

# **An Automated Approach to Remote Protein Homology Classification**

Timothy James Dallman

Department of Biochemistry and Molecular Biology  
University College London

A thesis submitted to the University of London in the Faculty of  
Science for the degree of Doctor of Philosophy

September 2007



UMI Number: U591450

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI U591450

Published by ProQuest LLC 2013. Copyright in the Dissertation held by the Author.  
Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against  
unauthorized copying under Title 17, United States Code.



ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346



# Abstract

The classification of protein structures into evolutionary superfamilies, for example in the CATH or SCOP domain structure databases, although performed with varying degrees of automation, has remained a largely subjective activity guided by expert knowledge. The huge expansion of the Protein Structure Databank (PDB), partly due to the structural genomics initiatives, has posed significant challenges to maintaining the coverage of these structural classification resources. This is because the high degree of manual assessment currently involved has affected their ability to keep pace with high throughput structure determination.

This thesis presents an evaluation of different methods used in remote homologue detection which was performed to identify the most powerful approaches currently available. The design and implementation of new protocols suitable for remote homologue detection was informed by an analysis of the extent to which different homologous superfamilies in CATH evolve in sequence, structure and function and characterisation of the mechanisms by which this occurs. This analysis revealed that relatives in some highly populated CATH superfamilies have diverged considerably in their structures. In diverse relatives, significant variations are observed in the secondary structure embellishments decorating the common structural core for the superfamily. There are also differences in the packing angles between secondary structures. Information on the variability observed in CATH superfamilies is collated in an established web resource the Dictionary of Homologous Superfamilies, which has been expanded and improved in a number of ways.

A new structural comparison algorithm, CATHEDRAL, is described. This was developed to cope with the structural variation observed across CATH superfamilies and to improve the automatic recognition of domain boundaries in multidomain structures. CATHEDRAL combines both secondary structure matching and accurate residue alignment in an iterative protocol for determining the location of previously observed folds in novel multi-domain structures. A rigorous benchmarking protocol is also

described that assesses the performance of CATHEDRAL against other leading structural comparison methods.

The optimisation and benchmarking of several other methods for detecting homology are subsequently presented. These include methods which exploit Hidden Markov Models (HMMs) to detect sequence similarity and methods that attempt to assess functional similarity.

Finally an automated, machine learning approach to detecting homologous relationships between proteins is presented which combines information on sequence, structure and functional similarity. This was able to identify over 85% of the homologous relationships in the CATH classification at a 5% error rate.

This thesis was gratefully supported by the Biotechnology and Biological Sciences Research Council.

# Acknowledgements

Firstly I would like to thank my supervisor Professor Christine Orengo for her inspiring supervision over the last four years. She never failed to encourage and motivate me throughout my time in her lab and has provided me with great guidance both academically and beyond.

Working in the Orengo lab, in our big office on the 6<sup>th</sup> floor, I've had the pleasure of working with so many weird and wonderful souls. With special mentions to Mark Dibley for putting up with my incessant badgering for advice, Tony Lewis for entertaining me with his 'special' sense of humour, Adam Reid for being the best smoking buddy you could wish for (until he gave up that is), Russell Marsden for his sense of perspective and Ollie Redfern for well just being Ollie! I'd also like to mention some of my past and present colleagues in the CATH group and beyond who've made the last four years so enjoyable; they include, Ali Cuff, Benoit Dessailly, Illhem Diboun, Jacob Hurst, Caroline Johnston, Stefano Lise, Lisa McMillan, Stathis Sideris, Ian Sillitoe, Corin Yeats, Sarah Addou, Adrian Akpor, Juan Antonio Ranea and Jesse Oldershaw.

I'd also like to thank the unerring support of my friends who consistently tried to stop their eyes glazing over when I tried to explain what this thesis was about and also putting up superbly with my petulant responses to that dreaded question "So how's the thesis going?". Special mentions to Laura, Lou, Chris, Danny, Paul and George, Daf & Soph, Mike & Jules, James & Amy, Mark & Becky, Jonny, Laura & Ed, Mo & Ellie, Josh & Dee and Peishan with her Pei-plan.

Finally I'd like to thank my family as without their love and support this thesis would never have been completed. So a final big thank you to mum, dad, Claire and Matt.

# Abbreviations

Abbreviation	Details
BaliBase	Benchmark Alignment Database
BLAST	Basic Local Alignment Search Tool
BLOCKS	Blocks of Amino Acid Substitution Matrices
BLOSUM	Blocks Substitution Matrices
CATH	Class, Architecture, Topology and Homologous superfamily
CATHEDRAL	CATH's Existing Recognition Algorithm
CE	Combinatorial Extension algorithm
COG	Cluster of Orthologous Groups
CORA	Consensus Of Residue Attributes
DALI	Distance Matrix Alignment
DDP	Double Dynamic Programming
DHS	Dictionary of Homologous Superfamilies
DNA	Deoxyribonucleic acid
EC	Enzyme Classification
EPQ	Error Per Query
FASTA	Fast All
FLORA	Functional Listing Of Residue Attributes
FSSP	Fold classification based on Structure-Structure alignment of Proteins
GO	Gene Ontology
GOA	Gene Ontology Annotation Project
HMM	Hidden Markov Model
HSSP	Homology derived Secondary Structure of Proteins
KEGG	Kyoto Encyclopaedia of Genes and Genomes
LSQMAN	Least Squares Alignment

<b>Abbreviation</b>	<b>Details</b>
MCC	Mathews Correlation Coefficient
NMR	Nuclear Magnetic Resonance
NN	Neural Network
PAM	Percent or Point Accepted Mutation
PDB	Protein Data Bank
PSI-BLAST	Position Specific Iterated-BLAST
PSSM	Position Specific Score Matrices
PRC	Profile Comparer
ROC	Receiver Operator Curve
RMSD	Root Mean Squared Deviation
SAM	Sequence Alignment and Modelling System
SAS	Structural Alignment Score
SAWTED	Structure Assignment With Text Description
SCOP	Structural Classification Of Proteins
SGI	Structural Genomics Initiative
SNNS	Stuttgart Neural Network Simulator
SSAP	Sequential Structural Alignment Program
SSEs	Secondary Structure Elements
SSE	Sum Squared Error
SSG	Structurally Similar Sub-Group
SSM	Secondary Structure Matching
STRUCTAL	Structural Alignment Server
SVM	Support Vector Machine
VAST	Vector Alignment Search Tool

# Table of Contents

Abstract .....	2
Acknowledgements.....	4
Abbreviations.....	5
Table of Contents.....	7
List of Figures .....	10
List of Tables .....	13
List of Equations .....	14
1 Introduction.....	15
1.1 Proteins .....	15
1.1.1 Primary Structure.....	16
1.1.2 Secondary Structure.....	17
1.1.3 Super-secondary structure.....	19
1.1.4 Tertiary Structure & Protein Domains.....	20
1.1.5 Quaternary Structure.....	20
1.2 Protein Evolution .....	21
1.3 Detection of Homology.....	22
1.3.1 Sequence Comparison Methods.....	23
1.3.2 Structural Comparison Methods .....	31
1.4 Protein Structure Family Resources.....	38
1.4.1 CATH database.....	39
1.4.2 SCOP.....	41
1.4.3 FSSP.....	41
1.5 Machine Learning Approaches to Bioinformatics .....	41
1.5.1 Artificial Neural Networks .....	43
1.5.2 Support Vector Machines .....	48
1.6 Aims.....	51
2 Analysis of Sequence, Structure and Functional Variability between Evolutionary Relatives in CATH Superfamilies .....	53
2.1 Background.....	53
2.2 Methods.....	60

2.2.1	Data Sets for Measuring Sequence, Structural and Functional Variability in CATH Superfamilies .....	60
2.2.2	Methods for Measuring Structural Similarity between Relatives.....	63
2.2.3	Methods for Measuring Sequence Similarity between Relatives .....	66
2.2.4	Predicting CATH relatives in UniProt.....	66
2.2.5	Extracting Functional Information from Public Resources for Sequences in the CATH-DHS. ....	67
2.2.6	Measuring the Variability in Secondary Structure Orientations – The EquivSEC Program.....	68
2.3	Results.....	70
2.3.1	The Extent of Structural Change in Domain Superfamilies and the Correlation between Sequence, Structural and Functional Similarity.....	70
2.3.2	Mechanisms of Structural Change .....	75
2.4	Discussion.....	88
3	Optimisation of the CATHEDRAL Algorithm to Classify Domains in CATH.....	91
3.1	Background and Aims.....	91
3.2	Methods.....	97
3.2.1	The CATHEDRAL Protocol.....	97
3.2.2	Data Sets Used for Optimising and Benchmarking CATHEDRAL .....	102
3.2.3	Comparing the Performance of CATHEDRAL in Aligning Single Domain Structures against Other Publicly Available Methods .....	103
3.2.4	Assessing the Performance of Fold Recognition Methods .....	104
3.3	Results.....	109
3.3.1	Structure Comparison Methods – Assessing the Performance of CATHEDRAL in Recognising the Correct Fold .....	109
3.3.2	Optimising CATHEDRAL to identify domains within Multi-domain Protein Structures.....	126
3.3.3	Comparison of CATHEDRAL performance in domain boundary assignment using Sequence Based HMMs .....	132
3.4	Discussion .....	133
4	Benchmarking Methods for Detecting Homologous Relationships between Proteins	135
4.1	Background and Aims.....	135

