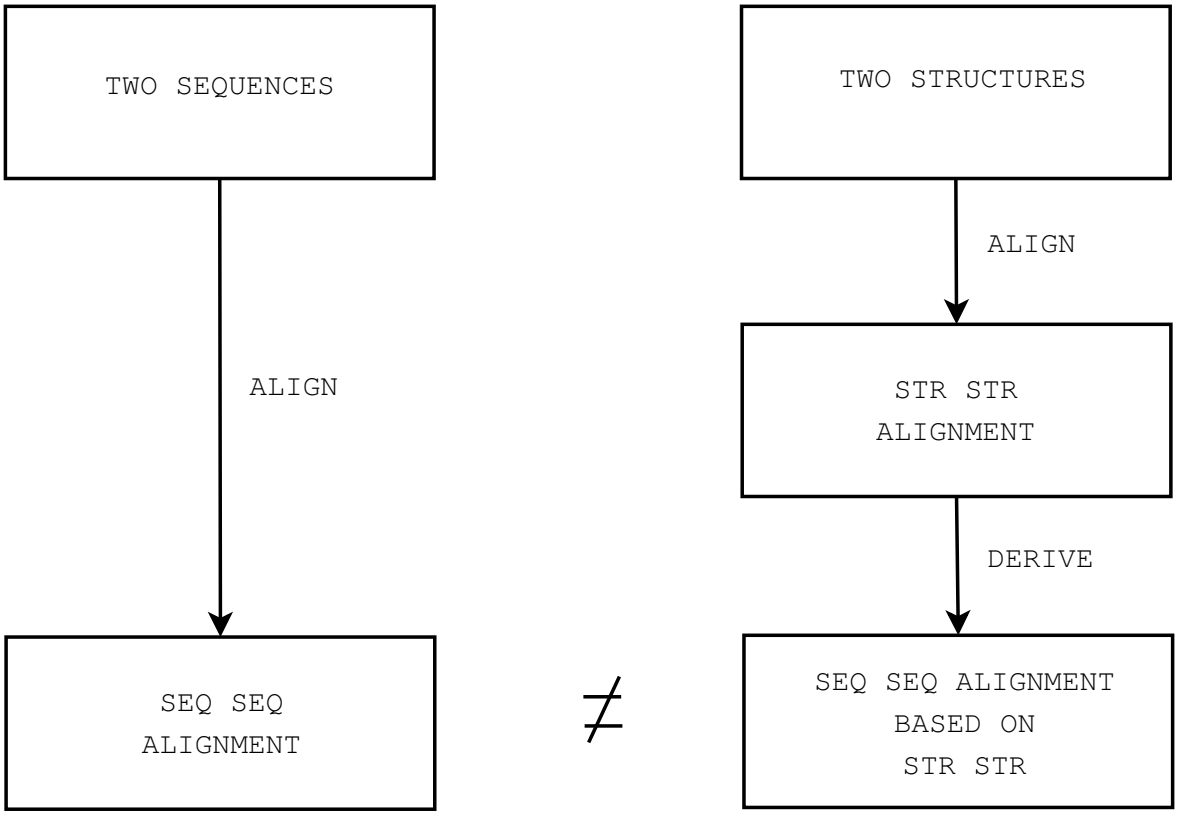


Figure 9: Sequence alignment and structural alignment do not always agree with one another.

locally was another reason why it was chosen for this research.

1.7 Why doesn't aligning by sequence always work?

For successful modelling, the sequence alignment derived from aligning the structures is the alignment that is required. However, as illustrated in figure 9, since the target structure is not available, one must use the standard sequence alignment in order to predict the structural alignment. Since the sequence alignment and structural alignment may differ, this can be a difficult problem.

Looking at figure 10 it can be clearly seen that if there is a high sequence identity between the two sequences then the alignment is likely to match the structural one. However if the sequence identity is low, 30% or less, then the sequence alignment can

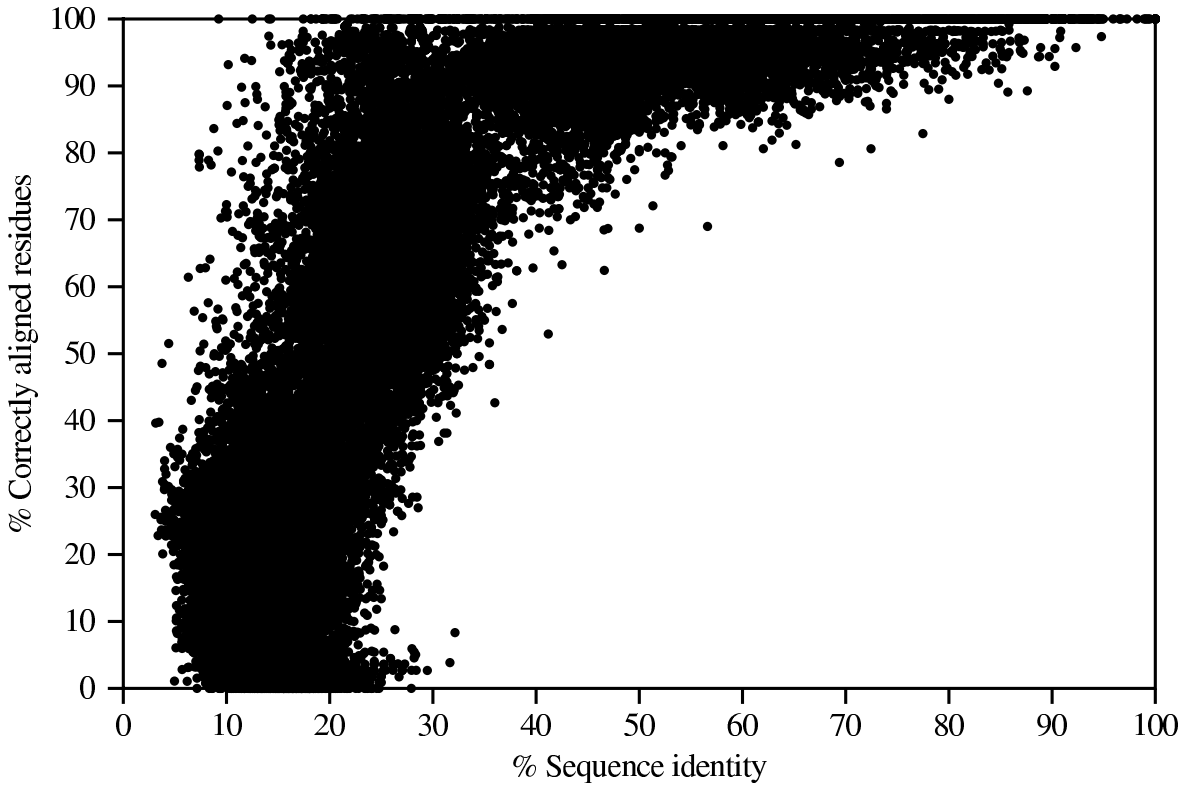


Figure 10: The relationship between percentage sequence identity and the percentage correct sequence alignment. The dataset used to compile this graph is the same initial dataset used in Chapter 3. Each pair of NReps within a CATH homologous family has been aligned by SSAP and Needleman and Wunsch. SSAP is taken as the gold standard, the correct alignment. Twelve outlying points have been removed after being identified as occurring due to errors in the CATH database (Dr. A.C.R. Martin, personal communication).

be completely different from the structural alignment. With a low sequence identity, the sequence alignment can totally fail to match the correct structural alignment.

Here sequence alignment is defined as an alignment created with the Needleman and Wunsch algorithm using the Dayhoff 78 matrix, a gap opening penalty of 10 and an extension penalty of 2. This calculates the optimum global sequence alignment between two sequences. Its concept of 'optimum' is based solely on a similarity matrix. It fails to take into account things such as:

1. secondary structures

2. charges

3. hydrophobicity

4. distance constraints affecting indels

These factors, which influence how the protein folds, may mean that the sequence alignment does not agree with the correct, structural alignment.

1.7.1 Suboptimal Alignments

A sub-optimal alignment is an alignment whose score is close to, but lower than, that of the optimal score (Naor and Brutlag, 1994). Since, the sequence alignment may deviate from the structural alignment, sub-optimal sequence alignments can be generated in order to see if they better reflect the structural alignment. However the range of sub-optimal alignments can be large and often inconsequential differences in alignment can be reported (Jaroszewski *et al.*, 2002).

When similar regions, or consensus elements, between two sequences are searched for, the alignment corresponds roughly to an ordered subset of the similarities (Vingron and Pevzner, 1995). This subset of similarities can then be used to give a range of alignments. Waterman (1984; 1986), Sobel and Martinez (1986), Karlin *et al.* (1988) and Vingron and Argos (1989) have all developed algorithms that search for these consensus elements (Vingron and Pevzner, 1995).

However it would not be possible to use suboptimal alignments in order to find the correct alignment when MLSAs (Misleading Local Sequence Alignments) occur (Chapter 3). In these areas the alignment is unambiguous and suboptimal alignments are unlikely to be identified.

1.8 Machine Learning

Machine-learning techniques are ideally suited for pattern recognition tasks where relatively large amounts of data are present and where the patterns are 'noisy' and not easily described by a compact set of rules (Nielsen *et al.*, 1999). The field of machine learning is very diverse as learning can accompany any kind of problem solving or process and so it can be studied in many different ways, such as decision making, classification, sensory signal recognition, problem solving, task execution, control or planning (Kodratoff and Michalski, 1990). There are a number of different methods of machine learning, including:

- Support Vector Machines (SVMs)
- Bayesian methods
- Decision Trees
- Neural Networks (NNs)

Support Vector Machines are based on Vapnik's Statistical Learning Theory (Vapnik, 1995). SVMs 'project' data into a higher dimensional space where they are linearly separable. The points closest to the dividing line are the support vectors. The technique tries to maximise the distance between the points and the dividing line as figure 11 illustrates.

This technique is less susceptible to over-fitting than other methods, and it achieves better results when dealing with new examples (Spinosa and de Carvalho, 2005). Other benefits of using SVMs are that they have good generalization accuracy and are fast to learn (Hearst *et al.*, 1998). However Support Vector Machines were not used within this research as they can be slow to train when using a large data set. SVMs are only binary classifiers, which in the case of this research also made them unsuitable as some of the problems dealt with had more than two outputs.

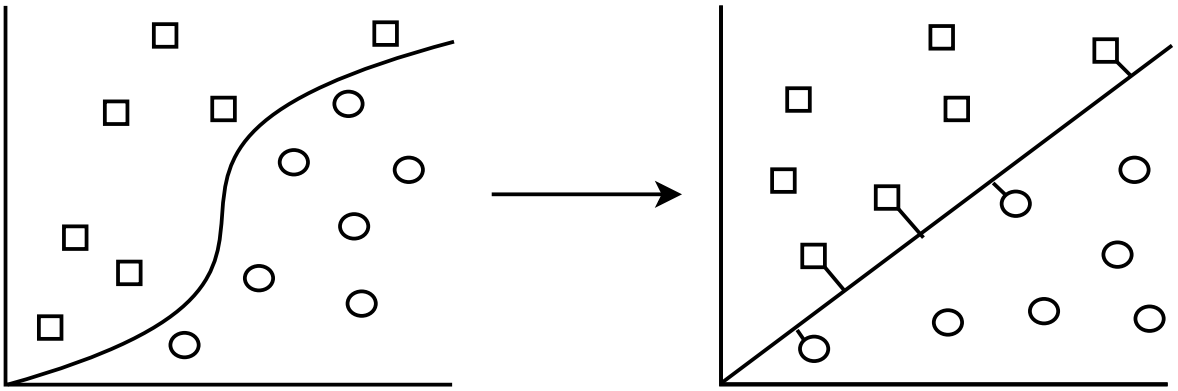


Figure 11: How a Support Vector Machine projects the data into a hyperspace where it is linearly separable and then maximises the distance between the support vectors and the dividing line.

Bayesian methods are based on Bayes' theorem:

$$Pr(E|H) = \left[\frac{Pr(H)}{Pr(E)}\right] Pr(H|E) \tag{4}$$

where $Pr(H|E)$ is the direct probability of a hypothesis conditional on a given body and $Pr(E|H)$ is the inverse probability of the data conditional on the hypothesis (See `http://plato.standford.edu/entries/bayes-theorem/`). Once new data is presented to this method it alters the existing probability based on that data, adjusting the weights by which it produces its predictions. It must therefore start with an existing prediction of the problem's distribution. This distribution is then altered with each new set of presented data (Sebastiani *et al.*, 2003). The benefits of using Bayesian methods are that they are simple to use and easy to interpret the results. However it does require prior knowledge of the distribution in order to create the initial distribution. This made it unsuitable for this research where it was not possible to know what the distribution would be like.

Decision Trees create a set of rules from which a classification can be made. They take as input an object or situation described by a set of properties, and output a set of yes/no decisions (Russell and Norvig, 2002), which is easily understood. A

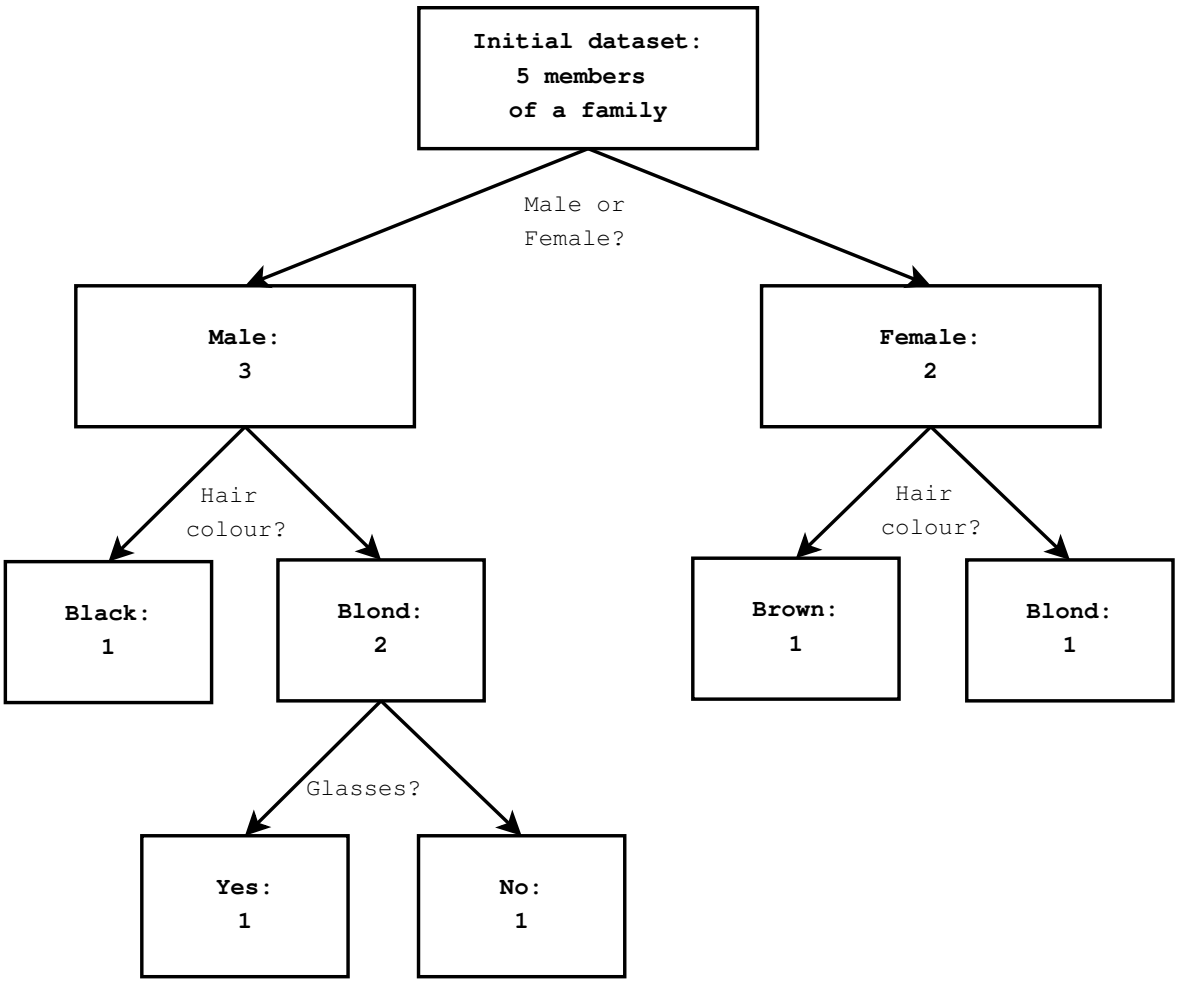


Figure 12: Example of how a decision tree works.

simple example of this method can be seen in figure 12. Each node of the decision tree corresponds to a particular attribute and each edge to alternative values for those attributes while the leaves of the tree (the final outcomes) correspond to objects with an identical classification (Michalski, 1983). Data starts at the top of the decision tree and is tested against the different attributes until it reaches a leaf and is classified. This type of machine learning is most often used for data mining and classification and would not have been practical for our purposes.

1.8.1 Artificial Neural Networks

Molecular biologists and information scientists have frequently used machine learning techniques aimed at problems, which could not yet be captured by explicit physical laws (Dietmann and Frömmel, 2002). Artificial neural networks (ANNs) are an example of this kind of technique. Neural networks have been shown to be reliable tools for protein structure prediction purposes (Rost and Sander, 1994) and attempt to mimic the learning processes of the brain in order to solve problems by learning that certain patterns of input should have a certain pattern of output. Neural networks resemble the brain in at least two respects:

1. knowledge is acquired by the network through a learning process, and inter-neuron connection strengths (known as synaptic weights) are used to store knowledge (Wei *et al.*, 1998; Reckwitz *et al.*, 1999; Forrest, 1993)

2. *"the NN is adaptive, fault tolerant, capable of very large-scale integration of information using neurobiological simulation principles, and produces a highly structured uniformity of analysis and architecture when finalised"* (Veltri *et al.*, 2002)

A neural network is made up of a number of artificial neurodes. Each neurode within the network is defined as:

- *"receiving a number of inputs from either the original input/external source or from the outputs of other neurodes from within the network"* (see `http://www.statsoft.com/textbook/stneunet.html` and `http://www.willamette.edu/~gorr/classes/cs449/ann-overview.html`).

- *"producing output passing the activation signal through an activation function (also known as a transfer function)"* (see `http://www.statsoft.com/textbook/stneunet.html`)

The architecture of a neural network is made up of the following six categories of topological data (Zupan and Gasteiger, 1993):

1. the number of inputs and outputs,
2. the number of layers,
3. the number of neurodes in each layer,
4. the number of weights in each neurode,
5. the way the weights are linked together within or between the layer(s),
6. which neurodes receive the correction signals.

The input layer is also known as the passive layer because it has no weights associated with it while the hidden and output layers are known as the active layers (each preceding set of weights being associated with the layer).

The neurodes each have one or more inputs and an output. The inputs come by way of a connection that has a strength (or weight); these weights correspond to synaptic efficacy in a biological neuron (see `http://www.statsoft.com/textbook/stneunet.html`). The weight of each connection is responsible for the ability of the network to predict an output given the correct input. In the training phase of creating a neural network the net is exposed to a number of different patterns. As it is exposed to these patterns the network adjusts the weights associated with each connection so that it will give the correct output in the testing phase. An example of these connections between the different nodes can be seen in figure 13.

The basic operation of a neurode is always the same; it collects a net input, y_j, and transforms it into an output signal, y_i, using a transfer function

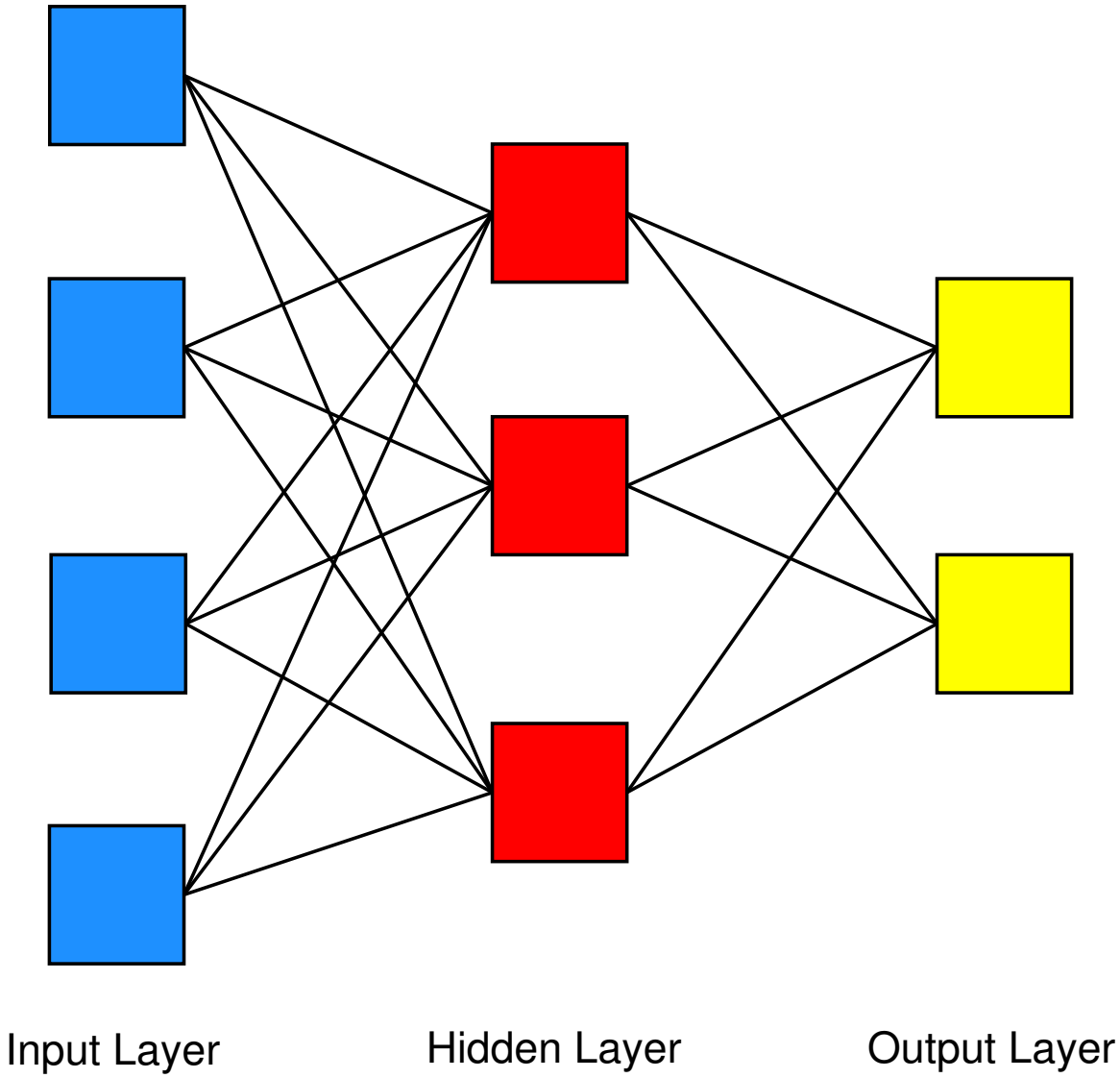


Figure 13: The layout of a neural network. The squares represent the different layers; blue being the original input layer, red being the hidden layer nodes and yellow representing the final output layer. The black lines indicate the connections between the different nodes.

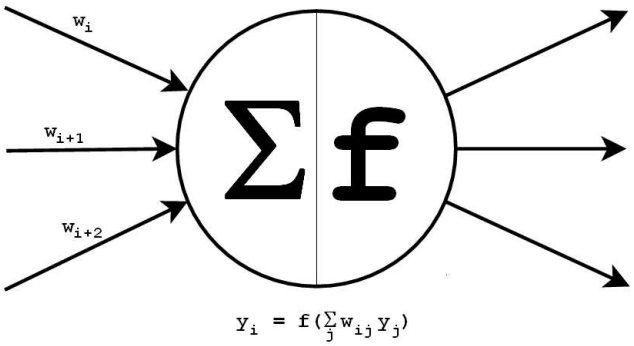


Figure 14: A neural network node. The node receives a number of inputs, w, weights them and combines them using the function, $f()$, in order to calculate its output.

(Zupan and Gasteiger, 1993), $f()$, of the weighed sum of its inputs (see `http://www.willamette.edu/~gorr/classes/cs449/ann-overview.html`):

$$y_i = f(\sum_j w_{ij} y_j) \tag{5}$$

where w represents the weights placed on the inputs. Figure 14 shows this graphically. These nodes can be built up in layers as can be seen in figure 13. Further details on neural networks and training methods can be found in Appendix A.

Typical neural net used here consisted of three layers: an input layer, a hidden layer and an output layer.

Chapter 2

Scoring Alignments With Empirical Potentials

2.1 Empirical potentials

A model must be assessed in order to evaluate the accuracy of modelling methods that were used in its construction. One of the bottlenecks for accurate prediction of protein structures and the structures of binding complexes is the immense number of possible conformations accessible to polypeptide chains (Dill and Chan, 1997; Dobson *et al.*, 1998; Honig, 1999). If the structure has also been solved experimentally then model and structure can be compared to calculate the RMSD. When dealing with sequences whose structures have not been solved by x-ray crystallography or NMR there is nothing with which to compare the model. Therefore the RMSD can not be used to used to represent the accuracy of the model.

Enthalpic, or molecular mechanics, potentials are of the form:

$$
\begin{aligned}
\mathrm{V} = \quad & \tfrac{1}{2}\sum_{b} k_b(b - b_0)^2 \\
& + \tfrac{1}{2}\sum_{\theta} k_\theta(\theta - \theta_0)^2 \\
& + \tfrac{1}{2}\sum_{\phi} k_\phi[1 - cos(m\phi - \delta)] \\
& + \sum_{i<j}\left[\frac{A_{i,j}}{r^{12}} - \frac{B_{i,j}}{r^6} + \frac{q_i q_j}{4\pi\epsilon_0\epsilon_r r}\right]
\end{aligned}
\tag{6}
$$

where b is the bond length, b_0 is the optimum bond length, θ is the bond angle and θ_0 is the optimum bond length. ϕ is the torsion angle, m is the periodicity, δ is the off-set, k are the weight constants and r is the separation between the two atoms. A and B are constants, q_i and q_j are charges on atoms, ϵ_0 is the permitivity of a vacuum and ϵ_r is the distant dependent dielectric.

These potentials aim to replicate the energetics within a protein structure, however they do not work for distinguishing correctly folded proteins as the hydrophobic effect is not calculated. Consequently they can yield comparable energies for native and completely unrelated folds (Casari and Sippl, 1992).

The failings of semi-empirical potentials were most clearly shown in a study by Novotny *et al.*. They showed that these semi-empirical potentials were unable to distinguish between the native folds and misfolded models (Novotny *et al.*, 1984; Novotny *et al.*, 1988). They constructed two incorrectly-folded proteins by taking the sequence of a mainly alpha-helix protein and threading it onto the structure of a mainly beta-strand protein of the same length, and vice versa. After energy minimization the incorrect structures were compared with their correctly-folded counterparts and an enthalpic potential of the form shown in Equation 7 from CHARMM (Brooks *et al.*,

1983) was used to see if it could find the correct structure. They found that the potentials were unable to distinguish between the correct and incorrect models. It was also discovered that the incorrect side-chains could be incorporated readily into both types of structures with only small structural adjustments.

Protein structure is governed by a number of forces such as hydrophobic interactions, electrostatic forces, the packing of side-chains and the optimization of hydrogen bonds. Therefore a database of protein structures solved by X-ray crystallography must contain, in some coded form, all the information on the many interactions that stabilise the native structure (Casari and Sippl, 1992). By statistical analysis of the contacts between atoms in known 3D structures (Labesse and Mornon, 1998) we can derive a set of mean-force, empirical potentials. These potentials form a useful tool for analyzing comparative models. These knowledge-based potentials are attractive because they are simple to construct and easy to use (Zhang *et al.*, 2004).

Two kinds of derivation methods are used for knowledge-based potentials:

1. The interactions in a database of known protein structures are assumed to obey a Boltzmann distribution,

2. *"The stability of the native folds relative to a manifold of misfolded structures is optimised"* (Zhang and Skolnick, 1998).

The earliest attempts to derive potentials from a database of known structures were reported by Tanaka and Scheraga (1976), almost thirty years ago (Sippl, 1993). The general definition of database-derived mean force potential is:

$$\mathrm{E(r)} = -kT\ln[f(r)] \tag{7}$$

where, r is the distance between two atoms, $E(r)$ is the energy at r, $f(r)$ is the probability density at r, k is the Boltzmann's constant and T is the absolute temperature

(Sippl, 1995; Sippl, 1990). The equation to calculate the mean force potential between atom type a and atom type b, where s is the separation between the two is calculated by:

$$E^{a,b,s}(r) = -kT\ln[f^{a,b,s}(r)] \tag{8}$$

where $f^{a,b,s}(r)$ is approximated by relative frequencies obtained from a data base of known structures (Sippl, 1995).

Potentials of mean force implicitly take into account all forces (electrostatic, van der Waals, etc.) acting between atoms as well as the influence of the surrounding medium on the interaction (Sippl, 1995). Because they take into consideration all the different forces acting on and between the atoms they should be able to differentiate between a correctly and an incorrectly folded protein structure. However to calculate the potential correctly, redundant information needs to be removed. The redundant information can be calculated by a suitably defined reference state (Sippl, 1995). It is an average over all atom and residue types (Sippl, 1995). For intra-molecular protein interactions, the reference state can be calculated by the equation:

$$E^{s}(r) = -kT\ln[f^{s}(r)] \tag{9}$$

Where:

$$f^{s}(r) = \sum_{a,b} f^{a,b,s}(r) \tag{10}$$

By removing the redundancy value the specific potential of the interaction can be calculated. The final equation is then:

$$\Delta E^{a,b,s}(r) = E^{a,b,s}(r) - E^{s}(r) = -kT\ln\left[\frac{f^{a,b,s}(r)}{f^{s}(r)}\right] \tag{11}$$

Empirical potentials attempt to quantify the free energy of a system (Pierce and

Winfree, 2002). Empirical potentials can be at an atomic or a residue level. Studies have shown that even simple residue level empirical potentials without atomic details may be sufficient to determine the overall fold of a protein (Crippen, 1991; Finkelstein and Reva, 1991; Maiorov and Crippen, 1992; Sippl, 1993; Kocher *et al.*, 1994; Matsuo and Nishikawa, 1994; Huang *et al.*, 1995; Park and Levitt, 1996; Thomas and Dill, 1996; Miyazawa and Jernigan, 2000). Thus a set of empirical potentials derived from a database of protein structures solved by x-ray crystallography can be used to distinguish between a group of models.

It is known that the most frequently observed states, the native folds, correspond to low energy states (Reva *et al.*, 1997). So if the potentials are applied to a model, a lower final value will indicate that the structure is more likely to be accurate than a higher final potential. In a study by Casari and Sippl (1992) a hydrophobic potential was able to identify the correct fold in 66% of all cases. In a further 14.7%, the native fold was among the five conformations of lowest energy (Casari and Sippl, 1992). Exceptions to this would include those that contain large prosthetic groups, Fe-S clusters, or polypeptide chains that do not adopt globular folds (Hendlich *et al.*, 1990).

2.2 RAM Potential

Since we hoped to be able to distinguish quite subtle variations in structure, it was felt that an atom level potential would be more effective and the RAM potential was chosen. It is a potential of mean force between all groups observed in native protein structures (Samudrala and Moult, 1998). It was downloaded from the ProStar potentials site (`http://prostar.umbi.umd.edu/index.shtml`). As with other empirical potentials it is assumed that the experimentally solved structures of proteins in the Protein DataBank represent the lowest energy of the preferred conformations of amino acids.

Other potentials such as PROSAII have been developed specifically for screening modelled structures. However the intention of this study was to develop a new potential based upon the original. As the code for PROSAII is not distributed as source code it would have been impossible to modify it to form a new potential. Also PROSAII works at the level of the residues and it was felt that an atom-level potential was needed to distinguish between very similar models. The RAM potential was implemented in two forms, a Perl program written by myself and a C version written by Dr. A.C.R. Martin.

2.3 Large Scale Analysis

To assess the ability of the RAM potential to pick out the more accurate model on a large scale, a computer farm was used. The Qlite (Martin, 2000a) software package was used to queue the jobs on the computer farm. All the proteins that were used had already had their structure solved by either NMR or x-ray crystallography. The dataset contained all the NRep pairs within each homologous family from the CATH database (v1.6) which contained 18,556 domains with 1028 homologous superfamilies, which could be further clustered into 672 fold groups and 36 distinct architectures (Pearl *et al.*, 2000). These domains were extracted from 13103 chains from 7703 PDB files.

Two types of alignment files were used for each parent and target alignment, one was aligned by sequence using the Needleman and Wunsch algorithm. The other was aligned by structure using SSAP (Taylor and Orengo, 1989). SSAP uses double-dynamic programming, at the residue level to align two protein structures (Harrison *et al.*, 2003). For each homologous family in CATH every representative at the N-level (near identical sequence representatives, NReps) was compared with every other NRep in that CATH family. This created a data set of approximately 56,000 alignment pairs. The data set was previously prepared by Dr. A.C.R. Martin.

These alignments were used to build two models for each pair; one from the sequence alignment and one from the structure alignment so that the potential energy value of both could be compared. It was expected that the models built from the structural alignment would be more accurate and that the RAM potential would rank these models as having lower energy.

The alignment files were run through a series of Perl programs that first prepared the correct files for MODELLER (v6). The model creation and analysis was controlled by a single program which submitted the files to the computer farm. Each model had a job file created for it, with a list of commands for the computer to carry out. An example of one of these files can be seen in figure 15. Qlite then used to queue the job files. The programs that were written and used for this can be found on the accompanying CD.

The job file, once run, caused a model to be created using MODELLER. The program Fitpdb.pl took that model and calculated its RMS using ProFit (Martin, 2001) to the corresponding structure in the Protein Data Bank. This value was then saved to a separate results file. The potentials_final.pl program calculated the potential energy of the model and wrote it to the same results file as the RMSD value. An overview of the potentials_final program can be seen in figure 16.

2.3.1 Testing the RAM potential

Using the RAM potential to analyze a large number of models it is possible to estimate how often it can correctly select the model based on the structural alignment by assigning it the lowest energy.

As can be seen in Table 2, 89.1% of the time the RAM potential was able to pick out the model created from the structural alignment by giving it the lowest potential value. One might expect that the RMSD for the model generated from the structural

```
f = /home/users/danielle/control_files
g = /home/users/danielle
modelname = 1hstA1awcAseq
pdbfile1 = 1hstA
topfile = $f/1hstA1awcAseq.top

cd $f
source /home/shared/ro/apps/modeller6/bashrc
/home/shared/ro/apps/modeller6/bin/mod6v2 $topfile
cd /home/users/danielle/
export DATADIR=/home/users/danielle/
$g/Fitpdb1.pl /data/pdb/pdb1hst.ent $f/${modelname}.B99990001.ent \
   A hst $g/data/ ${modelname}.zone
$g/potential -d $g/RAM.gen.par $f/${modelname}.B99990001.ent \
   >> $g/data/output.${modelname}
rm $f/${modelname}.B99990001.ent
rm $f/${modelname}.D00000001
rm $f/${modelname}.V99990001
rm $f/${modelname}.alignment
rm $f/${modelname}.ini
rm $f/${modelname}.log
rm $f/${modelname}.rsr
rm $f/${modelname}.sch
rm $f/$topfile
rm $f/${modelname}.zone
rm $f/${pdbfile1}${modelname}.pf
rm $f/profit.${pdbfile1}${modelname}
rm $f/$CompareCreateModeller
rm $f/${pdbfile1}${modelname}.txt
```

Figure 15: An example of a job file containing all the commands needed to create a single model and calculate the RMS and potential energy of that model.

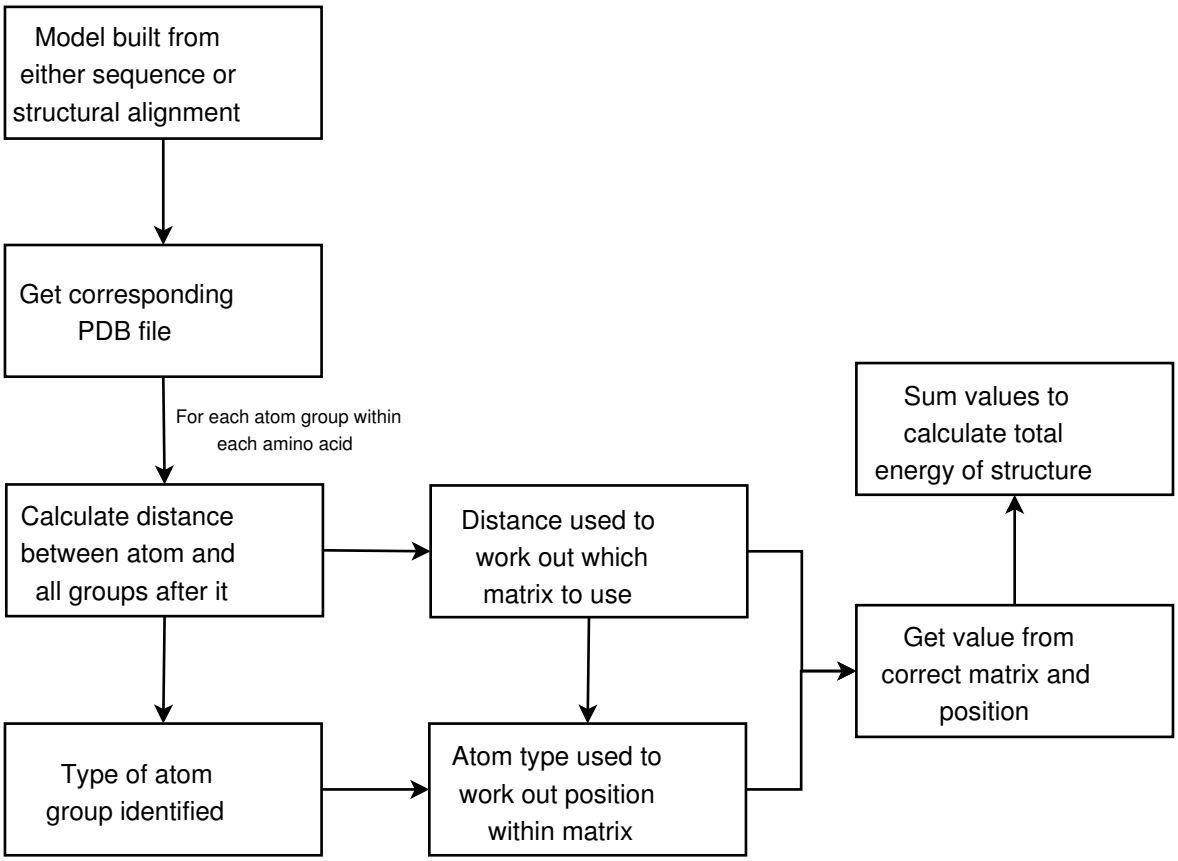


Figure 16: Overview of the potentials_final.pl program.

	Structural alignment model RMS lower	Structural alignment model RMSD higher	TOTAL
Structural alignment model potential energy lower	67.0%	22.1%	(89.1%)
Structural alignment model potential energy higher	6.1%	4.7%	(10.8%)
TOTAL	(73.1%)	(26.8%)	

Table 2: Percentage of proteins where the potential energy of the model from the structural alignment was lower or higher than its sequence aligned counterpart (these figures have been rounded down to one decimal place).

Sequence alignment

```
Seq 1   ACCHGGGILIKPRSVYILL
Seq 2   CCCHGG-IIIGGGRTTILL
```

Structural alignment

```
Seq 1  ACCHGGGILIKPRSVYILL
Seq 2  CCCHGGI-IIGGGRTTILL
```

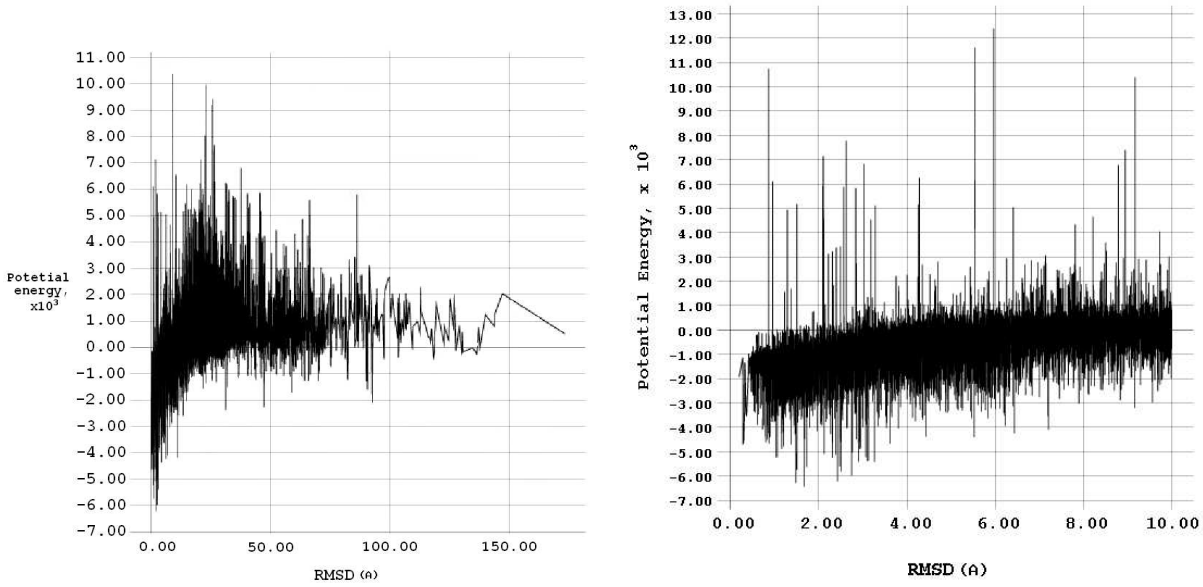
Figure 17: The single gap shift that caused the sequence alignment-based models to have a lower RMS than the structural alignment-based models.

alignment would always be lower than that from the sequence alignment. However this was only true 73.1% of the time. As this was unexpected some of the cases where this happened were examined by hand. In these cases it was found that the difference in RMSD between the two models was usually less than 0.05Å.

A program was written to examine all of the cases where the sequence alignment formed a better model than the structural one. In all cases it was the shift of a single gap between structural and sequence alignment that formed the only difference between the alignments as illustrated in figure 17.

In fewer than 5% of cases were both the potential energy and RMS deviation higher for the model produced from the parent and target being aligned by sequence. All the RMSD values are given in Ångströms and calculated over the carbon alpha chain of the protein structures.

Looking at a graph (Figure 18a) of the potential energy plotted against the RMSD for all proteins regardless of whether they were aligned by sequence or structure, no clear trend of lower potential energy being equivalent to a lower RMSD can be seen. However this graph contains the values for several protein models that are partially unfolded due to tail regions. These tail regions have been taken into account when calculating the RMSD and the potential energy, therefore giving very high values for both. If only those models with a RMSD of 10Å or lower are plotted (Figure 18b)



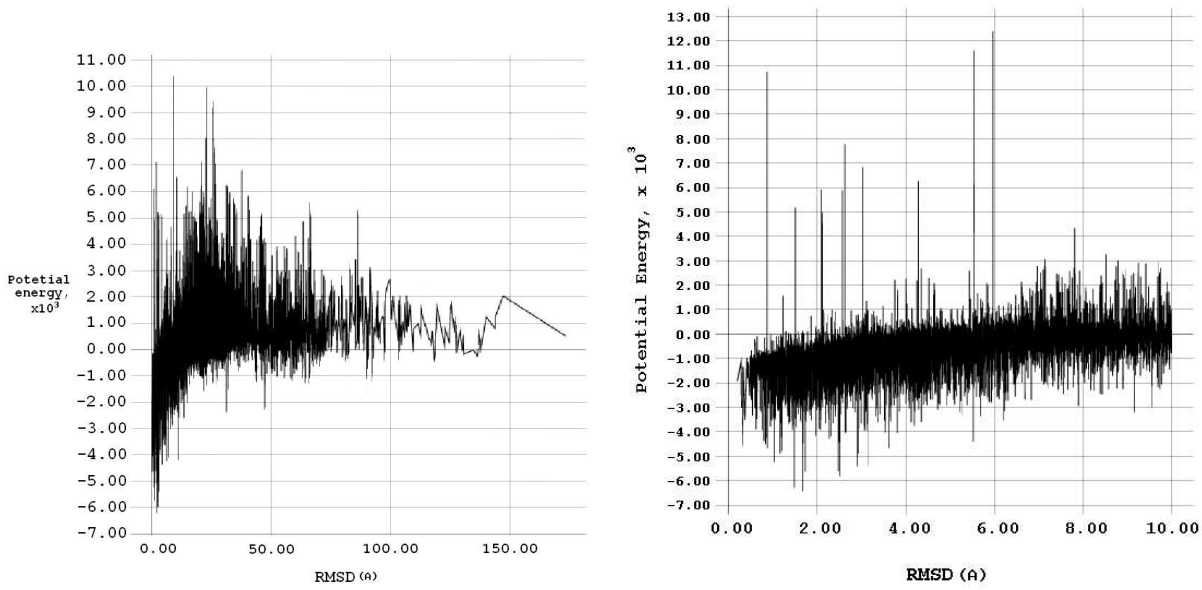
a) b)

Figure 18: The potential energy (in kcal/mol) of models created from both structural alignment and from sequence alignment plotted against the RMSD (in Ångströms). a) all models b) only those models with RMSD < 10.0Å.

then a trend for lower potential energies equating to lower RMSD values is much more clearly seen.

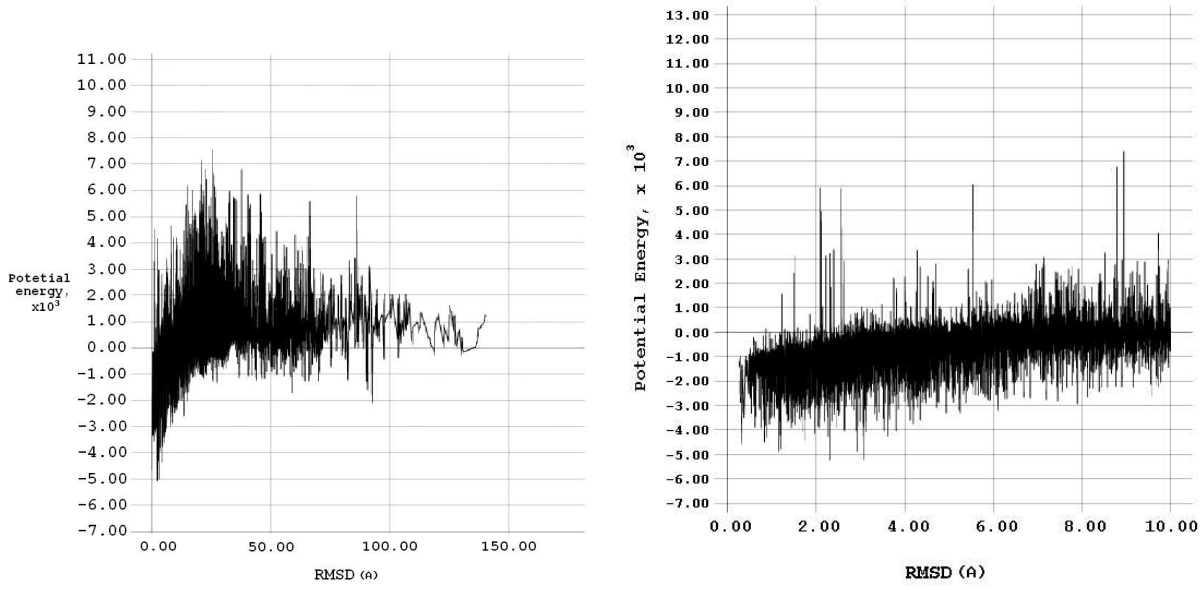
Separating the models into those aligned by sequence and those aligned by structure the same trend can be seen. Figure 19 shows the RMSD against the potential energy for models produced from a sequence alignment. Figure 20 shows the RMSD against potential energy for models produced for structural alignments.

The distribution of the potential energies is approximately normal. Figure 21 shows this distribution for all the models created, regardless of their RMSD value or whether they were created from a sequence or structural alignment. The only limit for the graph is that all the models had a potential energy of less than 200 kcal/mol. There were only a few models, those with unfolded portions, which had a potential energy in excess of 200 kcal/mol. Once again separating the results of the sequence (Figure 22) and structural (Figure 23) alignments the same normal distribution can be seen. The peak for the structurally aligned models is slightly more negative than for the models



a) b)

Figure 19: The potential energy (in kcal/mol) of models created from sequence alignment plotted against the RMSD (in Ångströms). a) all models b) only those models with RMSD < 10.0Å.



a) b)

Figure 20: The potential energy (in kcal/mol) of models created from structural alignment plotted against the RMSD (in Ångströms). a) all models b) only those models with RMSD < 10.0Å.

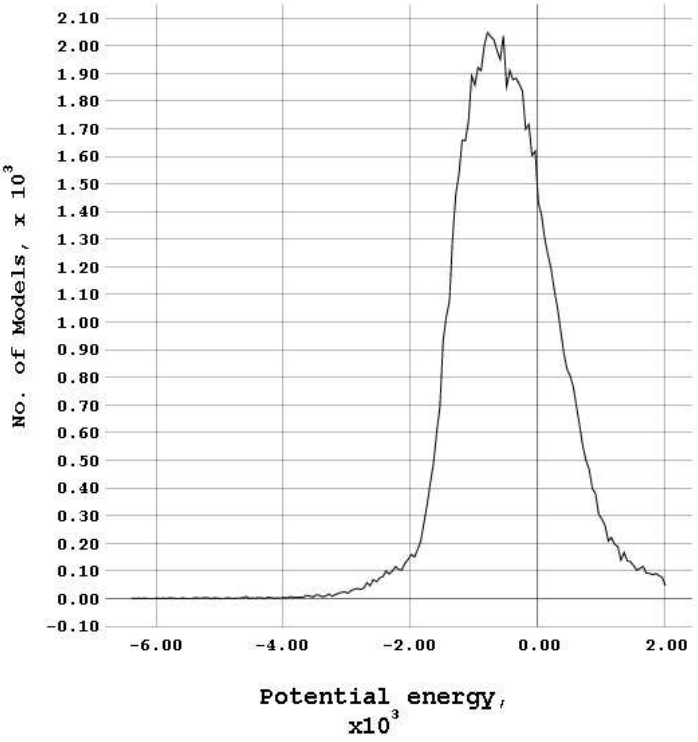


Figure 21: The distribution of the potential energies (in kcal/mol) of all models with a potential energy of less than 2000.

created by a sequence alignment. This indicates that on average the models produced from a structural alignment had a lower energy as evaluated by the RAM potential than their sequence aligned counterparts.

On the whole the RAM potential seems capable of selecting between models by assigning the one from the structural alignment with the lower energy. However it is not able to do this consistently. As this section has shown there is a trend in the models of lower potential energy being equivalent to a lower RMSD value. However there are exceptions to this. Some of these exceptions could be due to the slight differences in RMSD values between sequence alignment-based models and structural

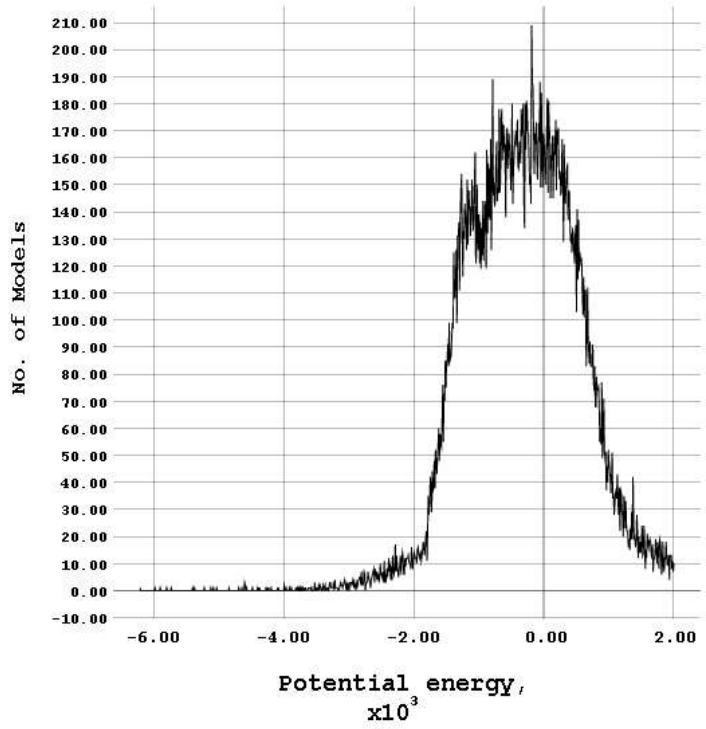


Figure 22: The distribution of the potential energies (in kcal/mol) of all models produced from a sequence alignment and with a potential energy of less than 2000.

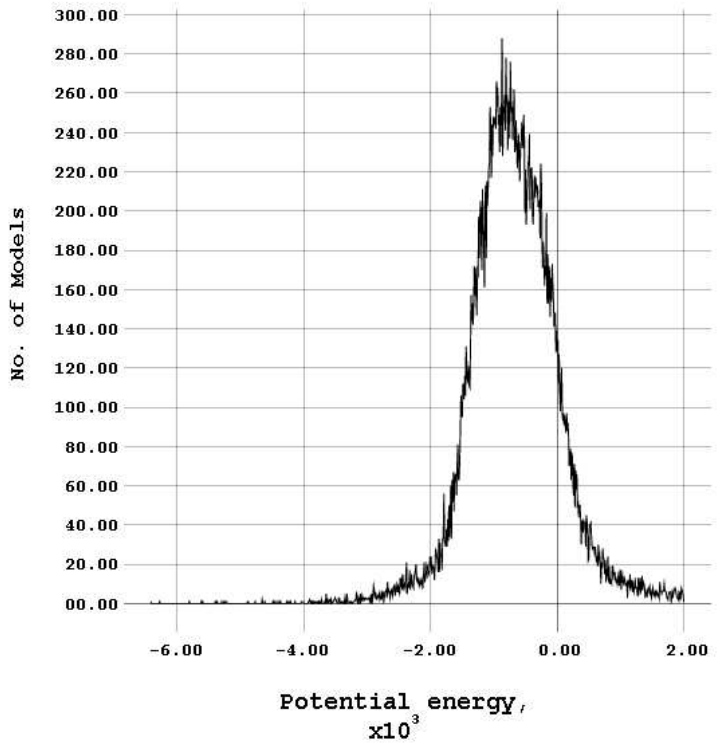


Figure 23: The distribution of the potential energies (in kcal/mol) of all models produced from a structural alignment and with a potential energy of less than 2000.

alignment-based models where only a single gap shift separates the two alignments.

2.4 Testing the potentials program with varying alignments

The potentials program was also tested to see if it could choose the most likely model of a protein from within a number of slightly different alignments. The alignments from which the assessed models were made had only slight differences between them. Therefore the models would only be slightly different. If the RAM potential was unable to choose between these subtle differences in structure it would not prove to be very useful for future work.

A Perl program called Random_alignment.pl, which can be found on the accompanying CD, was written that was able to create alternative alignments from a single alignment file. It takes an alignment and introduces a random insertion somewhere within it. The program randomly chooses the position and length (between 1 and 5) of the insertion in both the parent and target sequences. An overview of this program can be seen in figure 24.

The A chain of the protein 1hst (Ramakrishnan *et al.*, 1993) (chicken histone protein) and the A chain of 1awc (Batchelor *et al.*, 1998) (mouse GA-binding protein) were chosen because of the large variation between the RMS and potential energy value of their structurally aligned and sequence aligned models. The A chain of 1hst was chosen to be the target with the A chain of 1awc as the parent structure. Both sequences have a structure that has been solved by x-ray crystallography, allowing for comparison between model and experimentally solved structure.

The Random_Alignment.pl program was run multiple times to create one hundred

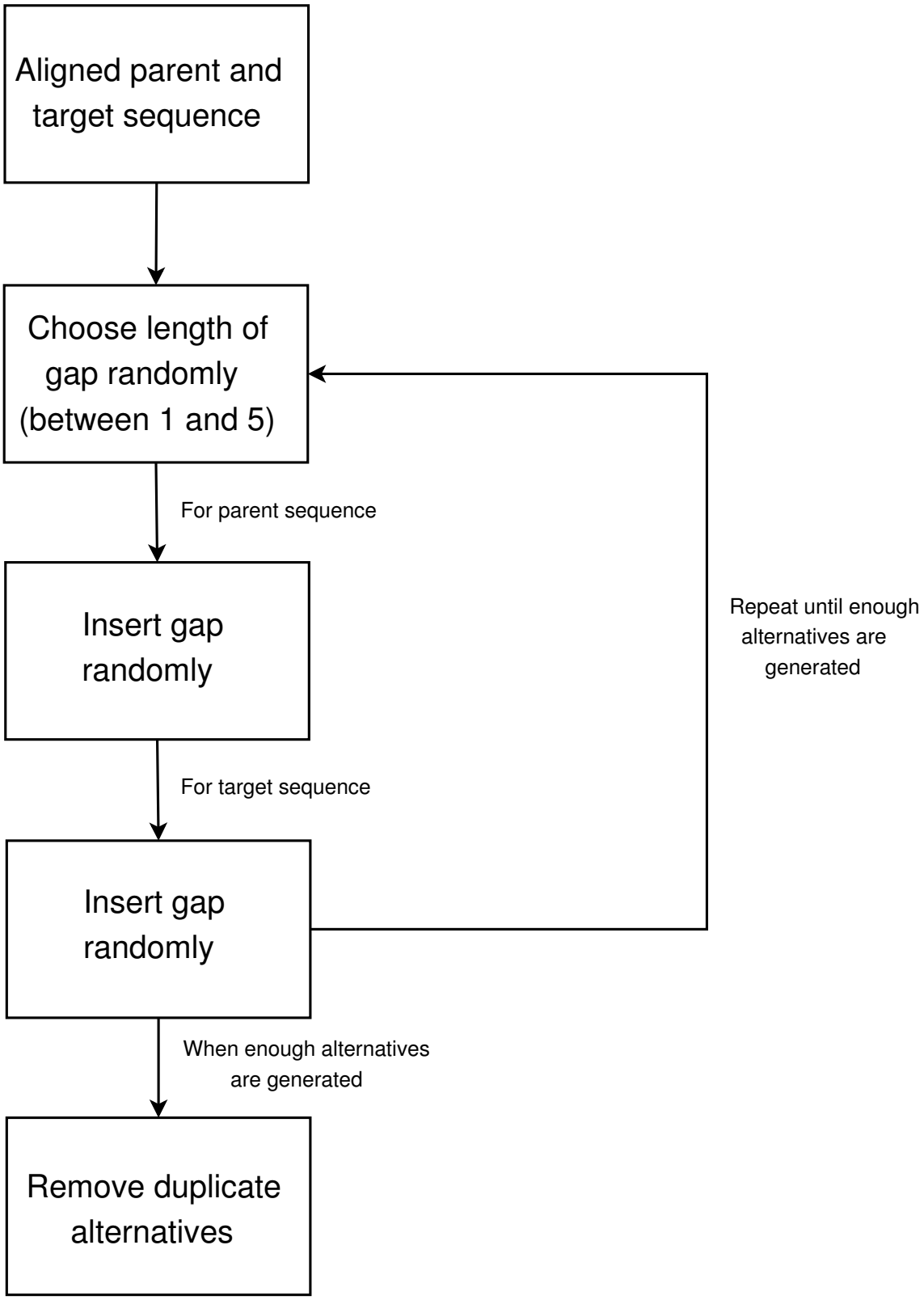


Figure 24: Overview of the Random_alignment.pl program. This program introduced random insertions into the sequence and structural alignments.

and seventy different alignments each from the original sequence and structural alignments. The program also created the control files needed and the models were once again produced by MODELLER. Each model was then run through the potentials program to calculate its potential energy, before the RMS deviation between it and the experimentally determined structure was calculated using ProFit.

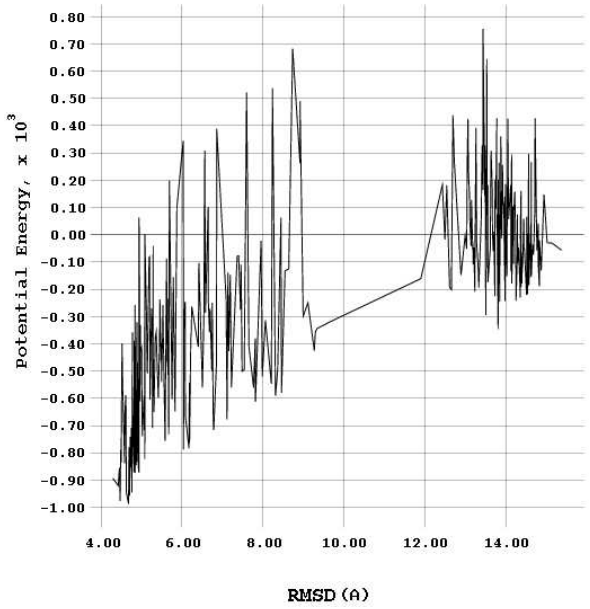
2.4.1 Results of testing with varying alignments

One hundred and seventy permuted alignments were created for both the structural and sequence alignment of 1hstA and 1awcA (the A chains of chicken histone and mouse GA-binding protein respectively). Unlike the comparison of models that were aligned either via structure or sequence the difference between these models was far more subtle, providing a more difficult test of the potential set.

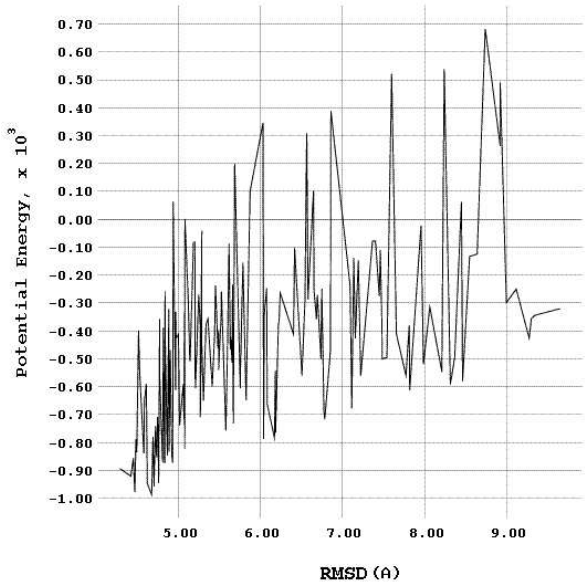
The potential energy of the models created from these variations on the original alignments can be seen in figure 25a. Figure 25b just shows the data for alignments based on the original structural alignment while figure 25c shows data for alignments based on the sequence alignment. The trends are quite good when based on the structural alignment but poor when based on the sequence alignment.

The distribution of the potential energies achieved by these models can be seen in Figure 26. Unlike the large scale analysis of radically different alignments the distribution does not appear to be normal. This was probably due to the difference in the number of models made, as far fewer were produced for the variable alignments.

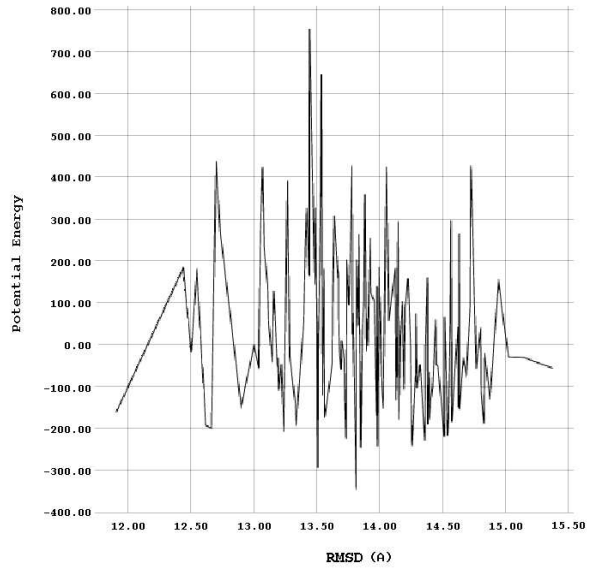
The RAM potential did not perform as well in distinguishing between subtly different structures as it had with the earlier larger scale analysis. This would suggest that while the RAM potential can select the better model when the differences between the models are significant it cannot perform as well when the differences are small.



a)

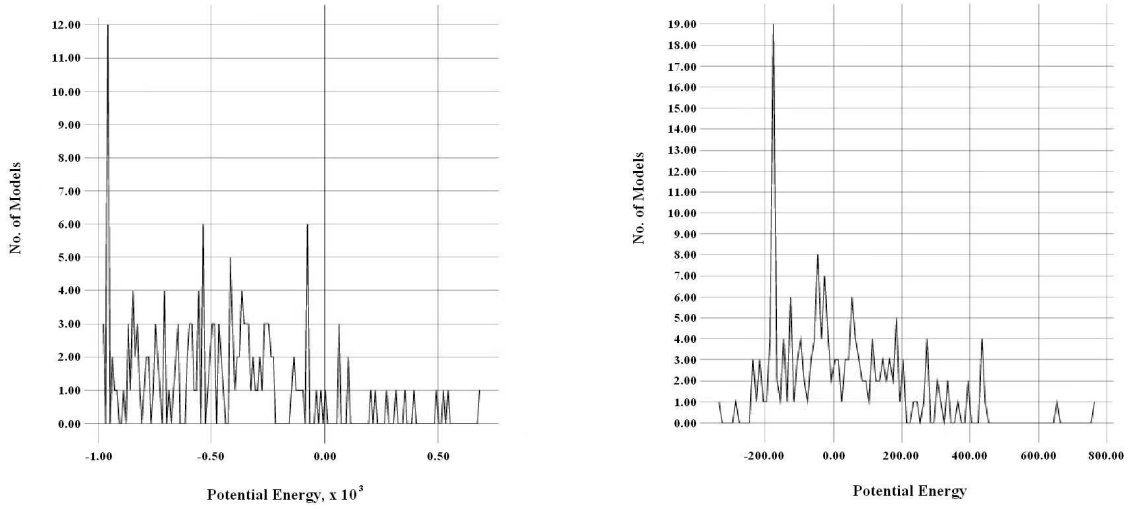


b)



c)

Figure 25: The potential energy (in kcal/mol) plotted against the RMSD (in Ångströms) of a series of models of 1hst, chain A, each with slightly different alignments, generated from a) the original sequence and structural alignments, b) the original structural alignment and c) the original sequence alignment.



a) b)

Figure 26: The distribution of the potential energies (in kcal/mol) calculated for the models based on a) the original structural alignment and b) the original sequence alignment.

2.5 CASP 5 Experiment

2.5.1 CASP

Every two years an experiment known as CASP (Critical Assessment of Structure Prediction) (Moult *et al.*, 1997; Moult *et al.*, 1999) is run to get a better idea of the current state of protein modelling. The CASP experiment attempts to address the problem of comparing and evaluating models in order to learn how accurate and reliable the model producing methods are (Siew *et al.*, 2000). There have been six experiments completed so far. CASP was set up partly because of claims in the literature at that time that the protein folding problem had been 'solved' without producing tangible benefits, since most of the 'solutions' included a strong dependence on the test set (Samudrala and Levitt, 2002). CASP provides a large-scale, blind environment for protein structure prediction techniques to be tested and analyzed. The number of groups participating in CASP has grown steadily as can be seen in table 3.

In CASP, a few dozen proteins of known sequence, but unknown structure, are

CASP Experiment	Year	No. of groups and prediction servers contributing
CASP1	1994	35
CASP2	1996	72
CASP3	1998	98
CASP4	2000	163
CASP5	2002	215
CASP6	2004	208 (unique)

Table 3: The increasing number of groups participating in the CASP experiment.

used as prediction targets (Fischer *et al.*, 2000). At the same time as groups attempt to model the proteins, others are solving the structure with x-ray crystallography or NMR for a final comparison. All types of methods for predicting protein structure are considered, including comparative modelling, fold recognition and *ab initio* prediction. The predicted structures must be submitted before the actual structure is released and are then compared with the solved structure.

In the most recent completed experiment, CASP6 held in 2004, there were 201 groups registered and a further 65 prediction servers with 87 target sequences to work on. Some of these sequences had relatively high sequence identity with proteins of known structure in the Protein Data Bank, so choosing a parent protein was not a difficult task. However other proteins had only low sequence identity with the sequences of existing structures, making their structure far more difficult to predict. Such sequences were modelled using threading, fold recognition or *ab initio* techniques.

For CASP5, the combined bioinformatics groups at Reading entered several models created over the course of a week through a combination of manually corrected alignments and MODELLER. The RAM potential was used in order to select between alternative alignments and its success was then evaluated. There was a total of 67 target sequences to work on in CASP5. These sequences were divided up between the group in order to find out which ones we were most likely to be able to model.

The sequences were then run through both BLAST and Psi-BLAST to identify parent structures.

Once the BLAST searches were completed the sequences were also run through a number of secondary structure prediction servers as well as threading and fold-recognition servers. This was done to see whether their results would agree with the templates found through Psi-BLAST and BLAST. The secondary structure prediction servers available for use were PSI-Pred (McGuffin *et al.*, 2000), JPred (Cuff *et al.*, 2000), NNPredict (Kneller *et al.*, 1990), GOR-IV (Garnier *et al.*, 1996) and PHD (Rost and Sander, 1993), while the threading and fold recognition servers used were GenThreader (McGuffin *et al.*, 2000), 3D-PSSM (Kelley *et al.*, 2000), FUGUE (Shi *et al.*, 2001), and SAM-T99 (Karplus and Hu, 2001). The results of using these servers and the BLAST and Psi-BLAST searches divided the target sequences into three groups, those that appeared as though they could be easily modelled, those that were possible with some work and those which had no obvious parent structures. The sequences in the first two categories were attempted while those in the third were set aside.

After a suitable parent structure had been found, its structure and function were looked at in greater detail. Important sections of secondary structure for stability and function could then be identified. These sections are the ones likely to be conserved between related proteins. Knowing where these secondary structures were would later help during the alignment phase of comparative modelling.

Parent and target sequences were aligned using ClustalW (Thompson *et al.*, 1994). The alignment was then examined and corrected by hand so that insertions and deletions did not occur in the middle of secondary structure elements, but where possible in loop regions. The information gathered about the parent structure was used in this phase so that close attention was paid to regions that were structurally or functionally important. For some sequences multiple alignments were created.

To produce the models, MODELLER was used through the interface Mint (Martin,

1995). This interface was used so that only an alignment file was needed in order to create a model. Mint itself creates the control file and calls the MODELLER program. When run on each alignment MODELLER had the DO_LOOPS option turned on. This option caused the program to create several structures that differed only in the loop regions from a single alignment.

The protein model output of MODELLER could then be examined using Rasmol (Sayle and Milner-White, 1995). This allowed for comparison between parent and model as well as looking for anything likely to be poorly predicted such as an unfolded tail. If the model did contain a section of unfolded chain then it was possible to enter the model, but specify that analysis should take place only over a certain number of amino acids which would not include the tail area.

The program Procheck (Laskowski *et al.*, 1993) was used to produce a Ramachandran plot to discover what percentage of the residues were in their most favoured positions. An example of a Ramachandran plot created for one of the CASP entries can be found in figure 27. Procheck identifies unusual bond angles which could indicate any areas where the model was incorrect.

Models were also evaluated using the RAM potential (Samudrala and Moult, 1998). The models with the lowest potential energy value were then chosen to be submitted to the CASP competition. Multiple models could be submitted, but had to be ranked by the submitters; the RAM potential was used for this purpose.

2.5.2 Results: CASP5

The group at Reading submitted forty-eight models in the CASP5 experiment, covering a total of nineteen target sequences. In some cases only a single model was submitted while in others up to five slightly different models where entered. The differences in the various models for each sequence came from varying alignments or from the same

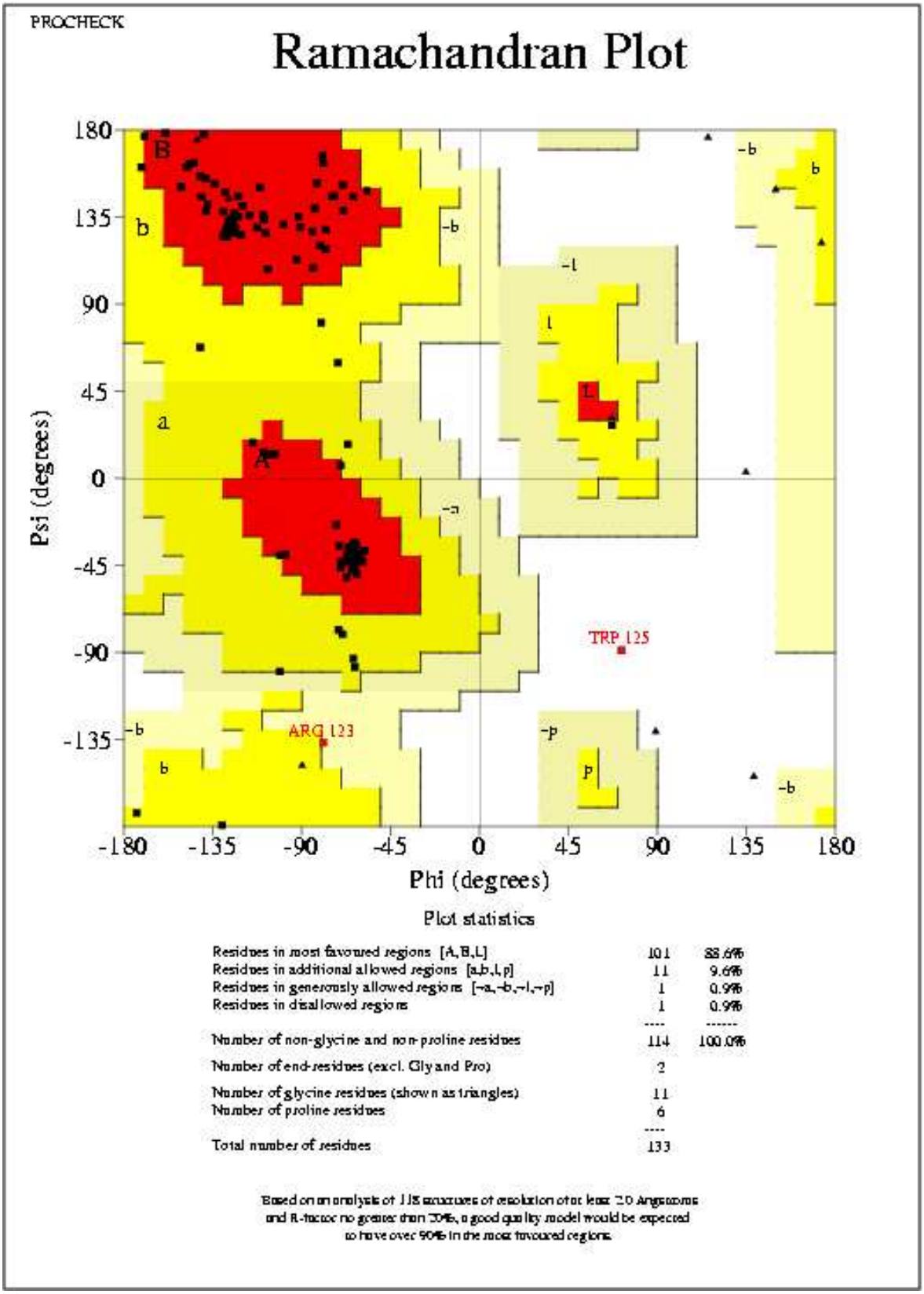


Figure 27: An example of a Ramachandran plot of CASP5 target T0149. The amino acids (black squares) that appear plotted in the red areas of the graph are in one of their most favoured positions. The tryptophan (TRP125) and arginine (ARG123) that are outside of these regions are in positions that are either not allowed by their structure or are allowed but only rarely due to clashes or strains.

alignment but with different loops created by MODELLER. The overall results of the CASP5 entry can be found in table 4.

Target name	Parent structure(s)	SeqID	RMSD	Total potential energy of model	Total potential energy of structure
T0130_1	1f5aA 1fa0A	15%	9.97	133.33	2746.25
T0130_2	1f5aA 1fa0A	15%	15.26	214.06	2746.25
T0130_3	1f5aA 1fa0A	15%	14.21	149.06	2746.25
T0130_4	1f5aA 1fa0A	15%	13.09	207.52	2746.25
T0130_5	1f5aA 1fa0A	15%	14.17	565.317	2746.25
T0133_1	1hg5A 1hxbA	13%	12.35	-1937.67	-6280.06
T0133_2	1hg5A 1hxbA	13%	12.22	-1842.45	-6280.06
T0137_1	2ansB	43%	1.02	-676.55	-1036.44
T0137_2	2ansB	43%	1.08	-677.07	-1036.44
T0137_3	2ansB	43%	1.30	-664.98	-1036.44
T0142	1i9zA	26%	3.49	-1294.62	-2888.51
T0149_1	1fts 2ffhC 1jpnA 1j8mF	15%	17.28	658.40	-5471.10
T0149_2	1fts 2ffhC 1jpnA 1j8mF	15%	17.03	1509.90	-5471.10
T0149_3	1fts 2ffhC 1jpnA 1j8mF	15%	17.40	1545.34	-5471.10
T0150_1	1ck8B	37%	2.70	-2337.05	-2635.34
T0150_2	1ck8B	37%	2.66	-2336.75	-2635.34
T0152_1	1bgbA	17%	5.22	-135.86	-1652.02
T0152_2	1bgbA	17%	5.26	-94.67	-1652.02
T0152_3	1bgbA	17%	5.21	-94.60	-1652.02
T0153_1	1euwA 1f7rA	34%	5.64	-1524.44	-1788.67
T0153_2	1euwA 1f7rA	34%	5.34	-1404.04	-1788.67
T0153_3	1euwA 1f7rA	34%	5.37	-1402.87	-1788.67
T0153_4	1euwA 1f7rA	34%	5.51	-1270.48	-1788.67
T0154_1	1ihoA	37%	6.95	-2921.82	-4439.26
T0154_2	1ihoA	37%	6.85	-2970.73	-4439.26
T0155	1dhn	33%	6.03	-1270.23	-1356.91
T0160_1	3mspB 2mspB 1grwD	22%	7.26	-744.34	-2330.65
T0160_2	3mspB 2mspB 1grwD	22%	6.83	-677.81	-2330.65
T0160_3	3mspB 2mspB 1grwD	22%	7.16	-743.96	-2330.65
T0160_4	3mspB 2mspB 1grwD	22%	7.43	-637.30	-2330.65
T0160_5	3mspB 2mspB 1grwD	22%	6.85	-675.51	-2330.65
T0164_1	2bnh 1a4yD	19%	6.00	1338.82	-1330.71

Table 4 The overall results of the group's CASP5 entry (all RMSDs are calculated in Ångströms using ProFit (calculated over the regions specified as confident), all potential energies in kcal/mol) to 2 d.p. Continued...

Target name	Parent structure(s)	SeqID	RMSD	Total potential energy of model	Total potential energy of structure
T0164_2	2bnh 1a4yD	19%	5.89	1357.18	-1330.71
T0164_3	2bnh 1a4yD	19%	5.92	1352.75	-1330.71
T0166_1	1jgsA	20%	cancelled	-2720.07	cancelled
T0166_2	1jgsA	20%	cancelled	-2102.18	cancelled
T0167_1	1jeoA 1jxaC	35%	5.44	-1213.22	-3003.94
T0167_2	1jeoA 1jxaC	35%	4.90	-1334.70	-3003.94
T0167_3	1jeoA 1jxaC	35%	5.15	-1221.40	-3003.94
T0167_4	1jeoA 1jxaC	35%	4.80	-1311.70	-3003.94
T0171_1	1a8q 1a88C 1broB 1brt 1a8uB	21%	9.20	-1917.65	-3319.92
T0171_2	1a8q 1a88C 1broB 1brt 1a8uB	21%	9.66	-1319.99	-3319.92
T0179_1	1jq3A	42%	5.45	-2122.61	-2878.57
T0179_2	1jq3A	42%	5.48	-2250.15	-2878.57
T0182	1mat	41%	1.42	-2444.99	-2932.44
T0184_1	1jfzC	33%	3.86	-3540.66	-4688.91
T0184_2	1jfzC	33%	3.88	-2505.20	-4688.91
T0188	1eo1A	29%	2.32	-1924.56	-2722.91

Table 4: (continued) The overall results of the group's CASP5 entry (all RMSDs are calculated in Ångströms using ProFit (calculated over the regions specified as confident), all potential energies in kcal/mol) to 2 d.p.

The group had varying degrees of success in the CASP5 experiment. In some cases the RMSD between model and actual structure was as low as 1.02 Å (from a model of target T0137, a fatty acid binding protein from *E. granulosus* with a sequence identity of just 43%). The majority of the models created had RMSD values of between 4 and 7 Å. In all cases the energy of the model, calculated by the RAM potential, is lower for the experimentally determined structure than it is for the modelled structures. Lower potentials, as calculated by the Ram potential, tended to equate to models with lower RMSD values. Overall it appears as though the models created from a single parent structure fared better than those created from multiple templates.

The models for the competition were sent with the request not to include the protein tails in the calculations.

Target 184 had a long unfolded tail (residues 165-240) which was requested not to be included in the calculations. If this tail was included, the RMSD increased from approximately 3.85 Å to more than 52 Å (figure 28).The potential energy also becomes more negative for the models if it is calculated over the folded section alone, rather than taking into account the unfolded region.

For those models where multiple models were submitted it is possible to see how the RAM potential performed distinguishing between subtle differences in the models. It would be hoped that the model with the lowest RMSD would also be the one with the lowest potential energy. Table 5 shows whether this was the case for those targets with multiple models. The table also has the ranking which is the count of how many models are better than the model with the lowest RAM energy plus one. Thus when the lowest energy model is also the lowest RMSD, then the ranking will be 1. The table shows how well the RAM potential did at distinguishing the models with the lowest RMSD by giving them the lowest energy.

In the majority of cases ($> 70\%$) the RAM potential failed to pick the model with the lowest RMSD. Again, this suggests that although the RAM potential is capable

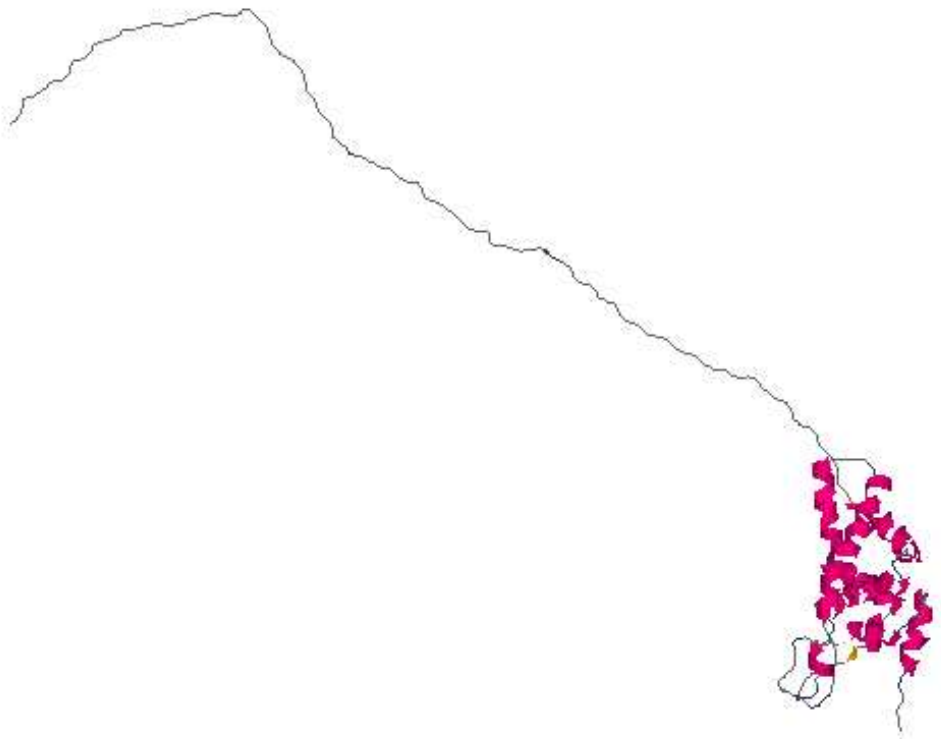


Figure 28: One of the modelled structures for CASP target 184. The reason for its high RMSD is the long unfolded tail region.

Target name	Lowest RMSD model	Lowest RAM potential model	Ranking
T0130	T0130_1	T0130_1	1
T0133	T0133_2	T0133_1	2
T0137	T0137_1	T0137_2	2
T0149	T0149_2	T0149_1	2
T0150	T0150_2	T0150_1	2
T0152	T0152_3	T0152_1	2
T0153	T0153_2	T0153_1	4
T0154	T0154_2	T0154_2	1
T0160	T0160_2	T0160_1	4
T0164	T0164_2	T0164_1	3
T0166	cancelled	cancelled	cancelled
T0167	T0167_4	T0167_2	2
T0171	T0171_1	T0171_1	1
T0179	T0179_1	T0179_2	2
T0184	T0184_2	T0184_1	1

Table 5: The lowest RMSD model and lowest potential (as calculated by the RAM potential) model.

of distinguishing between models with large differences it does not perform so well choosing between models with only subtle differences. This supports the view from the experiments on 1hstA/1awcA.

The majority of the models produced and submitted to the CASP5 experiment appear to have been fairly good. Those few exceptions with unusually high RMS deviations appear to be due to unfolded tails, incorrectly modelled loop regions or alignments that were not correct. CASP5 was a useful blind test of the potentials program as it highlighted the RAM potential's inability to distinguish between models with only small differences.

2.6 Conclusions and Discussion

Models created from a structural alignment are more correct than those created from a sequence alignment. In the cases where a model from the sequence alignment was

'better', the two models were, in fact, virtually identical. The RAM potential is capable of distinguishing correctly between these models the majority of the time.

When random inserts were introduced into the sequence and structural alignments of 1hstA and 1awcA it lead to the creation of models that had only subtle differences between them. When these models were analyzed the RAM potential was no longer capable of distinguishing between them consistently.

This was again shown in the CASP5 experiment where the RAM potential was used to choose between a number of models built from different possible alignments between the targets and the parents. The models that were built from these chosen alignments did not always follow the pattern of lowest potential energy = lowest RMSD.

Overall it seems as though the RAM potential can be used to distinguish between different models where the differences are quite pronounced. However where the differences are more subtle the RAM potential is not as capable of choosing between the different models correctly.

Chapter 3

Misleading Local Sequence Alignments (MLSAs)

As previously stated, one of the major causes of error in modelling proteins is obtaining the correct alignment between parent and target. If a model is to be accurate then it is critically important that the threading of the target sequence onto the parent(s) is correct. A misalignment is where the sequence alignment and the structural alignment differ in some way. Some misalignments are minor while other misalignments are far more severe and can affect the final model greatly. Even a mean alignment shift of a single residue results in a best possible RMSD of around 3.8 Å(the distance between two adjacent alpha-carbon atoms) (Martin *et al.*, 1997).

MLSAs (Misleading Local Sequence Alignments) are an extreme case of misalignment which were originally studied by Saqi *et al.* (1998). They are regions where structural similarity is clear and where the optimal sequence similarity was substantially higher than that seen in the structure alignment (Saqi *et al.*, 1998). These misaligned areas are not restricted to protein pairs with particularly high or low overall sequence similarity. A good example of a MLSA occurs within an aligned pair of cytochrome B reductase from pig and *E.coli* (Saqi *et al.*, 1998). As figure 29 shows, the

```
Pig      V D L V I K V Y F K D T H P -
           \ \     \ \ \ \ \     \ \
E.coli   - F D L L V K V Y F K N E H P
```

Figure 29: The structural alignment of cytochrome b reductase from two different species. The lines connecting the two sequences indicate what the sequence alignment would be. Figure adapted from Saqi *et al.* (1998).

sequence alignment seems obvious, nine out of fourteen residues are the same in the two sequences. However the true structural alignment shows that the two sequences are actually shifted by one residue.

3.1 Identifying MLSAs

Software, written by Dr. A.C.R. Martin, was used to extract MLSAs. It was decided only to look at the most extreme MLSAs to see the basic reasons why they occur. Also it was necessary to have a data set small enough for manual analysis.

Structural alignments were created using SSAP while the sequence alignments were created using nw (Martin, 1990), an implementation of the Needleman and Wunsch algorithm. By aligning each pair of NReps within each homologous superfamily in the CATH database (v1.6), 56,510 pairs were created. Each NRep is a representative of a near identical group of sequences having $\geq 95\%$ sequence identity (Martin, 2000b).

MLSAs were searched for by looking for short segments of 10 amino acids containing at least 5 matches which were not aligned. If it is assumed that all amino acids appear with the same frequency within a sequence then the probability of obtaining a given number of matches within a protein window of a given length is given by the equation:

$$P =_N C_n \times p^n \times (1-p)^{N-n} \tag{12}$$

where p is the probability of a match by chance, in this case $1/20$, N is the size of

the window, n is the number of matches and ${}_NC_n$ is the number of combinations of n elements chosen from N. So for a window of 10 residues with 5 matches and a p value of 1/20:

$$P = 5.0935 \times 10^{-5} \tag{13}$$

Thus the chance of finding a group of 5 matches within a window of ten residues by chance is very small ($P < 0.0001$).

The 10-residue window was slid along sequence 1 of the structural alignment. For each position i, a 10-residue window was placed at positions i - $10 \leq j \leq i + 10$ in the second sequence of the structural alignment. A MLSA was recorded if when comparing these windows, $i \neq j$, the window scored at least 5 matches in sequence 1 and if it scored more matches with both windows at position i or position j. If a MLSA was identified then the window was jumped along by a full ten residues to avoid identification of multiple overlapping MLSAs. An example of this can be seen in figure 30.

Of the 56,510 pairings, 8%, or roughly 4,500 pairs, fit the initial parameters for a MLSA (5 out of 10 residues misaligned). This number was reduced by looking at those pairings that had six or more amino acids mis-aligned within the ten residue window. This gave a data set of eighty-two, of which thirty-one pairs scored at least twice as well in the sequence alignment as they did in the structural alignment. These were then analyzed manually to identify genuine MLSAs (table 6), here, and elsewhere, domains are identified as in CATH. 22 sequences were discarded at this final stage for a variety of reasons. Some were due to errors in CATH domain assignments. Others were caused by errors in the SSAP structural alignments. Still others were caused by arbitrary structural alignments, such as having highly flexible regions within the protein structure. After this process, only nine protein pairs remained in the data set: the most extreme cases of MLSA.

```
            Structural alignment

                     10        20        30
                     |         |         |
            **********
      Seq 1  ACTSRVNMYLSVDSTLRSTRSCVSLMNPQL
      Seq 2  TCRACTSRPRSVDYHILACRSCVSCMNTMS

Windows at i - 10 < j < i + 10 in Seq 2

   j          Seq2 window    NMatch
   i + 0      TCRACTSRPR     1
   i + 1      CRACTSRPRS     0
   i + 2      RACTSRPRSV     0
   i + 3      ACTSRPRSVD     5
   i + 4      CTSRPRSVDY     0
   i + 5      TSRPRSVDYH     1
   i + 6      SRPRSVDYHI     1
   i + 7      RPRSVDYHIL     1
   i + 8      PRSVDYHILA     0
   i + 9      RSVDYHILAC     0
   i + 10     SVDYHILACR     0
```

Figure 30: An example of how the sliding windows identified MLSAs. In this case the window in Seq 1 being examined is ACTSRVNMYL. Looking at the windows in sequence 2 it can be seen that $j = i + 3$ scored 5 matches. Where $j = i$, there was only one match. If the window in Seq 2 at $j = i$ is compared with its currently aligned window (SRVNMYLSVD) it only scores 3 matches. Therefore this would be flagged as a MLSA.

Structure Pair		Sequence Score	Structure Score	Structure Alignment	Sequence Alignment
1noyA1	1waj01	10	0	KRMEDIGLEA	KRMEDIGLEA
				WI--KRMEDI	KRMEDIGLEA
1ucyH0	1etrH1	10	0	IVEGQDAEVG	IVEGQDAEVG
				GQDAEVGLS-	IVEGQDAEVG
2hntC0	1ahtH1	10	1	IVEGSDAEIG	IVEGSDAEIG
				------IVEG	IVEGSDAEIG
1envA0	1eboA0	10	4	QIEDKIEEIL	QIEDKIEEIL
				EILSKIYHIE	QIEDKIEEIL
1envA0	1eboA0	10	4	KIYHIENEIA	KIYHIENEIA
				EIARIKKLIG	KIYHIENEIA
1ltsC0	1xtcC0	9	2	VKRQIFSDYQ	VKRQIFSDYQ
				YQSDIDTHNR	VKRQIFSGYQ
1aksA0	1sluB1	8	1	IVGGYTCAAN	IVGGYTCAAN
				SVVALPSSCA	IVGGYTCQEN
1mmd06	1mma06	8	1	GFPNRIIYAD	GFPNRIIYAD
				FPNRIIY--A	GFPNRIIY--
1sriA0	1swfA0	8	2	GRYDSAPASG	GRYDSAPASG
				DSAPATDGSG	GRYDSAPATD
1envA0	1eboA0	8	4	IKKLIGEARQ	IKKLIGEARQ
				ADGLIEGLRQ	IKKLIGEADG
1aksA0	1a0jA1	7	0	IVGGYTCAAN	IVGGYTCAAN
				SAVALPSSCA	IVGGYECRKN
1hgu00	1hwgA0	7	0	KQTYAKFDNS	KQTYAKFDNS
				QTYSKFD---	KQTYSKFD--
1ab9B0	1azzA1	7	1	IVNGEEAVPG	IVNGEEAVPG
				SWVGLPSTDV	IVGGVEAVPN
1hgu00	1hwgA0	7	1	–ESIPTPSN	–ESIPTPSN
				SIPTPS---N	FSESIPTPS-
1isaA0	3sdpA0†	7	1	VTNLNNLIKG	VTNLNNLIKG
				YVVNLNNLVP	VVNLNNLVPG
1bmfD3	1skyE3†	7	3	VLIMELINNV	VLIMELINNV
				VGKTVLIQEL	VLIQELIHNI
1ab9B0	1fujA1	6	0	IVNGEEAVPG	IVNGEEAVPG
				AQPHSRVQLP	IVGGHEAQPH
1isaA0	3sdpA0†	6	0	ELPALPYAKD	ELPALPYAKD
				PPLPYAHDA-	--PPLPYAHD
1vewA0	3sdpA0†	6	0	TLPSLPYAYD	TLPSLPYAYD
				PPLPYAHDA-	--PPLPYAHD
5ldh01	9ldtA1	6	0	ATLKEKLIAP	ATLKEKLIAP
				-ATLKDQLIH	ATLKDQLIHN
1mfeL2	1ospL1†	6	1	QPKSSPSVTL	QPKSSPSVTL
				----DIQMSQ	QSSSSFSVSL

Table 6 Details of the thirty-one sequences identified as MLSAs. †The most extreme of the MLSAs. Continued...