4.1.1	Using Sequence Similarity to Assess Homology.....	135
4.1.2	Using Structural Similarity to Assess Homology .....	137
4.1.3	Using Functional Similarity to Assess Homology.....	138
4.2	Methods.....	142
4.2.1	Benchmarking Sequence Comparison Methods .....	142
4.2.2	Benchmarking Structural Comparison Methods.....	145
4.2.3	Implementing and Benchmarking Function Comparison Methods .....	148
4.3	Results.....	152
4.3.1	Performance of Sequence Comparison Methods in Recognising Homology 152	
4.3.2	Benchmarking Structure Comparison Methods.....	155
4.3.3	Assessing the Performance of GOSIM and SAWTED in recognising functionally related homologues.....	164
4.4	Discussion.....	169
5	A Machine Learning Approach to Homologue Recognition.....	171
5.1	Background and Aims.....	171
5.2	Methods.....	175
5.2.1	Data Sets .....	175
5.2.2	Data Generation &Feature Selection .....	175
5.2.3	Optimisation and Benchmarking Procedure .....	177
5.2.4	Superfold Classifier .....	179
5.3	Results.....	181
5.3.1	Feature Selection.....	181
5.3.2	Optimisation of the Neural Network.....	182
5.4	Discussion .....	196
6.	Conclusions.....	198
	References.....	201
	Appendix.....	212



# List of Figures

Figure 1.1. The basic structure of a biological amino acid.....	16
Figure 1.2. A Venn diagram describing the chemical and physical properties of amino acids. ....	17
Figure 1.3. The Needleman and Wunsch dynamic programming algorithm.....	25
Figure 1.4. Overview of Hidden Markov Model (HMM), showing transition probabilities between match (M), delete (D) and insert (I) states.....	29
Figure 1.5. Flowchart of the SSAP algorithm.....	35
Figure 1.6. The DALI method of Holm and Sander (1993) .....	37
Figure 1.7. Diagram of the CATH hierarchy.....	40
Figure 1.8. Schematic diagram showing an example of the architecture of a feed-forwards layered neural network.....	44
Figure 1.9. Support Vector Machines Class Boundaries. ....	49
Figure 1.10 a) Separating two classes of data using a linear hyperplane. The soft margin (C parameter) is shown by the dotted lines. b) Two classes of data that cannot be separated in two dimensions using a line. c) By squaring the x feature in b) using the 'kernel trick', a linear solution can be found. d) A line separating two classes of data, which is linear in 4 dimensions, but not in 2.....	50
Figure 2.1. Schematic representation of the progression from close homologues, through more remote (twilight zone) and very remote (midnight zone) homologues and finally analogous/homologous structural relatives .....	55
Figure 2.2. Screenshot from DHS website showing a CORAPLOT multiple alignment of the S35Reps of the DNA helicase RuvA, C-terminal domain superfamily.....	64
Figure 2.3. A 2DSEC plot of the ATP-grasp large domain.....	65
Figure 2.4. Screenshot of the DHS website showing the functional annotations available. ....	68
Figure 2.5. Scatter plot showing the relationship between sequence, structure and function of all homologues in enzyme superfamilies.. ....	71
Figure 2.6. The relationship between the number of functional groups, defined by COG annotation and the number of structural subgroups in a superfamily.....	72
Figure 2.7. Structural similarity measured by SSAP versus EC conservation for all homologous pairs in CATH with EC classifications.. ....	73
Figure 2.8. Percentage variability in domain size of relatives in relatives of the same superfamily. ....	74
Figure 2.9. Percentage frequency of insertions comprising one or more secondary structures. ....	77
Figure 2.10. MOLSCRIPT representations of smallest and largest (in terms of secondary structures) for selected superfamilies from the most frequently embellished architectures .....	78
Figure 2.11. Three domains from the galectin-type carbohydrate recognition domain superfamily. ....	79
Figure 2.12. Three domains of the ATP-Grasp Superfamily.....	81
Figure 2.13. The observed frequency of changes in angle for each type of consensus, contacting secondary structure pairs.....	84

Figure 2.14. Average SSAP score plotted against the average deviation in contacting, consensus secondary structure orientation for each superfamily with $\geq 9$ S35Reps.	85
Figure 2.15. An example of the EquivSEC output for a particular domain superfamily (1.10.10.10, “winged helix” DNA binding domain).	87
Figure 3.1. Flow chart of CATHEDRAL algorithm for assigning folds and domain boundaries to protein chains.	99
Figure 3.2. Diagram showing how the CATHEDRAL algorithm uses the information from the secondary structure matching step to guide the dynamic programming to find the optimal alignment.	101
Figure 3.3(a-b). ROC curves showing the performance of the 6 structural comparisons methods in their ability to identify correct fold matches.	111
Figure 3.4. ROC curves plotted for different structural comparison methods based on geometric scores where a positive match represents a true fold match	114
Figure 3.5(a,b). Plot of the percentage of correct folds matched against the ranked native score for the (a) CATH and (b) CATH-SCOP dataset.	117
Figure 3.6(a,b). Plot of the percentage of correct folds matched against the SAS score for the (a) CATH and (b) CATH-SCOP dataset.	119
Figure 3.7. Plot showing the percentage of fold alignments within a particular threshold for the SAS(a), SI <sub>MIN</sub> (b) and SI <sub>MAX</sub> (c) measures	121
Figure 3.8. Average number of aligned residues for a given SAS score.	124
Figure 3.9. Graph showing how the alignments of each method compared to manually validated BaliBase alignments.	125
Figure 3.10. Comparison of the performance of the GRATH (GT), SAS, RMSD and SVM scores for assigning domains within multi-domain chains.	128
Figure 3.11. Percentage of domain assigned (blue) and the percentage of domain boundaries within 10 residues of verified boundaries (pink) at a range of SVM score cut-offs.	129
Figure 3.12. Percentage of correct folds identified at a particular rank for varying numbers of putative folds (NF) selected by the GRATH.	130
Figure 3.13. The percentage of correctly assigned domain boundaries (within 15 residues of the manually validated boundary) against varying number of superfamily representatives.	131
Figure 4.1(a-c). ROC curves plotting coverage against error per query for BLAST, PSI-BLAST, SAM, HMMer and PRC against three different datasets.	154
Figure 4.2(a,b,c,d). ROC curves plotted for different structural comparison methods where a positive match represents a true superfamily match.	157
Figure 4.3(a,b,c,d). Plot of the percentage of correct superfamily relatives matched against the ranked native score for the (a) CATH and (b) CATH-SCOP dataset and the ranked SAS score for (c) CATH and (d) CATH-SCOP dataset.	160
Figure 4.4. ROC curve showing the performance of the profile based CORALIGN method against SSAP in the recognition of remote protein homologues.	162
Figure 4.5. Graphs showing the distribution of scores from CORALIGN(a) and SSAP(b) on non-related proteins (pink), analogous proteins (green) and homologous proteins (blue).	163
Figure 4.6. ROC curves showing the performance of the GOSIM method of comparing the semantic similarity of GO terms on four differently annotated data-sets.	Figure