Structure Pair		Sequence Score	Structure Score	Structure Alignment	Sequence Alignment
1vom06	1br2A6	6	1	GFPNRI----	GFPNRI----
				VFQEFRQRYE	GFPNRIVFQE
1vom06	1mma06	6	1	--IYADFVKR	--IYADFVKR
				KATDAVLKHL	-IIYADFQKA
1ak200	1akeA0†	6	2	LKATMDAGK-	LKATMDAGK-
				-SELGKQAKD	AKDIMDAGKL
1faiL1	1tcrA1	6	2	DRVTISCRAS	DRVTISCRAS
				ASLQLRCKYS	ARVTVSEGAS
1gafL2	1ae6L1†	6	2	TVAAPSVFIF	TVAAPSVFIF
				---DIVMTQA	TQAAPSVPVT
1ikfL1	1tcrA1	6	2	DRVTISCRAS	DRVTISCRAS
				ASLQLRCKYS	ARVTVSEGAS
1mamL1	1tcrA1	6	2	DRVTISCRAS	DRVTISCRAS
				ASLQLRCKYS	ARVTVSEGAS
1nmbL0	1tcrA1	6	2	DRVTISCRAS	DRVTISCRAS
				ASLQLRCKYS	ARVTVSEGAS
1vgeH1	1hyxH1†	6	3	VKLLEQSGAE	VKLLEQSGAE
				EVKLLESGGG	VKLLESGGGL
2fbjH1	1vgeH1†	6	3	VKLLESGGGL	VKLLESGGGL
				KLLEQSGAEV	VKLLEQSGAE

Table 6: (continued). Details of the thirty-one sequences identified as MLSAs. †The most extreme of the MLSAs.

Sequence pairs	Residues involved
1isaA0 3sdpA0	35–44:35–44
1bmfD3 1skyE3	164–173:166–175
1isaA0 3sdpA0	5–12:5–12
1vewA0 3sdpA0	3–12:5–13
1mfeL2 1ospL1	111–120:6–15
1ak200 1akeA0	65–73:49–57
1gafL2 1ae6L1	109–188:5–14
1vgeH1 1hyxH1	2–11:2–11
2fbjH1 1vgeH1	2–11:2–11

Table 7: Details of the sequences identified as containing the most extreme MLSAs.

The process of whittling down the initial data set from roughly 56,000 pairs to the final nine can be seen in figure 31. The details of these sequences can been seen in table 7.

3.2 Analysis of MLSAs

3.2.1 Visual analysis

Firstly the domains that were involved in the nine MLSAs were examined visually, looking at where they occurred within a sequence and where they occurred within the structures using Rasmol.

One apparent MLSA (1ak200 1akeA0) was identified as occurring within a hinge region. Looking at the sequence identified no obvious indels in either sequence to account for a MLSA occurring at that point within that protein pairing as can be seen in figure 32. The residue that acts as the fulcrum for a hinge must be flexible. Indeed it has been shown that mutating the amino acid that acts as the fulcrum can stabilise the protein and prevent the occurrence of swapping between one conformation and another (Odaert *et al.*, 2002). In the p13suc1 protein from *Schizosaccharomyces pombe* there is a hinge region with a proline acting as its fulcrum. If the proline is mutated

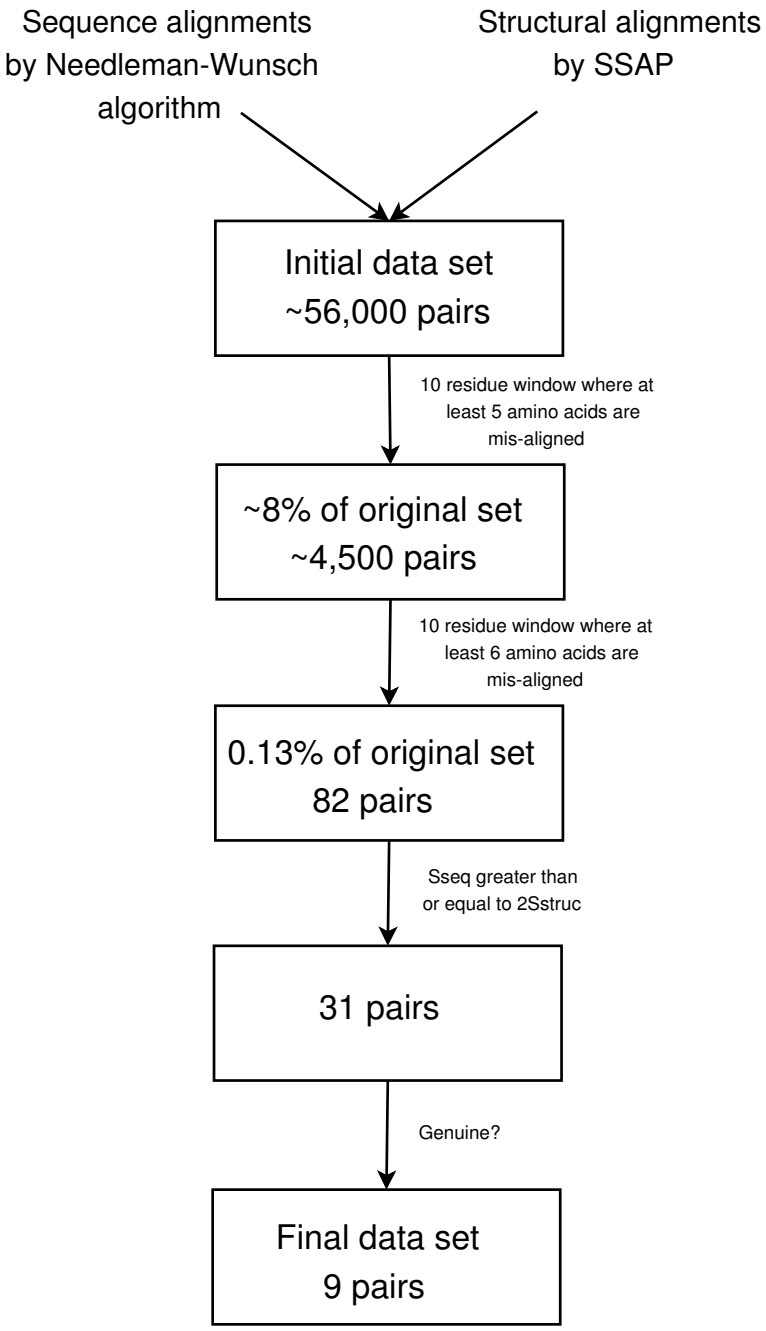


Figure 31: How the nine most extreme and genuine MLSAs were found.

Structure alignment

LKATMDAGK-
-SELGKQAKD

Sequence alignment

LKATMDAGK-
AKDIMDAGKL

Figure 32: The sequence alignment and structural alignment over the window of ten amino acid residues in the protein pairing 1ak200 1akeA0.

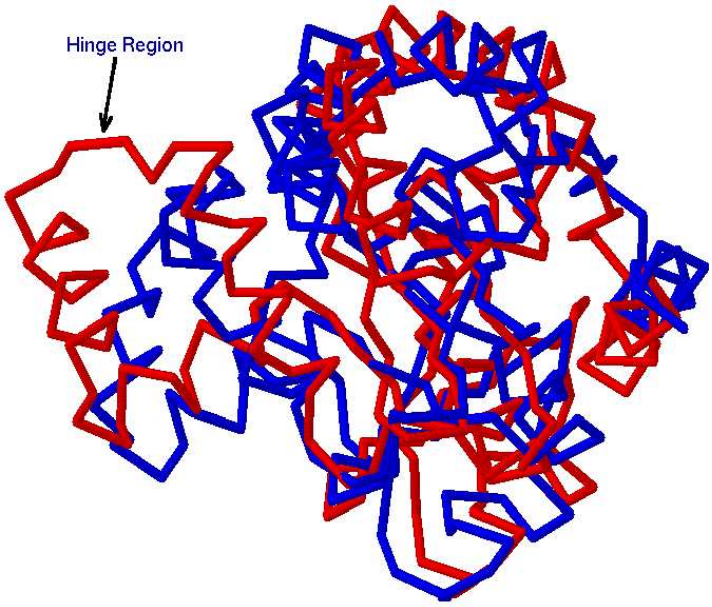


Figure 33: The superimposed structures of the 1ak200 and 1akeA0 protein domains with the hinge region shown to the left, as indicated by the arrow.

into an alanine the hinge no longer works.

The structures of 1ak200 and 1akeA0 can be seen in figure 33. It can be seen that the position of the hinge region in each structure is different. This is because each structure has been solved with the hinge in a different conformation. Fitting the structures with SSAP resulted in a distortion of the structural alignment. It appears that the two parts of the structure separated by the hinge should be treated as separate domains.

Structural alignment

```
                  **********
1vewA0   --------SYTLPSLPYAYDALEPHFDKQT
3sdpA0   ----------PPLPYAHDA--LQPHISKET
```

Sequence alignment

```
                  **********
1vewA0   --------SYTLPSLPYAYDALEPHFDKQT
3sdpA0   ------------PPLPYAHDALQPHISKET
```

Figure 34: The MLSA within the 1vewA0 3sdpA0 pairing showing that it occurs near the terminal of the protein. The sections indicated with asterisks are the MLSA regions.

In the remaining eight identified most extreme MLSAs, the domain structures involved revealed that six of the eight occurred in terminal regions, (figure 34). At the termini regions constraints are not present to the same extent as they are in other areas of protein. As a result when indels occur the constraints are not present to force the amino acids to take up a different position and make the necessary changes elsewhere as demonstrated in figure 35.

Looking further at the sequences and structures showed something unusual in two of the sequences. Both had already been identified as occurring at terminal regions by the visual analysis, but further analysis also suggested that the MLSAs were influenced by the presence of bulky residues and glycine. In both the 1vgeH1 1hyxH1 pairing and the 2fbjH1 1vgeH1 pairing there was a glycine present in a conformation that only it can adopt. Glycine is the smallest of all the amino acids with its side-chain being made up of a single hydrogen atom. In the structures of the proteins involved the domain is structured in such a way that if it were to exist as the sequence alignment suggests it would force another, larger residue into the position occupied in the true structure by the glycine.

In the case of the 2fbjH1 1vgeH1 pairing, no other amino acid would be able to fit into the space without its side-chain interfering with the rest of the structure. The

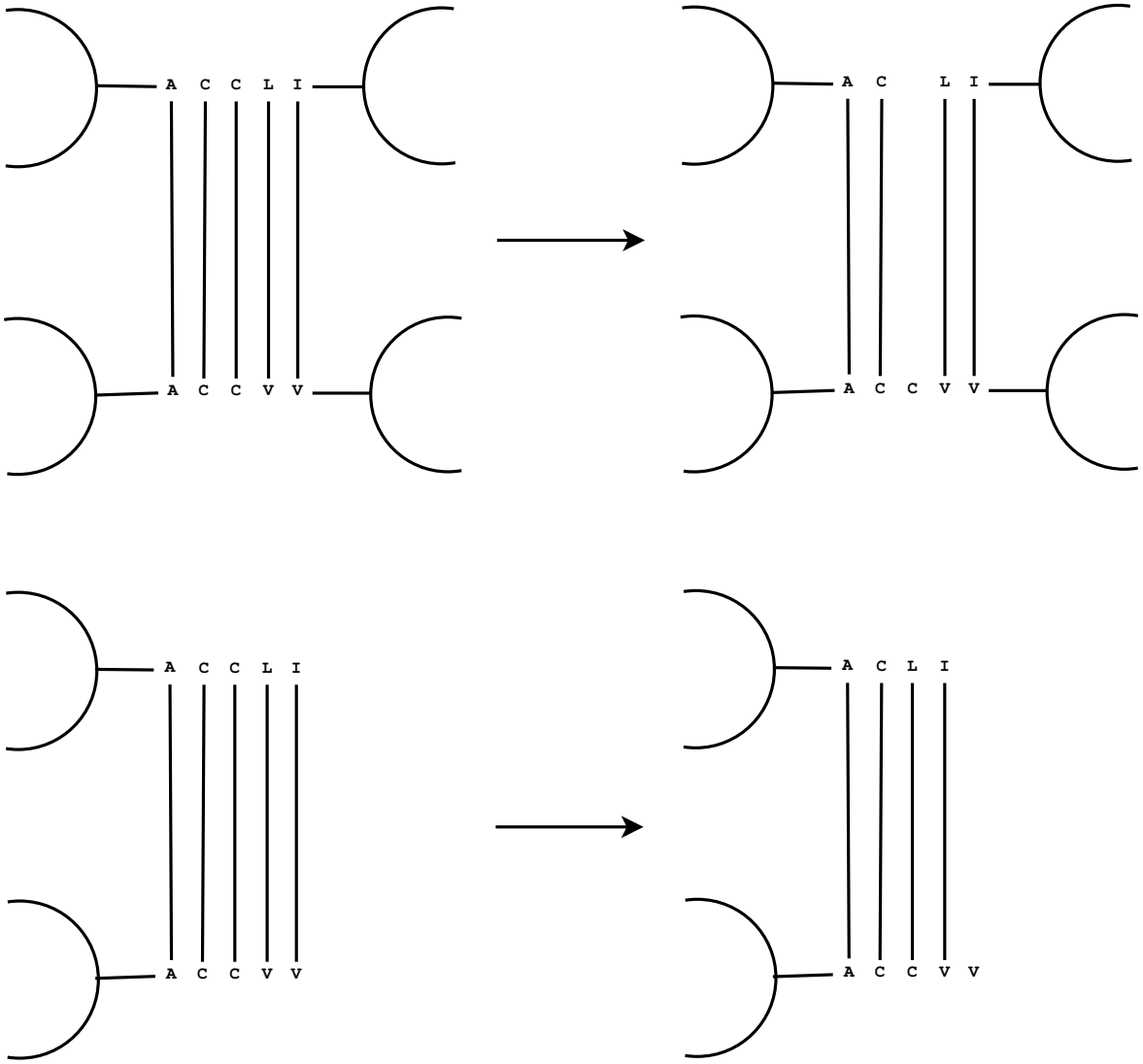


Figure 35: How lack of force constraints affect termini structure. The lack of force constraints at the termini mean that indels are not accommodated elsewhere as the residues are not forced into different positions.

proteins are aligned so as to keep the prolines in the same position in each case. Further towards the terminal end of the protein the protein structure has been conserved between the two due to the presence of some larger residues like glutamine and glutamate. However this forces a kink into the protein structure between the prolines and the larger residues. Only a glycine could adopt the conformation necessary to fit within the kink due to backbone configuration. Hence the shift between the structural alignment and the sequence alignment.

The is same is true of the second pairing, 1vgeH1 and 1hyxH1. In this case the glycine is once again the only residue capable of being in the position that it is. Because of larger, bulkier residues on either side the glycine is forced to adopt a conformation that no other amino acid would be able to because of the size of their side-chains.

3.2.2 Hydrophobic and hydrophilic residues

The placing of hydrophobic and hydrophilic residues within the structure of a protein is a primary driving force in protein folding. A major contribution to the free energy of the natural structure of a protein is due to interactions between hydrophobic amino acids that tend to form a core in the three-dimensional structure shielded from the surrounding environment by hydrophilic residues (Jiang *et al.*, 2003). Exposing hydrophobic residues to the aqueous environment surrounding a protein destabilises its overall structure. Therefore a protein will tend to cluster its hydrophobic residues away from the outside. It is possible that the reason that MLSAs occur is partly to keep the hydrophobic and hydrophilic residues pointing either outwards or inwards to stabilise the protein.

For each pairing of the remaining six most extreme genuine MLSAs, both the sequence alignment and the structural alignment were examined and their hydrophobic and hydrophilic residues identified. This identification can be seen in figure 36. If

the presence of the MLSA is partly caused by the positioning of hydrophilic and hydrophobic residues, then it should be seen that the structural alignment ensures that more hydrophilic residues are facing outwards. Looking at figure 36 it is clear that the hydrophobic and hydrophilic residues are not always aligned in the same place in the structural and sequence alignments. Examining these alignments, it would appear that the hydrophobic residues are better aligned in the sequence alignment rather than the structure alignment. The other alignments appeared to show a similar trend. This would suggest that the positioning of the hydrophobic and hydrophilic residues is not related to the occurrence of MLSAs.

However in order to investigate further whether the positioning of the hydrophilic and hydrophobic residues affects the structures and therefore the MLSA it was necessary to look at the 3-D structures of the proteins and not just the alignments. Figure 37 shows a section of the two structures of the aligned pairing 1bmfD3 and 1skyE3 using Rasmol. It is clear that the proteins have structured themselves in such a way as to minimise exposure of the hydrophobic residues. The hydrophilic residues are facing the aqueous environment while the hydrophobic residues are buried in most cases. If the sequence alignment had been the correct one, then more hydrophobic residues would have been exposed. Such factors cannot be considered in simple sequence alignment.

This analysis was extended in a quantitative fashion using the PRIFT hydrophobicity scale (Cornette *et al.*, 1987) to investigate the hydrophobicity of the residues that were more than 20% accessible to the outside aqueous environment. By calculating values for both protein domains in both the sequence alignment and the structural alignment these values could be compared. Table 8 shows the PRIFT hydrophobicity scale used in the analysis. Figure 38 shows how the hydrophobicity was calculated for both the structural and sequence alignments.

For each case, the MLSA region alone was examined and then the extended region surrounding it. As figure 38 shows, the four values were calculated for each pairing

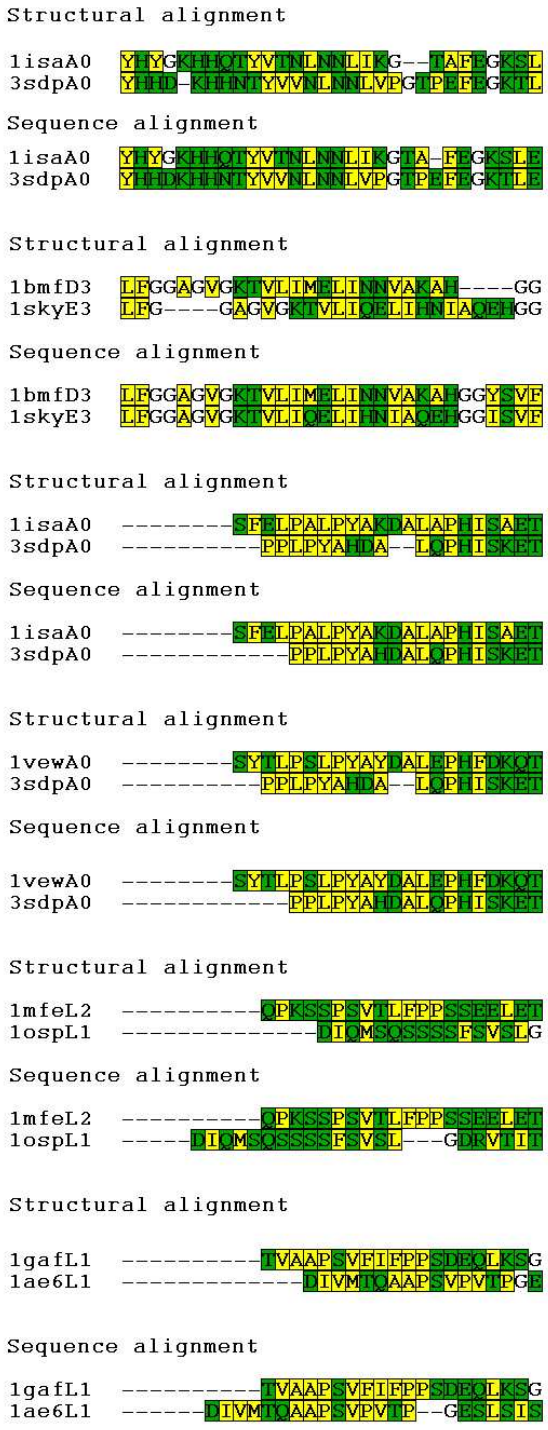


Figure 36: The sequence and structural alignments of the remaining six pairs of protein domains containing MLSAs with hydrophilic and hydrophobic residues shown in green and yellow respectively. The MLSAs are the ten residue wide boxes in the middle of each sequence pairing.

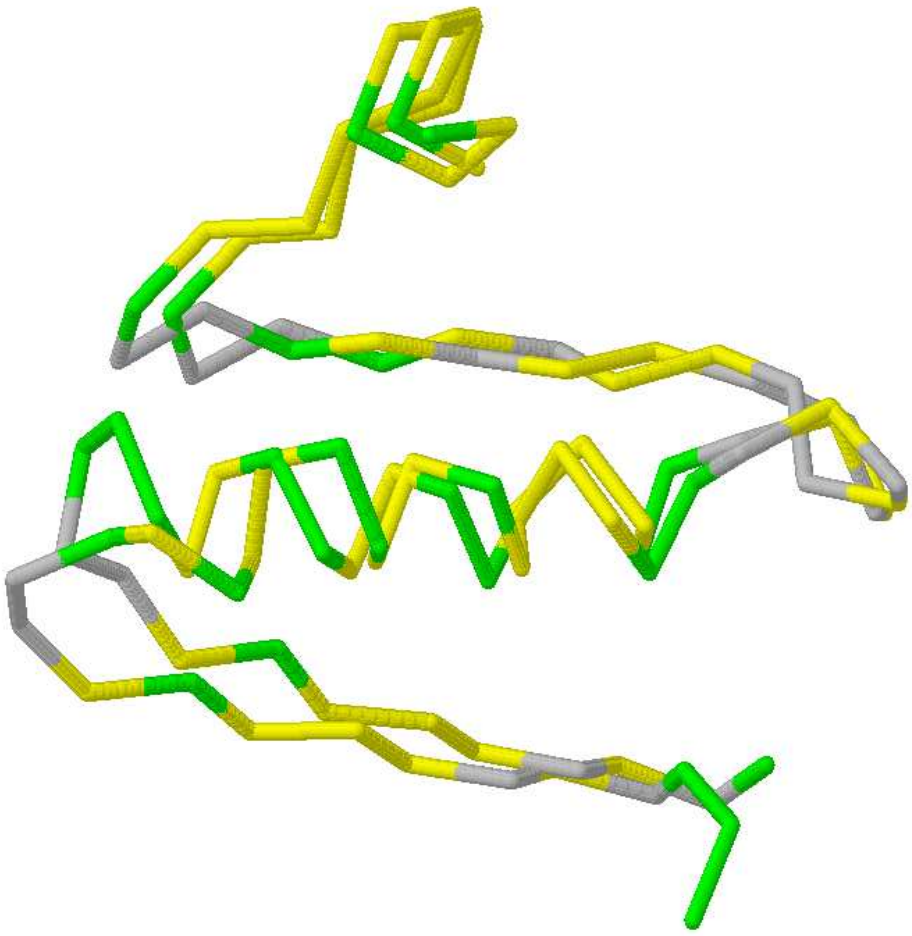


Figure 37: The aligned structures of 1bmfD3 and 1skyE3, with the hydrophobic and hydrophilic residues marked on in yellow and green respectively. Only the partial structure near the MLSA region is shown.

Amino Acid	Hydrophobicity
ALA	0.22
ARG	1.42
ASN	-0.46
ASP	-3.08
CYS	4.07
GLN	-2.81
GLU	-1.81
GLY	0.00
HIS	0.46
ILE	4.77
LEU	5.66
LYS	-3.04
MET	4.23
PHE	4.44
PRO	-2.23
SER	-0.45
THR	-1.90
TRP	1.04
TYR	3.23
VAL	4.67

Table 8: The PRIFT hydrophobicity scale.

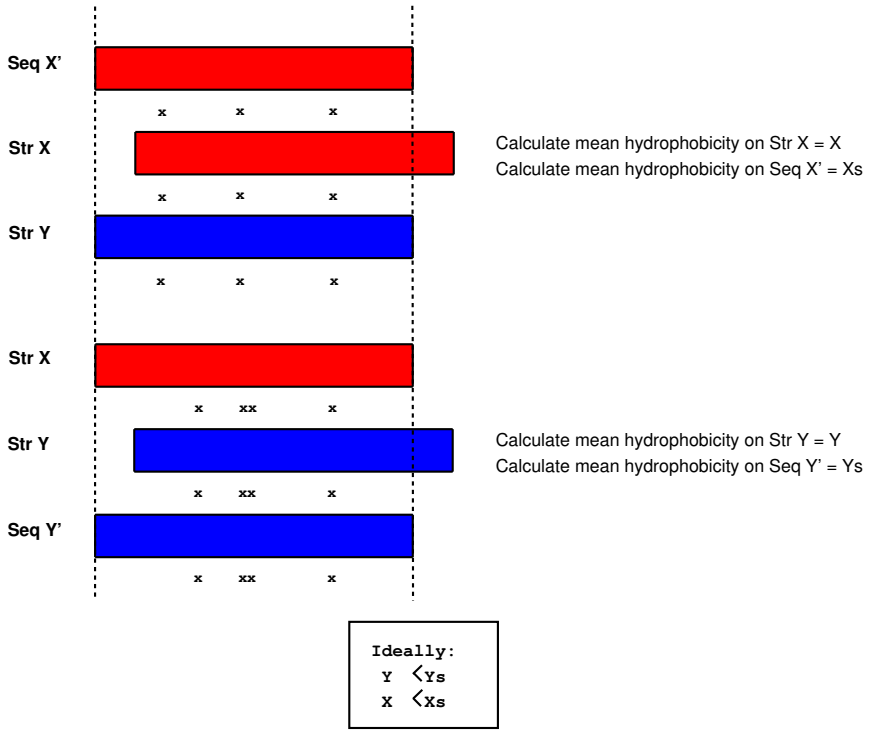


Figure 38: How the hydrophobicity was calculated for those residues greater than 20% accessible for both the sequence and structural alignment, 'x' represents a site with greater than 20% accessibility.

were:

1. X: the averaged hydrophobicity for structural alignment X
2. Xs: the averaged hydrophobicity for sequence alignment X'
3. Y: the averaged hydrophobicity for structural alignment Y
4. Ys: the averaged hydrophobicity for sequence alignment Y'

Analysis involved comparing the MLSA region of the first structurally aligned protein domain (str Y) with the section that aligned by sequence (seq X') and by structure (str X) in the second sequence. Accessible residues in str Y were then marked and the equivalent residues in str X and seq X noted. By referencing the PRIFT hydrophobicity scale the values of X and Xs could be calculated. This would then be repeated

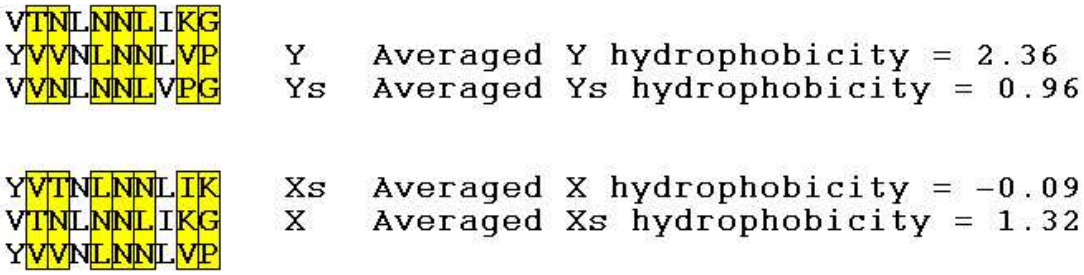


Figure 39: Hydrophobicity for the MLSA in the alignment pairing 1isaA0 3sdpA0, amino acids 35-44:35-44

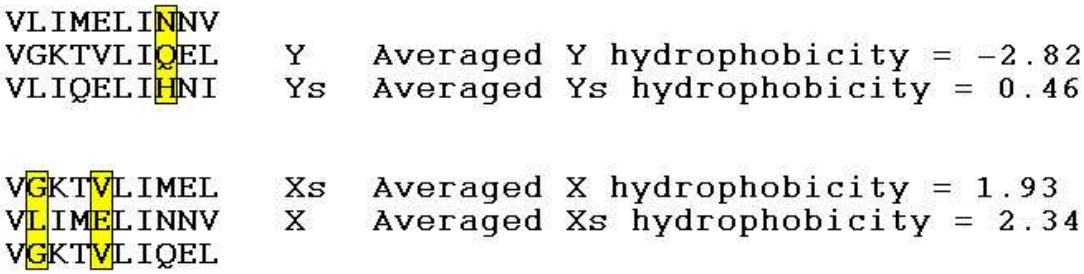


Figure 40: Hydrophobicity for the MLSA in the alignment pairing 1bmfD3 1skyE3, amino acids 164-173:166-175.

using the MLSA structure for X to calculate Y (for the structural alignment) and Ys (for the sequence alignment).

If the MLSA has led to increased burial of hydrophobic residues, the value of X should be less than that of Xs, and the value of Y should be less than Ys.

Figures 39, 40, 41, 42, 43 and 44 show the results of the hydrophobicity scores for each of the remaining MLSAs. This data are compiled in table 9. In half the cases we can see that the scores have $X < Xs$ and $Y < Ys$. This is what would be expected by chance. Thus, while hydrophobicity is therefore not a dominant effect it is quite possible that it is partially responsible for some of the MLSAs. Due to the limited amount of data it was not possible to do any further statistical tests.

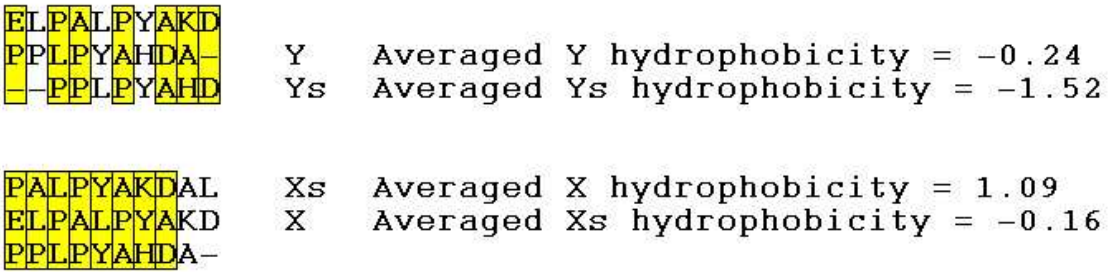


Figure 41: Hydrophobicity for the MLSA in the alignment pairing 1isaA0 3sdpA0, amino acids 5-12:5-12.

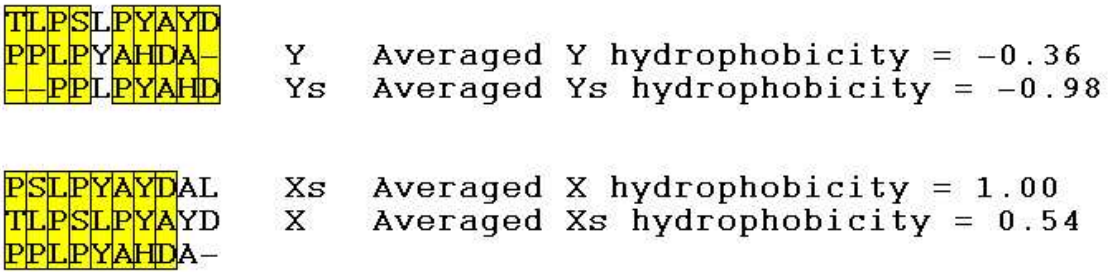


Figure 42: Hydrophobicity for the MLSA in the alignment pairing 1vewA0 3sdpA0, amino acids 3-12:5-13.

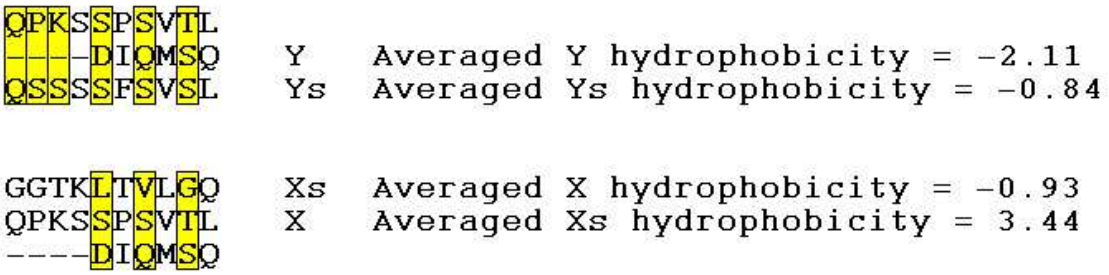


Figure 43: Hydrophobicity for the MLSA in the alignment pairing 1mfeL2 1ospL1, amino acids 111-120:6-15.

Alignment pairing	Residue region	X score	Xs score	Δ X and Xs scores	Y score	Ys score	Δ Y and Ys scores
1isaA0 3sdpA0	35–44:35–44	-0.09	1.32	1.41	2.36	0.96	-1.40
1bmfD3 1skyE3	164–173:166–175	1.93	2.34	0.41	-2.81	0.46	3.27
1isaA0 3sdpA0	5–12:5–12	1.09	-0.16	-1.25	-0.24	-1.52	-1.28
1vewA0 3sdpA0	3–12:5–13	1.00	0.54	-0.46	-0.36	-0.98	-0.62
1mfeL2 1ospL1	111–120:6–15	-0.93	3.44	4.37	-2.11	-0.84	1.27
1gafL2 1ae6L1	109–118:5–14	2.16	-1.63	-3.79	-0.10	-1.43	-1.33

Table 9: The averaged hydrophobicity of the accessible residues in the MLSAs.

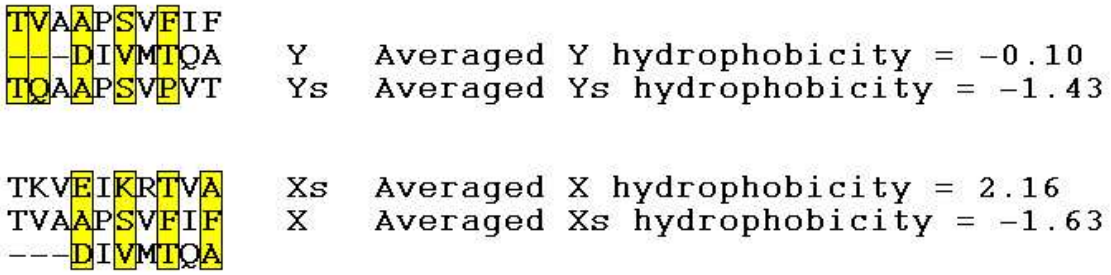


Figure 44: Hydrophobicity for the MLSA in the alignment pairing 1gafL2 1ae6L1, amino acids 109-118:5-14.

3.2.3 Hydrogen bonds

Hydrogen bonds are the most important of all directional intermolecular interactions (Steiner, 2002). They are formed when a hydrogen atom is effectively 'shared' by two other atoms. The atom to which the hydrogen is more tightly linked is known as the hydrogen donor with the second atom being the hydrogen acceptor. The bond between the hydrogen and the acceptor is the hydrogen bond.

Hydrogen bonds are necessary in determining molecular conformation, molecular aggregation, and the function of a vast number of chemical systems ranging from inorganic to biological (Steiner, 2002). In proteins they are especially important in the 3-D structures as well as in some protein functions. It is the hydrogen bonds in alpha helices and beta sheets that stabilise these structures. It is possible that the difference between sequence and structural alignments that results in an MLSA is caused in part by optimizing hydrogen bonding.

Once again, looking at the six remaining protein pairings with the most extreme genuine MLSAs, the hydrogen bonds were examined visually through Rasmol using both PDB files. The hydrogen bonds were examined using Rasmol's hbond command. This displays backbone to backbone hydrogen bonds in regions of defined secondary structure (See `http://www.umass.edu/microbio/Rasmol/rasbonds.htm#hbonds`).

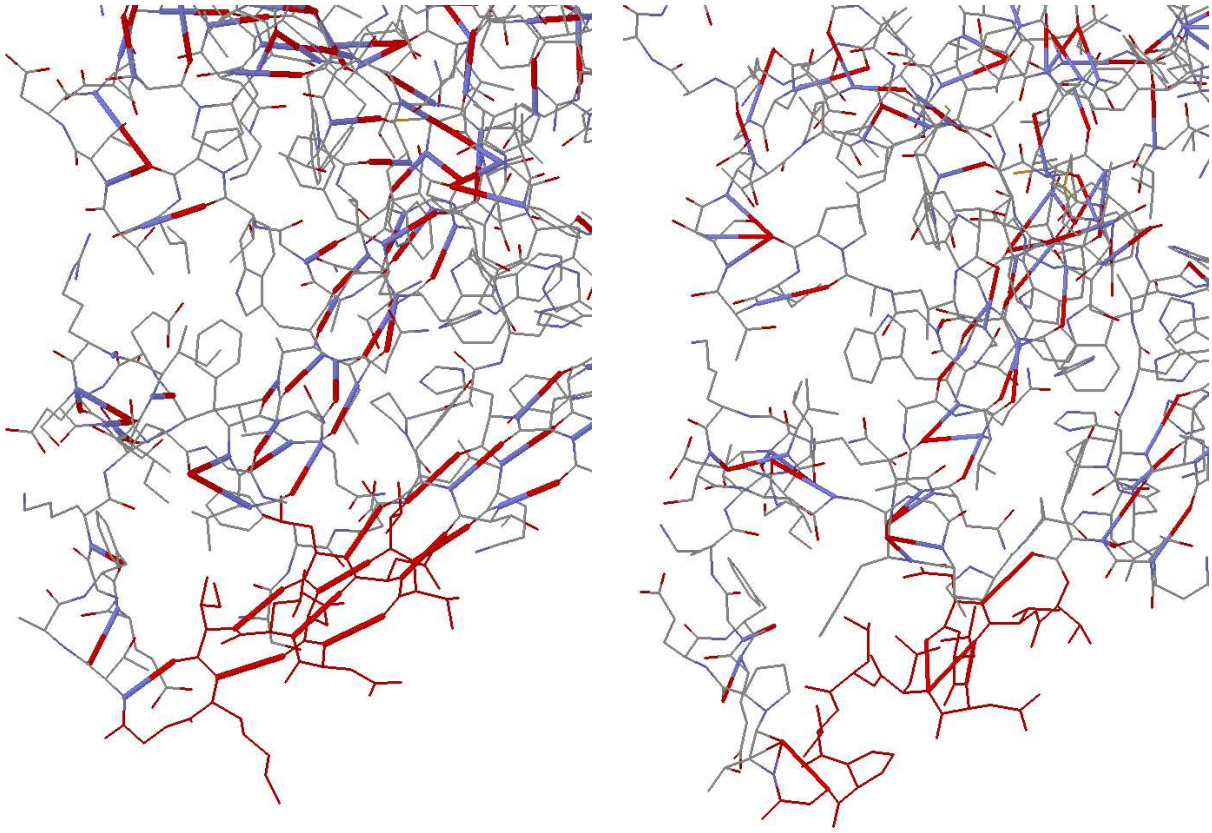
Alignment pairing	Residue region	Hydrogen bonds within MLSA of first protein	Hydrogen bonds near MLSA in first protein	Hydrogen bonds within MLSA of second protein	Hydrogen bonds near MLSA in second protein
1isaA0 3sdpA0	35-44:35-44	11	26	2	13
1bmfD3 1skyE3	164-173:166-175	17	17	18	16
1isaA0 3sdpA0	5-12:5-12	5	12	1	5
1vewA0 3sdpA0	3-12:5-13	4	12	2	4
1mfeL2 1ospL1	111-120:6-15	5	20	13	10
1gafL2 1ae6L1	109-118:5-14	4	23	9	5

Table 10: Numbers of hydrogen bonds in the region of the MLSA in each protein. Hydrogen bonds that are 'near' are those involving residues within ten amino acids upstream or downstream of the MLSA.

The area of the MLSA and the region surrounding it were looked at to determine how the hydrogen bonds might be contributing to the structure of the domain.

Table 10 shows the number of hydrogen bonds in and around each MLSA for each protein in the alignment pairs. The table shows that there are quite a number of hydrogen bonds involved in the area. Most of the MLSAs regions contain a number of hydrogen bonds with either the acceptor atom or the donor in the region. Figure 45a shows the structure of the MLSA region of 1isaA0 and figure 45b the structure of the MLSA region of 3sdpA0. Both figures show the hydrogen bonds.

There are quite a number of hydrogen bonds in and around the areas where MLSAs occur. Some of the protein pairs have more hydrogen bonds in one structure than in the other. For example in 1gafL2 there are a total of 27 hydrogen bonds in or around the area of the MLSA. In 1ae6L1 there are only 14 hydrogen bonds. The difference in the number of hydrogen bonds could be leading to a difference in the structure of the protein in that area, therefore the stabilizing effects of these bonds could be a contributing factor in the formation of the MLSAs. In other protein pairs such as 1mfeL2 1ospL1 there are a similar number of hydrogen bonds in both structures which



a) b)

Figure 45: The structure of the MLSA region of a) 1isaA0 and b) 3sdpA0, showing the hydrogen bonds. The thickest lines represent the hydrogen bonds.

Positively charged residues	Negatively charged residues	Neutral residues
arginine	aspartate	alanine
histidine	glutamate	asparagine
lysine		cysteine
		glutamine
		glycine
		isoleucine
		leucine
		methionine
		phenylalanine
		proline
		serine
		threonine
		tryptophan
		tyrosine
		valine

Table 11: Charges of amino acids. Histidine is only positively charged at low pH.

suggests that hydrogen bonds may not be playing as large a part in the formation of hydrogen bonds.

3.2.4 Charges in and around the MLSAs

Some amino acids are negatively or positively charged, they can be seen in table 11. All buried charged residues must have a partner. It was possible that the charges or their interactions were partially responsible for the remaining MLSAs. By comparing where the charged residues were in the sequence alignments and the structural alignments it was possible to see how the MLSAs affected the positioning of the charges.

Figure 46 shows what happens to the charged residues in the domains involved when they are aligned by sequence as well as structure. In some of the protein pairs the shifts in the charged residues between the two alignments are fairly minor, such as in the first 1isaA0 3sdp30 MLSA. However in others such as the 1bmfD3 1skyE3 pair the charged residues are shifted quite significantly. In those alignments where there is

a significant shift in the charged residues it could suggest that the charged interactions do play a part in causing the MLSA.

Although the charges are seen to shift slightly between the sequence alignments and the structural alignments it is the interactions between these charges that may prove whether they do indeed play a part in MLSAs or not. If the charged residues are not interacting with other residues or with the environment then whether they are shifted or not within the MLSA area is unlikely to be important. However if the charges are interacting with other charges then the shifts between sequence and structural alignment are very likely to be caused by them.

Figure 47 shows the charges and their interactions in the MLSAs and surrounding areas. It also shows whether or not there are equivalent interactions at the same point in the aligned residue. So, for example, in the 1isaA0 3sdpA0 pairing there is an interaction between alanine 12 and lysine 19 in 3sdpA0 and an equivalent interaction between lysine 11 and glutamic acid 24 in 1isaA0. Whereas in the same pairing there is no interaction involving alanine 20 even though its aligned equivalent, glutamic acid 24, is part of a charge interaction.

From looking at figure 47 there are a few charge interactions within or near the MLSA region. Some have equivalent interactions to their aligned partners, some do not. The charges may stabilise the structure of the protein around the MLSA region.

After looking at the results achieved thus far it seemed likely that both MLSAs 1isaA0 3sdpA0 (amino acids 5-12:5-12) and 1vewA0 3sdpA0 (amino acids 3-12:5-12) were most likely also to be caused by their closeness to the terminus of the protein domains involved. The analysis done up to that point did not seem to indicate any other significant cause for them and proximity to the terminal region of a protein had already been used explain other MLSAs identified. However further analysis was conducted on the remaining four proteins looking more closely at the 3-D structures and charged residues involved.

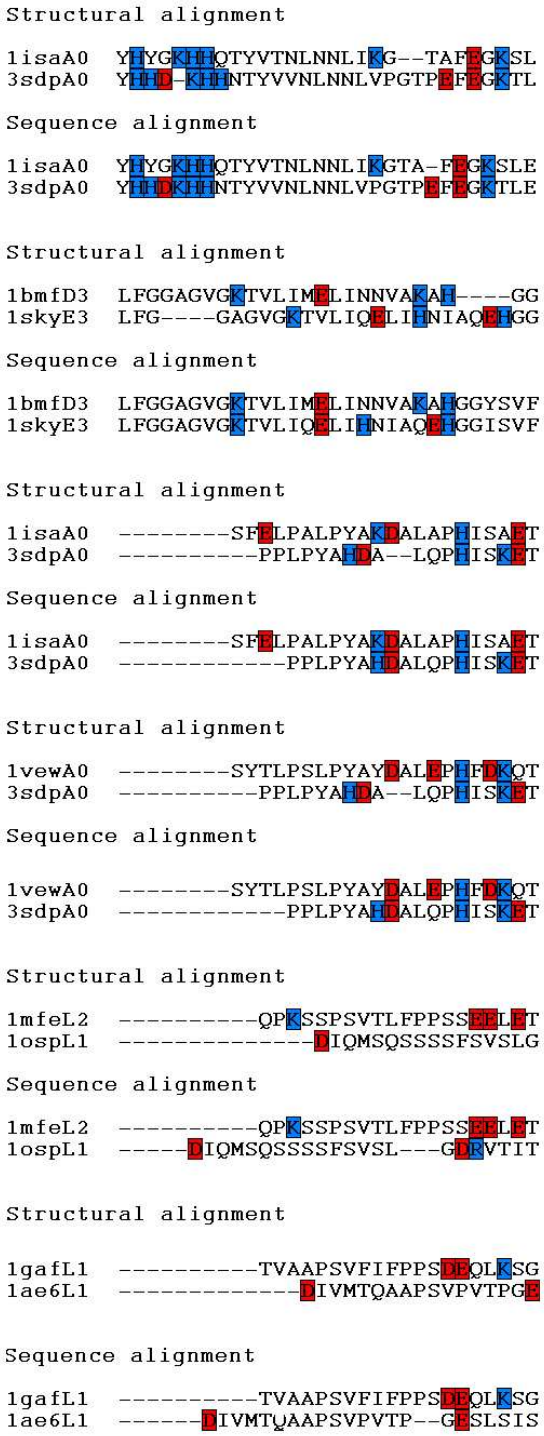


Figure 46: The positioning of the charged residues in and around the MLSAs in the sequence and structural alignments. Positively charged residues are highlighted in blue, negatively charged residues are highlighted in red.