(a) shows the performance based on homology and Figure (b) show the performance based on identifying functionally related homologues as defined by their enzyme classification. ....	165
Figure 4.7. ROC curves showing the performance of the SAWTED method of scoring text vectors extracted from the PDB on both the full corpus and a restricted corpus only containing SWISSPROT keywords.. ....	167
Figure 4.8. ROC curves showing the performance of the SAWTED (Full Corpus) and GOSIM (GOA_S35 dataset) methods.. ....	168
Figure 5.1. Graph showing the optimisation of the architecture of the hidden layer. ...	183
Figure 5.2. ROC curves showing the performance of the neural networks at homology recognition compared to the sequence comparison methods (PRC), the structure comparison method (CATHEDRAL (native score)) and function comparison methods (PDB-SAWTED & EC Conservation) on the S35(a), S20(b) and S10(c) datasets. ....	186
Figure 5.3. EPQ curves showing the performance of the neural networks created on the S35, S20 and S10 datasets using the CATH-All dataset. ....	188
Figure 5.4. ROC curves showing the performance of the neural networks trained specifically on each of the 'Superfolds' against the general 'S35' trained network.. ....	193
Figure 5.5. ROC curves showing showing the performance of the NNS35 network, the Superfold neural network and the individual methods for the CATH-All dataset. ....	194

# List of Tables

Table 2.1. Table summarising the 3 datasets used for measuring sequence, structural and functional similarity in this chapter. ....	62
Table 2.2. Table showing the % of superfamilies which have an average variation in secondary structure orientation , greater than 25°, between 16 and 25°, 8 and 15° and less than 8°.. ....	82
Table 3.1 The percentage of residues aligned by each method relative to SSAP for all genuine fold matches. ....	123
Table 4.1. Curated exceptions for PRC on the ‘S35’ dataset at an E-value cut-off of 0.01, compared with those produced in the same conditions using the SAS-8 heuristic.. ....	145
Table 5.1. The rank based on the normalised sum squared error for each of the feature sets.....	182
Table 5.2. Optimal parameter of $\eta$ and $d_{max}$ for the Vanilla back-propagation learning algorithm for each dataset used.....	184
Table 5.3. The performance of the three neural networks created on the S35, S20 and S10 dataset in homology recognition on the CATH-All dataset in terms of Accuracy (TP+TN/TP+FP+TN+FN) and Mathews Correlation Coefficient (MCC).....	189
Table 5.4. The performance of the three neural networks created on the S35, S20 and S10 dataset in homology recognition on the CATH-All dataset in terms of Accuracy (TP+TN/TP+FP+TN+FN) and Mathews Correlation Coefficient (MCC).....	194

# List of Equations

Equation 1.1 Root Mean Square Deviation (RMSD). .....	32
Equation 2.1 The 2DSEC percentage variability score. SSE standing for Secondary Structure Elements. ....	65
Equation 3.1. SAS (Structural Alignment Score). N represents the number of aligned residues. ....	106
Equation 3.2. $SI_{MAX}$ , N represents the number of aligned residues, and $L_1, L_2$ the number of residues in the respective domains. ....	106
Equation 3.3. $SI_{MIN}$ , N represents the number of aligned residues, and $L_1, L_2$ the number of residues in the respective domains. ....	106
Equation 5.1. Accuracy equation. ....	179
Equation 5.2. Sensitivity equation. ....	179
Equation 5.3 Specificity equation. ....	179
Equation 5.4 Matthew's Correlation Coefficient Equation. ....	179

# 1 Introduction

## *1.1 Proteins*

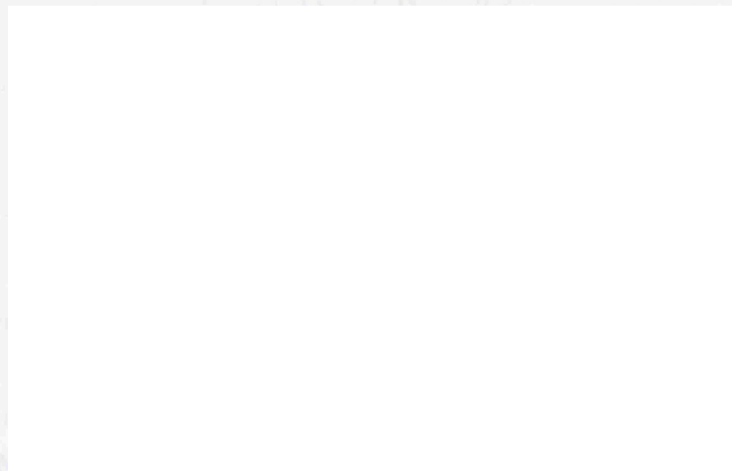
Proteins are a class of ubiquitous organic molecules that are vital for nearly every biological process in nature. They exhibit a diverse array of functions including enzyme catalysis, providing mechanical support within and between cells and also the control of gene expression and signalling pathways that dictate the vast amount of processes observed in organisms. The role or function of a particular protein is closely linked to the three dimensional structure, through the formation of clefts to provide the optimal environment to bind an enzyme substrate, surfaces to interact with other proteins or the particular orientation of secondary structures to bind DNA. Therefore, the elucidation of protein structures is vital in understanding the molecular basis of protein functions and holds great potential in revealing the molecular mechanisms of disease and providing opportunities for rational drug design.

The first three dimensional protein structure solved was that of myoglobin in 1958 by Max Perutz and Sir John Cowdery Kendrew (1958) characterised using X-ray diffraction. Since then a great deal of research has led to the determination of many protein structures using techniques such as X-ray diffraction and Nuclear Magnetic Resonance (NMR). As of June 2007 there are over 44,000 co-ordinates of protein structures deposited in the protein data bank structural database and new technologies developed by the international Structural Genomics Initiatives (SGIs) have permitted this number to increase exponentially (Berman et al. 2000). However, despite the recent advances in protein structure determination the process remains relatively expensive and time consuming compared to DNA sequencing. This is highlighted by the vast number of sequences in the UniProt protein sequence database (Wu et al. 2006), where there are over 4.5 million sequences. It is unlikely with the increase in genomic data that this chasm between sequence and structure data will be bridged and therefore it is of great

importance for the SGI projects to be selective in their target selection procedures to attempt to characterise unelucidated regions of fold space.

### ***1.1.1 Primary Structure***

The primary structure of a protein describes the sequence of amino acids along the polypeptide chain. In general, an amino acid is any molecule that contains both amine (NH<sub>2</sub>) and carboxyl (COOH) groups, however the type of amino acid found in biological systems is the alpha amino acid where the amine and carboxyl groups are attached to the same  $\alpha$ -carbon. There are 20 'standard' amino acids found in nature which differ by a variable side chain (R) also connected to the  $\alpha$ -carbon (Figure 1.1).



**Figure 1.1.** The basic structure of a biological amino acid. With the central  $\alpha$ -carbon connected to an amino group, a carboxyl group and a variable side chain (Branden, Tooze 1999).

All 20 naturally occurring amino acids can be broadly grouped according to their physical and chemical properties (Figure 1.2). Often amino acids are assigned to one of three classes (Branden, Tooze 1999); those with hydrophobic sidechains, those with charged sidechains and those amino acids with polar sidechains. Glycine is an exception to this as its sidechain consists of a single hydrogen atom. Polypeptide chains are

polymerised on the ribosome in a condensation reaction between the carboxyl group of one amino acid and the amino group of the next with the removal of a water molecule.

Figure 1.2. A Venn diagram describing the chemical and physical properties of amino acids (Taylor, 86). The residues are alanine (A), cysteine (C), aspartic acid (D), glutamic acid (E), phenylalanine (F), glycine(G), histidine (H), isoleucine (I), lysine (K), leucine (L), methionine (M), asparagine (N), proline (P), glutamine (Q), arginine (R), serine (S), threonine(T), valine (V), tryptophan (W) and tyrosine (Y).

### ***1.1.2 Secondary Structure***

To satisfy energetic constraints, water soluble, globular proteins are driven to fold into their three dimensional structures by the packing of hydrophobic residues into the core of the structure leaving those amino acids with hydrophilic sidechains exposed to the aqueous environment. The burial of hydrophobic residues is also accompanied by the



burying of the polar amine and carboxyl groups. These polar groups are stabilised by the formation of hydrogen bonds and this is the driving force between secondary structure formations. There are two main types of secondary structures the alpha helix and the beta sheet.

The backbone dihedral angles along the polypeptide chain describe the fold of the protein, and therefore the secondary structures. Dihedral angles are defined as the angle between two planes. The angle  $\phi$  describes the angle involving the residues  $C' - N - C\alpha - C'$  and the  $\psi$  angle involving the backbone residues  $C\alpha - C' - N - C\alpha$ .

An alpha helix can be described as a right handed coiled structure. Its formation is stabilised by the hydrogen bonding of every amine group to the backbone carboxyl group of the amino acid four residues earlier in the helix. Each amino acid contributes a  $100^\circ$  turn in the helix which corresponds to one full turn of the helix being equivalent to 3.6 residues. Residues in alpha helices adopt on average backbone dihedral angles of  $-60^\circ$  for the  $\phi$  angle and  $-50^\circ$  for the  $\psi$  angle.