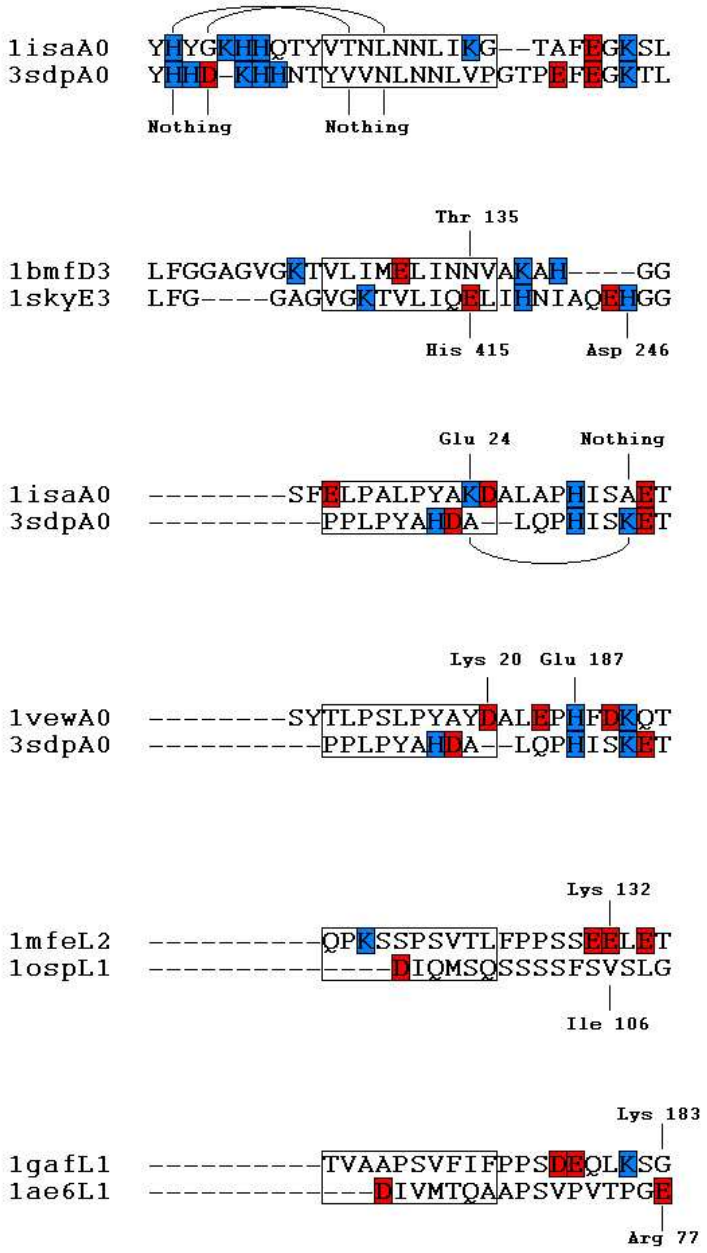


Figure 47: The interactions of the charged residues in and around the MLSAs. Positively charged residues are highlighted in blue, negatively charged residues are highlighted in red. The black lines show charge interaction between residues, either within the region of the structure illustrated or between these and other residues. For example in the 1isaA0 3sdpA0 alignment pair, in 1isaA0 there is a lysine in the MLSA which may be interacting with Glu 24 (not part of the MLSA). The equivalent amino acid in 3sdpA0 is an alanine is not charged and so is shown not to be participating in a charged interaction.

Positively charged residues	Negatively charged residues
Lys 11	Asp 12/Glu 21/Glu 24
Lys 43	Glu 48
Lys 50/Arg 57	Glu53

Table 12: Charge interactions between amino acids 1-61 in 1isaA0.

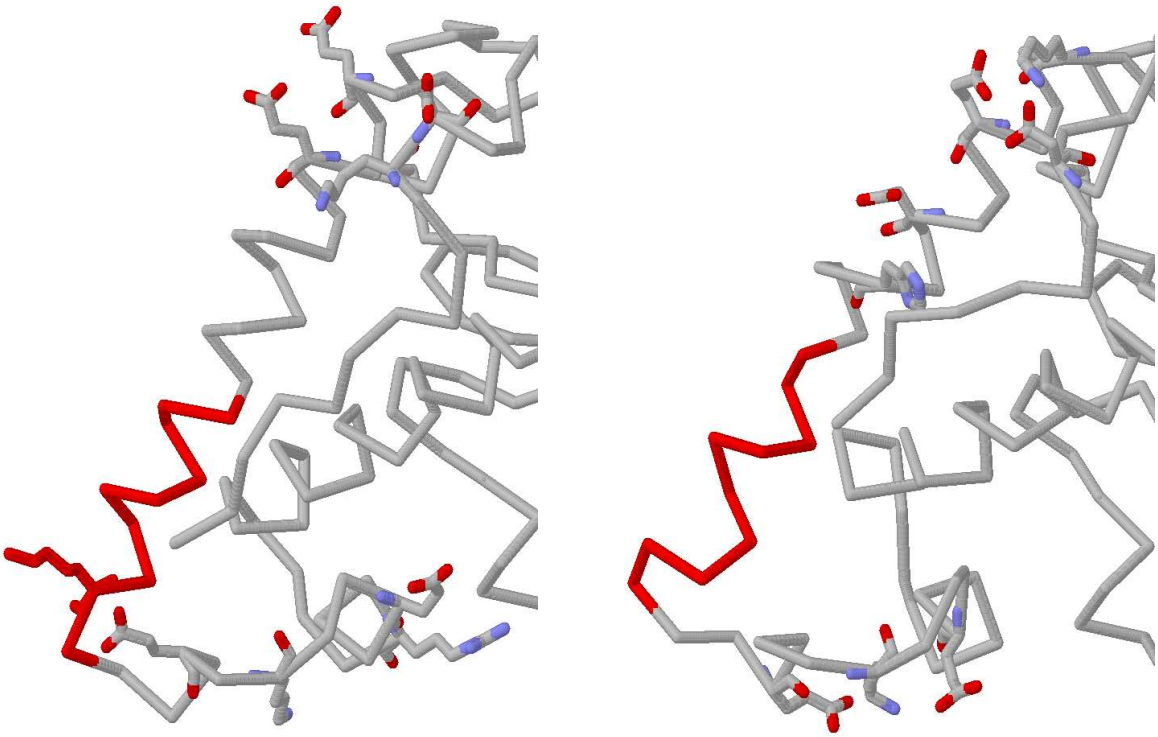
Positively charged residues	Negatively charged residues
Lys 20	Asp 12/Glu 24
His 31	Asp 28
Lys 51	Glu 47/Glu 55

Table 13: Charge interactions between amino acids 1-61 in 3sdpA0.

For 1isaA0 and 3sdpA0 (for the MLSA at residues 35-44:35-44) figure 46 shows the whereabouts of positively and negatively charged residues both in and near the MLSA regions of this protein pair and the others examined. Residues 1-61 were examined for the presence of interactions between charged residues that might be contributing to the formation of MLSAs.

Figure 48 shows the actual 3-D structures of these protein sections and the charged interactions that appear to be taking place. In both 1isaA0 and 3sdpA0 there appear to be three charged interactions occurring in or around the MLSA region. Table 12 and table 13 list the charge pairs in 1isaAo and 3sdpA0 respectively that appear to be interacting with one another. The number of charged interactions remains the same between the two. This suggests that charged interactions may be playing a part in this particular MLSA as if one had used the sequence alignment (i.e. shifted the residues in 1isaA0 to match the alignment in 3sdpA0) then these charged pairings might not have been maintained.

The MLSA which occurs within the protein pairing of 1bmfD3 1skyE3 (amino acids 164-173:166-175) was investigated in the same way as the previous one. As before the



a) b)

Figure 48: 3-D structure of a) 1isaA0 and b) 3sdpA0, with the MLSA region highlighted in red, showing the charge interactions (residues involved have their sidechains visible).

Positively charged residues	Negatively charged residues
His 179	Glu 136
His 173/His415	Glu 170/418

Table 14: Charge interactions between amino acids in 1skyE3.

Positively charged residues	Negatively charged residues
Lys 151	Asp 330

Table 15: Charge interaction between amino acids in 1bmfD3.

region looked at for charged interactions was extended beyond what had previously been examined. Tables 14 and 15 show the number of charge interactions in domains 1skyE3 and 1bmfD3. There appear to be more charged interactions in 1skyE3 than in 1bmfD3. The differences in the number of charged residues could account partially for the MLSA as they may be stabilizing a structure in 1skyE3 that does not exist in 1bmfD3.

The third protein alignment pairing to be examined in closer detail with regards to charged residues was 1mfeL2 1ospL1 (amino acids 111-120:6-15). There were very few charged interactions in the extended region (residues 108-135:1-75) that was looked at for this pairing, there was only one charged interaction in each 3-D structure. This might indicate that the structure of the area is not very dependent upon charges and their interactions for stability.

Although the lack of charged interactions might suggest that they do not play a large role in causing these MLSAs they were examined to confirm this. Tables 16 and 17 show the different charged interactions within the structures. Each domain had a single charge interaction in or around the MLSA region. Maintaining this charge interaction in each structure could be the reason for the MLSA. Using the sequence alignment to

Positively charged residues	Negatively charged residues
Lys 132	Glu 127

Table 16: Charge interaction between amino acids in 1mfeL2.

Positively charged residues	Negatively charged residues
Lys 24	Asp 70

Table 17: Charge interaction between amino acids in 1ospL1.

build models of these protein structure rather than the structural alignment could be enough to lose these interactions. Without the interaction in each domain the protein may be unable to form the correct structure.

The final protein alignment pairing looked at was 1gafL2 1ae6L1 (amino acids 109-118:5-14). The final set of charge interactions between the protein pair was investigated and the results found in table 18 and 19. As with 1ospL1 1mfeL2 there was only a single charged interaction in the 3-D structure of each. However looking at figure 49 it appears that the residues highlighted do not in fact form a charged interaction as their side-chains are pointing in opposite directions.

There are charged interactions in and around the region of the MLSAs in these aligned protein pairs. Although it would only take one charged interaction to make a difference in the structure there seems to be little difference in the number of interactions near the MLSAs. Possibly there may be charged interactions further away from

Positively charged residues	Negatively charged residues
Lys 103	Asp 85

Table 18: Charge interaction between amino acids in 1gafL2.

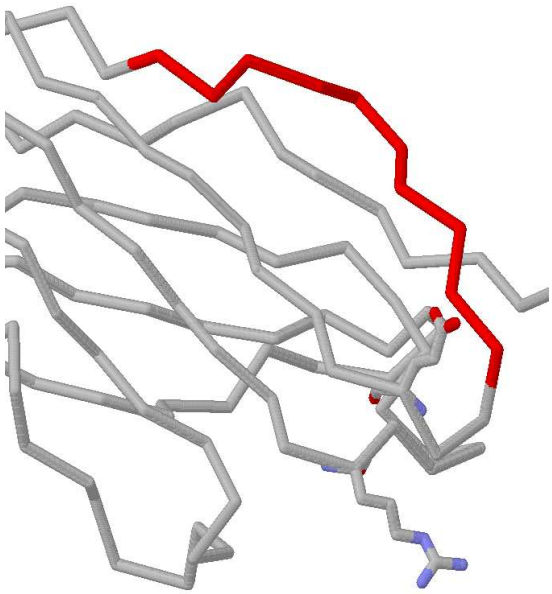


Figure 49: 3-D structure of 1ae6L2 (the MLSA region is highlighted in red) showing the charge interactions (residues involved have their sidechains visible).

Positively charged residues	Negatively charged residues
Arg 77	Glu 17

Table 19: Possible charge interaction between amino acids in 1ae6L1.

the MLSA region that are having some kind of effect upon the structures of the protein domains.

3.2.5 Secondary structure

Looking at the complete 3-D structures of the domains 1isaA0 and 3sdpA0 (for the MLSA at residues 35-44:35-44) it can be seen that the MLSA occurs mainly within alpha-helices as figure 50a and b show. More of the structure of 1isaA0 is in the alpha-helix conformation than 3sdpA0.

For 1bmfD3 1skyE3 once again the MLSA occurs in an alpha helix in both structures as figure 50c and d show. The MLSA in 1bmfD3 is positioned at the end of an alpha-helix. In 1skyE3 the MLSA is not so close to the end of the alpha-helix.

In the case of 1mfeL2 1ospL1 the MLSA again mainly occurred within sections of secondary structure. Unlike before the secondary structure was beta-strand rather than alpha-helix (figure 51a and b). More of the MLSA is in beta-strand conformation in 1mfeL2 than in 1ospL1.

Figures 51c and d show that the MLSA in 1gafL1 also occurs within a beta strand. However in 1ae6L1 the MLSA occurs in a loop region which is bordered by a region of beta-strand. The difference in secondary structure between the two protein domains may suggest why this particular MLSA occurs.

All four MLSAs have occurred within regions of regular secondary structure, it could be that maintaining these regions of alpha helix and beta strand are the reason for the misaligned areas. MLSAs are therefore caused by a combination of the factors which have been investigated individually.

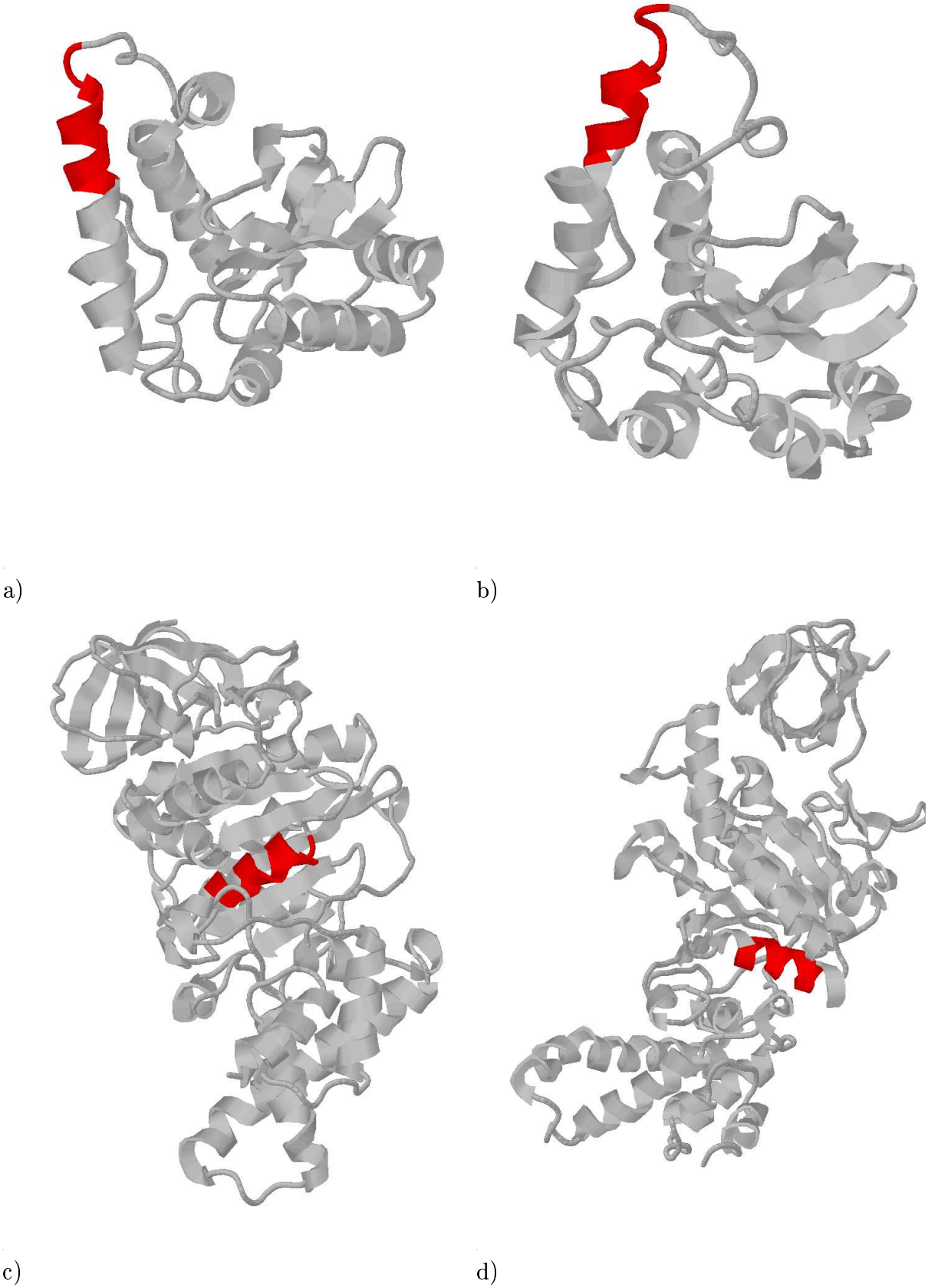


Figure 50: 3-D structures of a) 1isaA0, b) 3sdpA0, c) 1bmfD3 and d) 1skyE3. In each case the MLSA region is highlighted in red.

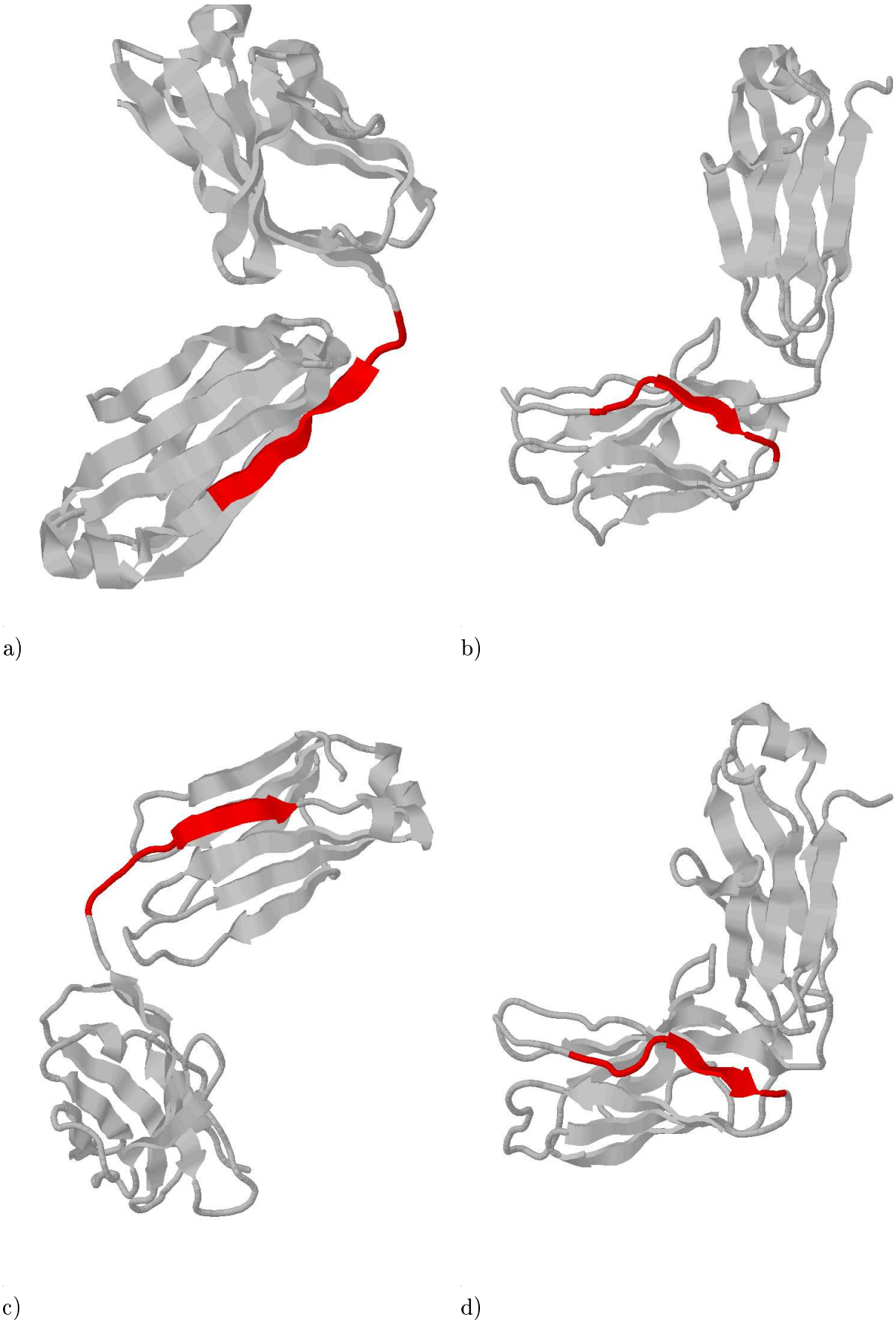


Figure 51: 3-D structures of a) 1mfeL2, b) 1ospL1, c) 1gafL1 and d) 1ae6L2. In each case the MLSA region is highlighted in red.

3.3 Summary: Where and Why do MLSAs Occur?

It appeared that the nine MLSAs examined could be divided up into three distinct groups depending on when and why they occurred. Of the nine, one occurred because of a hinge section of protein structure as figure 33. The two structures that made up the MLSA each had the hinge region present, but in different conformations leading to differences in the structural alignment. It should therefore have been removed during the filtering of the original 56,510 protein pairings because the MLSA was caused by this highly flexible protein region. Indeed the two regions separated by the hinge should have been classified as separate structural domains in CATH.

Of the remaining identified MLSAs, some had more than one reason for occurring. One reason that occurred in six of the pairings was because the MLSAs occurred in terminal regions of the alignments. This kind of MLSA occurs because it is 'easier' for the pairing to accommodate a change at the terminal region of the protein rather than further in the structure where conserved regions relating to structure or function are more likely to occur.

When looking at the four protein pairings, 1isaA0 3sdpA0, 1bmfD3 1skyE3, 1mfeL2 1ospL1 and 1gafL2 1ae6L2, it became clear that all four MLSAs occurred within regions of secondary structure. The first two occurred within alpha helices and the second two within beta strands. These structures can be seen in figure 50 and figure 51. Maintaining these secondary structures appears to be the major reason for the difference between the sequence alignment and the structural alignment.

Another reason for the MLSAs to occur may to be in order to minimise the exposure of hydrophobic residues as figure 37 shows. Hydrophobic residues are usually kept away from the aqueous environment that surrounds most proteins. Obviously if these residues were exposed they would destabilise the protein structure. Therefore it is better for the alignment to alter slightly rather than expose these hydrophobic residues.

Alignment pair	Residue region	Causes
1isaA0 3sdpA0	35–44:35–44	Secondary structure
		Accessibility of hydrophobic residues
		Charge interactions
1bmfD3 1skyE3	164–173:166–175	Secondary structure
		Accessibility of hydrophobic residues
		Charge interaction
1isaA0 3sdpA0	5–12:5–12	Occurs in terminal region
1vewA0 3sdpA0	3–12:5–12	Occurs in terminal region
1mfeL2 1ospL1	111–120:6–12	Secondary structure
		Accessibility of hydrophobic residues
		Occurs in terminal region
1ak200 1akeA0	65–73:49–57	Involved in hinge region
1gafL2 1ae6L1	109–118:5–14	Secondary structure
		Accessibility of hydrophobic residues
		Occurs in terminal region
1vgeH1 1hyxH1	2–11:2–11	Occurs in terminal region
2fbjH1 1vgeH1	2–11:2–11	Occurs in terminal region

Table 20: Summary of MLSAs and their possible causes

In these extreme examples of MLSAs we did not find much data to support this idea but it is possible that in some cases it may play a part.

Table 20 shows a summary of the MLSAs and their possible causes.

Chapter 4

Sequence-Structure Mis-Alignments (SSMAs)

We have defined SSMAs (Sequence-Structure Mis-Alignments) as less extreme cases of misalignment than MLSAs (Misleading Local Sequence Alignments). A SSMA is where the structural alignment and the sequence alignment of a pair of aligned sequences do not agree. A SSMA region is a continuous section of an alignment where the structural and sequence alignment do not match.

The aim of this chapter was to study SSMAs in the hope that it would be possible to predict them with the aid of neural networks.

4.1 What are SSMAs?

As defined above, SSMA regions are sections where the structural alignment and the sequence alignment do not agree. The structural alignment is taken to be the correct alignment, the one that will give us a close model to the actual structure. Figure 52 shows a good example of a SSMA region where the alignments differ between 1igmH0 (Immunoglobulin m (IgM) Fv fragment from *Homo sapiens*) and 1ap2A0 (Monoclonal

```
                        1ap2A0   DIVMTQSPSSLTVTAGEKVTM

1igmH0   Sequence alignment   EVHLLESGGNL-VQPGGSLRL
1igmH0 Structural alignment   EVHLLESG-GNLVQPGGSLRL
                                       ****
```

Figure 52: An example of a SSMA region found within 1igmH0. This occurs when aligned with 1ap2A0, the section highlighted by * is the misaligned region, made up of four SSMA positions. The figure compares the same sections of the structural and sequence alignment of 1igmH0 so the difference between its sequence alignment and its structural alignment can be seen.

antibody c219 from *Mus musculus*).

This type of misalignment is a lot more common than MLSAs. Although there are many, less extreme cases of MLSAs than the ones that were looked at in the previous chapter, they are still relatively rare when compared with SSMAs.

Predicting where these SSMAs occur would be a possible way of improving the step of sequence alignment in comparative modelling. Sequence alignment is one of the steps that introduces the most error into a comparative model of a protein with unknown structure. If this error could be lessened then the final model would be more accurate and therefore of more use. The prediction of the SSMAs was done using neural networks. However the networks needed enough relevant input to make a prediction. Obviously the sequences of the protein pairs formed part of this input but the SSMAs themselves were looked at to discover whether there was anything else that needed to be added to the input of the neural nets. The more relevant information that could be added to the input files, the better the level of prediction was hoped to be.

4.2 Finding the SSMAs

The same data set was used as had been previously used as the initial data set for the MLSA research, namely all the pairs of NRep sequences within each H-family within

the CATH database (v1.6). This produced a total data set of approximately 56,000 protein pairs. Unlike when investigating the MLSAs, this data set was not filtered or restricted.

The data set of protein pairs was used to produce both sequence and structural alignments. The sequence alignments were produced by the Needleman and Wunsch algorithm while structural alignments were produced by SSAP.

A program called Checkalignment.pl was written to run through the directory where the alignments were stored. A flow diagram of this program is shown in figure 53. This program compared the two alignments to discover where they differed from one another, in other words, where the SSMAs existed. This program extracts all the structural alignment files and enters them into an array. It then goes through each structural alignment in turn and checks that the equivalent sequence alignment also exists within the directory. Once both files had been identified the program extracts the alignments from these files. This gives the program four strings of data to work with. The program then compares the two data strings for each protein domain.

During the comparison the sequences are printed out and marked with an * where the alignments differed, i.e. the SSMAs. The secondary structures of the proteins were also printed out.

The output from this file could then be analyzed to derive data such as how many SSMAs occur within each alignment and where the misalignments begin and end.

4.3 Distribution of SSMAs in a Sequence Pairing

The first analysis was to discover the distribution of the number of SSMAs within a protein domain sequence. Each output file from Checkalignment.pl was examined in turn to count the number of regions of SSMAs using the program ssma_analysis.pl. A program called analysis.pl gathered this information and then used it to produce a

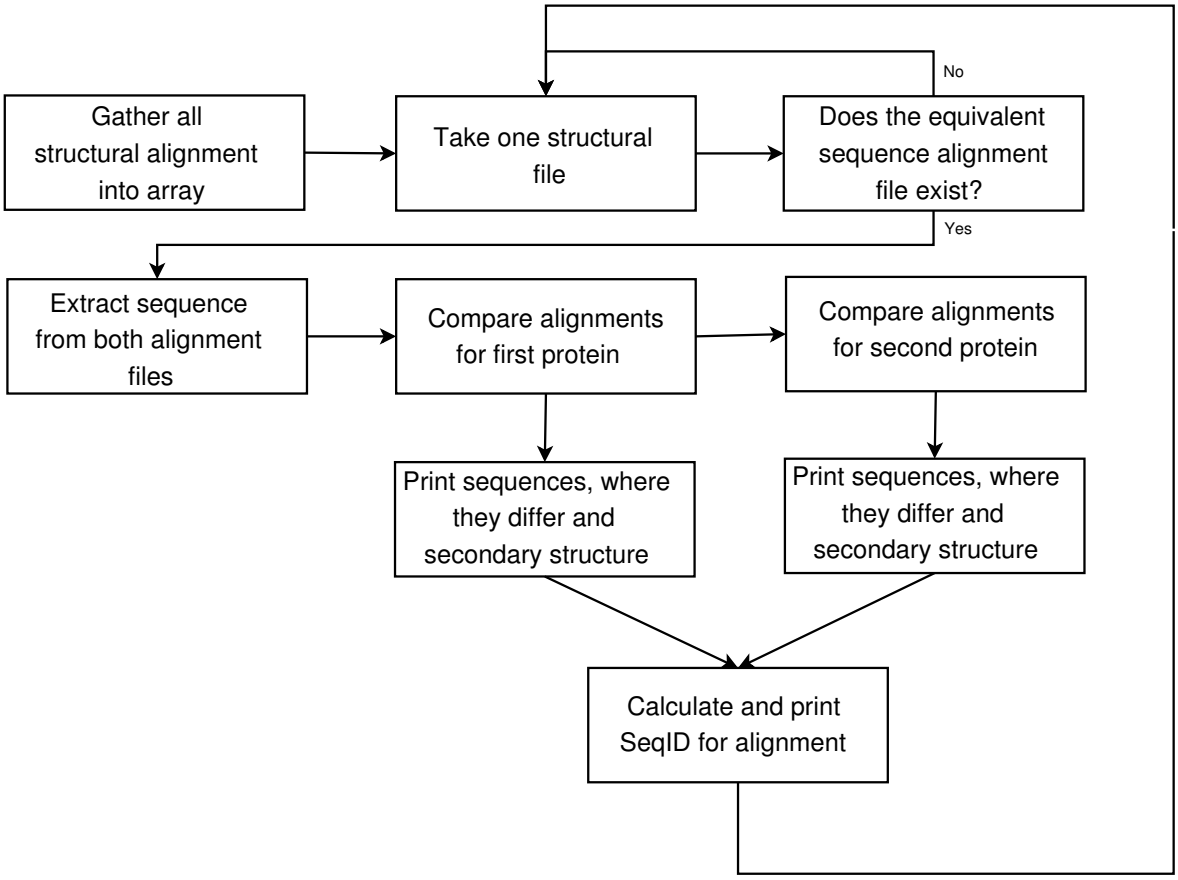


Figure 53: A flow diagram of how the program Checkalignment.pl works to find the SSMAs.

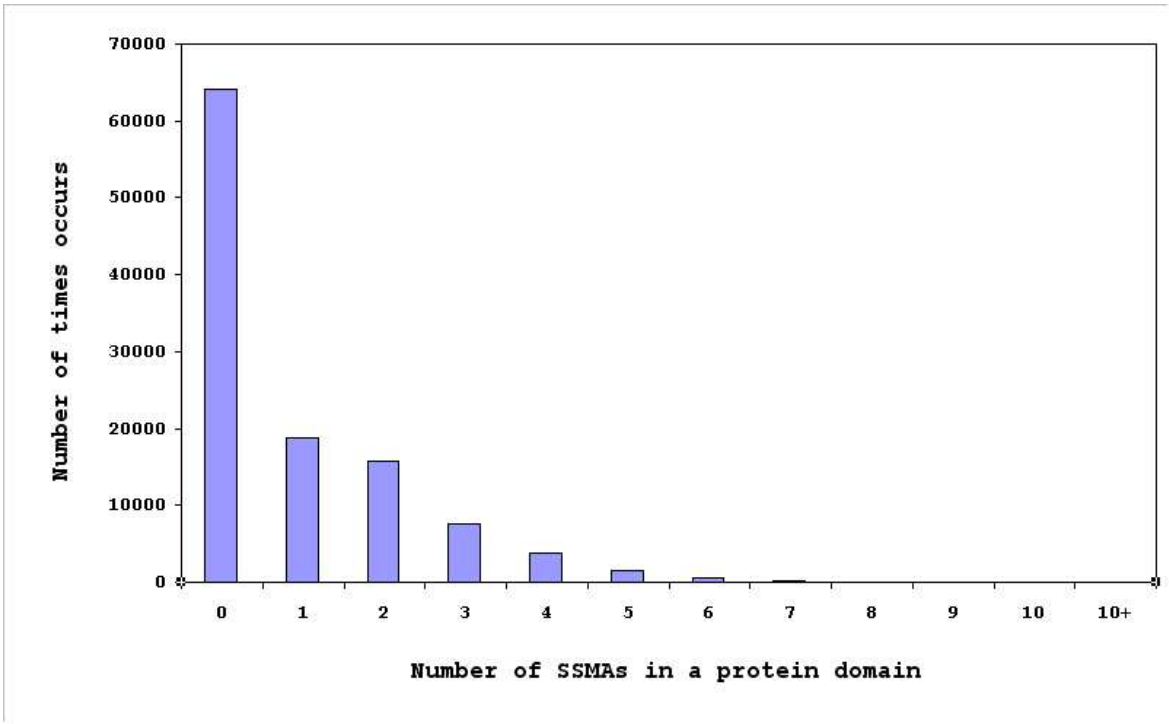


Figure 54: Distribution of the number of SSMAs within a protein domain sequence.

graph of distribution, as can be seen in figure 54.

Most comparisons of the sequence and structure alignment for a protein domain had no SSMAs within them. Obviously the length of a protein domain plays some part in this as a domain sequence can only contain a certain number of SSMAs. If the sequence and structural alignments of a domain disagree significantly they are more likely to form one large SSMA than many little ones. The alignment is more likely to be shifted into disagreement in one or two places (which can affect a large or small section of protein) than it is to be shifted many times over.

Secondly, the distribution of the total percentage of the protein sequence within a SSMA was examined. The results of this analysis can be seen in figure 55. Only those alignments which contained SSMAs were used in this analysis.

As the graph shows most of the proteins had less than 20% of their sequence's total length taken up with sections of SSMA. This would seem to indicate that most SSMAs are fairly small sections of alignment. Those pairs of sequence and structural

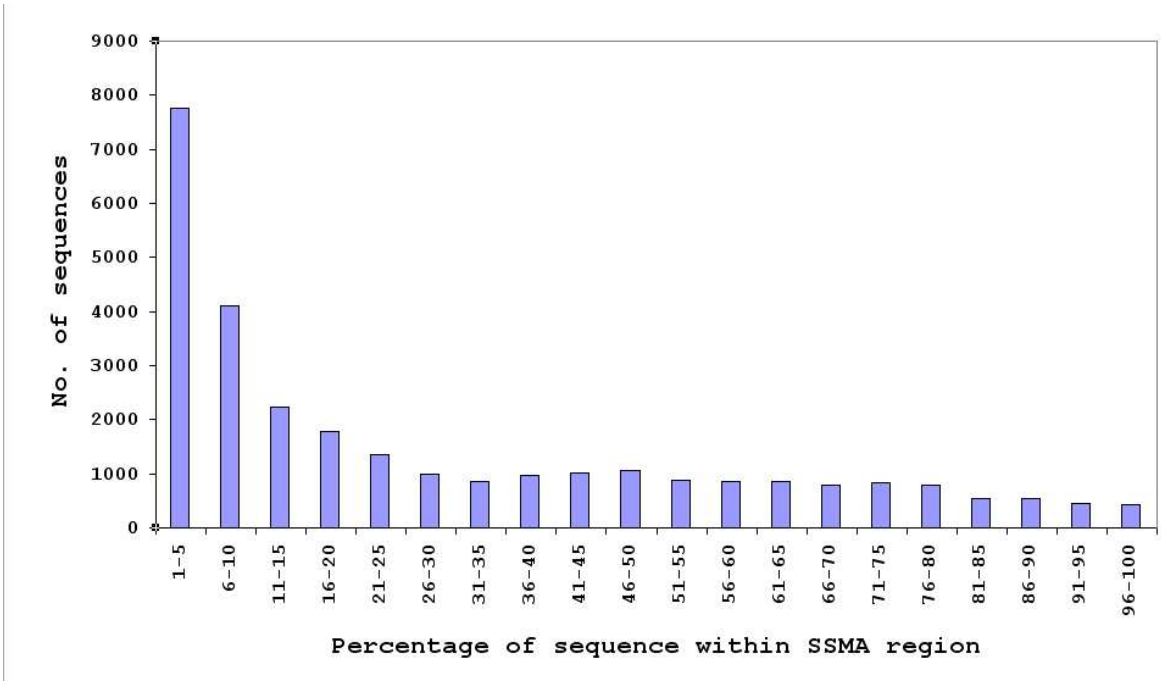


Figure 55: Distribution of the percentage of protein sequence which was within a SSMA region.

alignment which had the majority of their sequence taken up by SSMA are most likely to be alignments where there is a shift in the alignment near the beginning which continues to the end or close to it.

4.3.1 SSMA Length Analysis

Next the lengths of the individual SSMA regions were examined. The analysis.pl program was altered so that it checked through each output of the Checkalignment.pl program and counted through each SSMA region to discover its length. The lengths of the seq-str misalignments could tell us if there was an optimum length for the SSMAs or if there was a cut-off point to their length which could be introduced into the alignment prediction program. These data could also be taken into account when writing a program to generate alternative alignments.

The results of this analysis can be seen in figure 56. As the graph shows, the lengths of SSMA regions vary, but the majority are fairly small. Cases of large SSMA

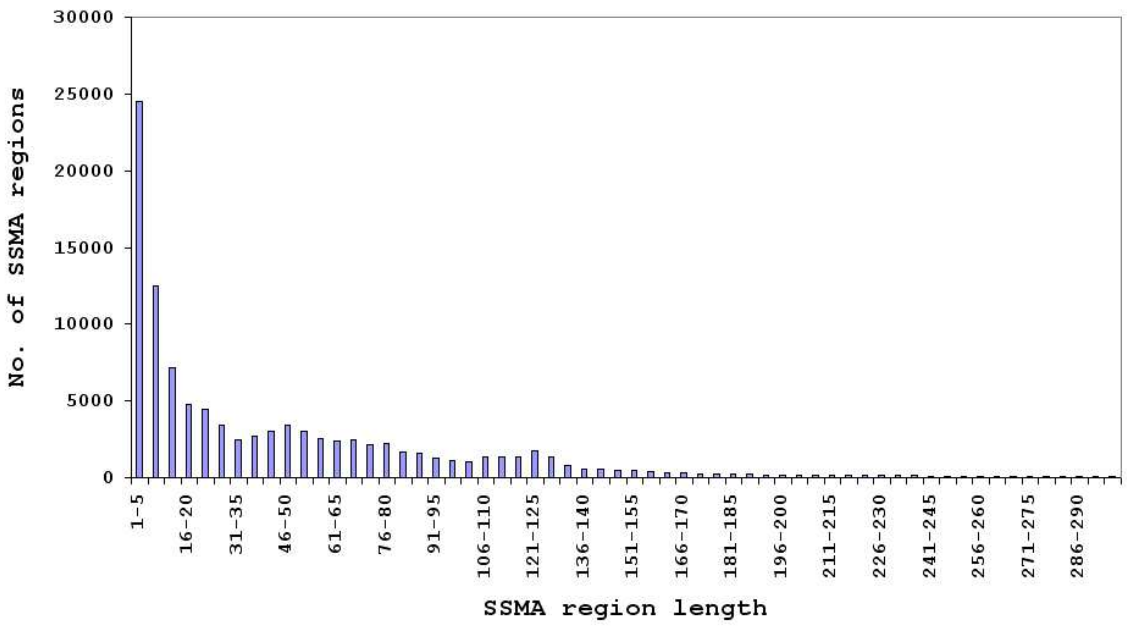


Figure 56: Distribution of the lengths of SSMA regions.

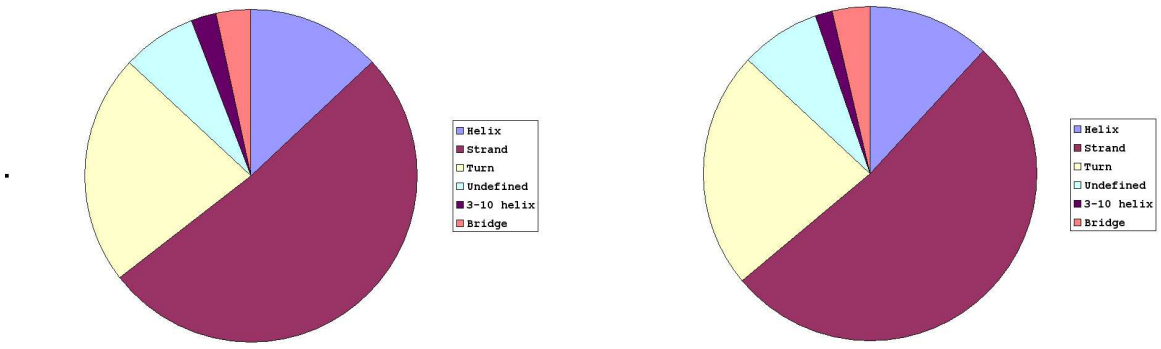
regions were caused by shifts near the beginning of the alignment that caused the entire sequences to be shifted in relation to one another.

4.4 SSMA Starting and Finishing Residues

In MLSAs it was observed that some occurred in regions of secondary structure. Secondary structure was examined to see if it played any part in SSMAs.

First, the SSMA regions were analyzed to look at the beginning of the misaligned sequence. For each one the first amino acid of the SSMA was taken and its secondary structure recorded. Secondary structure data were precalculated using the program SS (Martin, 1999) based on the SSTRUC program (Smith and Thornton, 1989) which is a modification of the DSSP algorithm (Kabsch and Sander, 1983). The secondary structures were divided into six categories, these were:

- alpha helix



a) b)

Figure 57: Secondary structures of a) the first and b) the last residue in a SSMA region. These data have been normalised to take into account the fact that the structures do not appear in even amounts within the total data set.

- beta strand
- turn
- undefined
- 3_{10} helix
- bridge (a single residue in a beta-strand structure)

The results of the analysis were normalised to take into account the fact that some secondary structures occur more often than others. For example more residues are in an alpha-helix than are in a β bridge conformation. The precalculated secondary structure data were used to calculate how many residues were in each of our six categories of secondary structure. The total number SSMA of regions beginning in each category was then divided by the total number of residues in that category.

The results of this analysis show that the majority of SSMA sequences start within a beta-strand. This can be seen in figure 57a. The second largest number of SSMA regions began in a turn, with the alpha helices coming next.

The same analysis was done for the final residue of each SSMA region. The results of this analysis can be seen in figure 57b.

The distribution is extremely similar and the majority of the final residues occur within beta strands. Slightly fewer SSMA regions finish within regions of 3_{10} helix.

The two charts suggest that secondary structure may play a significant role in causing the SSMA region.

Clearly SSMAs are not randomly distributed between secondary structures so including secondary structure assignments in the input for the neural networks gives those networks more relevant information on which to base their predictions.

Chapter 5

Predicting SSMAs Using Neural Networks

Building upon the work detailed in the previous chapter it was decided to examine the 'alignability' of a sequence. In other words, could a neural network be used to predict which regions of an individual sequence were likely to be in a region of SSMA when aligned with a relatively distant homologue.

5.1 Using a neural network to predict SSMAs

In order to try to predict where SSMAs may occur in a single sequence, a neural network was used. SNNS (Stuttgart Neural Network Simulator) (Zell *et al.*, 1995) was chosen as it has an easy graphical interface which allows interactive design of network topologies and allows the use of many learning algorithms.

Because there was a strong preference for SSMAs to begin and end within certain types of secondary structure (figure 57), input to the neural network consisted of a sequence window together with secondary structure assignments. Pre-pattern files were generated that were capable of presenting the neural net with a nine residue 'window'

```
1abc HHAGGGTHP HHHHHEEEE 0
1abc HAGGGTHPP HHHHEEEEE 0
1abc AGGGTHPPY HHHEEEEEH 1
1abc GGGTHPPYI HHEEEEEHH 1
1abc GGTHPPYII HEEEEEHHH 1
1abc GTHPPYIIR EEEEEHHHH 0
1abc THPPYIIRT EEEEHHHHH 0
```

Figure 58: An example of part of a pre-pattern file. The first column refers to the name of the protein, the second column is the protein sequence, the third to the secondary structure of the protein and the fourth to whether there is a SSMA or not. 1 = SSMA 0 = non-SSMA

of a sequence, along with its secondary structure and whether the middle residue was a SSMA or a non-SSMA. The secondary structure classes used were the same as those used in the previous chapter (α-helix, β-strand, turn, undefined, 3_{10} helix and bridge).

The pre-pattern does not explicitly identify the middle amino acid as either a transition or a non-transition. Transition refers to whether or not the middle amino acid position marks the point at which a sequence enters or leaves a region of SSMA. However, if a previous window was marked as a non-SSMA and the present window as a SSMA, then that is an 'in' transition, as it marks the points where a SSMA begins. If the present window is a non-SSMA and the previous window as a SSMA, then that is an 'out' transition, as it marks the point where a SSMA ends. An example of a pre-pattern file can be seen in figure 58.

The pre-pattern file was then converted into a SNNS pattern file in the form shown in figure 59. Using a series of zeros and ones it describes the nine-residue window that was laid out in the pre-pattern file. The pattern file can be split into nine lines of twenty digits (representing the amino acid sequence), nine lines of six digits (representing the secondary structure) and the last four digits which indicate the presence of a SSMA and a transition.

The neural networks were each set up to have a twenty by nine input unit layer

```
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
0 1 0 0 0 0
0 1 0 0 0 0
0 1 0 0 0 0
0 1 0 0 0 0
1 0 0 0 0 0
1 0 0 1
```

Figure 59: Pattern file representing the sequence and corresponding secondary structure; CFFGDWNRK HHHHEEEEH. The first set of nine lines (twenty numbers long for each possible residue that it could be) each represent a single amino acid residue in binary format. The second nine lines (each six numbers long for the six different secondary structure assignments used:helix, strand, turn, undefined, 3_{10} helix and bridge) represent the secondary structure of each of the nine amino acids in turn. Again the secondary structure is in binary. Of the final line the first two digits are for whether the central residue of the 9-residue window is a non-SSMA or an SSMA. As the binary code is '1 0', this is not an SSMA which would be represented by the binary code '0 1'. The second two digits of the final line are for where the central residue is a transition or a non-transition. '0 1' indicates that it is a non-transition, '1 0' would indicate a transition.

for the sequence, then a six by nine input unit layer for the secondary structure. As described above, the secondary structure was split up into six categories; α-helix, 3_{10} helix, β-strand, turn, bridge (single residue in β-strand conformation) and unidentified.

The programs that were used to generate these pre-pattern and pattern files can be found on the accompanying CD. They were:

- NNprepare.pl
- chooserandom.pl
- makepattern.pl (written Dr. A.C.R. Martin)

The networks had a twenty node hidden layer fully connected to the input and output layers (figure 60). The network was trained to give a yes or no answer to the questions of "Is this residue in a SSMA?" and "Is this residue the transition?". The basic layout of the neural nets can be seen in

Later networks were implemented to distinguish between transitions into a SSMA and out of a SSMA. These were identical except that they had a five node output layer. As before, the first two numbers indicated whether the central residue was a SSMA or not. The last three digits represented transitions (1 0 0 = In-transition, 0 1 0 = Out-transition, 0 0 1 = non-transition).

5.2 Training and test data sets

Previous work had used pairs of NReps within CATH homologous families. A pair of NReps can show SeqID of up to 95%. At such high sequence identities, SSMAs are unlikely to occur. For all future work SReps were used. The S family (Sequence family) level of the CATH database is set between the Homologous superfamily level and the NReps (Non-identical) level. As described on the CATH website

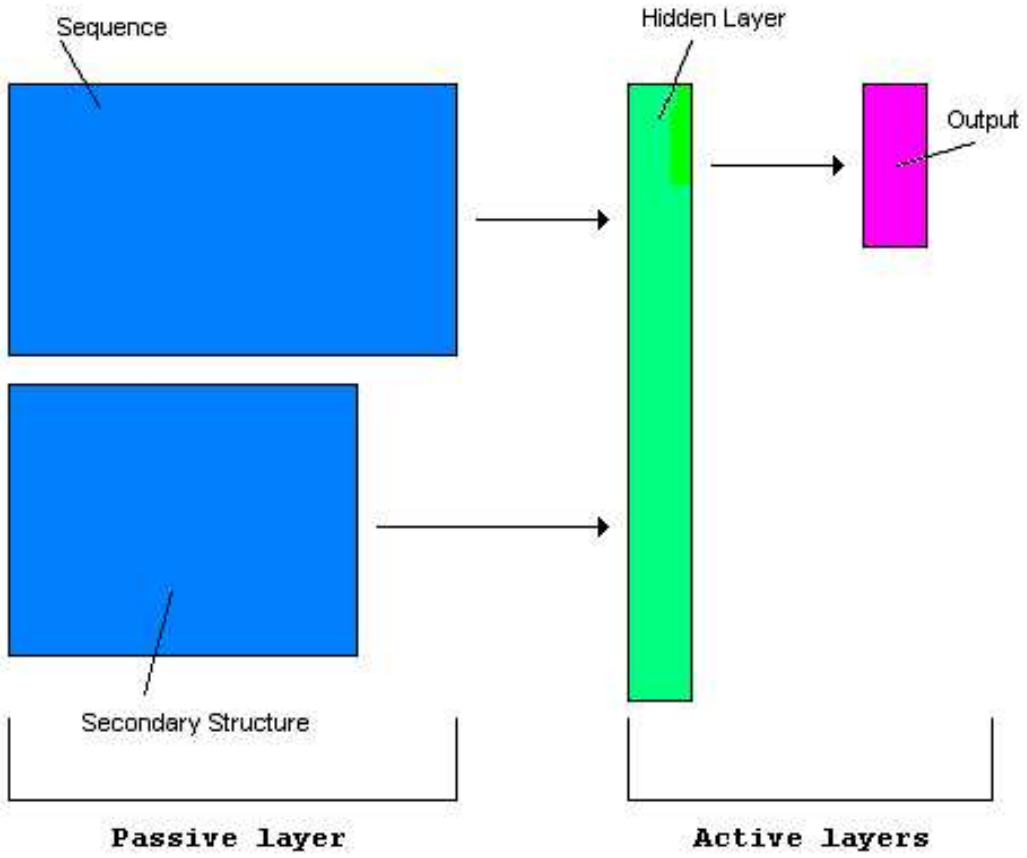


Figure 60: A typical single-hidden-layer neural network. Some neural networks had more than one hidden layer, but they were the same size as all others, consisting of twenty nodes.

(`http://cathwww.biochem.ucl.ac.uk/`), this level of the database clusters domains that have a sequence identity of $\geq 35\%$. As structure is conserved more than sequence during evolution, domains that share more than 35% identical residues nearly always have highly similar structures.

Each SRep in a homologous family was aligned against all others within that family. The programs listed above then used these alignments as input in order to create one large data set within a single file comprising 99,290 SSMA patterns and 5,524,115 non-SSMA windows. From this single file random patterns were drawn using pickrandom.pl to form the prepattern file.

The pickrandom.pl program takes the number of SSMAs (N_s), the number of non-SSMAs (N_n) and the target number of patterns (T) required for training or testing. In order to obtain a set with an approximately 1:1 ratio of SSMAs and non-SSMAs, this target number is divided by two to obtain the target number of each type of pattern ($t_s = t_n = T/2$). The program then runs through each pattern. If the pattern is a SSMA, then it generates a random number, r_s, where $0 \leq r_s < N_s$. Then if $r_s < t_s$, the pattern is output and the count of SSMAs is incremented. Similarly, if the pattern is a non-SSMA, it generates r_n, where $0 \leq r_n < N_n$; if $r_n < t$, the pattern is output and the count of non-SSMAs is incremented. Patterns not output are written to a list of unused patterns from which further training and test sets can be drawn.

If transitions are also being considered, the algorithm is slightly more complex. By definition, there will be an equal number of SSMA and non-SSMA transitions. Therefore, the program takes the number of transitions (N_t) and deducts half of this value from the number of SSMAs and the number of non-SSMAs ($N_s \leftarrow N_s - N_t/2$; $N_n \leftarrow N_n - N_t/2$). As before, the program runs through each pattern. Given a target number of transition patterns, t_t, (either 10% or 50% of the total target size, T) if the pattern is a transition, then the program generates a random number r_t, where $0 \leq r_t < N_t$. If $r_t < t_t$, the pattern is output and the count of SSMAs or non-SSMAs is

Dataset	SSMA patterns	Non-SSMA patterns	Transition patterns	Total patterns
Set 1	46380	45886	7719	92266
Set 2	46702	46181	7569	92253
Set 3	58070	47059	9476	105129
Set 4	46616	49354	9239	95970
Set 5	14080	13598	7904	27678
Set 8	11147	12046	8208	23193
Set 7	9991	11792	8843	21783
Set 8	13768	12965	8911	26733

Table 21: Datasets used in the training and testing of neural networks.

incremented as appropriate (there is an equal probability that a transition is a SSMA or a non-SSMA). If the pattern is not a transition, then the algorithm proceeds as before except that the target number of transition patterns, t_t, will be a subset of the total target number of patterns, T, so $t_s = t_n = (T - t_t)/2$.

Once the pickrandom.pl program had finished the prepattern file was then converted using makepattern.pl into the correct pattern file for training or testing a neural network. The 'Unused' file could then be used to create further data sets while being certain it held none of the patterns already used.

Four training and testing sets were used in this research. The details of these sets can be found in table 21. These were chosen at random (by the pickrandom.pl Perl program) and no pattern could be used in more than a single dataset. Sets 1 to 4 had an approximate ratio of SSMA patterns to non-SSMA patterns of 1:1, while the ratio of transition patterns to non-transition patterns was 1:9. Sets 5 to 8 had a 1:1 ratio of SSMA patterns to non-SSMA patterns and a ratio of transition patterns to non-transition patterns of approximately 1:1. The first set of neural networks were trained using one of the datasets 1 to 4 and then tested using the other three. The networks trained to focus more on predicting transitions used datasets 5 to 8 in the same way.

The nets were trained with a 100,000 series of nine-residue patterns, complete with secondary structure and an indication of whether they were a SSMA or not, transition or not. The testing pattern file sets were of a similar size. A larger size of testing and training pattern files was attempted, but computer memory limitations prevented this. The only exceptions to this size of training and testing set were those where the ratio of transitions to non-transitions was 1:1. Due to there only being around 36,000 transition sequence windows the test and training sets had to be reduced to 20,000. In all other test pattern files the ratio of non-transitions to transitions was 9:1. However in order to see if a more even number of transitions and non-transitions would improve that area of prediction the ratio was altered. The ratio of SSMAs to non-SSMAs in the pattern files was always 1:1.

5.3 Different Parameters of Neural Nets

Different parameters for the neural networks were used in an effort to improve their predictive ability. For each series of neural nets a 'control' network was also trained, this being a network where all the parameters were left as the default or as 0. This gave an idea of how well each net performed when compared to an unparameterised network. The basic layout of the neural networks can be seen in figure 60, containing one passive (input) layer and two active (hidden and output) layers.

Some neural networks had more than one hidden layer. The second hidden layer was connected to the secondary structure input layer and the output layer. In these neural networks both hidden layers contained twenty nodes. A diagram of this layout can be seen in figure 61.

The original series of neural networks were trained starting with the control and then altering the parameters slightly each time based on previous results. Each new set of parameters was selected with the previous sets in mind. These sets of parameters

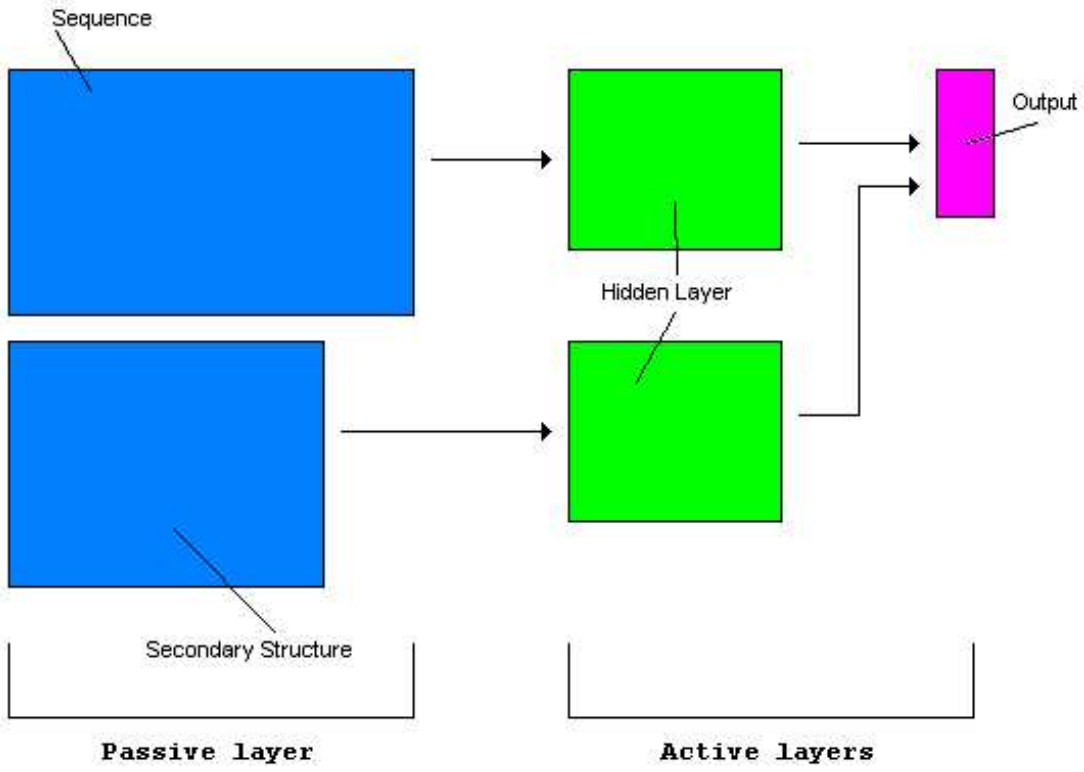


Figure 61: Neural network layout consisting of a single passive layer and three active (two hidden and one output) layers. The two hidden layers handle the sequence and secondary structure data respectively.

can be seen in table 22. The default parameters meant that the training method was left as the default (standard back-propagation) and all values were left as 0 except for the number of cycles trained. This control would allow for comparison with the other neural networks to see how much of an improvement each change in parameters would cause.

Some networks were 'jogged' during the training process to avoid the network getting stuck in an energy minimum. Jogging adds a small random number, between specified bounds, to the weights before each cycle of training. These limits were set to either ± 0.1 or ± 0.01.

Neural nets were trained for 1000 cycles under a variety of other conditions. The Rprop (Resilient backpropagation) training method was found to work well at correctly predicting the position of SSMAs. Nets were at first trained using just the recommended settings for the training method; $\delta_0 = 0.1$ (the initial update-value), $\delta_{max} = 50.0$ (the limit for the maximum step-size) and $\alpha = 4$ (the weight-decay exponent). The non-standard setting for the Rprop training method were $\delta_0 = 0.2$, $\delta_{max} = 50.0$ and $\alpha = 4$.

5.4 Results of Training Sets

Owing to the very large amounts of available data, jack-knifing or cross-validation was not necessary - multiple separate large test and validations sets could be created.

Once trained, the neural nets were then tested using three pattern files other than the one with which they were trained. These testing pattern files did not contain any of the same protein windows as the training file. The output of the testing gives confidence values which were then compared with the original pattern file to discover whether the position was correctly predicted for SSMAs and transitions. These then allowed for the calculation of percentages of correctly predicted SSMAs and correct

Network name	Training Parameters
SSMAtrained191103.net†	All parameters left as default, no jogging, single hidden layer, trained for 1000 cycles
SSMAtrained201103.net	Rprop with recommended settings, single hidden layer, no jogging, 1000 cycles
SSMAtrained211103.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
SSMAtrained221103.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.1 to 0.1, 1000 cycles
SSMAtrained241103.net	Rprop with recommended settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
SSMAtrained011203.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
SSMAtrained021203.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.1 to 0.1, 1000 cycles
SSMAtrained031203.net	Rprop with recommended settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
SSMAtrained041203.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 2000 cycles
SSMAtrained041203double.net	Rprop with recommended settings, double hidden layer, jogged every epoch -0.01 to 0.01, 2000 cycles
SSMAtrained081203.net	Rprop with non-standard settings, single hidden layer, jogged every epoch -0.1 to 0.1, 1000 cycles

Table 22: Parameters of the first trained neural networks. †SSMAtrained191103.net was used as a control. Datasets used in this training had a 1:1 ratio of SSMAs:non-SSMAs, and a 9:1 ratio of non-transitions:transitions.

transitions.

The Matthews' correlation coefficient (Matthews, 1975) (MCC) was calculated for the results of each testing set. The coefficient is calculated using the equation:

$$\text{MCC} = \frac{TP_x TN_x - FP_x FN_x}{\sqrt{(TP_x + FN_x)(TP_x + FP_x)(TN_x + FP_x)(TN_x + FN_x)}} \tag{14}$$

where TP_x, TN_x, FP_x, FN_x are the numbers of true positives, true negatives, false positives and false negatives for state x. The value of MCC is between -1 and +1. A value of -1 indicates total disagreement, +1 total agreement and 0 a completely random prediction and yields easy comparison with respect to a random baseline (Baldi *et al.*, 2000).

The results of the initial neural networks trained can be seen in table 23. The values in the table represent the averaged data from the three testing files. The control net (SSMAtrained191103.net) performed poorly, only predicting the presence or absence of SSMAs correctly 50% of the time. This gave a good series of values with which to compare the other neural network results. Two networks performed very well, predicting the SSMAs over 86% correctly and the transitions over 96% correctly: SSMAtrained211103.net and SSMAtrained031203.net (See table 22). Of all the networks, SSMAtrained031203.net predicted the SSMAs correctly most often while SSMAtrained081203.net predicted the transitions better by pure percentages.

Although all of the nets appear to have performed well in predicting the transitions it must be remembered that each testing set had a ratio of 9:1 non-transitions to transitions. This means that the network could always predict a position as being a non-transition and be correct approximately 90% of the time. Consequently, further networks were trained using a 1:1 ratio of transitions to non-transitions.

Network name	SSMAs predicted correctly	MCC SSMAs predicted correctly	Transitions predicted correctly	MCC Transitions predicted correctly
SSMAtrained191103.net†	50.0%	0.0019	94.9%	0.7338
SSMAtrained201103.net	59.0%	0.1897	92.2%	0.6276
SSMAtrained211103.net	86.4%	0.7285	96.5%	0.8052
SSMAtrained221103.net	75.9%	0.5183	95.5%	0.7628
SSMAtrained241103.net	64.3%	0.7023	86.0%	0.8344
SSMAtrained011203.net	82.6%	0.7230	96.7%	0.8464
SSMAtrained021203.net	78.1%	0.6441	95.5%	0.7917
SSMAtrained031203.net	89.1%	0.7978	96.5%	0.8592
SSMAtrained041203.net	83.9%	0.7343	96.4%	0.8070
SSMAtrained041203double.net	84.3%	0.7442	96.4%	0.8372
SSMAtrained081203.net	83.1%	0.7179	97.0%	0.8500

Table 23: Results of first trained neural networks. †SSMAtrained191103.net was used as a control.

5.4.1 Predicting SSMA Transitions

Five nets were trained using the 20,000 pattern files with the 1:1 ratios for both SSMAS:non-SSMAs and transitions:non-transitions (datasets 5 to 8 in table 21). The first four nets were different to one another to see if any one particular series of parameters would improve the number of correctly predicted sequence windows. Single and double hidden layers were used, as were the recommended and non-standard parameters for Rprop. As before the recommended settings for Rprop refer to $\delta_0 = 0.1$, $\delta_{max} = 50.0$ and $\alpha = 4$, while the non-standard settings refer to $\delta_0 = 0.2$, $\delta_{max} = 50.0$ and $\alpha = 4$. The final net was designed as a control using parameters that had previously given low numbers of correct SSMA predictions. The parameters of these nets can be seen in table 24.

As before the networks were tested with different pattern files to those that they were trained with. Testing was once again performed using three pattern files, this time each of 20,000 patterns due to lack of transitions. The averaged results of the three testing files for these networks can be seen in table 25.

Network name	Training Parameters
Half110204.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Half120204.net	Rprop with non-standard settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Half130204.net	Rprop with recommended settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Half140204.net	Rprop with non-standard settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Half150204.net†	Rprop (parameters set to zero), single hidden layer, no jogging, 1000 cycles

Table 24: Parameters for neural networks trained with pattern files containing the 1:1 ratio of non-transitions:transitions. †Half150204.net as used as a control.

Network name	SSMAs predicted correctly	MCC SSMAs predicted correctly	Transitions predicted correctly	MCC Transitions predicted correctly
Half110204.net	83.5%	0.6697	94.8%	0.8882
Half120204.net	83.8%	0.6766	95.1%	0.8962
Half130204.net	84.2%	0.6846	94.1%	0.8736
Half140204.net	84.7%	0.6948	95.5%	0.9030
Half150204.net†	53.8%	0.0760	82.8%	0.6443

Table 25: Results of training the neural networks with a 1:1 ratio of non-transitions to transitions. †Half150204.net as used as a control.

The control performed worse than the rest of the networks as expected. The neural networks predicted the presence or absence of the transitions very well, though not as well as some of the previous networks. They did tend to give slightly better values for the Matthews' correlation coefficient though. The SSMAs were also consistently predicted well even with the much smaller pattern files. However they still did not do as well as some of the initial series of networks did.

5.5 Predicting in/out transitions

The neural nets had been able to predict the occurrence of SSMAs extremely well. They had also been capable of predicting where the transitions occur. The next neural nets were altered from four outputs (SSMA, non-SSMA, transition, non-transition) to five outputs (SSMA, non-SSMA, in-transitions, out-transition, non-transition), to train and test nets with the intention of predicting an in-transition and an out-transition separately.

For the first set of neural networks training and test pattern files using a 9:1 non-transition:transition ratio were used. This meant that there were approximately 5% in-transitions and 5% out-transitions in each pattern file. New datasets had to be created to train and test these neural networks, these can be seen in table 26. The parameters of all the neural nets trained with these pattern files can be seen in table 27. The percentages of transitions and SSMAs predicted correctly can be seen in table 28. As these neural networks produced three interdependent outputs the values for the Matthews' correlation coefficient equation had to be altered. The modified equation to calculate the MCC values is:

$$\mathrm{MCC} = \frac{(p_i + p_j)n - (u_i + u_j)o}{\sqrt{((p_i + p_j + o)(p_i + p_j + u_i + u_j)(n + o)(n + u_i + u_j))}} \qquad (15)$$

Dataset	SSMA patterns	Non-SSMA patterns	In-Transition patterns	Out-Transitions patterns	Total patterns
Set 1	51600	48631	4507	4759	100231
Set 2	49726	49028	3871	4692	98754
Set 3	49104	47737	4622	5027	96841
Set 4	50928	48914	4524	3816	99842

Table 26: Datasets used in the training and testing of neural networks where in-transitions and out-transitions were predicted separately.

Network name	Training Parameters
Alteredtrained070504.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Alteredtrained080504.net	Rprop with non-standard settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Alteredtrained090504.net	Rprop with recommended settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Alteredtrained100504.net	Rprop with non-standard settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Alteredtrained110504.net†	Rprop (no settings), single hidden layer, no jogging, 1000 cycles

Table 27: Parameters of the neural networks trained with pattern files containing the 1:1 ratio of in-transitions:out-transitions. †Alteredtrained110504.net was used as a control.

where p_i = the number of true positives, p_j = half the number of positions correctly predicted as transitions but incorrectly predicted as an in/out transition when it is an out/in transition, n = the number of true negatives, o = the number of false positives and u_i = the number of false negatives, u_j = half the number of positions correctly predicted as transitions but incorrectly predicted as an in/out transition when it is an out/in transition. The MCC values given in this way can be seen in table 28.

The previous neural nets were trained with only 5% in-transitions and a similar number of out-transitions. The next set of neural nets was trained with the same parameters as the previous five nets but with different training and testing pattern files, seen in table 29. The parameters of each neural network can be seen in table 30. Instead

Network name	SSMAs predicted correctly	In-Transitions predicted correctly	Out-Transitions predicted correctly	Non-transitions predicted correctly	MCC
Alteredtrained070504.net	85.0%	61.0%	65.7%	99.4%	0.8170
Alteredtrained080504.net	85.1%	55.7%	66.1%	99.4%	0.8050
Alteredtrained090504.net	82.8%	64.8%	66.1%	99.4%	0.8340
Alteredtrained100504.net	75.4%	65.4%	69.3%	99.2%	0.8250
Alteredtrained110504.net†	86.2%	43.6%	72.6%	99.3%	0.7778

Table 28: Results of the neural networks trained with pattern files containing the 1:1 ratio of in-transitions:out-transitions. †Alteredtrained110504.net was used as a control.

Dataset	SSMA patterns	Non-SSMA patterns	In-Transition patterns	Out-Transitions patterns	Total patterns
Set 1	9457	9952	3784	5004	19409
Set 2	11674	9739	4855	4706	21413
Set 3	9708	10468	4638	4271	20176
Set 4	10041	9510	3994	4087	19551

Table 29: Further datasets used in training and testing neural networks for predicting In/Out-Transitions.

Network name	Training Parameters
Alteredtrained180504.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Alteredtrained190504.net	Rprop with non-standard settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Alteredtrained200504.net	Rprop with recommended settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Alteredtrained210504.net	Rprop with non-standard settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
Alteredtrained220504.net†	Rprop (no settings), single hidden layer, no jogging, 1000 cycles

Table 30: Parameters for the neural networks trained with pattern files containing the 1:1:2 ratio of in-transitions:out-transitions:non-transitions. †Alteredtrained220504.net was used as a control.

of only using 5% of each transition, pattern files were created with only approximately 20,000 protein windows. Within each pattern file 10,000 protein windows were transitions. Therefore each pattern file had roughly 25% in-transitions, 25% out-transitions and 50% non-transitions.

The networks were tested with three test sets, each containing none of the patterns from the training file or the other test files. These files contained 20,000 protein windows with the transitions in the same ratios as they were in the training file. The results of these neural nets can be seen in table 31.

Comparing the results in table 28 and table 31 it would seem that the non-transitions are always predicted well. However in order to predict whether a transition is into a SSMA or out of a SSMA it is necessary to increase the number of

Network name	SSMAs predicted correctly	In-Transitions predicted correctly	Out-Transitions predicted correctly	Non-transitions predicted correctly	MCC
Alteredtrained180504.net	84.8%	87.5%	90.0%	95.8%	0.8748
Alteredtrained190504.net	84.6%	84.6%	82.9%	95.8%	0.8481
Alteredtrained200504.net	84.4%	89.6%	90.7%	95.8%	0.8849
Alteredtrained210504.net	82.2%	85.9%	86.5%	95.8%	0.8594
Alteredtrained220504.net†	85.7%	87.7%	89.0%	95.6%	0.8726

Table 31: Results for the neural networks trained with pattern files containing the 1:1:2 ratio of in-transitions:out-transitions:non-transitions. †Alteredtrained220504.net was used as a control.

these types of transitions that the neural network is exposed to during the training phase. This is as expected.

Altering the transition ratios does not seem to have had an effect predicting the absence or presence of SSMAs correctly. Both sets of networks predicted the SSMA positions between 75% to 86% of the time.

5.6 Confidence Scores

In addition to the Matthews' Correlation coefficient a confidence was also calculated. Outputs for predictions are not binary, but are a real value between 0 and 1. These values for SSMA and non-SSMA can be combined to give a confidence score rather than simply selecting the higher value.

In order to measure the confidence that the neural networks had in their predictions the equation in figure 16 was used. The confidence score returned from this equation was between -1 and 1. A value close to -1 indicated a confident prediction of a non-SSMA and a value close to 1 indicated a SSMA.

$$\text{Confidence score} = 2 \times (\frac{Py}{Py + Pn} - 0.5) \tag{16}$$

where Py = predicted probability of the position being SSMA and Pn = predicted probability of the position being non-SSMA.

The neural networks from table 24 were used for the calculation of confidence scores. These nets were used because there was a good level of difference between the control nets and the actual nets. If the neural nets were predicting SSMA positions well then the graphs of confidence would favour the -1 and +1 sides of the graph. If the neural nets were predicting the SSMA positions poorly because there was only a small difference between its SSMA and non-SSMA score from the results file, then the distribution would tend towards the values closer to 0.

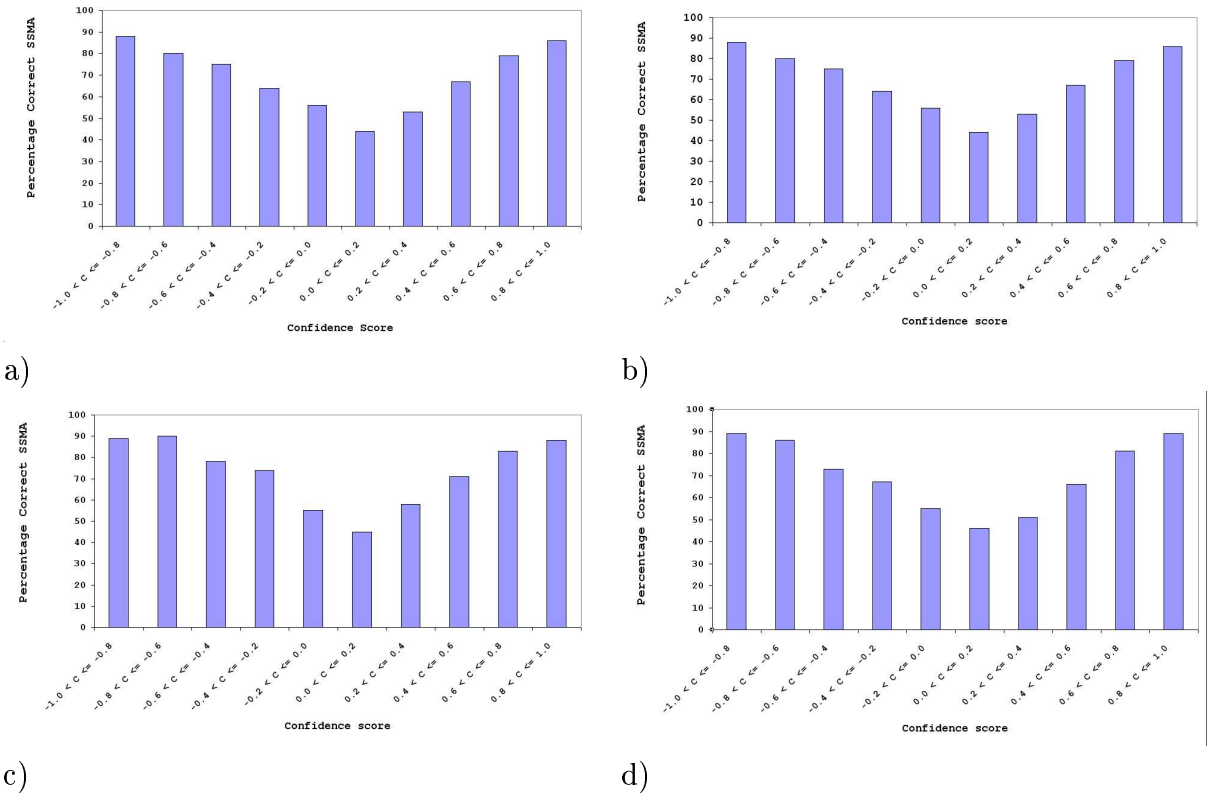


Figure 62: Distribution of confidence scores for networks trained with a 1:1 ratio of transitions to non transitions: a) half110204.net b) half120204.net c) half130204.net and d) half140204.net

Figure 62 shows the distribution of confidence scores for the neural nets (figure 63 is a control). The mean can be found in table 32. The nets were tested with a 20,000 protein window pattern file. All bar the control net figure 63 show the confidence values favouring the -1 and +1 extremes. Only the confidence scores resulting from the control neural net (Half150204.net) give a different pattern of results. As the control net is supposed to perform poorly, confidence scores of close to 0 were expected.

The confidence of the other trained neural nets was calculated in the same way. Of all the networks, SSMAtrained031203.net performed best for SSMA prediction as figure 64a shows. The other networks had means closer to 0 and larger standard deviations. This, coupled with its very good correct prediction rates and MCC values, made this the best of the single sequence neural networks that trained. The only network that came close was SSMAtrained081203.net, its confidence scores graph can

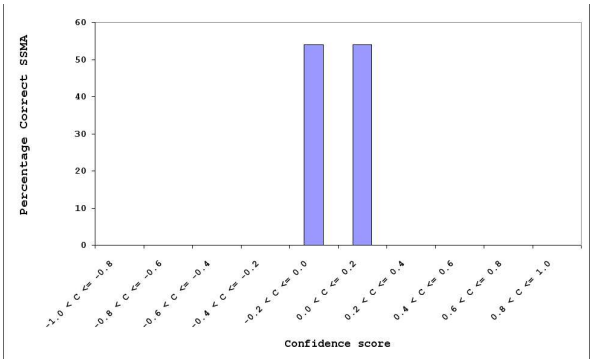
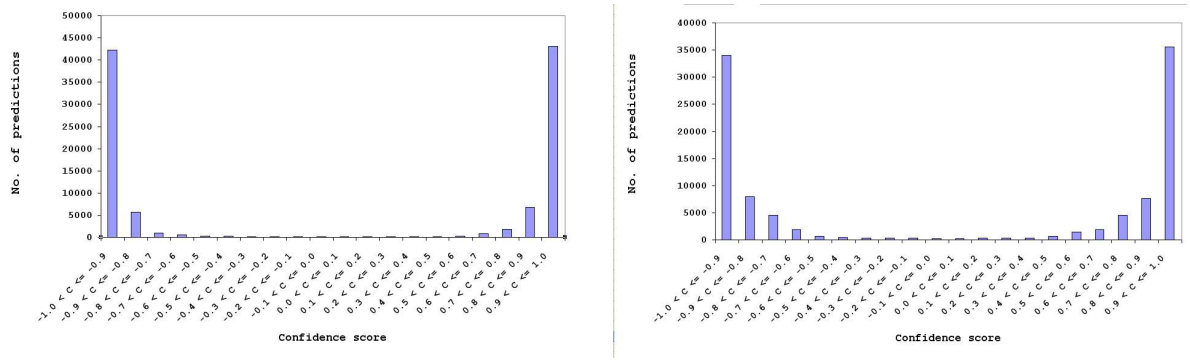


Figure 63: Distribution of confidence scores for the control network, half150204.net

Network name	Mean for values less than 0	Mean for values greater than than 0
half110204.net	-0.841	0.823
half120204.net	-0.880	0.872
half130204.net	-0.822	0.810
half140204.net	-0.848	0.839
half150204.net†	-0.059	0.091
SSMAtrained031203.net	-0.952	0.966
SSMAtrained081203.net	-0.926	0.916

Table 32: The mean values for data presented in figures 62, 63, 64. †half150204.net was used as a control.



a) b)

Figure 64: Distribution of confidence scores for networks trained with a 1:9 ratio of transitions:non-transitions. a) SSMAtrained031203.net and b) SSMA-trained081203.net

be seen in figure 64b. For these reasons these two networks were later used for the networks detailed in the later chapter "Dual Sequence Prediction".

5.7 ROC Plots

Receiver operating characteristic (ROC) (Peterson, 1954; Swets, 1988; Gribskov and Robinson, 1996) analysis is a powerful and widely used technique for assessing predictive methods (Ison and Blades, 2005). A ROC curve is defined as a plot of test sensitivity versus its 1-specificity or false positive rate (Park *et al.*, 2004) for different threshold points of a parameter (See `http://www.medcalc.be/manual/mpage06-13b.php`). In this research the threshold was the confidence of the neural network in its predictions of a position being considered a SSMA or a non-SSMA. The neural networks predicted two values, one for the chance of it being a SSMA and one for the chance of it being a non-SSMA. If the prediction for it to be a SSMA position was higher than the prediction for it to be a non-SSMA position then it was marked as predicted as a SSMA position. This equates to using a threshold (confidence score) of 0.

A ROC plot was calculated for the best of each set of trained networks. In each one, different threshold values of the confidence score between -1 and +1 were used

and the true and false negatives and positives calculated. The true and false negatives and positives were then used to calculate the sensitivity and selectivity of the networks using that cutoff value. Sensitivity is calculated using the equation:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{17}$$

where TP is the number of true positives and FN is the number of false negatives. The selectivity of the neural network at the threshold points were calculated using:

$$\text{Selectivity} = \frac{TN}{FP + TN} \tag{18}$$

where FP is the number of false positives and TN is the number of true negatives.

Figure 65 shows the ROC plot for SSMAtrained031203.net. As the graph shows using a cutoff of 0 in this case seems to have been ideal as it has a high sensitivity showing that it has the majority of the true positives. At the same time it has a low $1 - Selectivity$ which shows that it is not returning a large number of false positives. The same is demonstrated again in figure 66 for half140204.net and in figure 67 for altered020504.net. A different cutoff for deciding whether a position was predicted as SSMA or non-SSMA would not have improved the number of true positives or decreased the number of false positives by much.

5.8 Prediction of an Alignment Pairing

The networks were tested using a randomly chosen set of 9-residue windows but were also tested using examples of actual domain sequences. The domain sequences chosen all contained SSMAs and were randomly chosen from all the pairings that made up the data set but had not been used in either the training or the testing pattern files. Eight pairings were chosen as seen in table 33.

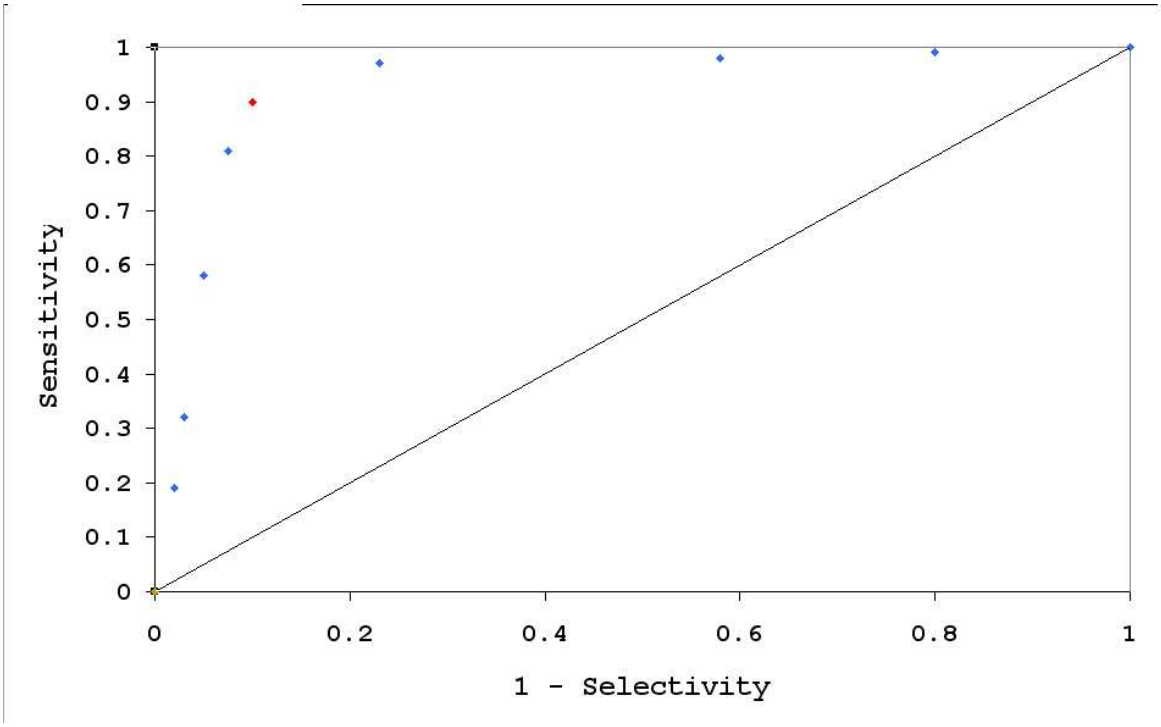


Figure 65: ROC plot for SSMAtrained031203.net. The red point is the value used in the confidence analysis, the black line is the value expected by chance.

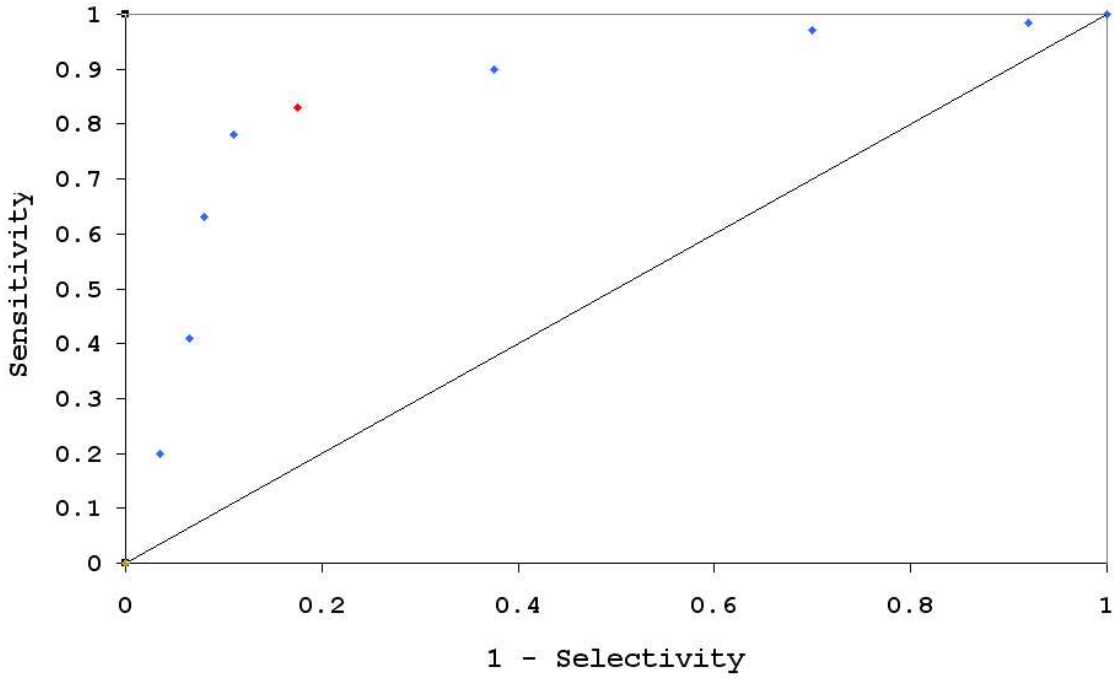


Figure 66: ROC plot for half140204.net. The red point is the value used in the confidence analysis, the black line is the value expected by chance.

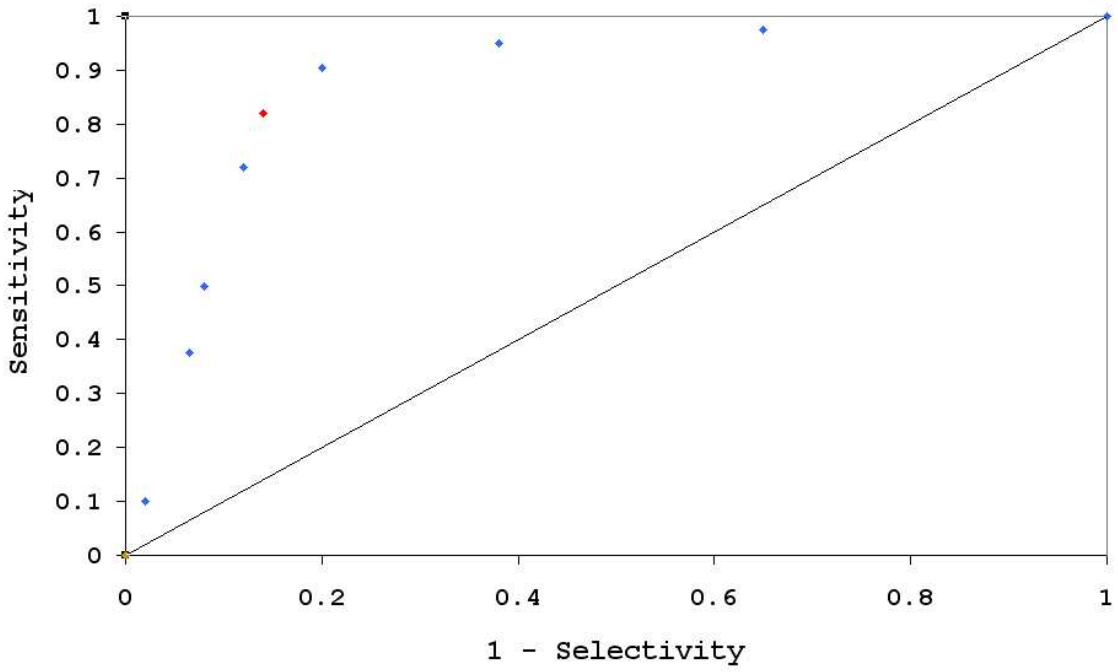


Figure 67: ROC plot for altered020504.net. The red point is the value used in the confidence analysis, the black line is the value expected by chance.

Sequence pairing	% predicted correct
1rtfB1 1sluB2	84.4
3rp2A2 1rtfB2	91.2
3hhrB1 1cfb01	77.9
1zxq02 1flrH2	90.9
1hrnA1 1b5fA0	80.6
2plv40 1bev40	63.2
1ytiA0 1hrnA1	80.7
1flrH2 12e8L1	92.9

Table 33: The eight alignments that were used in the neural network testing and the % SSMA positions predicted correctly.

Prediction confidence	Symbol
$0.0 \leq \|C\| \leq 0.2$	nothing
$0.2 < \|C\| \leq 0.4$	.
$0.4 < \|C\| \leq 0.6$	:
$0.6 < \|C\| \leq 0.8$	|
$0.8 < \|C\| \leq 1.0$	#

Table 34: The symbols used by the graphical.pl program to represent the different levels of confidence in a SSMA prediction.

```
         ++      ++   ++     ++       ++++  +          Predicted transitions
          ******* **** ****** ******** * ***********   Predicted SSMAs
##################################################   Confidence
          ******* **** ****** ******** * ***********   Actual SSMAs
PWQAAIFAKHRGERFLCGGILISSCWILSAAHCFQERFPPHHITVILGRTYR   Domain sequence


++  + +      + +     + + +      ++   +++    ++++
 ***  ******* *******  * ****** ****** ****** * ****
|################################.##########################
 ***  ***************  ******** ****** ****** * ************
VVPGEEEQKFEVEKYIVHKEFDDDTYDNDIALLQLKSDSSRCAQESSVVRTNYLDWIRDN
```

Figure 68: Example of the output of the program graphical.pl for the protein domain 1rtfB2.

The program graphical.pl, which can be found in the accompanying CD, was used to take the raw output of the neural network and convert it into a visually comprehensible annotated sequence. As well as indicating where the sequences were correctly and incorrectly predicted as SSMA or transitions it also gave a graphical indication of how confident the network's predictions were. An example of the output of this program can be seen in figure 68. The confidence (as calculated by Equation 17) is indicated by symbols shown in table 34.

Chapter 6

Predicting With Two Sequences At Once

The single sequence prediction neural networks in the previous chapter identified areas of a sequence likely to be misaligned. The neural networks trained in this chapter were designed to indicate where the alignment between two sequences was correct or incorrect. Rather than using a single sequence with its secondary structure, the input for these neural nets was created from an aligned pair of domain sequences and the secondary structures of those sequences. The layout of these nets can be seen in figure 69. Both SSMAs and transitions were predicted using these networks.

6.1 Training and testing data files

The datasets were created from the same SRep alignments as were used in the previous chapter. These networks were trained and tested with pattern files of approximately 100,000 patterns. Once again a single training pattern file was used for each neural network and tested with three test sets. Each pattern could appear only once in any of the pattern files to ensure that if a network was trained with a certain pattern it

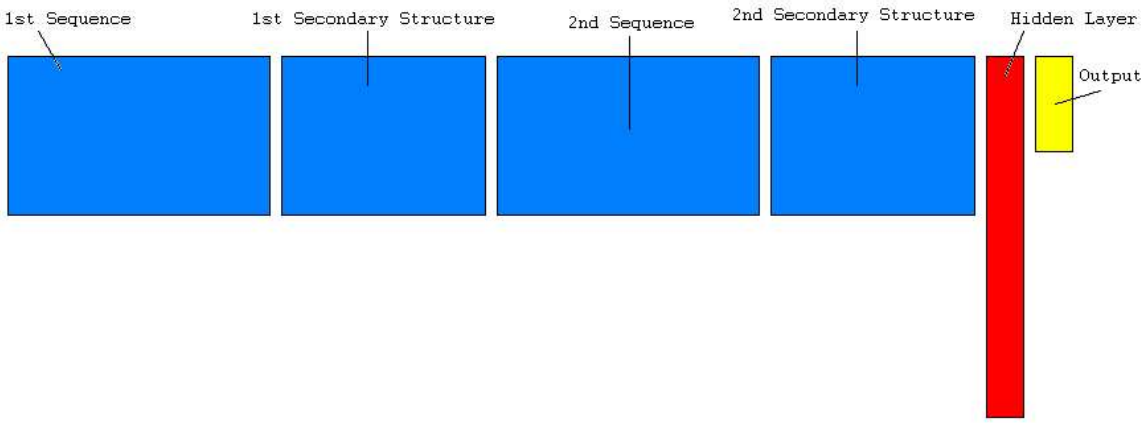


Figure 69: The layout of the dual sequence neural networks. The blue boxes represent the input layer, the red the hidden layer and and the yellow the output layer.

could not be tested with that same pattern.

The ratio of SSMAs to non-SSMAs was maintained at approximately 1:1. The transitions were kept at a ratio of 9:1 non-transitions:transitions. This low level of transitions was done even though previous nets had shown that a ratio of 1:1:2 in-transitions: out-transitions:non-transitions predicted the transition positions better. This was owing to the fact that only 20,000 protein windows could be used with that ratio of transitions. Because of the increasing complexity of the patterns, a 100,000 pattern file was necessary for training these neural networks. There were not enough in- and out- transitions to create four pattern files of 100,000 windows with 1:1 ratios.

These pattern files were generated by slightly altering the same set of programs used in the single sequence SSMA neural networks. The datasets used for training and testing can be seen in table 35.

6.2 Different Parameters of Neural Nets

The control network used Rprop but its parameters were left at 0 and trained for 1000 cycles. Other training parameters were then set up based on previous experiences with

Dataset	SSMA patterns	Non-SSMA patterns	Transition patterns	Total patterns
Set 1	46684	46529	6559	93213
Set 2	48439	45691	8631	94130
Set 3	49818	50742	9654	99923
Set 4	46832	46784	6602	93616

Table 35: Datasets used in the training and testing of neural networks when predicting two sequences at once.

training the single sequence prediction networks. As the networks were trained and tested the parameters were developed for successive networks based on the earlier ones. All of the parameters can be seen in table 36.

Most of the networks had a single hidden layer of 20 nodes, if no other value is specified in table 36 then the network had this type of hidden layer. In some cases double hidden layers were used which were made up of two layers, each made up of 20 nodes. Two other types of hidden layer were tried, one small (5 node) and one large (50 node) to try to improve the predictions of the neural networks.