Beta sheets are formed by three or more beta strands. These strands form hydrogen bonds between  $C=O$  and  $N-H$  groups on adjacent strands culminating in the formation of planar sheets. The direction of strands that make up a beta sheet can be parallel or anti-parallel. Parallel beta sheets have an average  $\phi$  angle of  $-119^\circ$  and an average  $\psi$  angle of  $113^\circ$ , while anti-parallel exhibit an average  $\phi$  angle of  $-139^\circ$  and an average  $\psi$  angle of  $135^\circ$ .

Although alpha helices and beta sheets represent the most common secondary structures there are other less stable secondary structures that are observed less frequently. One example is the  $3_{10}$  helix. These helices are always short, no longer than a couple of turns, and found almost exclusively at the end of regular alpha helices. The internal hydrogen bonding is between the amine group of residue  $i$  and the carboxyl group of  $i+3$  (as apposed to  $i+4$  for regular alpha helices) and results in the dipole being less well aligned and therefore a less stable, and hence rarer structure. An even rarer and more unstable

helix formation is the  $\pi$ -helix where the hydrogen bonding is formed between the amine group of residue  $i$  and the carboxyl group of  $i+5$ . Finally the beta-turn is a more common secondary structure element arising when the protein chain turns back upon itself. It is stabilised by an intramolecular hydrogen bond between a proline residue side and its main chain nitrogen atom. Such regions are also often glycine-rich, which confers little steric hindrance and promotes flexibility in the protein chains.

### ***1.1.3 Super-secondary structure***

Secondary structures adjacent to one another can assemble into regular motifs known as super secondary structures. These motifs often form the building blocks of larger structural assemblies or can be associated with a specific functional role as in the case of the helix-loop-helix super-secondary structure associated with DNA-binding proteins.

Beta-hairpins are one of the simplest motifs comprising two small anti-parallel strands joined by a loop region. This motif is often observed as part of a more complex beta sheet or occasionally in isolation from other secondary structure elements. The conformation of beta hairpins is dependent on the length and sequence of the composite strands and studies have shown that 70% of beta-hairpins are less than 7 residues in length with two-residue turns forming the most frequent conformation (Sibanda, Thornton 1985). Consecutive anti-parallel beta-hairpins form a super-secondary structure known as a beta-meander.

Super-secondary structure arrangements that are involved in a more specific functional role include the helix-turn-helix motif and the helix-loop-helix motif. The helix-turn-helix motif, also referred to as the EF hand, is often implicated in calcium binding with the carboxyl sidechains and main chain carbonyl groups mediating the interaction. This was first observed in parvalbumin. The helix-loop-helix motif is regularly reused in a variety of folds that bind DNA and was first observed in prokaryotic DNA binding proteins such as the cro repressor from phage lambda. The cro repressor forms a dimer

with each subunit consisting of an anti parallel, three stranded beta-sheet with helical elements inserted between the first and second strands. On dimerisation, the second helices from each subunit are located on one side of the beta sheet and their orientation allows them to fit into adjacent major grooves of the DNA (Luscombe et al. 2000).

#### ***1.1.4 Tertiary Structure & Protein Domains***

The tertiary structure of a protein describes the overall 3D conformation adopted by the protein chain. These complex formations are stabilised by a combination of electrostatic, Van de Waals forces and covalent disulfide bonds. The tertiary structure comprises one or more globular domains, which according to Richardson (1981) form semi-independent folding units, with a well packed hydrophobic core. Secondary structure elements are rarely shared between domains (Taylor 1999) and this leads to an increase in sequence variability in the loop regions between domains, since they do not impact on the overall fold of the protein. The domain is considered to be the primary evolutionary unit and domains are often observed with a variety of other domain partners in different proteins.

#### ***1.1.5 Quaternary Structure***

Some proteins are composed of more than one polypeptide chain and the description of this complex can be termed the quaternary structure. Individual chains associate through electrostatic and covalent interactions to form larger oligomeric formations. These complexes may be transient associations that further increase the functional repertoire of proteins and facilitate regulatory networks. In addition, new active sites can also form at the interfaces of protein chains (Liu, Eisenberg 2002).

## ***1.2 Protein Evolution***

Evolution through a process of random mutation and natural selection has given rise to the dramatic diversity of species observed in nature. On a molecular level, it is the recombination and mutation of DNA that generates this diversity and mutations in genes manifest themselves as changes in the amino acid sequences they encode. These changes accumulate over time and are the mechanism by which proteins can evolve new functions.

Proteins related by evolution, therefore descending from a common ancestral protein, are termed homologues. As homologues have evolved through divergent evolution from a common ancestor, close relatives will often have similar sequences. However, as relationships become more distant the primary sequence changes as the two homologues have mutated independently from the ancestral gene even if their function remains similar. With some homologous proteins the evolutionary relationship is distant enough that there is no significant sequence similarity and only structural similarity remains, these are often termed remote homologues. Homologous proteins that arise through speciation events are termed orthologues and frequently share the same function.

An alternative mode of protein evolution is gene duplication and homologues that arise through such events are termed paralogues. The new copy of the gene is not subjected to the same evolutionary constraints as the parent gene and this can potentially enable it to exploit a different functional niche. New functions can arise through amino acid mutations of key active site residues or through modifications of functionally important regions of the protein structure. Once the constraints of function have been removed, paralogues can often diverge in sequence beyond the limits of current detection.

Two protein domains that share similar three dimensional structures cannot automatically be considered homologous. Due to a limited number of ways that  $\alpha$ -helices and  $\beta$ -sheets can pack three dimensionally there may also be convergence of evolutionary unrelated

proteins to adopt similar folds (Chothia 1992; Orengo et al. 1994). Two proteins with similar structures but no evidence of an evolutionary relationship can be described as analogues. An interesting example of structural convergence is between the analogous proteins thermolysin and mitochondrial processing peptidase (Makarova, Grishin 1999). These proteins have a very similar arrangement and packing of secondary structure elements and they also show striking similarity in their active site residues. The connectivity of the structure however is completely different making it very unlikely they have evolved from a common ancestor.

It is often difficult to make the distinction between remote homologues and analogues. Very remote homologues often have no significant sequence homology detectable by current comparison methods. Furthermore, in some families of homologous proteins significant structural divergence has been observed, even up to the extent that the fold has changed (Grishin 2001; Krishna, Grishin 2005). Therefore, with the difficulty in separating remote homologues and analogues, it is of little surprise that as methods to detect homologous relationships improve, structures previously classified as analogues are subsequently identified as very distant homologues. In fact it could be said that the definition of what constitutes a remote homologue is intrinsically linked to the ability of methods to identify them. Examples include the  $\alpha/\beta$  TIM barrels, some of which have very limited sequence similarity but similar structures and therefore previously thought of as products of convergent evolution, but now increasingly shown to be descended from a common ancestor (Copley, Bork 2000).

### ***1.3 Detection of Homology***

The characterisation of proteins, be it by their primary sequence or their three dimensional structure, has been accelerated by high throughput techniques. This has led to a massive expansion in both the protein sequence and structure repositories, although there are still an order of magnitude fewer structures than sequences. However, this plethora of new protein information is frequently poorly annotated in terms of function.

Currently one of the most common and easily accessible methods for obtaining functional information for a new sequence or structure is to identify an experimentally characterised homologue and protein family resources (eg Pfam (Finn et al. 2007)) are being increasingly used to do this.

### ***1.3.1 Sequence Comparison Methods***

Sequence similarity is often the most useful indicator of homology. The traditional way to compare two sequences is to align them against one another and compute the residue similarity. For two sequences that differ by a couple of point mutations this is a fairly trivial task, but for more distant relatives that exhibit extensive residue insertions and deletions (indels), this becomes more problematic. Alignment methods tend to be optimised to produce either local or global alignments. Global alignments optimise equivalence across the entire length of the proteins being compared, whilst local alignment methods seek local patterns of sequence similarity.

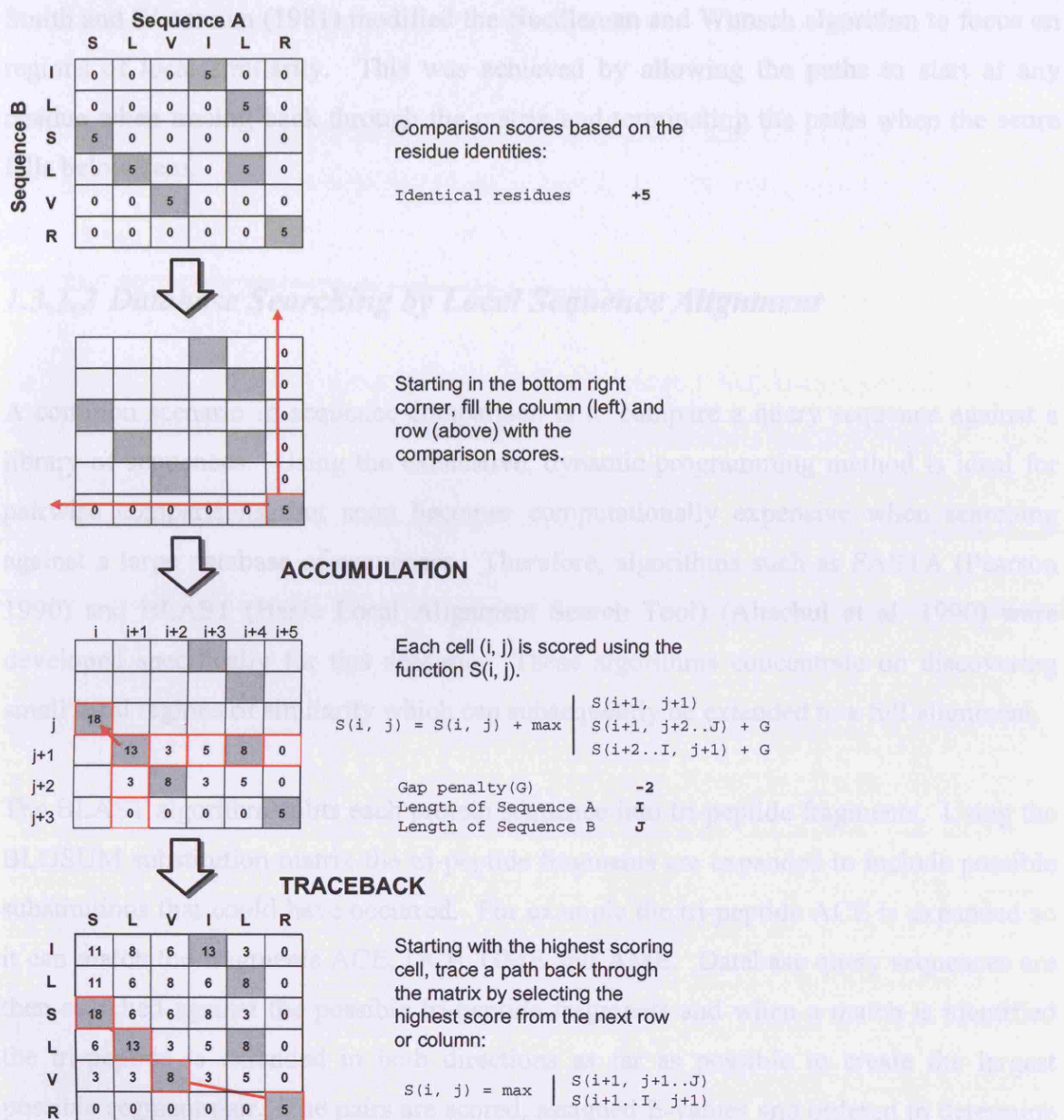
Sequence alignment methods make use of substitution matrices, which calculate the probability of a specific residue mutating into another residue over evolution. As mentioned in Section 1.1.1, amino acids can be classified into three types based on their physicochemical properties; for example those with hydrophobic sidechains, those with charged sidechains and those amino acids with polar sidechains. It can be assumed that the substitution of one residue for another with similar properties is more likely to be tolerated. For example, a mutation from valine to leucine is likely to have minimal effect on the proteins stability and function as both residues have similar hydrophobic properties and molecular size. Such information can be encoded into a residue substitution matrix to guide the alignment.

An alternative approach is to assess the evolutionary probability of specific residue mutations and was pioneered by Dayhoff and co-workers (Dayhoff 1978). They used a database of protein families to generate alignments of close evolutionary relatives (>85%

sequence identity). By examining a large number of alignments the probabilities of mutations between all 20 amino acids can be calculated and used to inform the scoring of an alignment. This method was further extended by Henikoff and co-workers (Henikoff, Henikoff 1991) who created the BLOSUM family matrices from regions of locally aligned sequences in the BLOCKS database at a variety of sequence thresholds. Proteins with a sequence identity greater than a given threshold are clustered together. Substitution values are calculated and used to populate a matrix, representing different evolutionary distances (e.g. BLOSUM50 clusters sequences at 50% identity). These matrices have been shown to be more effective in searching for homologous relationships than PAM matrices (Henikoff et al. 1993).

#### ***1.3.1.1 Global Alignment***

Needleman and Wunsch employed a computational technique called dynamic programming to optimally align two pairs of protein sequences (Needleman, Wunsch 1970). The algorithm finds the optimal alignment by considering every possible combination of residues, including potential indels. The method begins by constructing a two dimensional matrix that reflects the similarity of all residues in protein A with those in protein B. This matrix is traversed from the bottom right corner to populate the accumulation matrix using the scoring function as depicted in Figure 1.3. The value of the scoring function  $S(i,j)$  is determined by the values of previous cells below and to the right. If the diagonal value  $S(i+1,j+1)$  is not selected it means a gap is inserted in the alignment and this incurs a gap penalty to penalise this indel. Once the accumulation matrix is completely populated the optimal alignment is found by tracing back through the matrix to determine the highest scoring path. Figure 1.3 summarises the dynamic programming as implemented by Needleman and Wunsch.



**Figure 1.3.** The Needleman and Wunsch dynamic programming algorithm. Each residue in sequence A and B is scored for similarity and these scores are used to populate a matrix. The accumulation step populates another matrix using the function  $S(i,j)$ , where gaps are penalised. The final traceback step looks for the highest scoring path.



Smith and Waterman (1981) modified the Needleman and Wunsch algorithm to focus on regions of local similarity. This was achieved by allowing the paths to start at any residue when tracing back through the matrix and terminating the paths when the score falls below zero.

### ***1.3.1.2 Database Searching by Local Sequence Alignment***

A common scenario in sequence comparison is to compare a query sequence against a library of sequences. Using the exhaustive, dynamic programming method is ideal for pairwise comparisons, but soon becomes computationally expensive when searching against a large database of sequences. Therefore, algorithms such as FASTA (Pearson 1990) and BLAST (Basic Local Alignment Search Tool) (Altschul et al. 1990) were developed specifically for this scenario. These algorithms concentrate on discovering small local regions of similarity which can subsequently be extended to a full alignment.

The BLAST algorithm splits each protein sequence into tri-peptide fragments. Using the BLOSUM substitution matrix the tri-peptide fragments are expanded to include possible substitutions that could have occurred. For example the tri-peptide ACE is expanded so it can match the fragments ACE, GCE, GME and AME. Database query sequences are then searched against the possible tri-peptide fragments and when a match is identified the tri-peptide is extended in both directions as far as possible to create the largest possible segment pair. The pairs are scored, assigned E-values and ordered to determine the highest scoring segment pair (HSP) for each sequence in the database. The blast algorithm was extended to account for gaps in the alignment, whereby high scoring segments in close proximity are linked together using dynamic programming to obtain the final alignment (Altschul et al. 1997).

### ***1.3.1.3 Profile Based Sequence Comparison***

Although sequence comparison methods such as BLAST and FASTA can detect most homologous relationships between close homologues (sequence identity greater than 30%), when the relationship is more remote (sequence identity below 30%) only half of the homologues can currently be identified (Brenner et al. 1998).

To address this problem, methods have been developed that match residue features conserved during evolution. These features are identified by examining multiple sequence alignments of related protein sequences and the variation of observed amino acids at each position can be modelled in a sequence 'profile'. A profile can be defined as a 'consensus primary structure model consisting of position-specific information' (Eddy 1996). Significance can be assigned to each alignment position based on its conservation in the protein family. This is useful as not all residues in a protein are of equal evolutionary importance, for example, those that are involved in the function of the protein or stabilise the three dimensional fold are subject to greater evolutionary conservation. Profile methods exploit the extra evolutionary information provided by a well-aligned set of homologues. Often there are positions in an alignment of homologues where the amino acids are highly conserved and putative homologues are likely to have the same amino acid conserved at this position. Other positions in the alignment may be more variable and thus the score for a putative homologue should not be greatly affected by variation at these positions. It has been found that methods using multiple sequences detect three times as many remote homologues as pairwise methods (Park et al. 1998).