6.3 Results of Training Sets

The results of the neural networks can be seen in table 37. The Matthews' correlation coefficient was calculated for both the number of correctly predicted SSMAs and the number of correctly predicted transitions. As the results show although all but one of the networks predicted above 60% they did not perform as well as the previous single sequence prediction networks. It is important to note that these nets are predicting something rather different, namely whether two sequences are correctly aligned, rather than the 'alignability' of a single sequence. As the training and test pattern files contained half SSMAs and half non-SSMAs if the network had predicted all the patterns as SSMA then it would have been correct 50% of the time. Comparing this value to

Network name	Training Parameters
dualtrained100804.net†	Rprop (no settings), single hidden layer, no jogging, 1000 cycles
dualtrained110804.net	Rprop with recommended settings, single hidden layer, no jogging, 1000 cycles
dualtrained120804.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
dualtrained130804.net	Rprop with non-standard settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
dualtrained160804.net	Rprop with recommended settings, single small (1 x 5) hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
dualtrained170804.net	Rprop with recommended settings, double hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
dualtrained180804.net	Rprop with recommended settings, single large (5 x 10) hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
dualtrained030904.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
dualtrained060904.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles, trained with four pattern sets
dualtrained130904.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles, trained with pattern file with no gaps
dualtrained160904.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles, based on dualtrained130904.net
dualtrained200904.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles, trained with second pattern file with no gaps
dualtrained210904.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 10000 cycles, trained with second pattern file with no gaps
dualtrained220809.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles, trained without secondary structure patterns
dualtrained240908.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 10000 cycles

Table 36: Parameters for the dual-trained neural networks. †dualtrained100804.net was used as a control.

Network name	SSMAs predicted correctly	MCC SSMAs predicted correctly	Transitions predicted correctly	MCC Transitions
dualtrained100804.net†	64.3%	0.2864	92.9%	0.6513
dualtrained110804.net	64.2%	0.2840	92.9%	0.6513
dualtrained120804.net	67.0%	0.3400	92.6%	0.6408
dualtrained130804.net	67.0%	0.3400	92.6%	0.6408
dualtrained160804.net	67.0%	0.3393	92.9%	0.6513
dualtrained170804.net	67.5%	0.3506	92.9%	0.6513
dualtrained180804.net	66.9%	0.3374	92.2%	0.6265
dualtrained030904.net	66.3%	0.3266	92.3%	0.6307
dualtrained060904.net	24.4%	-0.5130	54.1%	0.0440
dualtrained130904.net	66.3%	0.3262	92.9%	0.6513
dualtrained160904.net	62.1%	0.2427	92.9%	0.6513
dualtrained200904.net	67.7%	0.3542	92.9%	0.6513
dualtrained210904.net	66.6%	0.3316	92.9%	0.6513
dualtrained220904.net	62.3%	0.3461	92.9%	0.6513
dualtrained240904.net	60.2%	0.2034	92.9%	0.6513

Table 37: Results for the dual trained neural networks. †dualtrained100804.net was used as a control.

the ones reached by these networks shows that the networks did not predict more than 10-17% above this random value.

The transitions were predicted correctly more than 92% of the time in all but one of the networks. However with the non-transition to transition ratio being 9:1 this means that had the network simply predicted all the patterns as non-SSMA it would have been correct 90%. Taking this into consideration the prediction of the transitions for these networks is not very good. Especially not compared to the transition prediction rate that some of the networks had in the single sequence prediction.

The Matthews' correlation coefficient values confirm this view. None of the SSMA predictions managed to get more than 0.3542 although transition prediction reached 0.6573.

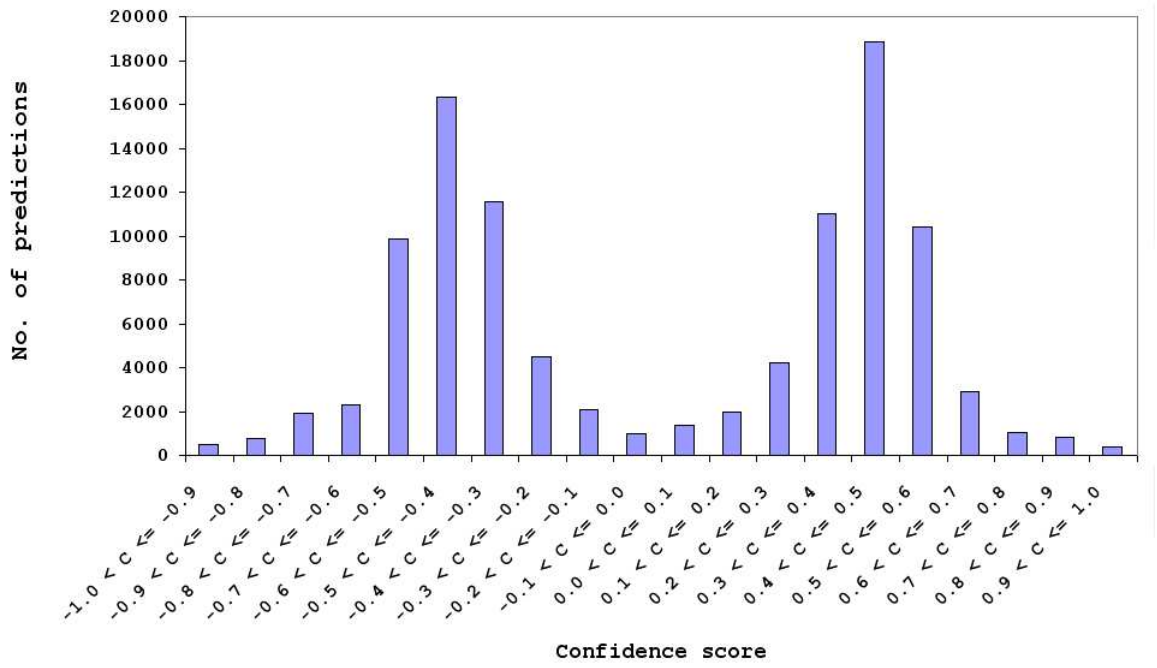


Figure 70: Distribution of confidence scores for dualtrained200904.net.

6.4 Confidence Scores

The confidence scores were calculated as before using equation 16.

Encouragingly, in addition to the low accuracy and MCC scores, the confidence scores were also low. Figure 70 shows the confidence scores of the best of the two sequence SSMA prediction networks (dualtrained200904.net). Unlike previous neural network confidence scores the majority of the confidence scores are clusters between -0.5 to 0.5 rather than at -1 and 1.

6.5 ROC Curves

As had been done before a ROC curve was calculated for the best of the networks trained in this chapter, dualtrained200904.net. Selectivity and Sensitivity were calculated according to the equations 18 and 19. Figure 71 shows the results of altering the cutoff value for whether a position is predicted as a SSMA or non-SSMA. Compared

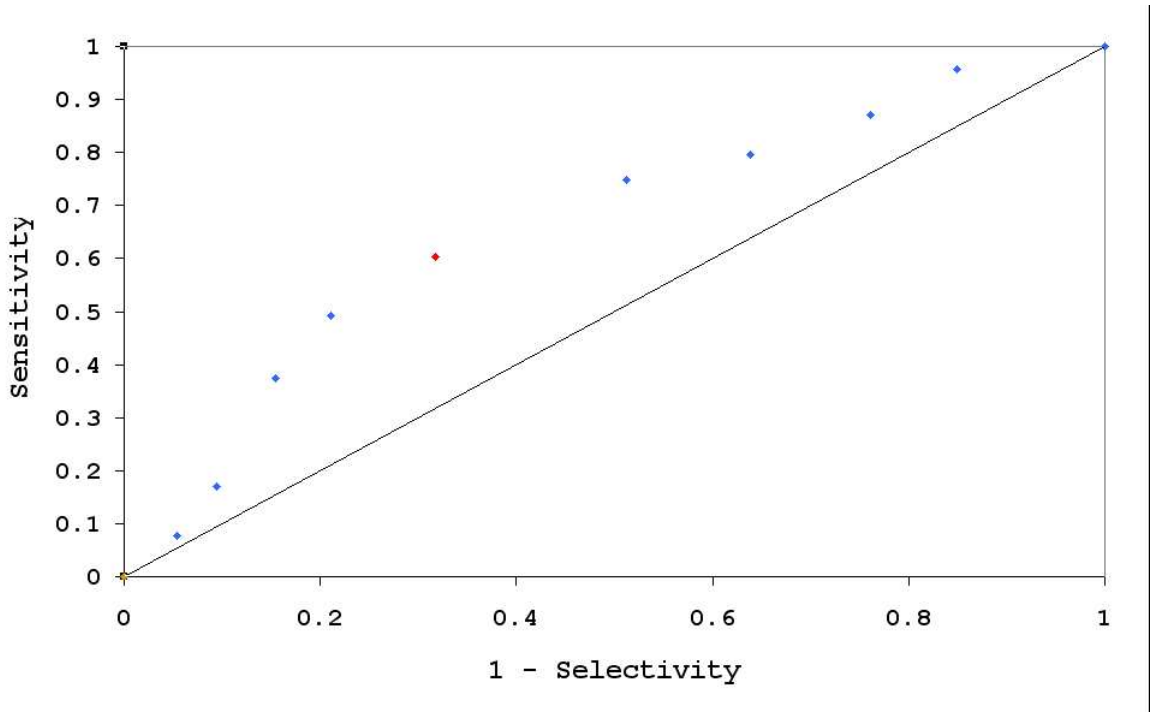


Figure 71: ROC plot for dualtrained200904.net. The red point is the value used in the confidence analysis, the black line is the value expected by chance.

with the ROC curves in the previous chapter, this graph shows that the neural network performed poorly.

6.6 Prediction of an Alignment Pairing

Using these networks to examine an alignment would be expected to give us a poor idea of where the SSMAs occur. Although the networks do predict correctly in places the percentages of correct SSMA predictions would lead us to believe that approximately 40% of the alignment would be incorrectly predicted by these neural networks. Figure 72 shows the alignment between 1rtfB1 (two chain tissue plasminogen activator from *Homo sapiens*) and 1sluB2 (anionic N143H, E151H trypsin complexed to A86H ecotin from *Rattus norvegicus*) and where the best of the neural networks (dualtrained200904.net) predicted it correctly. The diagram was produced using the graphical.pl program as used in the previous chapter.

```
                +                                       Predicted transitions
                                    ***  *   *          Predicted SSMAs
                         ..:::::::::::::::::.::::::::   Confidence
                                                        Actual SSMAs
-----------------------PYQVSLNSGY--HFCGGSLINDQWVVSAAH   1sluB2
IKGGLFADIASHVCLPPADLQLPDWTECELSGYGKHEALSPFYSERLKEAHVR   1rtfB1

                               +       +     ++        ++
   ***             * **       **               ***     *     *    *
.......                     ..:::.....................:::::::.....
                              ***   ******** *************** ******
CYKSRIQVTLGEHNINVLEGNEWFVNAAKIIKHPNFDRKTLNNDIMLIKLSSPVKVATNYVDWIQDT
LYPSSRCTSQHLLNRTVTD-NMLCAGDTRSNLH---DACQGDSGGPLVCLNDGRMTLVGIISWGLGC



*     **
 ............
*****
IA-----------*
GQKDVPGVYTKVT*
```

Figure 72: The SSMA and transition predictions of dualtrained200904.net for the alignment pairing 1sluB2 and 1rtfB1. All the predictions are at quite low confidence. See table 34 for the confidence symbols.

As figure 72 shows, these nets failed to correctly predict some of the position of the SSMAs in this case. The confidence is also shown to be low.

Chapter 7

Dual Sequence Prediction

Since the networks that were designed to predict SSMAs using two aligned sequences at once did not give the expected results, a different approach was taken, based on the successful single sequence SSMA networks. This process used a sequence of two neural networks, one that predicted where a single sequence was likely to have SSMAs and one that then combined this information with an alignment in order to predict where the SSMAs occurred within the alignment. A flowchart of this can be seen in figure 73. A series of programs were used in order to achieve this (see the accompanying CD), they were:

- genssmafile.pl (written by Dr. A.C.R. Martin) — using the sequence and structural alignment files, genssmafile.pl generates a file containing the true SSMA positions in a sequence
- wrapgenssm.pl — given the directory name where the sequence and structural alignments exist, this program takes each pair in turn and presents them to genssmafile.pl to generate SSMA files for all protein domain pairs in a directory
- getss.pl (written by Dr. A.C.R. Martin) — when presented with a PDB identifier, getss.pl generates a file containing the secondary structure of the appropriate

sequence. Secondary structure information is extracted from a pre-calculated SST file generated by SSTRUC (Smith and Thornton, 1989).

- wrapgetss.pl — given the directory name where the sequence and structural alignments exist, wrapgetss.pl takes the name of each structural alignment file and pulls out the PDB identifiers to present to getss.pl, one at a time, to generate secondary structure files for all protein domain pairs

- makeinput1.pl (written by Dr. A.C.R. Martin) — takes an output file from genssmafile.pl and the matching output file from getss.pl and combines the SSMA position information, alignments and secondary structure into a single entry in a separate file

- wrapmakeinput1.pl — using the name of the directory containing the sequence and structural alignment, this program uses makeinput1.pl to make a file containing a table of SSMA positions, alignments and secondary structure for all protein domain pairs

- blackboxSSMA.pl — using a sequence alignment file (in FASTA format) this program uses a pre-made wrapmakeinput1.pl file to create a pattern file and submit it to a trained single sequence prediction neural network. This is then predicts the single sequence SSMAs within a sequence and creates a new table within a new file containing SSMA position information, the predicted SSMA position information, alignments and secondary structure

- blackboxSSMA2.pl — uses blackboxSSMA.pl to predict the single sequence SSMAs for each sequence in an alignment and then uses the results to prepare and run a pattern file through a trained dual sequence prediction neural network to predict the dual sequence SSMAs for a protein domain pair and print the information to an output file

- wrapblackbox.pl — when presented with a directory name, goes through each sequence alignment and structural alignment pair of files and presents the files to blackboxSSMA2.pl to generate dual sequence SSMA predictions

The pattern files for the dual sequence networks were significantly different from those used in the previous chapter's neural nets. It was hoped that by increasing the amount of information provided to the network that there would be a greater amount of success. Also by giving the second neural network the results of the single sequence SSMA predictions it was hoped that this would help in SSMA prediction in an alignment. The pattern files contained:

- The sequence of the first domain as it appears in the alignment
- The sequence of the first domain without any gaps
- The secondary structure of the first domain as modified by the alignment
- The secondary structure of the first domain without any gaps
- The single sequence neural network SSMA prediction for the first domain
- The sequence of the second domain as it appears in the alignment
- The sequence of the second domain without any gaps
- The secondary structure of the second domain as modified by the alignment
- The secondary structure of the second domain without any gaps
- The single sequence neural network SSMA prediction for the second domain

The layout of the dual sequence neural networks can be seen in figure 74. The trained neural network for the single sequence SSMA prediction used was SSMA-trained031203.net. It had the best combination of SSMAs predicted correctly, highest

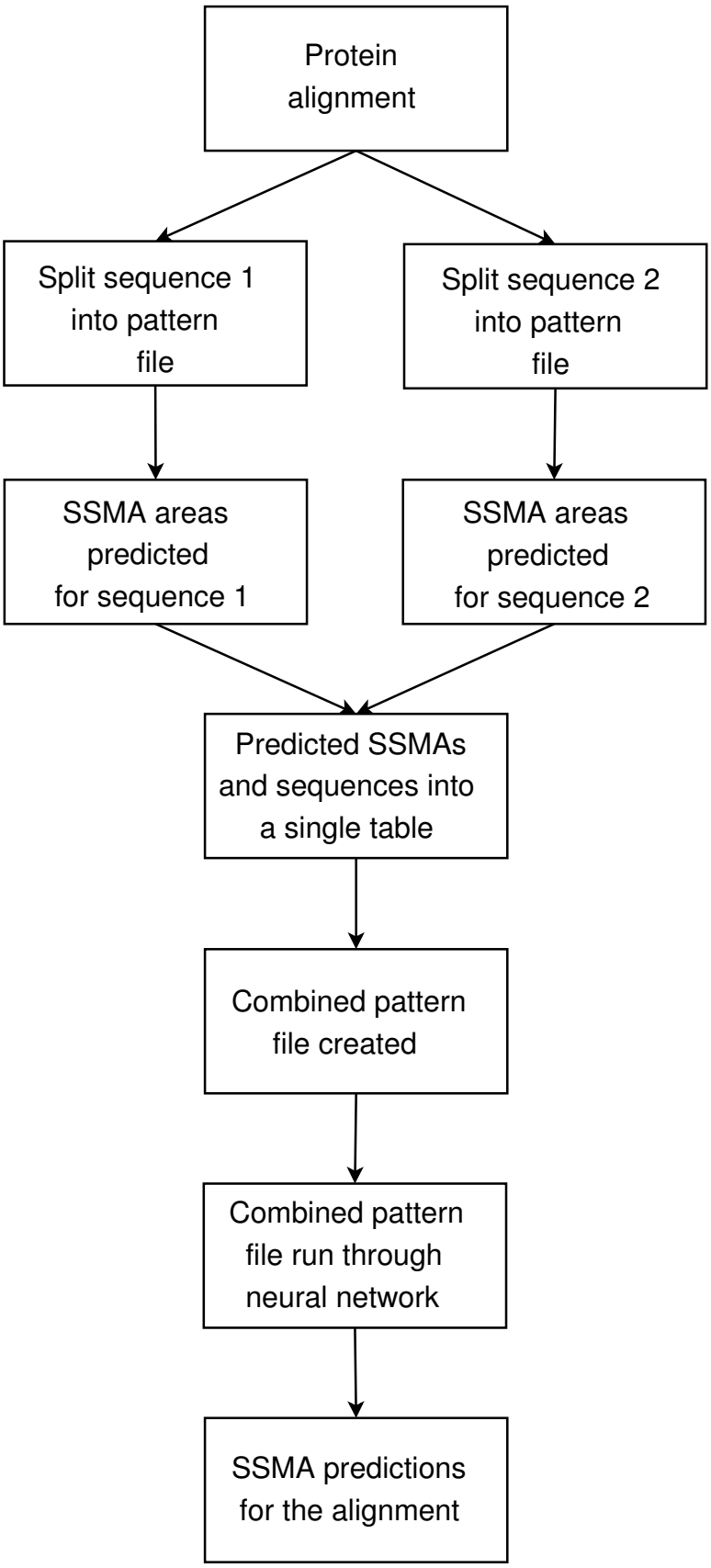


Figure 73: A flowchart of the neural network setup used in this chapter.

Dataset	SSMA patterns	Non-SSMA patterns	Total patterns
Set 1	53635	49787	103422
Set 2	53356	50165	103521
Set 3	52751	51078	103829
Set 4	53264	49792	103056

Table 38: Datasets used in the training and testing of neural networks for dual sequence prediction.

MCC values and best confidence scores. Only in one set of parameters was a different single sequence SSMA prediction network used; SSMAtrained081203.net. It achieved results that were almost as good as SSMAtrained031203.net. The layout of these networks can be seen in figure 74.

7.1 Different Parameters of Neural Nets

As before pattern files of approximately 100,000 9-residue windows were used to train and test the networks. A 9-window pattern could only appear in one testing or training set to prevent the networks being exposed to the same pattern on more than one occasion. The datasets used for training and testing of these neural networks can be seen in table 38.

The first network trained was once again a control, using Rprop but leaving the parameters set to zero. Other networks were tried with differing training parameters that were based on those that had been used before (see section 5.3). These parameters can be found in table 39.

7.2 Results of Training Sets

These neural networks only predicted the presence or absence of SSMAs, not transitions. Each neural network was tested with three pattern files and the values then

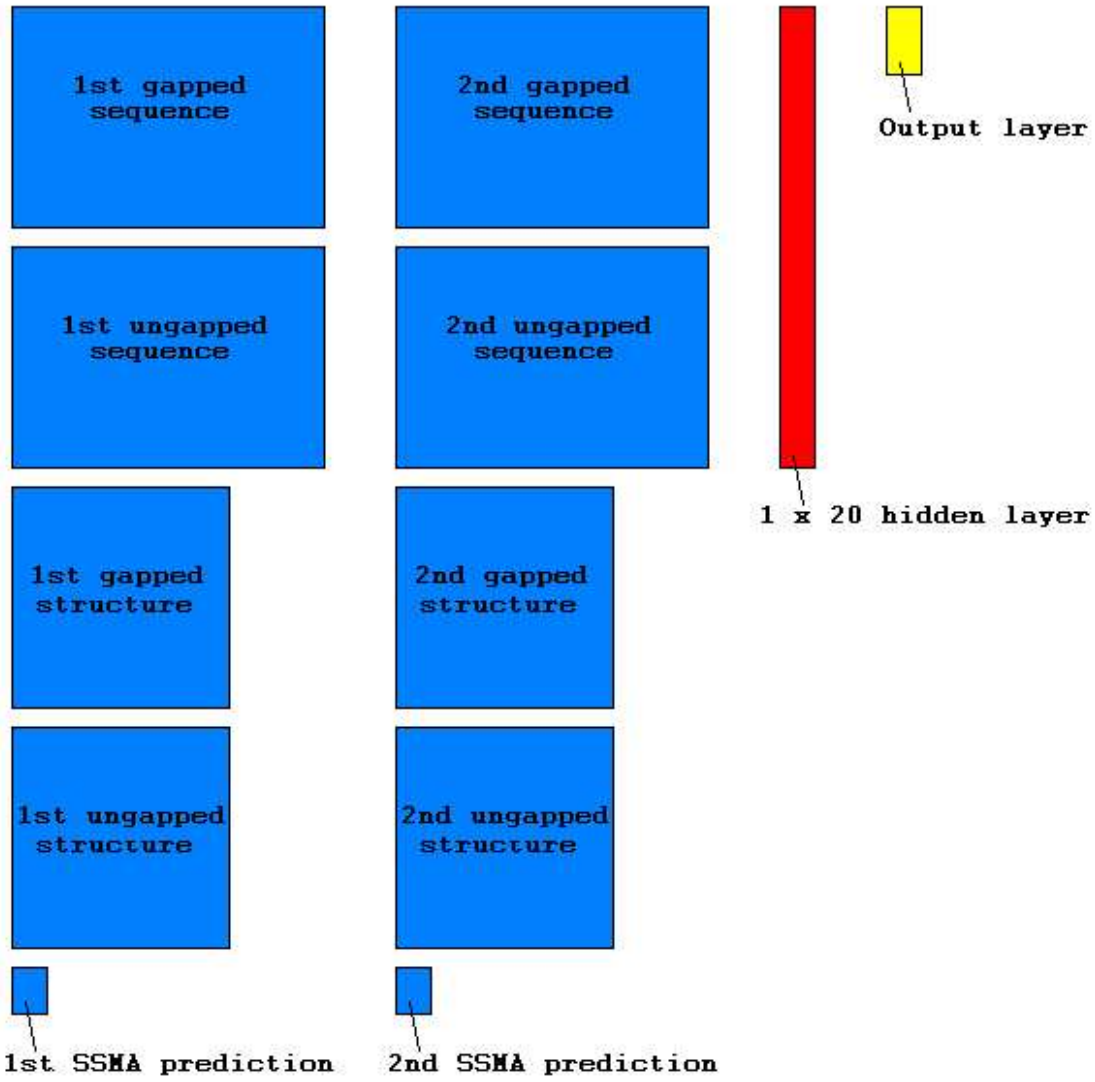


Figure 74: The layout of a typical pattern file used in the dual sequence neural networks. The inputs used were the sequences, secondary structures and SSMA prediction for both windows of the aligned sequences.

Network name	Training Parameters
negcontrol	untrained neural network
smoothing291104.net (control)	Rprop (no settings), single hidden layer, no jogging, 1000 cycles
smoothing021204.net	Rprop with recommended settings, single hidden layer, no jogging, 1000 cycles
smoothing031204.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles
smoothing101204.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.1 to 0.1, 1000 cycles
smoothing131204.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.001 to 0.001, 1000 cycles
smoothing161204.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 2000 cycles
smoothing161204.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 4000 cycles
smoothing130105.net	Rprop with recommended settings, single hidden layer, jogged every epoch -0.01 to 0.01, 1000 cycles, based off SSMAtrained081203.net

Table 39: Parameters of the dual sequence neural networks.

Network name	SSMAs predicted correctly	MCC SSMAs predicted correctly
negcontrol	48.5%	-0.0243
smoothing291104.net (control)	69.9%	0.4045
smoothing021204.net	70.7%	0.4147
smoothing031204.net	83.3%	0.6660
smoothing101204.net	92.9%	0.6481
smoothing131204.net	82.7%	0.6542
smoothing161204.net	82.5%	0.6503
smoothing171204.net	85.1%	0.7027
smoothing130105.net	82.9%	0.6585

Table 40: Results of the dual sequence neural networks.

averaged. The results of the testing can be seen in table 40. The results of these neural networks were much better than those found in the previous chapter with smoothing101204.net predicting the SSMAs correctly 92.9% of the time. However, the Matthews' correlation coefficient values are not as good as those that the single sequence neural networks achieved.

A negative control was introduced into this series of networks to compare with the others. It was an untrained network, the same one that the others were trained from. As expected it performed extremely poorly, making predictions at random ($\tilde{5}0\%$ correct predictions, MCC $\tilde{0}$). Comparing the other networks to that value it is possible to see how much better the others performed.

Using SSMAtrained081203.net as the single sequence SSMA prediction neural network did not improve the training of the dual sequence network. It performed as well as some of the other networks trained using SSMAtrained031203.net.

Of all the neural networks smoothing101204.net predicted SSMAs the best. It achieved a correct prediction rate of 92.9%, better than even the single sequence SSMA prediction networks were capable of. Of course those networks were predicting areas in a single sequence that were likely to have SSMAs when aligned with another. These

networks were predicting where an alignment was correct and where it was incorrect.

The Matthews' correlation coefficients for smoothing101204.net were not as good as for some of the other neural networks. Another network, smoothing171204.net, had an MCC average value of 0.7027, but predicted the positions as SSMA or non-SSMA correctly 85.1% of the time. These values should be compared with the best performance of 67.5% (MCC = 0.3542) seen in table 37 of the previous chapter.

7.3 Prediction of an Alignment Pairing

Taking smoothing101204.net as the neural network that performed best it was used on an alignment pairing. As before the alignment between 1rtfB1 (two chain tissue plasminogen activator from *Homo sapiens*) and 1sluB2 (anionic N143H, E151H trypsin complexed to A86H ecotin from *Rattus norvegicus*) was used. The alignment was converted into a pattern file using the programs listed above. This pattern file was then fed into smoothing101204.net and the results analyzed.

As figure 75 shows the neural network was able to predict the presence of the SSMAs in the alignment. Thus the method indicates where the alignment should be altered as the sequence and structural alignments disagree.

7.4 Confidence Scores

These networks performed very well in terms of predicting the SSMA regions correctly. The MCC values were not as good as the single sequence neural networks achieved but still meant that the predictions seemed significant in terms of true and false negatives and positives. The confidence values were also calculated to see how confident the networks were. These in combination with the prediction rates and MCC values would determine how well the networks had done.

```
                                                  *                Predicted SSMAs
##############################################|||#############   Confidence
                                                                   Actual SSMAs
----------------------PYQVSLNSGY--HFCGGSLINDQWVVSAAHCYKSRIQ   1sluB2
IKGGLFADIASHVCLPPADLQLPDWTECELSGYGKHEALSPFYSERLKEAHVRLYPSSRC   1rtfB1


                     ** ****   *********** ** *******   ****
##################|||:|#######################################
                        ***   ******** ************** ******
VTLGEHNINVLEGNEWFVNAAKIIKHPNFDRKTLNNDIMLIKLSSPVKVATNYVDWIQDT
TSQHLLNRTVTD-NMLCAGDTRSNLH---DACQGDSGGPLVCLNDGRMTLVGIISWGLGC


 ****
##############
*****
IA-----------*
GQKDVPGVYTKVT*
```

Figure 75: The SSMA predictions of smoothing101204.net for the alignment pairing 1sluB2 and 1rtfB1. See table 34 for the confidence symbols.

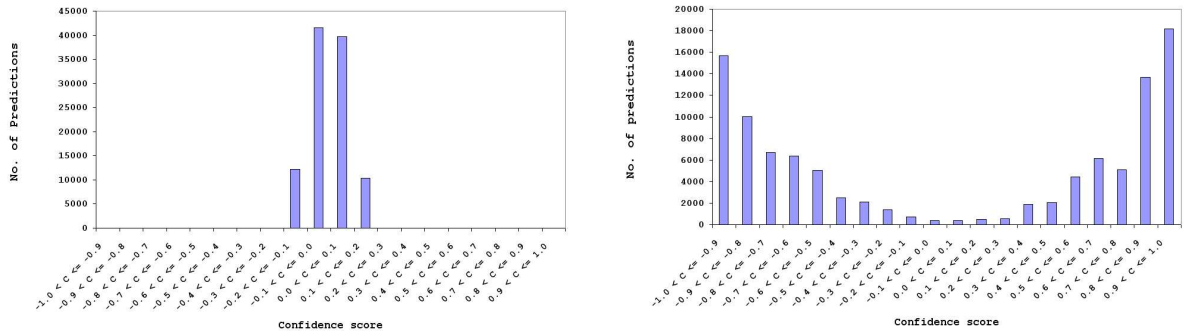
SSMAs	53751	51.8%
non-SSMAs	50078	48.2%
Total patterns	103829	

Table 41: Counts and percentages for the smoothing141204.pat pattern file.

As before, the confidence scores were calculated using Equation 16.

All of the networks were tested using the same pattern file, smoothing141204.pat which contained no patterns that any of them had been exposed to during the training phase. The single sequence SSMA predictions that formed part of the neural network input were based on predictions made by SSMAtrained031203.net. The numbers of SSMAs and non-SSMAs used for training this network can be found in table 41.

Figures 76 and 77 show the graphs of the confidence scores for the nine nets tested. The untrained negative control obviously performed very poorly, (figure 76a) showing that the neural network had little confidence in its predictions.



a) b)

Figure 76: Distribution of confidence scores for a) negcontrol (negative control) and b) smoothing291104.net (control).

The neural network that had performed well in the earlier tests, smooth101204.net did very well in the confidence scoring as well (figure 77c). The majority of its predictions were close to 1 or -1, showing that the level of confidence in its predictions was high. This does however mean that when it predicted incorrectly, which it did for only 7.1% of the data, it was also confident that it was correct. However, these incorrect predictions may have been within the lower confidence ranges of $0.5 <\mid C \mid \leq 0.6$ or $0.8 <\mid C \mid \leq 0.7$.

7.4.1 ROC Curves

Once again a ROC plot was calculated for the best of this set of neural networks, smoothing101204.net. Selectivity and Sensitivity were calculated as before, using Equations 17 and 18. The graph of this can be seen in figure 78.

As the graph shows, using a cutoff of 0 to analyze this set of neural networks was once again optimal.

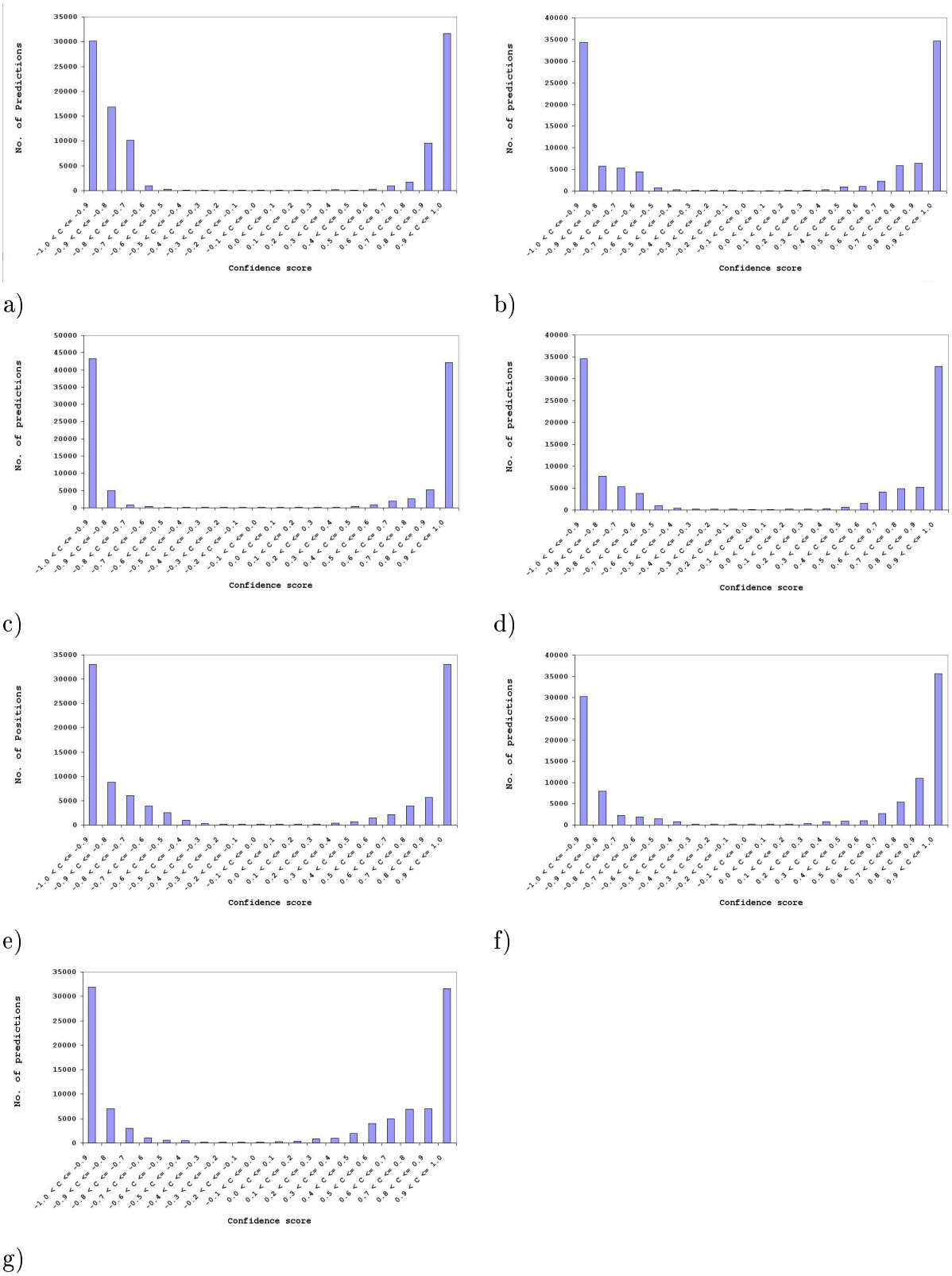


Figure 77: Distribution of confidence scores for the smoothing networks: a) smoothing021204.net b) smoothing031204.net c) smoothing101204.net d) smoothing131204.net e) smoothing161204.net f) smoothing171204.net g) smoothing130105.net.

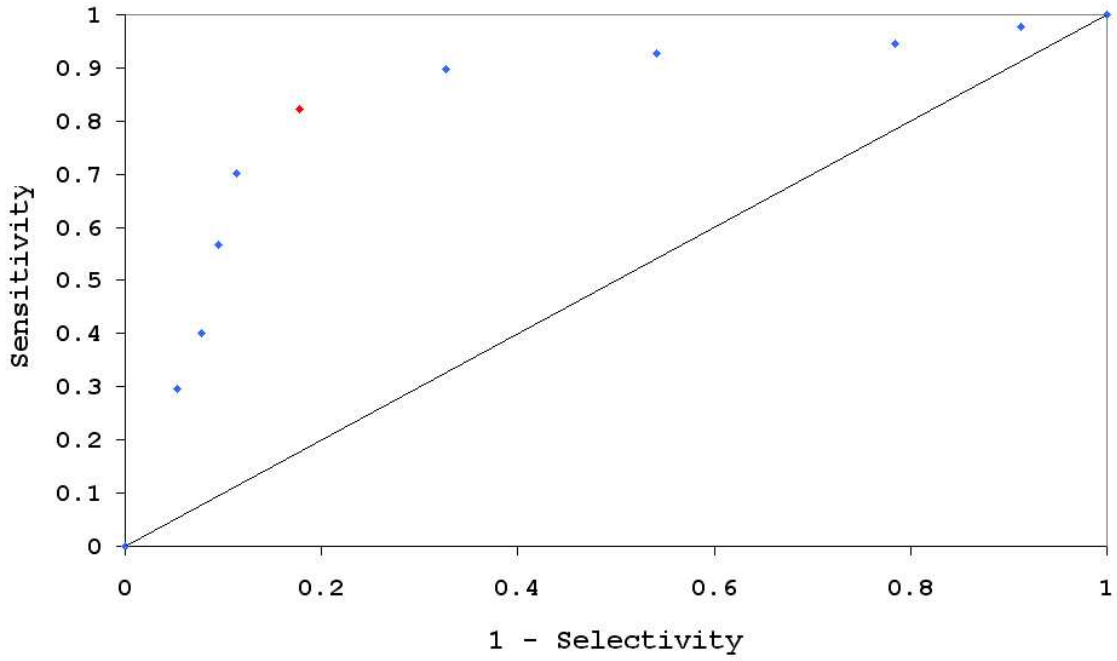


Figure 78: ROC plot for smoothing101204.net. The red point is the value used in the confidence analysis, the black line is the value expected by chance.

7.5 SSMA Prediction Website

A website was created that could be given an aligned pair of sequences as input and predict where the SSMAs were. This website can be found at `http://www.bioinf.org.uk/~danielle/`. As figure 79 shows the website works by being supplied with an existing protein domain alignment and at least one CATH domain identifier. Using this information it can work on either two protein domains of known structure or, more likely, an alignment comprised of a protein domain with known structure and a domain of unknown structure.

The program checks if the CATH identifier actually exists. It also checks if the sequence matches the one to which the CATH identifier is tied; this ensures that the sequence has been entered correctly. If there is a problem at this stage the website informs the user that there has been an error and also whether it relates to the sequence or the identifier. The CATH identifier is also used to find the secondary structure of

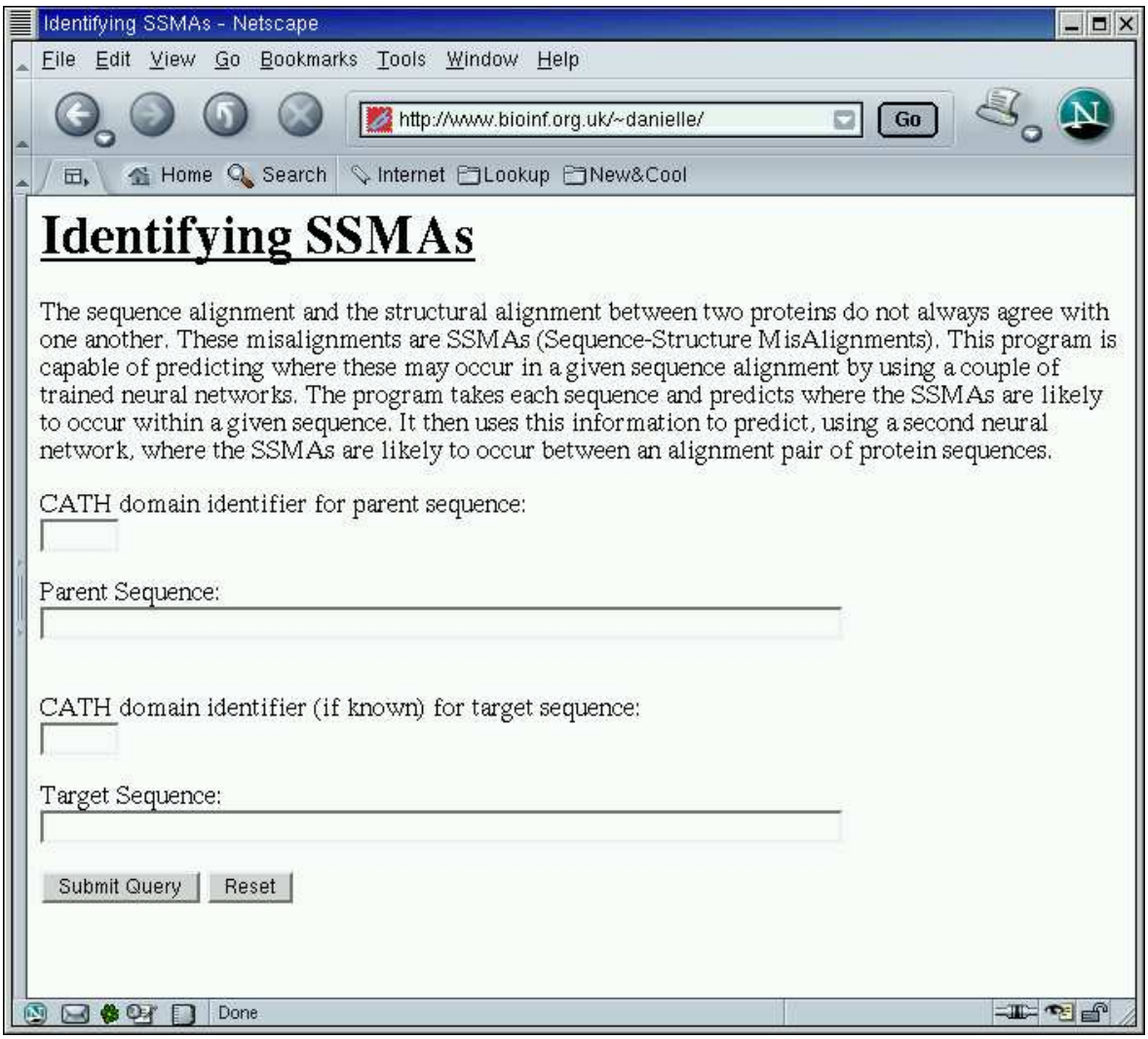


Figure 79: A screenshot of the SSMA prediction website.

the sequence(s) from a pre-made set of secondary structure files.

If only one CATH domain identifier has been included the program will use the structural data of the known domain for both sequences when creating the pattern files needed by the neural networks. It then feeds the pattern file into the neural networks.

The output of the website indicates where SSMAs have been predicted along the length of the original alignment. It presents the original alignment used as input and uses asterisks to show where the program has predicted a SSMA position as shown in figure 80. These data could then be used to adjust the original alignment.

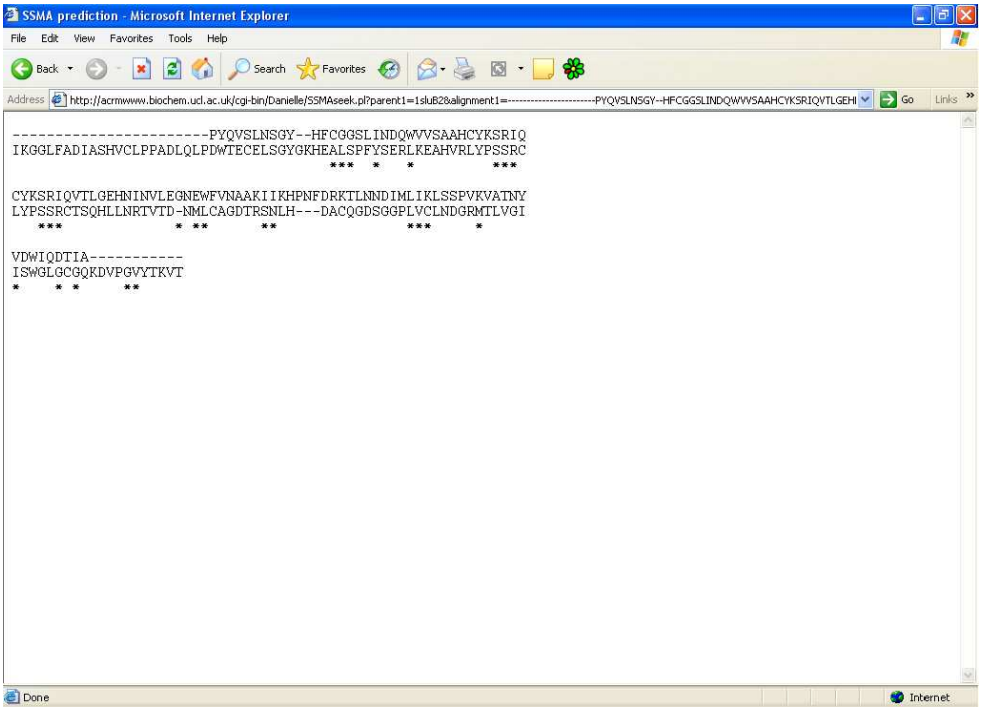


Figure 80: A screenshot of the result of the SSMA prediction website. The alignment of 1sluB2 and 1rtfB1 was used to generate this output page.

The website uses the best performing SSMAtrained031203.net and smoothing101204.net to make its predictions.

Chapter 8

Applying Predictions To Modelling

8.1 Creating alternative alignments

In order to apply the trained neural networks to improving the alignment between protein sequences, the creation of a program capable of creating sensible alternative alignments was necessary. The program needed to examine any given protein domain alignment, predict the presence of SSMAs and then base permutations around its findings.

The first program, alternative.pl, looks at each sequence using the previously trained neural networks in order to identify possible SSMAs. If a predicted SSMA is found, the program uses a number of random variables to alter the alignment. It begins by choosing one sequence of the pairing to alter first. Then, based on the length of the SSMA, it decides upon a number of gaps to introduce into the sequence. The longer the region of SSMA, the more gaps that are introduced. The minimum number of gaps that can be introduced is two, one that marks the beginning of the SSMA region and one that marks the end.

The gap at the beginning of the SSMA region is then inserted, the length of which is also randomly determined. A gap of the same length is then inserted into the second

```
    Sequence 1 AGHILLLGHPHNSCTYGGGLILL
    Sequence 2 AGHLLLLGGGHRSCTT-GGIIIC
    SSMA               *****    ***

Stage 1
           ...LLG--HPHNSCTY...
 Gap inserted at the beginning of the SSMA Sequence 1

Stage 2
           ...LLGGGH--RSCTT...
 Gap of equal length inserted randomly within the SSMA
 region of Sequence 2

Stage 3
           ...LLG--HPHNSCTY...
           ...LLGGGH--RSCTT...
 First insert completed
```

Figure 81: How gaps are introduced in the alternative.pl program.

sequence of the alignment to maintain the overall length of the alignment. Unlike the first sequence the position of this second gap is randomly chosen but is still within the predicted SSMA region. An example of how this works can be seen in figure 81.

After this the same is done with the point that marks the end of the SSMA region. Which sequence it is introduced into, the length and position of the matching sequence are all once again determined by random within the program.

Once the gaps have been introduced into each section where a SSMA region is predicted to be, the program then compares the two sequences of the alignment. It then removes gaps where both sequences have a gap at the same position.

The program then repeats this process for a set number of times in order to create a number of alternative alignments based around the position of SSMAs. An overview of this program can been seen in figure 82.

The first permutation program had a problem in that it could create a number of

alignments that differed only slightly and could generate the same alignment multiple times.

A second program, altalign.pl, was created by Dr. A.C.R. Martin that attempted to create a wider variety of sensible alternative alignments based on the predicted positions of the SSMA regions. The working of this second program can be seen in figure 83. Unlike the first program it works by first smoothing the SSMA prediction data. This means that it will remove a SSMA prediction if it is a single residue predicted as an SSMA. It also joins together any SSMA regions that are separated by two or fewer residue positions. This makes the SSMA predictions less fragmented.

The program then splits the alignment into blocks of either SSMA regions or non-SSMA regions, so that they can be dealt with separately. Each SSMA block is then dealt with individually. Firstly all gaps within the block are removed and the lengths of the remaining residues compared. If the lengths of each alignment are different then a suitable length gap is reintroduced. The reintroduction of the gap is done at each possible position within the block to create the initial variety within the SSMA block. Also, when the lengths differ, a gap from within the block is chosen and moved to a different position at random.

If the two protein sequences within the SSMA block were the same length, then a gap is introduced at random in each sequence, as in the first program of this type. Again this stage is repeated for all the alignments previously created by the program.