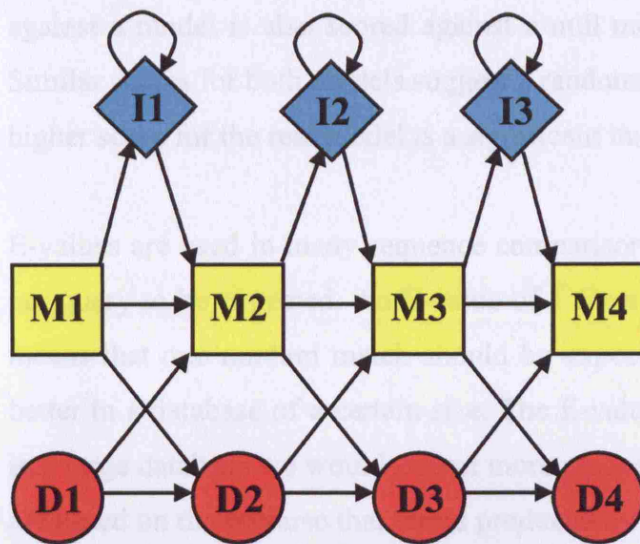
#### ***1.3.1.3.1 PSI-BLAST***

PSI-BLAST (Altschul et al. 1997) is a profile based method that is an extension of the BLAST algorithm described above. PSI-BLAST uses an iterative approach to refine a profile of the original query sequence. The profiles generated by PSI-BLAST are called Position Specific Score Matrix (PSSM) as they combine positional information with

residue exchange probabilities to generate a sequence ‘fingerprint’. The first step of the algorithm is a BLAST database search to identify close relatives from which the PSSM can be generated. These sequences are aligned and a PSSM generated from the residue propensities at each position in the multiple alignment. Subsequently instead of searching with a single sequence, the database is now searched with the PSSM and this enables the identification of more distant homologues. After each iteration the multiple alignment is rebuilt by including the newly identified sequences and the PSSM regenerated. PSI-BLAST iterates through this process until no more relatives can be found below a given E-value cut-off or a specified number of iterations has been reached.

#### ***1.3.1.3.2 HMMs***

Hidden Markov Models (HMMs) are an alternative type of profile which has been shown to outperform PSI-BLAST for recognising very remote homologous relationships (Park et al. 1998; Eddy 1996; Karplus et al. 2005; Park et al. 1998). HMMs can be considered a more formal approach to the profile methodology with the key incorporation of position-specific gap penalties. HMMs implement a statistical framework which is based on state-transition probabilities in a multiple sequence alignment. Each column in the multiple sequence alignment can be characterised by three states; match, delete and insert. The match state models the distribution of residues allowed at a particular position in the alignment, the delete state models having no residue at this position and the insert state models the insertion of one or more residues after this position (Figure 1.4). These states are connected by state-transition probabilities and a sequence of states is generated by moving from the start to end point according to these probabilities. A trained model can be used to emit sequences based on the parameters it has acquired from the domain family and also to assess the likelihood that some sequence has been emitted by the model.



**Figure 1.4. Overview of Hidden Markov Model (HMM), showing transition probabilities between match (M), delete (D) and insert (I) states.**

The two most common HMM methods used for sequence comparison are HMMer (Eddy 1996) and SAM-T (Karplus et al. 2005). SAM-T builds a HMM from either a single seed sequence or a seed alignment using a large sequence database. After the initial scan of the database, a model is generated from the alignment of related sequences, which can then be used for further database scans.

HMMs representing protein domains are often used to identify domains within sequences of unknown structure and function. In order to achieve this, a score is required which represents how well the sequence matches to each model. This can be achieved using either the Viterbi or Baum-Welch algorithms. Viterbi calculates the most probable path of a sequence through the model, whereas Baum-Welch calculates the sum of the probabilities of all possible paths through the model.

Null models are used in scoring sequences with HMMs in order to account for the fact that some sequences have a composition which is close to the background frequency. In such cases a sequence is scored highly by finding a path through an unrelated model due to the background frequencies assigned to the emitting states. Each sequence scored

against a model is also scored against a null model, which represents a random match. Similar scores for both models suggest a random match to the real model. A significantly higher score for the real model is a significant match.

E-values are used in many sequence comparison methods and give the number of errors per query to be expected. An E-value of 1 for a match between a model and a sequence means that one random match should be expected among sequences with that score or better in a database of a certain size. The E-value is dependent on database size because in a large database we would expect more random matches than in a small one. E-values are based on the premise that errors produced by sequence comparison methods follow an Extreme Value Distribution (EVD).

#### ***1.3.1.3.3 Profile-Profile Methods***

A further enhancement in sequence comparison has been the advent of profile-profile sequence comparison methods. Soding (2005) has shown that by constructing a profile of a query sequence as well as the database library of profiles, more homologous relationships can be identified. This is because both sides of the comparison contain information on the evolutionary variation of the sequences.

The profile-profile method COMPASS (Sadreyev, Grishin 2003) adapts PSI-BLAST's scoring system and E-value calculation for profile-profile comparisons. Alternatively *prof\_sim* (Yona, Levitt 2002) uses an entropy based method to measure the similarity between profiles.

A more recent addition to profile-profile methods is HMM-HMM comparisons. HHSearch (Soding 2005) and PRC (Madera 2006) are the most widely used HMM-HMM methods. Such methods align and score two HMMs on the basis of their joint emission probability. That is do they score the same sequence highly? PRC approaches this by aligning the domain family HMMs in the form of a pair HMM, and allowing a score to be

derived using the Viterbi algorithm. Each state of the pair HMM corresponds to pairs of domain family HMM states (matches M, inserts I and deletes D), and a transition in the pair HMM models simultaneous transitions in both domain family HMMs.

### ***1.3.2 Structural Comparison Methods***

As homologous proteins diverge their sequences can change beyond the detection limits of sequence comparison methods. However, Chothia and Lesk (1986) demonstrated that even when there is no discernable sequence similarity the three-dimensional structures remain similar. More recent analysis of several hundred well populated homologous superfamilies in the CATH database showed that even in very remote homologues (<20% sequence identity) at least 50% of the structure remains conserved (Chapter 2 and (Reeves et al. 2006)). Therefore, the alignment of protein structures provides an important tool for identifying remote homologous relationships.

As with sequence comparison, the alignment of protein structures is determined in two stages. Firstly the pairwise similarity between residues, or secondary structure elements is calculated between the two proteins. This is followed by an optimisation strategy to find an alignment that maximises the score of the aligned positions. The majority of methods compare the geometric properties of  $C_\alpha$  or  $C_\beta$  atoms and/or secondary structure information using distances or intramolecular vectors.

#### ***1.3.2.1 Calculating Structural Similarity***

Irrespective of the method used for aligning two protein structures, a way of quantifying the structural similarity of the proteins is required. The most widely used measure of structural similarity is the Root Mean Square Deviation (RMSD). Once equivalent residues in two protein structures have been defined by the alignment, a transformation matrix can be calculated to superimpose them in the same co-ordinate framework to

minimise the RMSD. RMSD is the square root of the average squared distances between equivalent positions (e.g. equivalent C<sub>α</sub> atoms) (Equation 1.1) so that proteins with similar three dimensional structures tend to have low RMSDs (<4.0 Å).

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

**Equation 1.1 Root Mean Square Deviation (RMSD).**

Using an RMSD value alone can be misleading for detecting homologues. It is known that RMSD values do not depend only on conformational differences but also for instance the sizes of the structures being compared. It is for this reason it is also important to consider the number of equivalent residues over which the RMSD has been calculated. Furthermore, higher RMSDs values are found when comparing two protein structures of high crystallographic resolution than observed when comparing two structures both at low resolution (Carugo 2003).

### ***1.3.2.2 Secondary Structure Based Structure Comparison Methods***

One approach to comparing protein structures is to assess the similarity in the composition and three dimensional orientations of their secondary structures. As the number of secondary structures is often an order of magnitude less than the number of amino acids, such an approach provides a fast and effective way of searching a database of structures to identify putative fold matches. Furthermore, as most amino acid mutations occur in the loop regions of proteins (Branden, Tooze 1999) secondary structure matching algorithms are effective for detecting fold similarities between remote homologues where significant indels have occurred.

Grindley and co-workers (1993) pioneered the use of graph theory to compare the secondary structure arrangements between two protein structures. A mathematical graph is a two dimensional set of points, termed nodes, connected by edges that describe the relationship between them.