All the alignments are kept from each stage so as to provide a wide variety of possible permutations.

In a final clean up stage unlikely or repetitious alternative alignments for each predicted SSMA block are removed. The cleaning up stage occurs in a number of ways. Firstly the program removes any gaps which are matched up against gaps in the second sequence, just as happened in the previous program. The program also goes on to remove unaligned gaps that are adjacent to one another, something that also

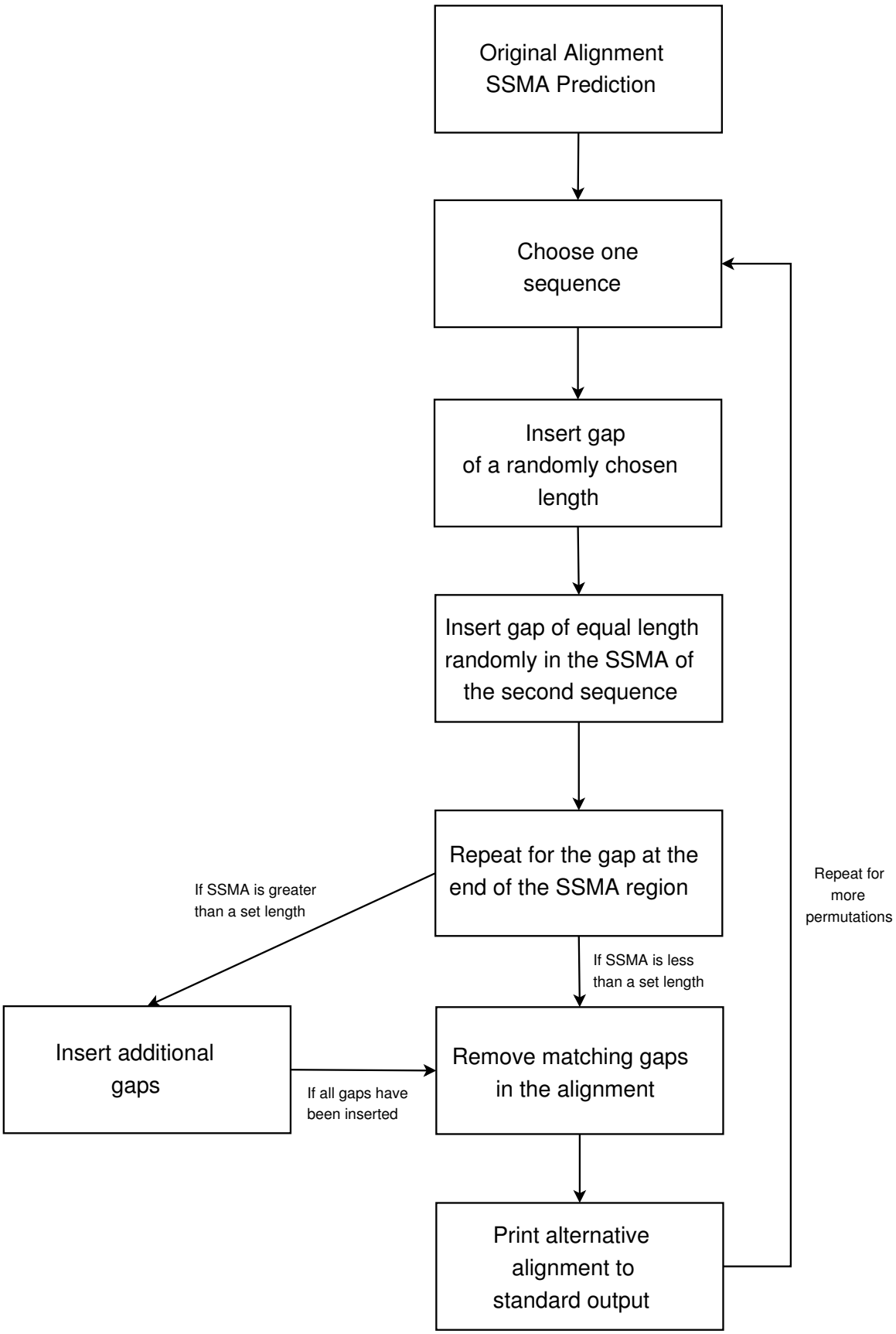


Figure 82: An overview of the alternative.pl program for creating permutations upon alignments.

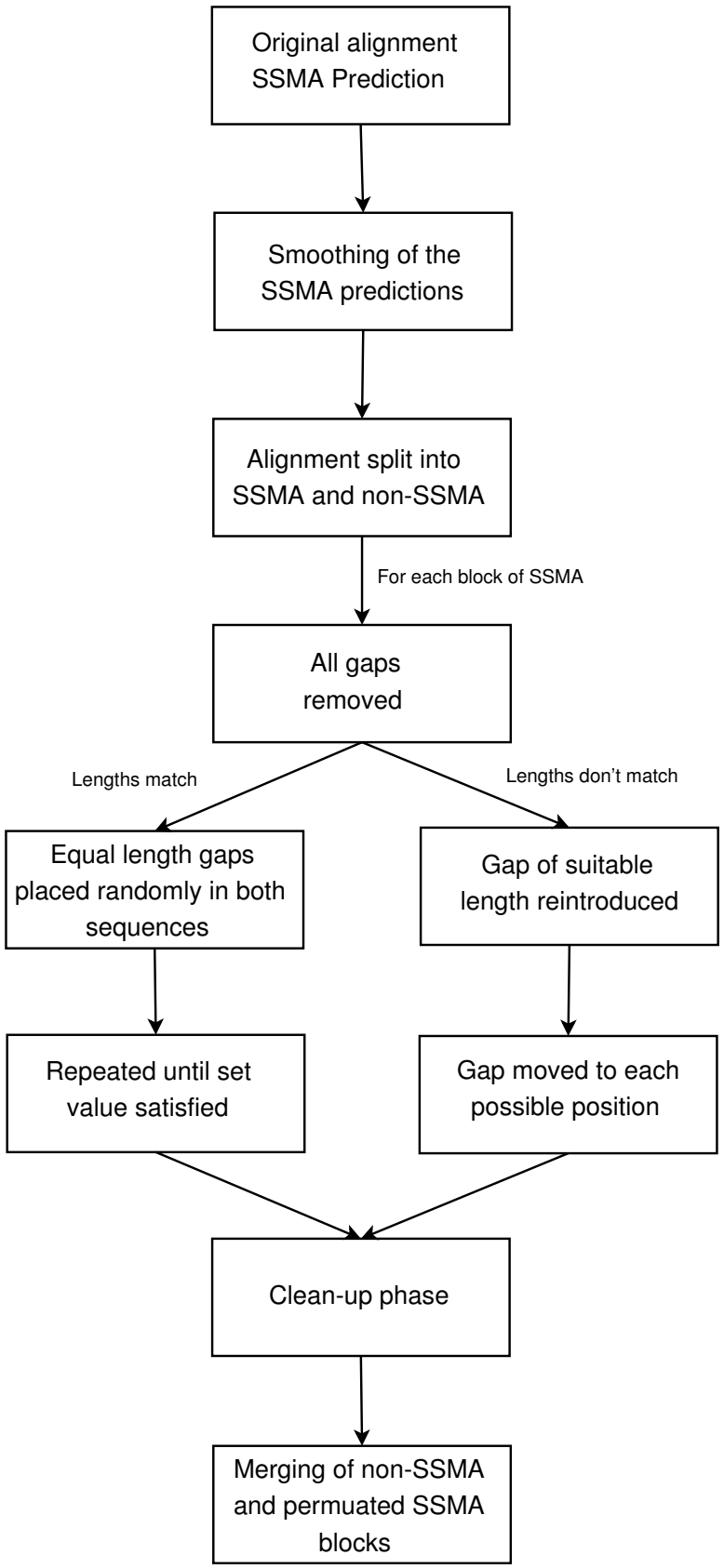


Figure 83: An overview of the altalign.pl program for creating permutations upon alignments.

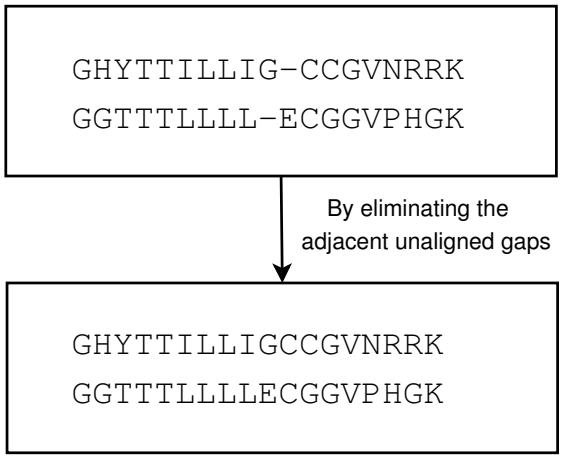


Figure 84: Removing unaligned gaps in altalign.pl.

```
GHYTTILL-I-GCGVNRRK
GGTTTLLLLEICGGVPHGK
```

Figure 85: An 'unlikely' alignment that would be removed by altalign.pl.

occurred in the first program. This can be seen in figure 84.

As well as removing those gaps that are aligned with other gaps and adjacent, though unaligned, gaps, the program also screens the permuted SSMA blocks for 'unlikely' alignments. The removed alignments are those with a lone non-terminal residue. An example of these unlikely permutations can be seen in figure 85.

The program also ensures that the set of permutations given as output are unique by removing any duplicates at this stage.

By this point the program can have created anything between two and a few hundred alignments for each predicted SSMA block. The final stage of the program merges each of the possible SSMA permutations with the original non-SSMA blocks to produce the final selection of alternative alignments. By altering the parameter values within the program it is possible to generate anywhere up to several million permutations of the original alignment. Clearly the number of possible permutations is highly dependent upon the number of predicted SSMAs within each alignment.

However the program does guarantee that there should be a good variation of alternative alignments produced. It also guarantees that the permutations are different from the very first stage.

8.2 Large scale testing

In order to test the effectiveness of the permutation program, large scale testing was done using the large dataset created from the SRep pairs within each H-family of the CATH dataset. Due to the fact that it would take a great deal of time to run the program for each protein alignment of the roughly 20,000 that made up the data set the permutation program was set so that it would produce fewer alignments. The program based on introducing random gaps into each SSMA region was limited to only creating twenty alignments for each pairing. The altalign.pl program had its parameter values set so that it would only produce the minimum number of alignments. Another program was created to 'wrap' around the alignment program and pick up the output for the large scale testing. It also analyzed the alignments and their permutations. Either permutation program could be inserted into the wrapper program. How this worked can be seen in figure 86.

The percentage alignment scores for this large scale testing of the random insert program did seem to back up earlier work that the neural nets are capable of picking up on correct alignments. Figure 87 shows the percentage correct alignment scores for all the original sequence alignments. Figure 88 shows the percentage correct alignment scores for those alignments that the neural networks (SSMAtrained031203.net and smoothing101204.net) predicted as not containing any SSMAs. As the graph clearly shows, in the majority of cases the neural network correctly identified those alignments which did not contain any SSMAs.

After removing from data shown in Figure 87 those alignments predicted not to

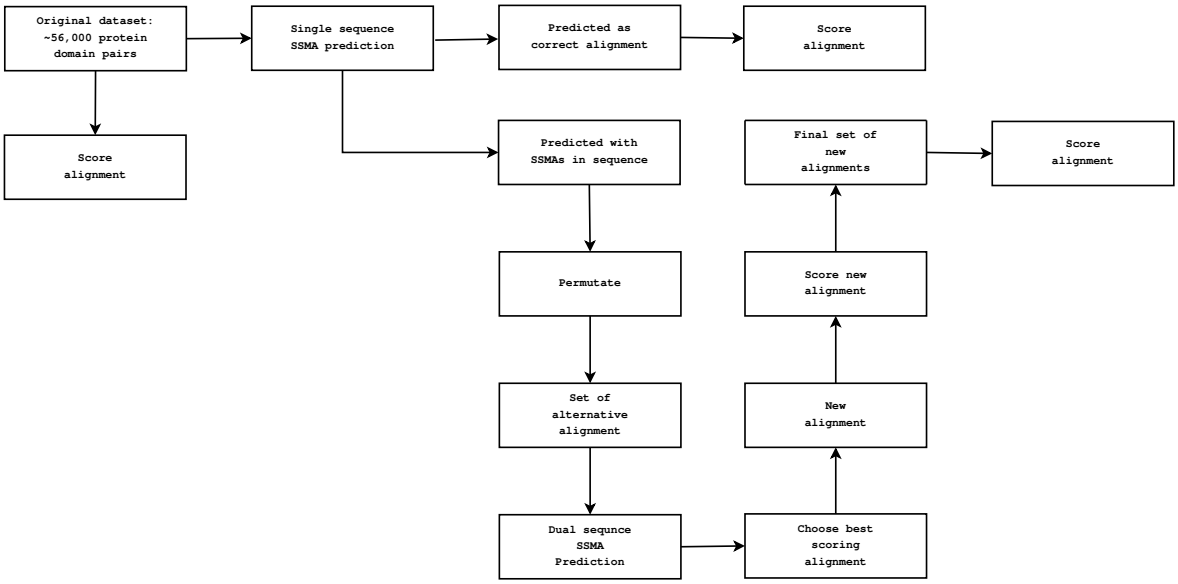


Figure 86: Overall wrapper program used for large-scale testing of the permutation programs.

contain any SSMAs (i.e. those shown in Figure 88), the remaining sequence pairs were permuted using alternative.pl. For each sequence pair, the permuted alignment predicted by the neural network to have the fewest SSMAs was selected as the 'best' alignment. However the graph in figure 89 shows that this 'best' permuted alignment for each pair of sequences performed badly.

Thus, while figure 88 shows that the neural network performs well in identifying the correct alignments, figure 89 shows that alternative.pl did not perform well in generating good alternative alignments. These alignments were not very different from one another, each tending to have similar percentage correct alignment scores. This can be seen in table 42, where the results returned from twenty different permutations of the same alignment are shown. The difference between them is rarely above 0.1%. This suggests that although the program is creating a number of different alignments and the neural net is capable of choosing between them, they are not sufficiently different as to have any large effect upon the overall alignment or, therefore, the structure.

Figure 90 shows the results form using altalign.pl in place of alternative.pl. It can

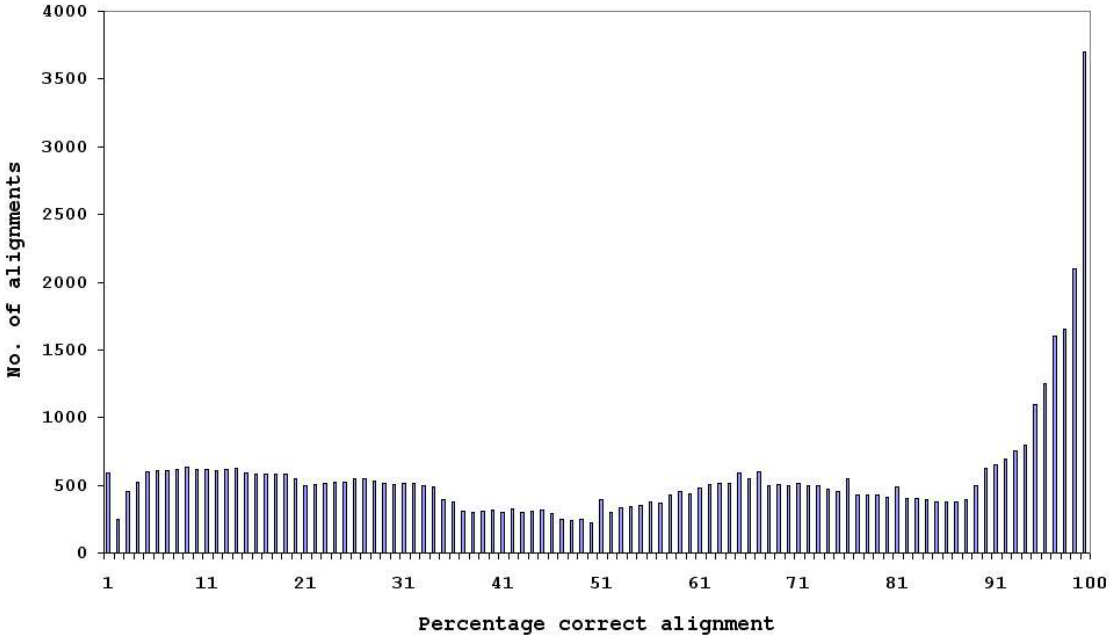


Figure 87: The percentage correct alignment scores for the original sequence alignments of the domain pairs.

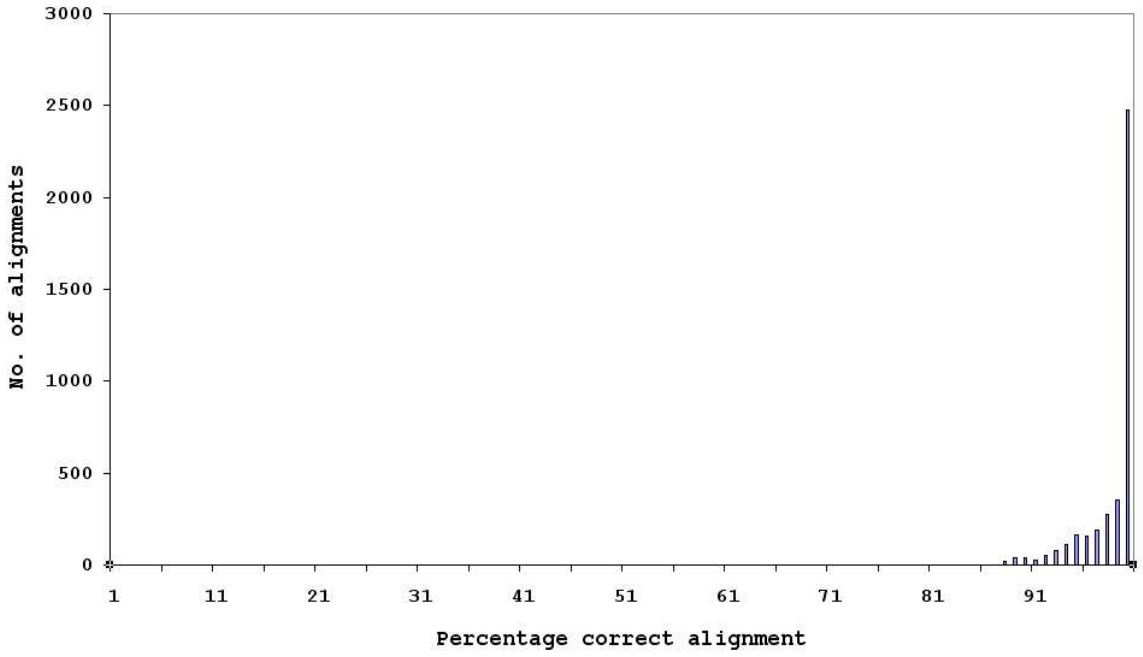


Figure 88: The percentage correct alignment scores for the alignments of domain pairs predicted by the neural network as not containing any SSMAs.

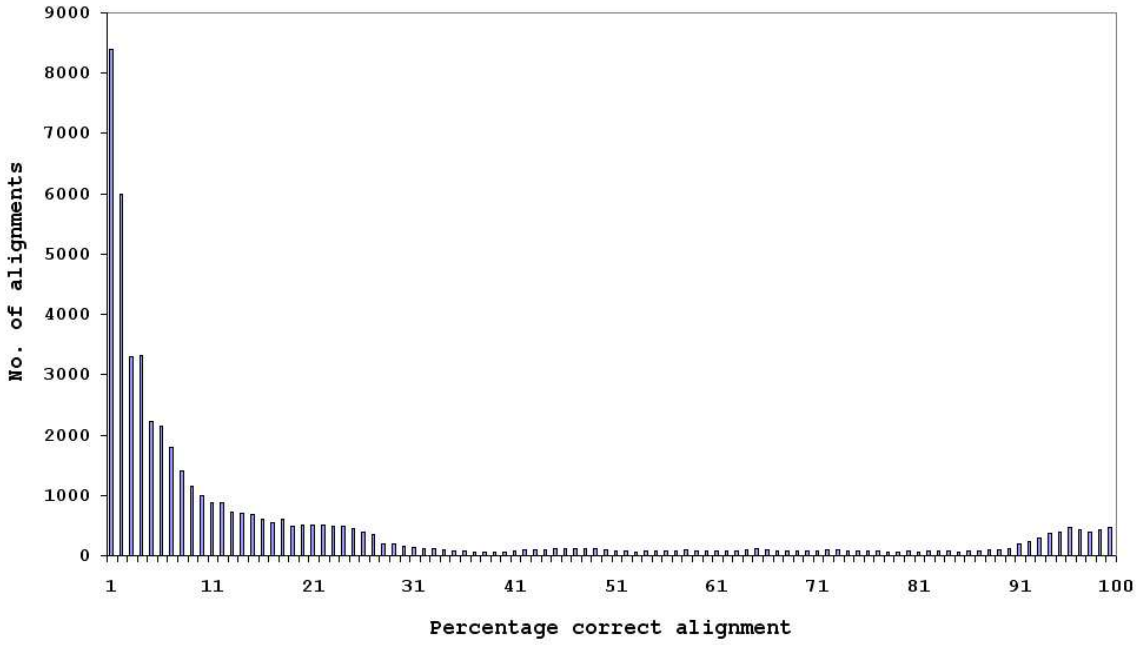


Figure 89: The percentage correct alignment scores for the alignment permutations created by alternative.pl and selected by the neural network as the best alignments. Original alignments between domain pairs predicted as correct were first removed.

Alignment section (1hrnA1 and 1bf5A0)	% correct alignment
PI-FDNII-SQG-VLKEDV-F-SFYYNRDSENS--LGG PVWY--NMLNQ-GLVKER-RFS-FWLNRNVD-EEE-GG	68.1%
PI-FD-NIISQGVLKEDV-F-SFYYNRDSENS--LGG PVWY-NML-NQGLVKER-RFS-FWLNRNVD-EEE-GG	68.4%
PIF--DNIISQGVLKEDVFSFYYNRDSE-NSL-GG PV-WYNMLNQGL-VKERRFSFWLNRNVDE--EEGG	67.7%
PIFDNIISQGVLKEDV-F-SFYYNRDSE-NSL-GG PVWYNMLNQGLVKER-RFS-FWLNRNVDEE--EGG	68.4%
PI-FD-NIISQG-VLKEDVFSFYYNRDSENS--LGG PVWY-NML-NQ-GLVKERRFSFWLNRNVD-EEE-GG	67.9%

Table 42: A selection of alignments created by alternative.pl for part of the SSMA-containing alignment pair 1hrnA1 1bf5A0

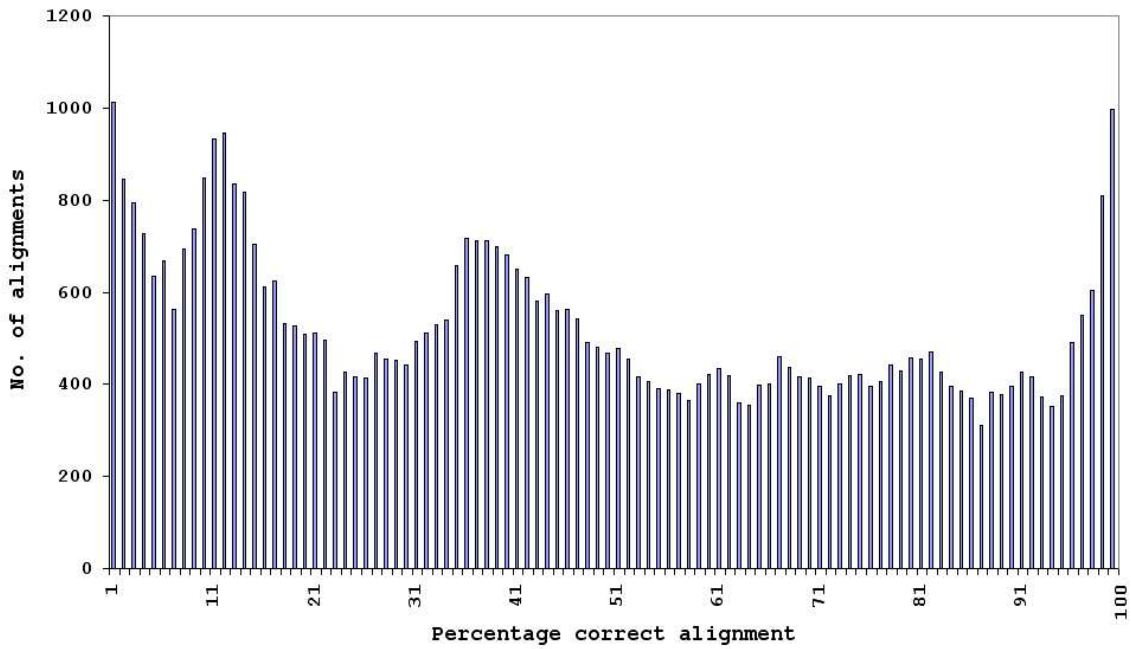


Figure 90: The percentage correct alignment scores for the permutations created by altalign.pl.

be seen that the distribution of alignment quality is very different from that seen in figure 89. As the graph shows the program performed better than alternative.pl but still did not perform well enough to generate the correct alignments with any consistency.

The program shown in figure 86 was run a second time, but this time using the structural alignments as the input. This was done to ensure that the neural net choosing the 'best' of the permuted alignments was functioning correctly. If the neural net was choosing badly between the alignments then it was expected that the percentage correct alignment scores for the permuted sequences would be poor. The results of using alternative.pl to permute the structural alignments can be seen in figure 91. As can be seen the scores for this run are very high, with the majority being above approximately 95% correct. This confirms that the neural network was generally choosing correctly between the alternative alignments that the random insert program was creating.

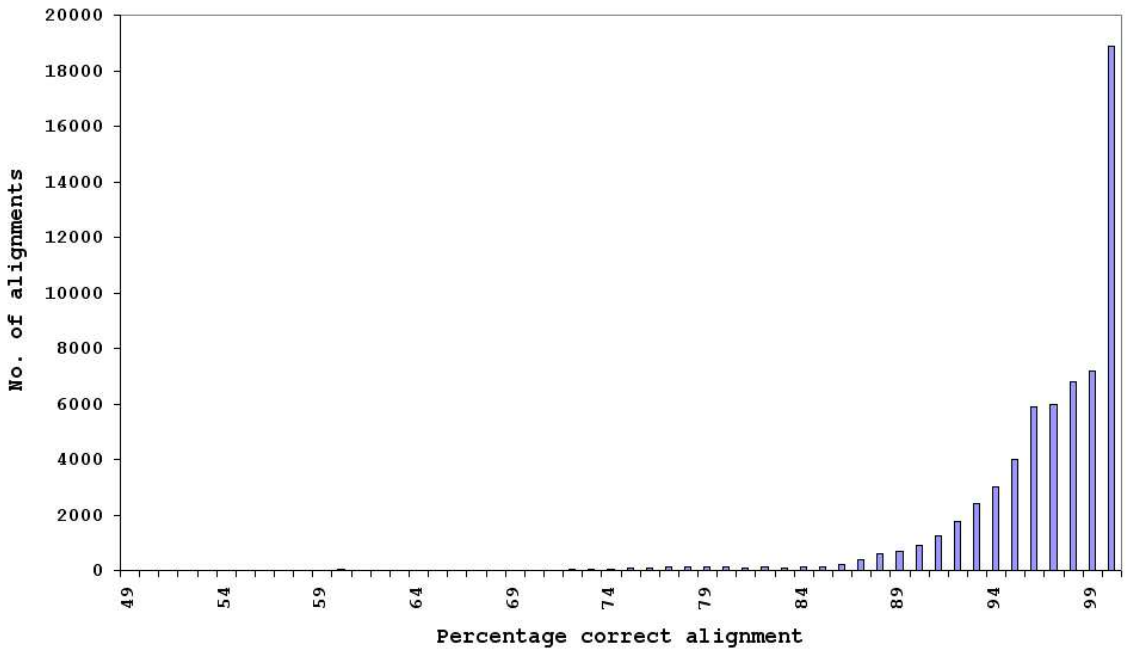


Figure 91: Distribution of percentage correct alignment scores for those alignments selected as 'best' by the neural network from permutations (generated by alternative.pl) of the original structural alignments.

8.3 CASP5 testing

Although both programs had been tested on a large scale data set it was decided to test them on a smaller, but familiar, data set, namely that of the sequences in the CASP5 experiment. The original structural models that had been entered did not take into account the possibility of SSMAs and the alignments were often altered by hand.

The RMSD of a selection of the CASP5 original protein models as well as the alternatives can be seen in table 43. The table also shows the average RMSD for each group of models. As can be seen the alternative models generated by alternative.pl and altalign.pl both improved the mean RMSD of the models with altalign.pl performing the best. This agrees with the earlier analysis that altalign.pl was the better of the two programs for generating alternative alignment for modelling. In table 44 the potential scores, calculated using the RAM potential, can be seen.

The alternative models created by the program alternative.pl are significantly better

Target name	Original model RMSD	alternative.pl model RMSD	altalign.pl model RMSD
T0130	9.966	10.703	8.962
T0130	15.260	10.715	9.970
T0133	12.345	12.540	9.238
T0133	12.224	13.157	8.905
T0137	1.020	2.885	1.057
T0142	3.485	3.458	2.961
T0149	17.276	9.986	6.354
T0149	17.034	10.281	7.192
T0149	17.398	10.307	7.011
T0150	2.700	2.736	2.748
T0150	2.662	2.280	2.104
T0153	5.335	7.164	6.552
T0153	5.373	7.065	6.590
T0154	6.946	6.157	6.138
T0154	6.847	4.538	4.095
T0155	6.031	1.084	2.226
T0160	7.260	3.859	3.304
T0160	6.826	2.732	2.527
T0160	7.163	2.905	2.613
T0167	4.796	7.439	5.720
T0171	9.200	6.477	5.910
T0171	9.656	6.526	5.827
T0179	5.450	2.698	2.469
T0179	5.475	2.599	2.300
T0182	1.416	1.333	1.308
T0184	3.863	3.875	3.884
T0188	2.315	2.220	2.206
Mean RMSD	7.605	5.841	4.821

Table 43: RMSD values of the CASP5 models created by the different methods of protein alignment.

Target name	Actual structure potential, kcal/mol	alternative.pl model potential, kcal/mol	altalign.pl model potential, kcal/mol	original model potential, kcal/mol
T0130	2746.25	133.33	153.87	1211.84
T0130	2746.25	214.06	141.09	862.06
T0133	-6280.06	-1937.67	-1855.41	-2175.55
T0133	-6280.06	-1842.45	-1886.25	-2417.10
T0137	-1036.44	-676.55	-349.01	-610.32
T0142	-2888.51	-1294.62	-1348.22	-1692.31
T0149	-5471.10	658.40	-2274.83	-3782.26
T0149	-5471.10	1509.90	-2305.57	-3003.89
T0149	-5471.10	1545.34	-2263.07	-3571.24
T0150	-2635.34	-2337.05	-2261.63	-2351.07
T0150	-2635.34	-2336.75	-2429.59	-2499.93
T0153	-1788.67	-1404.04	-1202.48	-1318.36
T0153	-1788.67	-1402.87	-1153.52	-1280.00
T0154	-4439.26	-2921.82	-3006.41	-3007.64
T0154	-4439.26	-2970.73	-3527.13	-3821.09
T0155	-1356.91	-1270.23	-1338.77	-1285.21
T0160	-2330.65	-744.34	-1549.54	-1558.48
T0160	-2330.65	-677.81	-1710.60	-1896.58
T0160	-2330.65	-743.96	-1592.47	-1807.12
T0167	-3003.94	-1311.70	-857.20	-992.57
T0171	-3319.92	-1917.65	-2319.04	-2983.54
T0171	-3319.92	-1319.99	-2265.85	-3030.82
T0179	-2878.57	-2122.61	-2271.97	-2493.25
T0179	-2878.57	-2250.15	-2337.11	-2596.10
T0182	-2932.44	-2444.99	-2506.28	-2544.93
T0184	-5907.72	-3540.66	-3494.85	-3451.28
T0188	-2722.91	-1924.56	-2006.32	-2111.63

Table 44: The RAM potential score of the CASP5 models created by the different methods of protein alignment.

than the originals in many cases. Of the 27 cases examined, the RMSD changed by less than 0.2Å in 5 cases. In 19 cases it got better by more than 0.2Å (in some cases by more than 10Å) and in only 3 cases did it get worse by more than 0.2Å. All of these cases, where the RMSD got worse, had an original RMSD of worse than 4.7Å and in no case did the RMSD get worse by more than 1.3Å.

The alternative alignment program, altalign.pl, proved to be better in this testing that alternative.pl. Again this was expected as it had performed much better in the large scale testing. In many cases it outperformed the original CASP5 models. Some of its alignments proved to be a large improvement compared to the original, such as target T0149, while in other cases it was only by a slim margin, such as target T0154.

8.4 Alternative alignment website

With two permutation programs investigated and the second program proving the better, it was hoped that it would be possible to create a website that would make it available to the public. However this did not prove to be the case as both programs took too long to run to make this feasible.

Chapter 9

Conclusions

9.1 Empirical Potential

9.1.1 Large scale analysis: Testing the RAM potential

The RAM empirical potential appears to be quite effective at picking out the more accurate of two protein models when alignments are highly different. 89% of the time in the large scale analysis, it selected the protein produced by a structural alignment over one produced by a sequence alignment. The structural alignments were produced by overlaying the structure of the parent and target and would be expected to give a better model than a sequence alignment.

However there were some cases where the potentials program and the RMSD values favoured the sequence aligned model. This happened when there was a single residue shift difference between the sequence alignment and the structural alignment.

The RAM potential energies of the modelled protein were normally distributed. The mean energy was around -100 kcal/mol for the models produced from those proteins aligned by sequence and around -1000 kcal/mol for the structural alignment models.

9.1.2 Large scale analysis: Testing the RAM potential with varying alignments

Varying the structural alignment by randomly adding insertions produced a multitude of similar models for the RAM potential to choose between. While there was a general trend of lower potential energy values relating to lower RMSD values, this was not as clear as when comparing the structurally aligned and sequence aligned models. This while the RAM potential was quite effective at selecting between very different models, it was less effective at selecting between a set of approximately correct models. The data set was a lot smaller than that used for analysis of widely different structures and it is possible that if more models were created the trend would become clearer.

9.1.3 Large scale analysis: RAM potential Conclusions

The RAM potential was capable of distinguishing between larger differences in structures but had difficulty with more subtle variation. This is similar to the result obtained by Pettitt *et al.* (2005) when they tested MODCHECK against three other model quality assessment programs (MQAPs). MODCHECK is based on classic threading potentials and uses a set of mean force pairwise potentials (Pettitt *et al.*, 2005; Hendlich *et al.*, 1990).

They concluded that MODCHECK was able to improve the top model quality selection ability for structure prediction servers that did not already attempt to incorporate information from the 3D structure of the template protein and that it was consistent in improving model rankings in these cases (Pettitt *et al.*, 2005). However MODCHECK also was shown to have a high level of false positives, as were the other MQAPS that it was tested against (Pettitt *et al.*, 2005).

Both MODCHECK and our own potentials program used potentials of mean force to try to choose between protein models. Since these types of potentials can choose

between protein models if the differences between them are great they could prove to be useful in quickly choosing a number of likely structures out of a large number of models.

9.1.4 CASP5

The CASP5 competition also showed that the RAM potential did not perform well in selecting between slightly different alignments and loop conformations in the blind test. It only picked the correct answer 25% of the time in those cases where the score could be calculated.

The models were submitted for the experiment asking that they were assessed only over those sections of the structure where we felt confident in our predictions. In this way the assessed RMS for the two models of T0184 was around 3.8 Å. However, if the two models had been assessed over their entire length they would both have had RMSDs of over 50 Å. At the same time they both would have had total potential energy values of less than -2000 kcal/mol whereas the correct structure had a potential energy of -5907.72 kcal/mol. When looking at the entire models using Rasmol it could quite clearly been seen that both models had long unfolded tails, the source of their high RMSD values.

Overall the group fared well in the CASP5 competition although the RAM potential did not contribute to improving the results. In some cases the protein models that were submitted were within 1.5 Å RMSD of the true structure.

9.2 MLSAs

Misleading local sequence alignments are an extreme case of misalignment and have a number of causes. These causes are:

- Occurring in terminal region / Absence of force constraints

- Presence of secondary structure
- Accessibility of hydrophobic residues
- Charge interactions

These causes were discovered from investigating the nine most extreme cases of MLSAs found by an automated protocol. Table 20 shows the nine pairs of aligned protein domains which contained the extreme MLSAs and the reasons that they occurred. It is possible that there are other causes of these misalignments that might be found through further study of these and less severe MLSAs. Other reasons for the occurrence of MLSAs may include the packing of residues, hydrogen bonds and charge-charge interactions. Study of the less severe MLSAs may bring to light even more reasons for why and where they occur.

Visual analysis was used first to look for obvious clues. This lead to the realization that one of the MLSAs (1ak200 1akeA0) occurred within a hinge region (figure 92). As such this was not a true MLSA: the large structural differences confused the structural alignment leading to arbitrary assignment of equivalent residues.

Six of the MLSAs occurred in the terminal regions of at least one protein domain of the pairings. In these terminal regions the force constraints that exist elsewhere in the protein structure do not exist to the same extent. Without the force constraints when an indel occurs the residues cannot be forced into the expected positions.

The presence of secondary structure also appeared to play a role in the appearance of MLSAs. Four of the MLSAs occurred within areas of secondary structure, two within beta-strands, two within alpha-helices. The true position of the indel has moved to the adjacent loop region to maintain the integrity of the secondary structure.

Hydrophobic residues need to be kept away from aqueous environment in order to maintain the stability of the protein's structure. In the same way proteins tend to

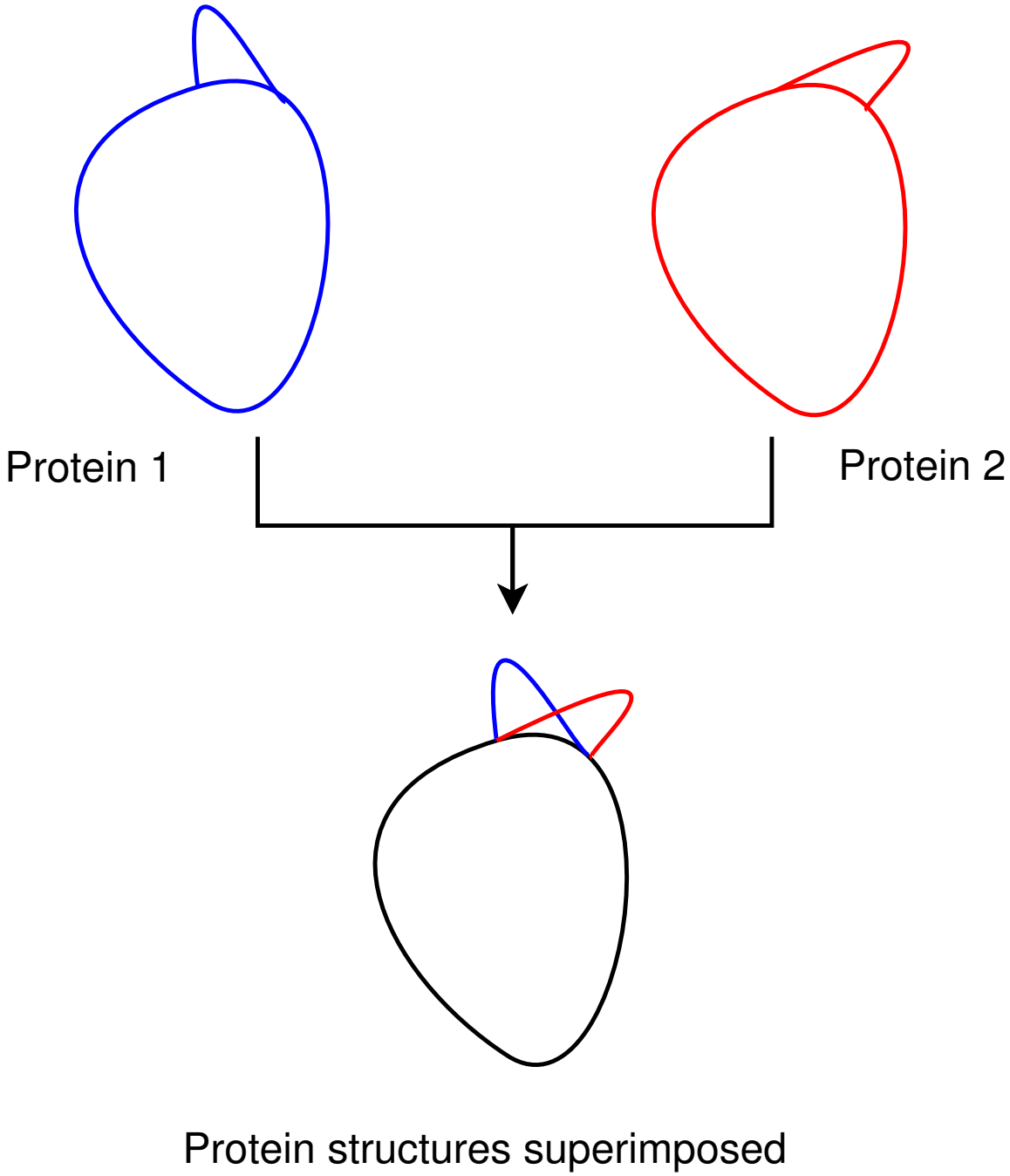


Figure 92: An example of how a hinge in a protein domain can cause a difference in aligned structures. When the two structures are superimposed upon one another the two structures do not match over the area over the hinge. The structural alignment between the residues in the hinge is thus arbitrary. If both had been experimentally solved with the hinge in the same configuration there would be no difference in structure and therefore no MLSA.

expose their hydrophilic residues. Examples were seen where the structural alignment increased the burial of hydrophobic residues compared with the sequence alignment.

There is no single reason why MLSAs occur. Rather they are brought about by a combination of causes, not all of which are present in all cases of MLSA. It is most likely that other reasons for this sort of misalignment would be brought to light by further study on the less extreme cases.

9.3 SSMAs

Sequence-structure misalignments (SSMAs) are not as extreme as MLSAs. They are simply areas where the sequence and structural alignments do not agree and are therefore a great deal more common than MLSAs. Within the data set of all the Srep pairings within each homologous superfamily there were a total of 28,208,763 9-residue windows. Of that total there were 226,812 that were identified as SSMAs. This large data set meant that neural networks could be trained to predict where these SSMAs might occur in both single sequences and in aligned protein domains.

9.3.1 Studying SSMAs

A number of programs (which can be found on the accompanying CD) were used to study the SSMAs in order to find why and where they occurred.

Examination of SSMAs showed that they could be of any length, though smaller SSMAs were more likely to occur than longer ones. The number of SSMAs within a protein domain was also studied. A domain was more likely to have two or fewer SSMAs than it was to have a large number of them. The higher the percentage of residues in a SSMA the less likely it was to occur (figure 54). SSMAs were more likely to consist of a small percentage of residues (figure 55).

The most significant conclusion of this study was that the SSMAs occurred more

often within certain types of secondary structure. Approximately half of all SSMAs began in areas that were in beta-strands. A little under a quarter of them began at residues that were in turns with alpha-helices being the third most likely conformation for the starting residue of a SSMA. A similar pattern was seen when looking at the final residues of SSMAs. This bias towards beginning and ending in certain types of secondary structure lead to the secondary structure of the sequences being included in the input to neural networks designed to predict the presence of SSMAs.

9.3.2 Predicting SSMAs

The SNNS program was used to create a number of neural networks. The neural networks can be divided into two distinct groups; the single sequence SSMA prediction networks and the dual sequence SSMA prediction networks. The first group predicted where a single sequence was likely to be misaligned when aligned with a homologue of $\leq 35\%$ sequence identity. The second group of networks predicted where an aligned pair of sequences was correctly and incorrectly aligned.

The training of the single sequence neural nets produced some surprisingly good networks. Two in particular, SSMAtrained031203.net and SSMAtrained081203.net, had good rates of prediction for both SSMAs and transitions, along with very good Matthews' correlation coefficient values and confidence scores. A number of different training and neural network setups were used but none of them gave the same combination of good prediction rates, MCCs and confidence values.

These networks were trained with 9:1 ratios of non-transitions to transitions. An additional set of networks were trained using a 1:1 ratio to give the training net equal exposure to both. However this meant reducing the total size of the training pattern file to around 20,000 patterns. The prediction rates for both SSMAs and transitions remained high using this smaller training set. However the Matthews' correlation

coefficients and confidence scores for the SSMAs indicated that the networks were not performing as well.

Another series of networks, predicting the position of SSMA regions using two sequences at once did not prove to work as well as the single sequence networks. Although they performed better than random they did not achieve the same levels of prediction, MCC or confidence as the single sequence networks.

A third set of neural networks was then trained that built on the success of the single sequence nets. By incorporating the predictions of the single sequence neural networks along with sequence and structural information these networks proved themselves very capable of predicting correctly and incorrectly aligned areas when given a sequence alignment. Although the MCC values of these trained nets were not at the same level as the initial single sequence SSMA prediction networks they still overall outperformed the second series of networks.

One of this final series of networks, smoothing101204.net and the single sequence SSMA prediction network SSMAtrained031203.net were used to create a website that was capable of predicting where SSMAs would occur. This website can be found at `http://www.bioinf.org.uk/~danielle/` and given two sequences and at least one CATH identifier will return a prediction of where any SSMAs may occur.

9.4 Permuted Alignments

With the neural networks capable of predicting where SSMAs are likely to occur the next step of improving the alignment phase of comparative modelling is to create sensible alternatives around the predicted areas of sequence-structure misalignment.

Two programs were written to create alternative alignments. The first of these programs was alternative.pl. This program worked by inserting a number of gaps into each SSMA region based on the length of that region. The smallest number of gaps

used was two, one to mark the beginning and one the end. The numbers of gaps introduced were based on the earlier study of SSMAs.

When tested on the CASP5 targets this program showed a slight improvement for some of the target sequences. However when tested upon a large number of alignments it proved to take too long to run and also showed little or no improvement.

The second program created was altalign.pl, created by Dr. A.C.R. Martin. This program smoothed the SSMA prediction data and divided an alignment up into sections of SSMA and non-SSMA. It then treated each of the SSMA blocks individually to create a number of alternative blocks which could then be fitted back together in a number of ways to produce a large number of sensible and different alignments which could then be screened using the neural network smoothing101204.net. This screening compared the number of predicted SSMAs remaining in the alternative alignments and returned the one with the smallest value as the most likely. This screening process was the same in both programs.

The alternative alignments produced by altalign.pl proved better than those of alternative.pl when used on the CASP5 targets. The average RMSD over all CASP 5 target predictions improved from 5.841Å to 4.821Å. It also performed better in the large scale testing. However the results indicated that much more work needed to be done on alternative alignment creation.

9.5 Discussion

This study has looked into the possibility of using potentials of mean force, namely the RAM potential, in order to distinguish between alternative comparative models. It has been shown that this potential is useful when there are large differences between the models. However the potential does not prove as successful when the differences between the models are more subtle. It would seem that this way of choosing between

models needs to be combined with another method in order to be most successful.

The main source of error for a modelled protein structure is the alignment. Comparing the sequence alignment and the structural alignment highlights the areas where alignment fails. Major errors like severe MLSAs are rare but difficult to identify when there is only a sequence alignment present. MLSAs have a very obvious sequence alignment that proves to be incorrect when compared with the structural alignment. These misalignments have several different causes and it is likely that further study of less extreme MLSAs will highlight other contributing causes.