More recently Harrison and co-workers (2002) developed the algorithm GRATH as a way of rapidly detecting fold similarities to classify structures in the CATH database. In this, a three dimensional protein structure is represented as a two dimensional, undirected, fully connected graph. Each node of the graph represents a secondary structure vector. The edges of the graph are labelled with the types of secondary structures they are connected to, the distance of closest approach between the vectors, the dot-product angle between the two secondary structures and the dihedral angle defined from the two vectors and their midpoint vectors. Two protein graphs can be compared to detect common secondary structure 'cliques' by identifying equivalent edges that are labelled with similar distances and angles. The GRATH algorithm derives a statistical measure (E-value) that accounts for protein size when identifying significant fold matches.

## **SSM**

SSM (Secondary structure matching) (Krissinel, Henrick 2004) uses a similar approach to GRATH. The algorithm labels edges between nodes with distances and angles to determine equivalent secondary structures with which to guide a rigid body transformation. SSM then iteratively finds corresponding  $C_{\alpha}$  atoms, one from each structure and uses these to guide an optimal superimposition. The major difference between SSM and GRATH is that GRATH seeks fully connected cliques whereas SSM uses sub-graph matching. However, SSM also takes into account connectivity and the size of the secondary structures matched.

Because there are usually an order of magnitude fewer secondary structures in a protein than residues, secondary structure matching methods are extremely fast at searching



databases of folds and often used to identify likely fold matches that can be more accurately aligned using residue based methods.

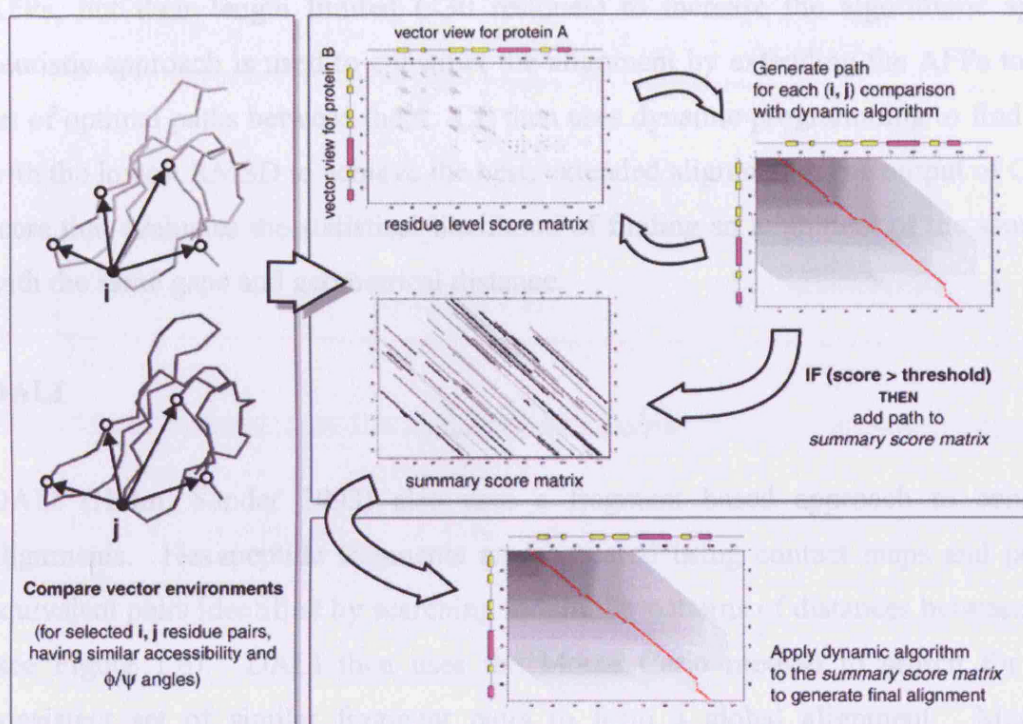
### ***1.3.2.3 Residue Based Structure Comparison***

The common goal of residue-based structure comparison methods is to identify residue pairs between two proteins that are structurally similar. There are two general strategies: (1) directly search for a good alignment, and (2) search for a transformation that optimally positions the two structures with respect to one another, and then use the transformation to find the best alignment. SSAP, DALI and CE belong to the first group whereas STRUCTIONAL and LSQMAN belong to the second.

#### **SSAP**

SSAP (Taylor, Orengo 1989) searches for an optimal alignment of two protein structures using dynamic programming. The dynamic programming algorithm, as described previously in Section 1.3.1.1, first scores the similarity between all residue pairs. This is achieved by comparing the residue ‘views’. Residue views are described by vectors between a specific C $\beta$  atom and all other C $\beta$  atoms within a structure and two residue views are scored for similarity. The number of residues compared is limited by selecting on secondary structure properties (e.g. accessibility, phi and psi angles). The SSAP algorithm utilises dynamic programming in two stages, termed Double Dynamic Programming (DDP). A ‘residue-level score matrix’ is constructed for each pair of putatively equivalent residues, containing scores that reflect the similarity of a given pair of vectors. The first level of dynamic programming is used to find the highest scoring path through these matrices. The second step is to amalgamate the information from the residue level matrices into a summary score matrix. Pairs of residues are determined to be potentially equivalent based on the score of the best path through their residue level matrix. All optimal paths returning scores above a given threshold are collated in the summary matrix (see Figure 1.5). The final level of dynamic programming is used to

find the overall best path or alignment through the summary score matrix. The native score of SSAP is a normalised logarithm of a measure which combines the similarity of the aligned residue views and the number of residues in the larger protein.



**Figure 1.5.** Flowchart of the SSAP algorithm. Vector environments are compared between pairs of potentially equivalent residues in each protein. A residue level score matrix is constructed for each pair and optimal paths (putative alignments) are calculated by dynamic programming. High scoring paths are then added to the summary score matrix. Dynamic programming is then applied to the summary matrix to generate the final optimal alignment of the two structures.


## CE

Another approach for comparing protein structures, is to split the structures into fragments of peptides, then find equivalent fragments and combine these through some optimisation protocol to give the final alignment.

CE (Shindyalov, Bourne 1998) constructs alignments by successively joining well aligned fragment pairs (AFPs). AFPs are defined as pairs of eight-residue fragments, which are identified as similar if their corresponding local geometry (defined by  $C_{\alpha}$  positions) is within a similarity threshold. Gaps are permitted between neighbouring AFPs, but their length limited (<30 residues) to increase the algorithmic speed. A heuristic approach is used to construct the alignment by extending the AFPs to define a set of optimal paths between them. CE then uses dynamic programming to find the pairs with the lowest RMSD to achieve the best, extended alignment. The output of CE is a Z-score that evaluates the statistical likelihood of finding an alignment of the same length, with the same gaps and geometrical distance.

## **DALI**

DALI (Holm, Sander 1993) also uses a fragment based approach to construct its alignments. Hexapeptide fragments are compared using contact maps and potentially equivalent pairs identified by searching for similar patterns of distances between residues (see Figure 1.6). DALI then uses the Monte Carlo method to search for the best consistent set of similar fragment pairs to form a global alignment. Many initial alignments are searched in parallel, to identify the optimum. DALI outputs a raw score and a normalised Z-score.



**Figure 1.6. The DALI method of Holm and Sander (1993). Proteins are fragmented into hexapeptides and their contact maps compared to find equivalent fragments. Fragments are concatenated and their RMSD checked to find valid extensions. Monte Carlo optimisation is used to guide the extension process to a full alignment.**

## **STRUCTAL**

STRUCTAL (Subbiah et al. 1993) assumes a series of initial alignments based on a correspondence of residues in the two structures, and uses a rigid-body transformation to superimpose the corresponding residues. It then finds the optimal alignment for the superposition. This is the beginning of an iterative process whereby the new alignment is used to superimpose the structures again to generate a new alignment to guide the next superposition until it converges on a local optimum. The local optimum, however, is dependent on the initial alignment so a variety of initial residue correspondences are used to increase the likelihood of finding the global optimum. Three of the initial correspondences include aligning the beginnings, the end and the mid-point of the structures without allowing gaps. Other correspondences include maximising the sequence identity or the  $C_{\alpha}$  torsion angle similarity. For a given correspondence, the

optimal transformation is the one with the minimum RMSD and for a given transformation, the optimal alignment is the one with the maximal STRUCTAL score obtained through dynamic programming. STRUCTAL provides a statistical measure of significance of the final alignment in the form of a p-value.