While MLSAs are severe cases of misalignment they are uncommon compared with SSMAs. These misalignments appear in many alignment pairings and are areas where structural and sequence alignments disagree. They seemed to favour starting and ending in beta-strands, followed by turns, then alpha-helices. As secondary structure seems to play an important role in their occurrence, this information was used in neural networks to predict their occurrence.

Neural networks have proved very successful in identifying the SSMAs in both single sequences and aligned protein pairs. Further work on these networks may further improve even upon the levels of prediction and certainty so far achieved.

With the SSMAs capable of being predicted, then improving on the initial alignments is a necessary next step to improving comparative modelling. It was hoped that creating varied and sensible alternative alignments that can be screened quickly would produce useful results. In this study, permutations of the alignments were made by randomly inserting gaps into the predicted SSMA regions. This did improve the protein models significantly, but took a great deal of time and left considerable room for improvement. A better way of improving the alternative alignments may be to use a genetic algorithm (Holland, 1975) instead.

Genetic algorithms simulate the process of biological evolution in computers (Lin *et al.*, 2005) and is a powerful tool for combinatorial problems of model optimization

and feature selection when the 'model space' is complex and has many local optima (Beiko and Charlebois, 2005). This type of algorithm has already been used successfully improve the alignments between two or more genomic sequences. The program GenAlignRefine (Wang and Lefkowitz, 2005) uses a genetic algorithm to improve on genomic alignments that are globally correct but fail to perform on a local level. It is possible that this kind of approach to protein alignment could improve sequence alignments.

A genetic algorithm begins with a pool of possible solutions. These possible solutions are scored according to their 'fitness' and mated with one another. This effectively produces a second generation of possible solutions which will again be scored as to their fitness. The hope would be that this second generation is closer to the actual solution. This generation are then mated and their offspring scored. Eventually an optimum solution is hopefully found, in this case the correct alignment.

If a number of possible alignments were used as the original population then it would be hoped that the mating of the alignments would eventually find the optimum alignment.

Another way of generating a number of alternative alignments, once the SSMA positions were predicted, would be to derive a number of sub-optimal alignments. By generating sub-optimal alignments based around the position of the predicted SSMAs, an alignment close to that of the structural alignment might be achieved.

9.6 Summary

The initial aim of this research was to answer two questions:

1. Can we identify correctly modelled proteins through potentials?

2. Can we identify correctly aligned sequences directly?

In answer to the first question; yes but only in a limited fashion. It was shown that the RAM potential is capable of distinguishing between correctly aligned models and incorrectly aligned ones when the difference between those models is large. However if the differences between the models are subtle it becomes unable to distinguish between them with any reliability.

In answer to the second question we posed, we can also identify where sequences are misaligned. After studying the severe cases of misalignment, the MLSAs, several possible contributing causes were identified. When the more common cases of misalignment were examined, the SSMAs, it was noted that they tended to begin and end within certain types of secondary structure, most notably beta-strand. This allowed us to incorporate secondary structure into the training and testing of neural networks which proved capable of predicting where SSMAs occur in single and aligned sequences.

Creating programs that were able to permute an alignment based around the predicted occurrence of SSMAs had a reasonable level of success. However, this method took far too long to run to allow creation of a viable web server. Using a genetic algorithm to create the correct alignment may be a way of improving on the results achieved. If the error gained through incorrect alignments could be eliminated from comparative modelling it may help to close the gap between the number of known sequences and the number of known protein structures.

While improving the generation of alternative alignments to be scored using the methods developed here remains the biggest challenge, the results were extremely encouraging. Applying the methods to the CASP5 models shown in Table 43 improved the average RMSD from 7.6Å to 4.8Å and in some cases, the RMSD was improved by more than 10Å. It is therefore clear that the methodology can contribute significant improvements to automated protocols for comparative modelling.

Appendix A

Neural Network Training

A.1 Back-propagation of errors

The most common learning method for neural networks is standard back-propagation of errors. Back-propagation is so called because the correction of its weights starts in the last layer and then continues backwards as can be seen in figure 93. The equations which correct these weights are (Zupan and Gasteiger, 1993):

Total weight change

$$\Delta w_{ji}^{l} = \eta \delta_j^l out_i^{l-1} + \mu \Delta w_{ji}^{l(previous)} \tag{19}$$

$$\text{error in the last layer } \delta_j^{last} = (y_j = out_j^{last}) out_j^{last} (1 = out_j^{last}) \tag{20}$$

$$\text{error in the hidden layer } \delta_j^l = (\sum_{k=1} r \delta_k^{l+1} w_{kj}^{l+1}) out_j^l (1 - out_j^l) \tag{21}$$

Required weight change

$$\text{in the last layer } \Delta w_{ji}^{last} = \eta (y_j - out_j^{last}) out_j^{last} (1 - out_j^{last}) out_j^{last-1} \tag{22}$$

$$\text{in the hidden layer } \Delta w_{ji}^{l} = \eta(\sum_{k=1}^{r} \delta_{k}^{l+1} w_{kj}^{l+1}) out_{j}^{l}(1 - out_{j}^{l}) out_{i}^{l-1} \tag{23}$$

where w is the weight, l is the index of the current layer, j identifies the current neuron and i is the index of the input source. In these equation δ_j^l is the error introduced by the corresponding neuron (Zupan and Gasteiger, 1993), *last* is the output layer, the constant η is the learning rate and μ is the momentum constant. The learning rate constant can prevent sudden changes in the direction in which corrections are made, while the momentum constant prevents the network from getting caught in shallow local minima (Zupan and Gasteiger, 1993).

A.2 Resilient back-propagation

The training method most used in this research was resilient back-propagation (Rprop) (Reidmiller and Braun, 1993). The resilient back-propagation algorithm is generally faster than traditional back-propagation (Liu *et al.*, 2002). It uses individual dynamically tuned learning rates during the training of the neural network. In a study by Schiffmann *et al.* (1993), Rprop was reported to outperform all other learning algorithms in both speed and quality.It is also one of the best learning methods in terms of accuracy and robustness with respect to its parameters (Anastasiadis *et al.*, 2003).

The basic principle of Rprop is to eliminate the harmful influence of the size of the partial derivative on the weight step (Anastasiadis *et al.*, 2003; Zell *et al.*, 1995). The equation which calculates the weight-specific $\Delta_{ij}^{(t)}$ value that the size of the weight change is determined by (Zell *et al.*, 1995):

$$\Delta_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & \text{if} \frac{\theta E^{(t)}}{\theta w_{ij}} > 0 \\ \Delta_{ij}^{(t)} & \text{if} \frac{\theta E^{(t)}}{\theta w_{ij}} < 0 \end{cases} \tag{24}$$

where $\frac{\theta E^{(t)}}{\theta w_{ij}}$ denotes the summed gradient information over all patterns in the pattern

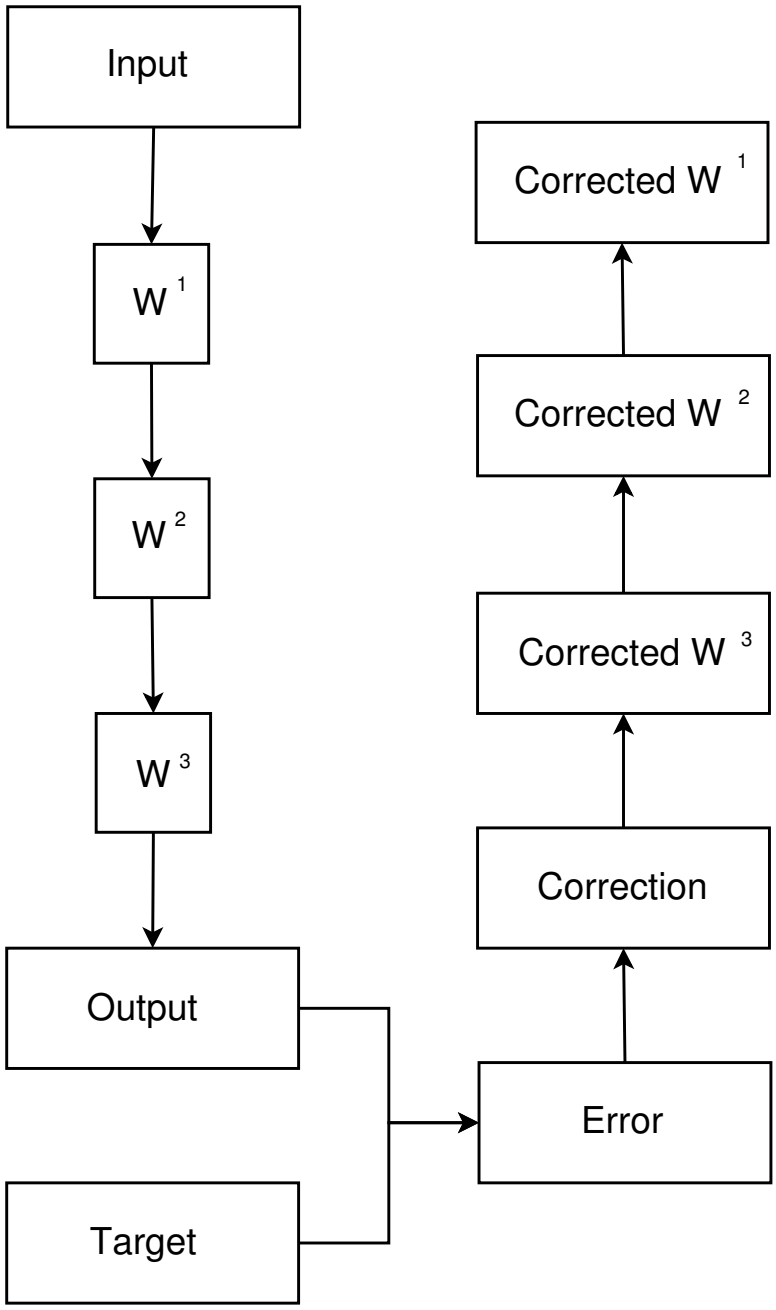


Figure 93: Weight (W) correction that occurs during back propagation training method.

file.

The second step of the learning method is to determine the new update-values $\Delta_{ij}^{(t)}$ (Zell *et al.*, 1995). This is calculated by:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ \times \Delta_{ij}^{(t-1)} & \text{if} \frac{\theta E^{(t-1)}}{\theta w_{ij}} \times \frac{\theta E^{(t)}}{\theta w_{ij}} > 0 \\ \eta^- \times \Delta_{ij}^{(t-1)} & \text{if} \frac{\theta E^{(t-1)}}{\theta w_{ij}} \times \frac{\theta E^{(t)}}{\theta w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)} & \text{otherwise} \end{cases} \tag{25}$$

where $\eta^- = 0.5$ and $\eta^+ 1.2$.

Bibliography

Abagyan RA, Batalov S 1997. Do aligned sequences share the same fold? *J. Mol. Biol.*, 273:355–368.

Altschul SF, Gish W 1996. Local alignment statistics. *Meth. Enzymol.*, 266:460–480.

Altschul SF, Gish W, Miller W, Meyers EW, Lipman DJ 1990. Basic Local Alignment Search Tool. *J. Mol. Biol.*, 215:403–410.

Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DP 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nuc. Ac. Res.*, 25:3389–3402.

Anastasiadis AD, Magoulas GD, Vrahatis MN, 2003. An efficient improvement of the Rprop algorithm. In *Proceedings of the 1st International Workshop on Artificial Neural Networks in Pattern Recognition, Florence, Italy*, pages 197–201. `http://www.dcs.bbk.ac.uk/~aris/Camera_ready_ANNPR03_final.pdf`.

Anfinsen CB 1973. Principles that govern the folding of protein chains. *Science*, 181:223–230.

Aszódi A, Gradwell MJ, Taylor WR 1995. Global fold determination from a small number of distance restraints. *J. Mol. Biol.*, 251:308–326.

Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N,

Yeh LSL 2005. The Universal Protein Resource (UniProt). *Nuc. Ac. Res.*, 33:D154–D159.

Bairoch A, Boeckmann B, Ferro S, Gasteiger E 2004. Swiss-Prot: Juggling between evolution and stability. *Bioinformatics*, 5:39–55.

Baker D, Šali A 2001. Protein structure prediction and structural genomics. *Science*, 294:93–96.

Baldi P, Brunak S, Chauvin Y, Andersen CA, Nielsen H 2000. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16:412–424.

Bartlett GJ, Todd AE, Thornton JM, 2003. Inferring protein function from structure. In Bourne PE, Weissig H (eds.), *Structural Bioinformatics*. Wiley.

Batchelor AH, Piper DE, de la Brousse FC, McKnight SL, Wolberger C 1998. The structure of GABPalpha/beta: an ETS domain- ankyrin repeat heterodimer bound to DNA. *Science*, 279:1037–1041.

Bates PA, Sternberg MJ 1999. Model building by comparison at CASP3: using expert knowledge and computer automation. *Proteins: Struct., Funct., Genet.*, 37:47–54.

Beiko RG, Charlebois RL 2005. GANN: Genetic algorithm neural networks for the detection of conserved combinations of features in DNA. *BMC Bioinformatics*, 6:36–36.

Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Rapp BA, Wheeler DL 2002. GenBank. *Nuc. Ac. Res.*, 30:17–20.

Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE 2000. The Protein Data Bank. *Nuc. Ac. Res.*, 28:235–242.

Bernstein FC, Koetzle TF, Williams GJ, Meyer EE, Brice MD, Rodgers JR, Kennard O, Shimanouchi T, Tasumi M 1977. The Protein Data Bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112:535–542.

Blake JD, Cohen FE 2001. Pairwise sequence alignment below the twilight zone. *J. Mol. Biol.*, 307:721–735.

Blundell TL, Sibanda BL, Sternberg MJ, Thornton JM 1987. Knowledge-based prediction of protein structures and the design of novel molecules. *Nature (London)*, 326:347–352.

Boeckmann B, Bairoch A, Apweiler R, Blatter M, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I, Pilbout S, Schneider M 2003. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nuc. Ac. Res.*, 31:365–370.

Bower MJ, Cohen FE, Dunbrack, R. L. J 1997. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *J. Mol. Biol.*, 267:1268–1282.

Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M 1983. A program for marcromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.*, 4:187–217.

Browne WJ, North ACT, Phillips DC, Brew K, Vanaman TC, Hill RC 1969. A possible three-dimensional structure of bovine lactalbumin based on that of hen's egg white lysosyme. *J. Mol. Biol.*, 42:65–86.

Bruccoleri RE, Karplus M 1987. Prediction of the folding of short polypeptide segments by uniform conformational sampling. *Biopolymers*, 26:137–168.

Burke DF, Deane CM 2001. Improved protein loop prediction from sequence alone. *Protein Eng.*, 14:473–478.

Burley SK 2000. An overview of structural genomics. *Nature: Struct. Biol.*, 7:932–934.

Cantor CR, Little DP 1998. Massive attack on high-throughput biology. *Nature Genetics*, 20:5–6.

Casari G, Sippl MJ 1992. Structure-derived hydrophobic potential. Hydrophobic potential derived from X-ray structures of globular proteins is able to identify native folds. *J. Mol. Biol.*, 224:725–732.

Chandrasekaran R, Ramachandran GN 1970. Studies on the conformation of amino acids. XI. Analysis of the observed side group conformation in proteins. *Int. J. Protein Res*, 2:223–233.

Chiu TL, Goldstein RA 1998. Optimizing potentials for the inverse protein folding problem. *Protein Eng.*, 11:749–752.

Chothia C, Lesk AM 1986. The relationship between the divergence of sequence and structure in proteins. *EMBO J.*, 5:823–826.

Contreras-Moreira B, Fitzjohn PW, Bates PA 2003. *In silico* protein recombination: enhancing template and sequence alignment selection for comparative protein modelling. *J. Mol. Biol.*, 328:593–608.

Cornette JL, Cease KB, Margalit H, Spouge JL, Berzofsky JA, DeLisi C 1987. Hydrophobicity scales and computational techniques for detecting amphipathic structures in proteins. *J. Mol. Biol.*, 196:659–685.

Crippen GM 1991. Prediction of protein folding from amino acid sequence over discrete conformation spaces. *Biochemistry*, 30:4232–4237.

Cristobal S, Zemla A, Fischer D, Rychlewski L, Elofsson A 2001. A study of quality measures for protein threading models. *Bioinformatics*, 2:5–5.

Cuff JA, Clamp ME, Siddiqui AS, Finlay M, Barton GJ 2000. Jpred: A consensus secondary structure prediction server. *Bioinformatics*, 14:892–893.

Day R, Beck DAC, Armen RS, Daggett V 2003. A consensus view of fold space: Combining SCOP, CATH, and the Dali Domain Dictionary. *Protein Sci.*, 12:2150–2160.

de la Paz P, Sutton BJ, Darsley MJ, Rees AR 1986. Modelling of the combining sites of three anti-lysozyme monoclonal antibodies and of the complex between one of the antibodies and its epitope. *EMBO J.*, 5:415–425.

De Maeyer M, Desmet J, Lasters I 1997. All in one: a highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. *Folding and Design*, 2:53–66.

Dietmann S, Frömmel C 2002. Prediction of 3D neighbours of molecular surface patches in proteins by artificial neural networks. *Bioinformatics*, 18:167–174.

Dill KA, Chan HS 1997. From levinthal to pathways to funnels. *Nature: Struct. Biol.*, 4:10–19.

Dobson CM, Šali A, Karplus M 1998. Protein folding: A perspective from theory and experiment. *Angew. Chem. Int. Ed. Engl.*, 198:868–893.

Donate LE, Rufino SD, Canard LH, Blundell TL 1996. Conformational analysis and clustering of short and medium size loops connecting regular secondary structures: a database for modeling and prediction. *Protein Sci.*, 5:2600–2616.

Dunbrack RL, Karplus M 1993. Backbone-dependent rotamer library for proteins: Application to side-chain prediction. *J. Mol. Biol.*, 230:543–574.

Engh RA, Huber R 1991. Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Crystallogr.*, A47:392–400.

Facchiano AM, Stiuso P, Chiusano ML, Caraglia M, Giuberti G, Marra M, Abbruzzese A, Colonna G 2001. Homology modelling of the human eukaryotic initiation factor 5A (eIF-5A). *Protein Eng.*, 14:881–890.

Feig M, Rotkiewicz P, Kolinski A, Skolnick J, Brooks CL 2000. Accurate reconstruction of all-atom protein representations from side-chain-based low-resolution models. *Proteins: Struct., Funct., Genet.*, 41:86–97.

Feng ZK, Sippl MJ 1996. Optimum superimposition of protein structures: ambiguities and implications. *Folding and Design*, 1:123–132.

Fidelis K, Stern PS, Bacon D, Moult J 1994. Comparison of systematic search and database methods for constructing segments of protein structure. *Protein Eng.*, 7:953–960.

Fine RM, Wang H, Shenkin PS, Yarmush DL, Levinthal C 1986. Predicting antibody hypervariable loop conformations. II: Minimization and molecular dynamics studies of MCPC603 from many randomly generated loop conformations. *Proteins: Struct., Funct., Genet.*, 1:342–362.

Finkelstein AV, Reva BA 1991. A search for the most stable folds of protein chains. *Nature (London)*, 351:497–499.

Fischer D, Elofsson AE, Rychlewski L 2000. The 2000 Olympic Games of protein structure prediction; fully automated programs are being evaluated vis-a-vis human teams in the protein structure prediction experiment CAFASP2. *Protein Eng.*, 13:667–670.

Fiser A, Do RK, Šali A 2000. Modeling of loops in protein structures. *Protein Sci.*, 9:1753–1773.

Fiser A, Šali A 2003. Modloop: automated modeling of loops in protein structures. *Bioinformatics*, 19:2500–2501.

Fletcher R, Reeves CM 1964. Function minimisation by conjugate gradients. *Comput. J.*, 7:149–154.

Flores TP, Orengo CA, Moss DS, Thornton JM 1993. Comparasion of conformational characteristics in structurally similar protein pairs. *Protein Sci.*, 2:1811–1826.

Forrest S 1993. Genetic algorithms: principles of natural selection applied to computation. *Science*, 261:872–878.

Frigerio F, Margarity I, Nogarotto R, Grandi G, Vriend G, Hardy F, Veltman OR, Venema G, Eijsink VGH 1997. Model building of a thermolysin-like protease by mutagenesis. *Protein Eng.*, 10:223–230.

Garnier J, Gibrat JF, Robson B 1996. GOR secondary structure prediction method version IV. *Meth. Enzymol.*, 266:540–553.

Gerstein M, Levitt M 1998. Comprehensive assessment of automatic structural alignment against a manual standard, the SCOP classification of proteins. *Protein Sci.*, 7:445–456.

Gibrat JF, Madej T, Bryant SH 1996. Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.*, 266:540–553.

Godzik A 1996. The structural alignment between two proteins: Is there a unique answer? *Protein Sci.*, 5:1325–1338.

Greer J 1981. Comparative model-building of the mammalian serine proteases. *J. Mol. Biol.*, 153:1027–1042.

Greer J 1990. Comparative modeling methods: application to the family of mammalian serine proteases. *Proteins: Struct., Funct., Genet.*, 7:317–334.

Gribskov M, Robinson NL 1996. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Comput. and Chem.*, 20:25–33.

Grunenfelder B, Winzeler EA 2002. Treasures and traps in genome-wide data sets: case examples from yeast. *Nature Rev. Genet.*, 3:653–661.

Guex N, Peitsch MC 1997. SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. *Electrophoresis*, 18:2714–2723.

Hadley C, Jones DT 1999. A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure*, 7:1099–1112.

Harrison A, Pearl F, Sillitoe I, Slidel T, Mott R, Thornton J, Orengo C 2003. Recognizing the fold of a protein structure. *Bioinformatics*, 19:1748–1759.

Hearst MA, Scholkapf B, Dumais S, Osuna E, Platt J 1998. Trends and Constroversies — Support Vector Machines. *IEEE Intelligent Systems*, 13:18–28.

Hendlich M, Lackner P, Weitckus S, Floeckner H, Froschauer R, Gottsbacher K, Casari G, Sippl MJ 1990. Identification of native protein folds amongst a large number of incorrect models. *J. Mol. Biol.*, 216:167–180.

Holland JH, 1975. *Adaptation in natural and artifical systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* The University of Michigan Press.

Holm L, Park J 2000. DaliLite workbench for protein structure comparison. *Bioinformatics*, 16:566–567.

Holm L, Sander C 1993. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233:123–138.

Honig B 1999. Protein folding: From the Levinthal paradox to structure prediction. *J. Mol. Biol.*, 293:283–293.

Huang ES, Samudrala R, Ponder JW 1999. *Ab Initio* fold prediction of small helical proteins using distance geometry and knowledge-based scoring functions. *J. Mol. Biol.*, 290:267–281.

Huang ES, Subbiah S, Levitt M 1995. Recognizing native folds by the arrangement of hydrophobic and polar residues. *J. Mol. Biol.*, 252:709–720.

Hubbard TJP 1999. (RMS/coverage) graphs: a qualitative method for comparing three-dimensional protein structure predictions. *Proteins: Struct., Funct., Genet.*, Suppl 3:15–21.

Hubbard TJP, Blundell TL 1987. Comparasion of solvent-inaccessible cores of homologous proteins: definitions useful for protein modeling. *Protein Eng.*, 1:159–171.

Ison JC, Blades MJ 2005. Rocplot: A generic software tool for roc analysis and the validation of predictive methods. *Appl Bioinformatics*, 4:131–135.

Jacobson M, Šali A 2004. Comparative protein structure modeling and its applications to drug discovery. *Annu. Rep. Med. Chem*, 39:259–276.

Jaroszewski L, Li W, Godzik A 2002. In search for more accurate alignments in the twilight zone. *Protein Sci.*, 11:1702–1713.

Jiang T, Cui Q, Shi G, Ma S 2003. Protein folding simulations of the hydrophobic-hydrophilic model by combining tabu search with genetic algorithms. *J. Chem. Phys.*, 119:4592–4596.

John B, Šali A 2003. Comparative protein structure modeling by iterative alignment, model building and model assessment. *Nuc. Ac. Res.*, 31:3982–3992.

Johnston CR, Shields DC 2005. A sequence sub-sampling algorithm increases the power to detect distant homologues. *Nuc. Ac. Res.*, 33:3772–3778.

Jones TA, Thirup S 1986. Using known substructures in protein model building and crystallography. *EMBO J.*, 5:819–822.

Kabsch W, Sander C 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637.

Kale L, Skeel R, Bhandarkar M, Brunner R, Gursoy A, Krawetz N, Phillips J, Shinozaki A, Varadarajan K, Schulten K 1999. NAMD2: greater scalability for parallel molecular dynamics. *J. Comput. Phys.*, 151:283–312.

Karlin S, Morris M, Ghandour G, Leung MY 1988. Efficient algorithms for molecular sequence analysis. *Proc. Natl. Acad. Sci. USA*, 85:841–845.

Karplus K, Hu B 2001. Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. *Bioinformatics*, 17:713–720.

Kawabata T, Nishikawa K 2000. Protein structure comparison using the markov transition model of evolution. *Proteins: Struct., Funct., Genet.*, 41:108–122.

Kelley LA, MacCallum RM, Sternberg MJE 2000. Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.*, 299:499–520.

Kihara D, Lu H, Kolinski A, Skolnick J 2001. TOUCHSTONE: an *ab initio* protein stucture prediction method that uses threading-based tertiary restraints. *Proc. Natl. Acad. Sci. USA*, 98:10125–10130.

Kleywegt GJ 1996. Use of non-crystallographic symmetry in protein structure refinement. *Acta Crystallogr.*, 52:842–857.

Kneller DG, Cohen FE, Langridge R 1990. Improvements in protein secondary structure prediction by an enhanced neural network. *J. Mol. Biol.*, 214:171–182.

Kocher JP, Rooman MJ, Wodak SJ 1994. Factors influencing the ability of knowledge-based potentials to identify native sequence-structure matches. *J. Mol. Biol.*, 235:1598–1613.

Kodratoff Y, Michalski R, 1990. *Machine Learning: An Artifical Intelligence Approach Volume 3.* Morgan Kaufmann Publishing Inc.

Kolinski A, Skolnick J 1998. Assembly of protein structure from sparse experimental data: an efficient Monte Carlo model. *Proteins: Struct., Funct., Genet.*, 32:475–494.

Kolodny R, Koehl P, Levitt M 2005. Comprehensive evaluation of protein structure alignment methods: Scoring by geometric measures. *J. Mol. Biol.*, 346:1173–1188.

Krissinel E, Henrick K 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr.*, 60:2256–2268.

Kyngäs J, Valjakka J 1998. Unreliability of the Chou-Fasman parameters in predicting protein secondary structure. *Protein Eng.*, 11:345–348.

Labesse G, Mornon J 1998. Incremental threading optimization (TITO) to help alignment and modelling of remote homologues. *Bioinformatics*, 14:206–211.

Lambert C, Leonard N, De Bolle X, Depiereux E 2002. ESyPred3D: prediction of proteins 3D structures. *Bioinformatics*, 18:1250–1256.

Laskowski RA, MacArthur MW, Moss DS, Thornton JM 1993. Procheck-a program to check the stereochemical quality of protein structures. *J. Appl. Cryst.*, 26:283–291.

Lesk AM, Chothia C 1980. How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *J. Mol. Biol.*, 136:225–270.

Li W, Liang S, Wang R, Lai L, Han Y 1999. Exploring the conformational diversity of loops on conserved frameworks. *Protein Eng.*, 12:1075–1086.

Liang S, Grishin NV 2001. Side-chain modeling with an optimized scoring function. *Protein Sci.*, 11:322–331.

Lin HN, Wu KP, Chang JM, Sung TY, Hsu WL 2005. GANA–a genetic algorithm for NMR backbone resonance assignment. *Nuc. Ac. Res.*, 33:4593–4601.

Liu ED, Yang GL, Tian BJ, Li ZW, Chen Y 2002. Application of resilient backpropagation neural network in predicting hydrophobic parameters of alkylbenzenes. *Se Pu*, 20:216–218.

Lo Conte L, Ailey B, Hubbard TJ, Brenner SE, Murzin AG, Chothia C 2000. SCOP: a structural classification of proteins database. *Nuc. Ac. Res.*, 28:257–259.

Lovell SC, Word JM, Richardson JS, Richardson DC 2000. The penultimate rotamer library. *Proteins: Struct., Funct., Genet.*, 40:389–408.

MacKerell AD, Bashford D, Bellott D, Dunbrack RI, Evanseck JD, Field MJ, Fischer S, Gao J, Guo H, Ha S, Joseph-McCarthy D, Kuchnir L, Kuczera K, Lau FTK, Mattos C, Michnick S, Ngo T, Nguyen DT, Prodhom B, Reiher WE, Roux B, Schlenkrich M,

Smith J, Stote R, Straub J, Watanabe M, Wiarkiewicz-Kuczera J, Yin D, Karplus M 1998. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem.*, 102:3586–3616.

Madej T, Gibrat JF, Bryant SH 1995. Threading a database of protein cores. *FEBS Lett.*, 373:13–18.

Maiorov VN, Crippen GM 1992. Contact potential that recognizes the correct folding of globular proteins. *J. Mol. Biol.*, 227:876–888.

Marti-Renom MA, Madhusudhan MS, Šali A 2004. Alignment of protein sequences by their profiles. *Protein Sci.*, 13:1071–1087.

Martí-Renom MA, Stuart AC, Fiser A, Sánchez R, Melo F, Ali A 2000. Comparative protein structure: modelling of genes and genomes. *Annu. Rev. Biophys. Biomolec. Struct.*, 29:291–325.

Martin ACR, 1990. Computer program: nw V3.11. `http://www.bioinf.org.uk/software/nw/`.

Martin ACR, 1995. Computer program: MINT V3.0 (modeller interface). `http://www.bioinf.org.uk/software/mint/`.

Martin ACR, 2000a. Computer program: QLite V1.0. `http://www.bioinf.org.uk/software/qlite/`.

Martin ACR 2000b. The ups and downs of protein topology: Rapid comparison of protein structure. *Protein Eng.*, 13:829–837.

Martin ACR, 2001. Computer program: ProFit V2.2. `http://www.bioinf.org.uk/software/profit/`.

Martin ACR, Cheetham JC, Rees AR 1989. Modelling Antibody Variable Loops: A combined algorithm. *Proc. Natl. Acad. Sci. USA*, 86:9268–9272.

Martin ACR, MacArthur MW, Thornton JM 1997. Assessment of comparative modeling in CASP2. *Proteins: Struct., Funct., Genet.*, Suppl. 1:14–28.

Martin, A. C. R. IL, 1999. Computer program: SS V1.0.

Matsuo Y, Nishikawa K 1994. Protein structural similarities predicted by a sequence-structure compatibility method. *Protein Sci.*, 3:2055–2063.

Matthews BW 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, 405:442–451.

McCammon JA, Harvey SC, 1987. *Dynamics of Proteins and Nucleic Acids.* Cambridge University Press, Cambridge.

McGuffin LJ, Bryson K, Jones DT 2000. The PSIPRED protein structure prediction server. *Bioinformatics*, 16:404–405.

Michalski RS, 1983. *Machine Learning: An Artifical Intelligence Approach Volume 1.* Tioga Publishing Company.

Miyazawa S, Jernigan RL 2000. Identifying sequence-structure pairs undetected by sequence alignments. *Protein Eng.*, 13:459–475.

Moult J, Hubbard T, Bryant SH, Fidelis K, Pedersen JT, Predictors 1997. Critical assessment of methods of protein structure and prediction (CASP): round II. *Proteins: Struct., Funct., Genet.*, Suppl. 3:2–6.

Moult J, Hubbard T, Fidelis K, Pedersen JT 1999. Critical assessment of methods of protein structure and prediction (CASP): round III. *Proteins: Struct., Funct., Genet.*, Suppl. 3:2–6.

Mückstein U, Hofacker IL, Stadler PF 2002. Stochastic pairwise alignments. *Bioinformatics*, 18:s153–s160.

Murzin AG, Brenner SE, Hubbard T, Chothia C 1995. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–540.

Naor D, Brutlag DL 1994. On near-optimal alignments of biological sequences. *J. Comp. Biol.*, 1:349–366.

Needleman SB, Wunsch CD 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453.

Nielsen H, Brunak S, von Heijne G 1999. Machine learning approaches for the prediction of signal peptides and other protein sorting signals. *Protein Eng.*, 12:3–9.

Novotny J, Bruccoleri R, Karplus M 1984. An analysis of incorrectly folded protein models. Implications for structure predictions. *J. Mol. Biol.*, 177:787–818.

Novotny J, Rashin AA, Brucceroli RE 1988. Criteria that discriminate between native folds and incorrectly folded models. *Proteins: Struct., Funct., Genet.*, 4:787–818.

Novotny M, Madsen D, Kleywegt GJ 2004. Evaluation of protein fold comparison servers. *Proteins: Struct., Funct., Genet.*, 54:260–270.

Nymeyer H, García AE 2003. Simulation of the folding equilibrium of alpha-helical peptides: a comparison of the generalized Born approximation with explicit solvent. *Proc. Natl. Acad. Sci. USA*, 100:13934–13939.

Odaert B, Landrieu I, Dijkstra K, Schuurman-Wolters G, Casteels P, Wieruszeski JM, Inze D, Scheek R, Lippens G 2002. Solution NMR study of the monomeric form of p13suc1 protein sheds light on the hinge region determining the affinity for a phosphorylated substrate. *J. Biol. Chem.*, 277:12375–12381.

Ogata K, Umeyama H 2000. An automatic homology modeling method consisting of database searrches and simulated annealing. *J. Mol. Graph.*, 18:305–306.

Oliva B, Bates PA, Querol E, Aviles FX, Sternberg MJ 1998. Automated classification of antibody complementarity determining region 3 of the heavy chain (H3) loops into canonical forms and its application to protein structure prediction. *J. Mol. Biol.*, 279:1193–1210.

Orengo C, Michie A, Jones S, Swindells M, Thornton J 1997. CATH — a hierarchic classification of protein domain structures. *Structure*, 5:1093–1108.

Orengo CA, Brown NP, Taylor WR 1992. Fast structure alignment for protein databank searching. *Proteins: Struct., Funct., Genet.*, 14:139–167.

Ortiz AR, Kolinski A, Skolnick J 1998. Fold assembly of small proteins using Monte Carlo simulations driven by restraints derived from multiple sequence alignments. *J. Mol. Biol.*, 277:419–448.

Panchenko AR, Marchler-Bauer A, Bryant SH 2000. Combination of threading potentials and sequence profiles improves fold recognition. *J. Mol. Biol.*, 296:1319–1331.

Park B, Levitt M 1996. Energy functions that discriminate X-ray and near native folds from well-constructed decoys. *J. Mol. Biol.*, 258:367–392.

Park SH, Goo JM, Jo CH 2004. Receiver operating characteristic (ROC) curve: practical review for radiologists. *Korean J. Radiol*, 5:11–18.

Pearl FM, Lee D, Bray JE, Sillitoe I, Todd AE, Harrison AP, Thornton JM, Orengo CA 2000. Assigning genomic sequences to CATH. *Nuc. Ac. Res.*, 28:277–282.

Pearson WR 1990. Rapid and sensitive sequence comparison with FASTP and FASTA. *Meth. Enzymol.*, 183:63–98.

Pearson WR 1996. Effective protein sequence comparison. *Meth. Enzymol.*, 266:227–258.

Pearson WR, Lipman DJ 1988. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, 85:2444–2448.

Peitsch MC 1996. ProMod and Swiss-Model: Internet-based tools for automated comparative modelling. *Biochem. Soc. Trans. (London)*, 24:274–279.

Peterson WW 1954. The theory of signal dectectability. *IRE Transactions on Information Theory*, 4:171–212.

Pettitt CS, McGuffin LJ, Jones DT 2005. Improving sequence-based fold recognition by using 3D model quality assessment. *Bioinformatics*, 21:3509–3515.

Pierce NA, Winfree E 2002. Protein design is NP-hard. *Protein Eng.*, 15:779–782.

Pillardy J, Czaplewski C, Liwo A, Lee J, Ripoll DR, Kazmierkiewicz R, Oldziej S, Wedemeyer WJ, Gibson KD, Arnautova YA, Saunders J, Ye YJ, Scheraga HA 2001. Recent improvements in prediction of protein structure by global optimization of a potential energy function. *Proc. Natl. Acad. Sci. USA*, 98:2329–2333.

Ponder JW, Richards FM 1987. Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.*, 193:775–791.

Press WH, Flannery, Brian P. andthroughput Teukolsky SA, Vetterling WT, 1986. *Numerical Recipies, The Art of Scientific Computing.* Cambridge University Press.

Ramakrishnan V, Finch JT, Graziano V, Lee PL, Sweet RM 1993. Crystal structure of globular domain of histone H5 and its implications for nucleosome binding. *Nature (London)*, 362:219–223.

Reckwitz T, Potter SR, Snow PB, Zhang Z, Veltri RW, Partin AW 1999. Artificial neural networks in urology: update 2000. *Prostate Cancer Prostatic Dis.*, 2:222–226.

Reidmiller M, Braun H 1993. A direct adaptive method for faster backpropagation learning: the Rprop algorithm. *Proceedings International Conference on Neural Networks*, pages 586–591.

Reva BA, Finkelstein AV, Sanner MF, Olson AJ 1997. Residue-residue mean-force potentials for protein structure recognition. *Protein Eng.*, 10:865–876.

Ring CS, Kneller DG, Langridge R, Cohen FE 1992. Taxonomy and conformational analysis of loops in proteins. *J. Mol. Biol.*, 224:685–699.

Rodriguez R, Chinea G, N. L, Pons T, Vriend G 1997. Homology modeling, model and software evaluation: three related resources. *Bioinformatics*, 14:523–528.

Rost B 1999. Twilight zone of protein sequence alignments. *Protein Eng.*, 12:85–94.

Rost B, Sander C 1993. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599.

Rost B, Sander C 1994. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Struct., Funct., Genet.*, 19:55–72.

Rufino SD, Donate LE, Canard L, Blundell TL 1996. Analysis, clustering and prediction of the conformation of short and medium size loops connecting regular secondary structures. *Pac. Symp. Biocomput.*, pages 570–589.

Russell S, Norvig P, 2002. *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Samudrala R, Levitt M 2002. A comprehensive analysis of 40 blind protein structure predictions. *BMC Struct. Biol.*, 2:3.

Samudrala R, Moult J 1998. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *J. Mol. Biol.*, 275:895–916.

Sánchez R, Šali A 1997. Evaluation of comparative protein structure modeling by MODELLER-3. *Proteins: Struct., Funct., Genet.*, 1:50–58.

Sander C, Schneider R 1991. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Struct., Funct., Genet.*, 9:56–68.

Saqi MAS, Russell RB, Sternberg MJE 1998. Misleading local sequence alignments: implications for comparative modelling. *Protein Eng.*, 11:627–630.

Saqi MAS, Wild DL, Hartshorn MJ 1999. Protein analyst–a distributed object environment for protein sequence and structure analysis. *Bioinformatics*, 15:521–522.

Sasin JM, Bujnicki JM 2004. COLORADO3D, a web server for the visual analysis of protein structures. *Nuc. Ac. Res.*, 32:W586–W589.

Saunder JM, Arthur JW, Dunbrack RL 2000. Large-scale comparison of protein sequnce alignment algorithims with structure alignments. *Proteins: Struct., Funct., Genet.*, 40:6–22.

Sayle RA, Milner-White EJ 1995. RASMOL: biomolecular graphics for all. *Trends Biochem. Sci.*, 20:374.

Schiffmann W, Joost M, Werner R, 1993. Comparison of optimized backpropagation algorithms. In Verleysen (ed.), *Proceedings of the European Symposium on Artificial Neural Networks, ESANN '93.* `http://citeseer.ist.psu.edu/schiffmann93comparison.html`.

Schwede T, Kopp J, Guex N, Peitsch MC 2003. SWISS-MODEL: An automated protein homology-modeling server. *Nuc. Ac. Res.*, 31:3381–3385.

Sebastiani P, H. YY, Ramoni M 2003. Bayesian machines learning and its potential applications to the genomic study of oral oncology. *Adv. Dent. Res.*, 17:104–108.

Shatsky M, Nussinov R, Wolfson HJ 2006. Optimization of multiple-sequence alignment based on multiple-structure alignment. *Proteins: Struct., Funct., Genet.*, 62:209–217.

Shetty RP, de Bakker PIW, DePristo MA, Blundell TL 2003. Advantages of fine-grained side chain conformer libraries. *Protein Eng.*, 16:963–969.

Shi J, Blundell TL, Mizuguchi K 2001. FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol.*, 310:243–257.

Shindyalov IN, Bourne PE 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, 11:739–747.

Shindyalov IN, Bourne PE 2000. An alternative view of protein fold space. *Proteins: Struct., Funct., Genet.*, 38:247–260.

Sierk ML, Pearson WR 2004. Sensitivity and selectivity in protein structure comparasion. *Protein Sci.*, 13:773–785.

Siew N, Elofsson A, Rychlewski L, Fischer D 2000. MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, 16:776–785.

Simons KT, Strauss C, Baker D 2001. Prospects for *ab initio* protein structural genomics. *J. Mol. Biol.*, 306:1191–1199.

Sippl MJ 1990. Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. *J. Mol. Biol.*, 213:859–883.

Sippl MJ 1993. Recognition of errors in three-dimensional structures of proteins. *Proteins: Struct., Funct., Genet.*, 17:355–362.

Sippl MJ, Jaritz M, Hendlich M, Ortner M, Lackner P, 1994. Applications of knowledge-based mean fields in the determination of protein structures. In *Statistical Mechanics, Protein Structure, and Protein Substrate Interactions*, pages 223–230. Plenum Press.

Sippl M 1995. Knowledge-based potentials for proteins. *Curr. Opin. Struct. Biol.*, 5:229–235.

Skolnick J, Kihara D 2001. Defrosting the frozen approximation: PROSPECTOR–a new approach to threading. *Proteins: Struct., Funct., Genet.*, 42:319–331.

Smith DK, Thornton JM, 1989. Computer program: SStruc.

Smith RF, Smith TF 1992. Pattern-induced multi-sequence alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modelling. *Protein Eng.*, 5:35–41.

Smith TF, Waterman MS 1981. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197.

Sobel E, Martinez H 1986. A multiple sequence alignment program. *Nuc. Ac. Res.*, 14:363–374.

Söding J, Biegert A, Lupas AN 2005. The HHpred interactive server for protein homology detection and structure prediction. *Nuc. Ac. Res.*, 33:W244–W248.

Spinosa EJ, de Carvalho ACPLF 2005. Support vector machines for novel class detection in Bioinformatics. *Genet. Mol. Res.*, 4:608–615.

Steiner T 2002. The hydrogen bond in the solid state. *Angew. Chem. Int. Ed. Engl.*, 41:49–76.

Subbiah S, Laurents DV, Levitt M 1993. Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. *Curr. Biol.*, 3:141–148.

Summers NL, Karplus M 1991. Modeling of side chains, loops, and insertions in proteins. *Meth. Enzymol.*, 202:156–204.

Sutcliffe MJ, Haneef I, Carney D, Blundell TL 1987a. Knowledge based modelling of homologous proteins. 1. Three-dimensional frameworks derived from simultaneous superposition of multiple structures. *Protein Eng.*, 1:377–384.

Sutcliffe MJ, Hayes FRF, Blundell TL 1987b. Knowledge based modelling of homologous proteins. 2. Rules for the conformations of sustituted side chains. *Protein Eng.*, 1:385–392.

Swets JA 1988. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293.

Tanaka S, Scheraga HA 1976. Medium and long-range interaction parameters between amino-acids for predicting three dimensional structures of proteins. *Macromolecules*, 9:945–950.

Taylor WR, Flores TP, Orengo CA 1994. Multiple protein structure alignment. *Protein Sci.*, 3:1858–1870.

Taylor WR, Orengo CA 1989. Protein structure alignment. *J. Mol. Biol.*, 208:1–22.

Tetko IV, Facius A, Ruepp A, Mewes HW 2005. Super paramagnetic clustering of protein sequences. *Bioinformatics*, 6:82–82.

Thomas PD, Dill KA 1996. An iterative method for extracting energy-like quantities from protein structures. *Proc. Natl. Acad. Sci. USA*, 93:11628–11633.

Thompson JD, Higgins DG, Gibson TJ 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nuc. Ac. Res.*, 22:4673–4680.

Tuffery P, Etchebest C, Hazout S, Lavery R 1991. A new approach to the rapid determination of protein side chain conformations. *J. Biomol. Struct. Dyn.*, 8:1267–1289.

Vapnik VN, 1995. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York.

Veltri RW, Chaudhari M, Miller MC, Poole EC, O'Dowd GJ, Partin AW 2002. Comparison of logistic regression and neural net modeling for prediction of prostate cancer pathologic stage. *Clinical Chemistry*, 48:1828–1834.

Vingron M, Argos P 1989. A fast and sensitive multiple sequence alignment algorithm. *Comput. Appl. Biosci.*, 5:115–121.

Vingron M, Pevzner PA 1995. Multiple sequence comparison and consistency on multipartite graphs. *Advances in Applied Mathematics*, 16:1–22.

Vitkup D, Melamud E, Moult J, Sander C 2001. Completeness in structural genomics. *Nature: Struct. Biol.*, 8:559–566.

Šali A, Blundell TL 1993. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, 234:779–815.

Šali A, Overington J 1994. Derivation of rules for comparative protein modeling from a database of protein structure alignments. *Protein Sci.*, 3:1582–1596.

Wang C, Lefkowitz EJ 2005. Genomic multiple sequence alignments: refinement using a genetic algorithm. *BMC Bioinformatics*, 6:200–200.

Waterman MS 1986. Multiple sequence alignment by consensus. *Nuc. Ac. Res.*, 14:9095–9102.

Waterman M 1984. General methods of sequence comparison. *Bull. Math. Biol.*, 46:473–500.

Webb B, 2005. About MODELLER. `http://salilab.org/modeller/`.

Wei JT, Zhang Z, Barnhill SD, Madyastha KR, Zhang H, Oesterling JE 1998. Understanding artificial neural networks and exploring their potential applications for the practicing urologist. *Urology*, 52:161–172.

Westhead DR, Thornton JM 1998. Protein structure prediction. *Curr. Opin. Biotech.*, 9:383–390.

Wu CH, Huang H, Arminski L, Castro-Alvear J, Chen Y, Hu ZZ, Ledley RS, Lewis KC, Mewes HW, Orcutt BC, Suzek BE, Akira T, Vinayaka CR, Yeh LSL, Zhang J, Barker WC 2002. The Protein Information Resource: an integrated public resource of functional annotation of proteins. *Nuc. Ac. Res.*, 30:35–37.

Zell A, Mamier G, Vogt M, Mache N, Hubner R, Döring S, Herrmann KU, Soyez T, Schmalzl M, Sommer T, Hatzigeorgiou A, Posselt D, Schreiner T, Kett B, Clemente G, Wieland J, 1995. Stuttgart neural network simulator. `http://www-ra.informatik.uni-tuebingen.de/SNNS/`.

Zhang C, Lu H, Zhou H, Zhou Y 2004. An accurate, residue-level, pair potential of mean force for folding and binding based on the distance-scaled, ideal-gas reference state. *Protein Sci.*, 13:400–411.

Zhang L, Skolnick J 1998. How do potentials derived from structural databases relate to "true" potentials? *Protein Sci.*, 7:112–122.

Zupan J, Gasteiger J, 1993. *Neural Networks for Chemists.* VCH Publishers (UK) Ltd.