## **LSQMAN**

LSQMAN (Kleywegt 1996), like STRUCTAL, searches iteratively for a rigid body transformation that optimally superimposes the two structures. The initial transformation is found by superimposing the first residue of each secondary structure element of the two structures. Once superimposed, LSQMAN starts by searching for a long aligned fragment pair, where matching residues are geometrically close ( $< 6\text{\AA}$ ) and the minimum fragment pair is not less than 4 residues. Given the alignment, an optimal transformation is calculated and a new iteration is started. The distance cut-off is increased during each iteration to extend the alignment further. Optimisation of the alignment is further guided by a similarity index. LSQMAN outputs a Z-Score to give a statistical interpretation of the alignments significance.

## ***1.4 Protein Structure Family Resources***

Since many remote evolutionary relationships can only be recognised by structure comparison, structure based protein family classifications are particularly valuable for understanding evolution.

Two of the most comprehensive structural classification, CATH (Orengo et al. 1997;Greene et al. 2006) and SCOP (Murzin et al. 1995;Hubbard et al. 1997) are hierarchical and organise proteins on the basis of both structural similarity and evolutionary relationships. With the exception of SCOP, all the databases here use an automated method for protein structure comparison at some point in the classification procedure.

### ***1.4.1 CATH database***

The CATH database has four main levels of classification (see Figure 1.7). The Class (C) level describes whether the protein domain is constituted of mainly alpha helices, beta strands or a combination of the two. Architecture (A) describes the orientation of these secondary structure elements in 3D space. The Topology (T) or fold level describes the connectivity of these secondary structures, whereas the homologous superfamily (H) level clusters proteins that have a clear evolutionary relationship. For example domain one of lactate dehydrogenase has the Class 'Alpha-Beta', the Architecture '3-Layer Alpha-Beta-Alpha Sandwich', the Topology 'Rossmann' and is a member of the 'NAD(P)-binding Rossmann-like domain' Superfamily.

The CATH resource uses a combination of manual and automated approaches. Robust structure comparison methods (SSAP, GRATH) are used to recognise structural relatives, and then evolutionary relationships are assigned if two of the following criteria are satisfied; (1) high sequence similarity (>35% sequence identity, or a significant E-value based on profile methods), (2) high structural similarity score (>80 measured by global structural comparison methods such as SSAP), (3) evidence of functional similarity (e.g. sharing of first 3 E.C. numbers).

The latest version of CATH, Version 3.1.0 released on January 2007 contains 93,885 domains clustered into 7794 homologous superfamilies, 2091 fold groups, 1084 architectures and 40 classes. Homologous superfamilies are further clustered based on sequence identity (e.g. >35%). Domains from the same sequence family (S35) generally share very high structural and functional similarity. Representative datasets of CATH usually comprise one representative from each S35 family, these are termed S35Reps.

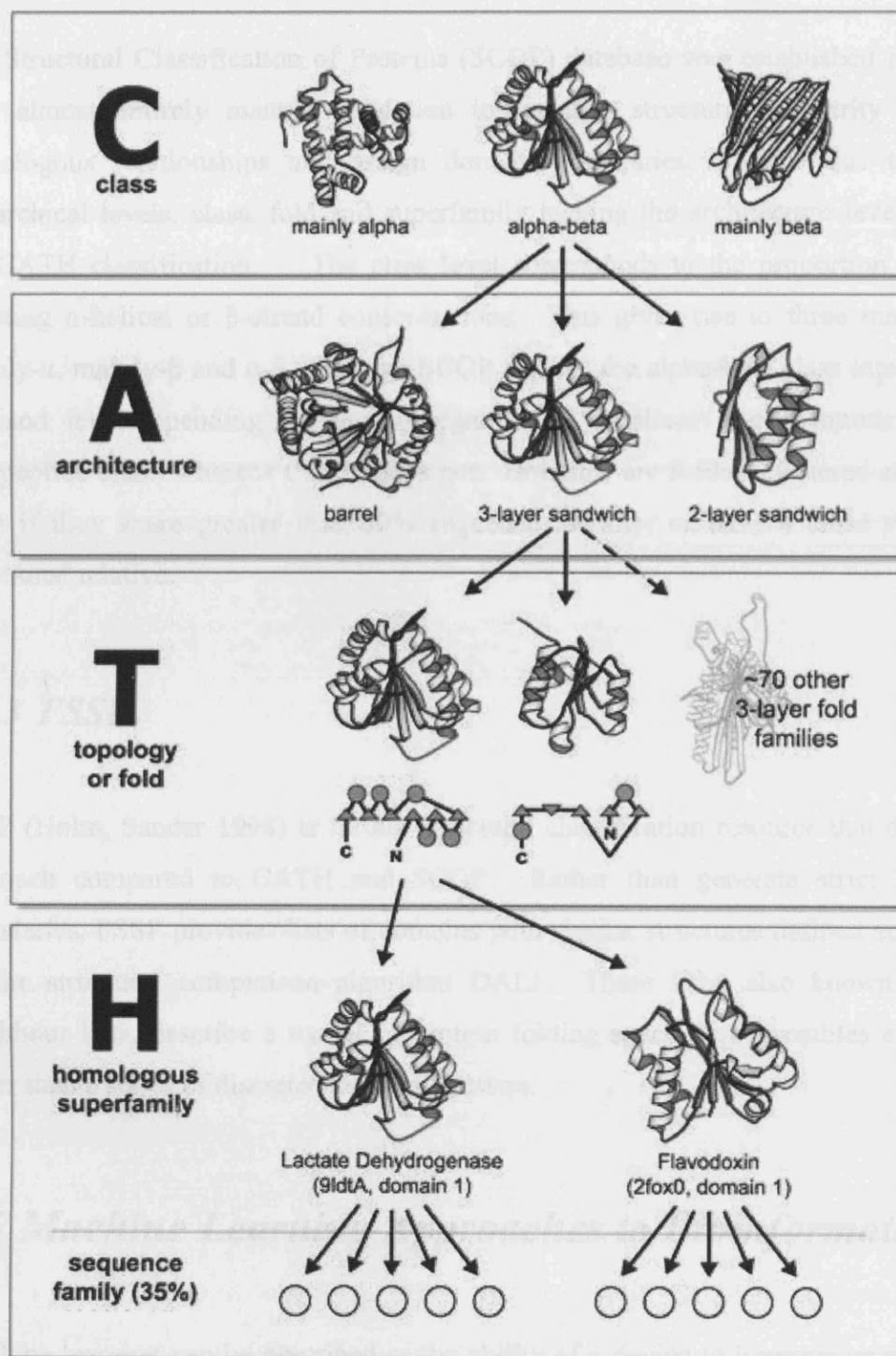


Figure 1.7. Diagram of the CATH hierarchy.

### ***1.4.2 SCOP***

The Structural Classification of Proteins (SCOP) database was established in 1995 and uses almost entirely manual validation to quantify structural similarity and define homologous relationships and assign domain boundaries. SCOP has three major hierarchical levels, class, fold and superfamily lacking the architecture level present in the CATH classification. The class level corresponds to the proportion of residues adopting  $\alpha$ -helical or  $\beta$ -strand conformations. This gives rise to three major classes, mainly- $\alpha$ , mainly- $\beta$  and  $\alpha$ - $\beta$ , although SCOP divides the alpha-beta class into alternating  $\alpha/\beta$  and  $\alpha+\beta$  depending on the segregation of  $\alpha$ -helices and  $\beta$ -strands along the polypeptide chain whereas CATH does not. Domains are further clustered at the family level if they share greater than 30% sequence identity, or have a close structural or functional relative.

### ***1.4.3 FSSP***

FSSP (Holm, Sander 1998) is further structural classification resource that differs in its approach compared to CATH and SCOP. Rather than generate strict hierarchical boundaries, FSSP provides lists of domains with similar structures defined automatically by the structural comparison algorithm DALI. These lists, also known as nearest neighbour lists, describe a model of protein folding space that resembles a continuum rather than a series of discrete structural clusters.

## ***1.5 Machine Learning Approaches to Bioinformatics***

Machine learning can be described as the ability of a device to improve its performance based on past performance. Machine learning systems generally use non-linear classification to combine information about existing data presented in training examples to facilitate a prediction on an unseen dataset. The machine learning field can be split



into two classes: *supervised* and *unsupervised* learning and both have been employed in biological applications.

In supervised learning, entities are classified using a set of well defined features with the end result being a classification solely based on the information described by these features. The entities used in the learning process are labelled with a specific class and the machine learning system learns how to combine the features associated with those entities to recreate the classification. The goal of supervised learning is to produce a machine learning system that can accurately predict class membership of new entities based on the available features. For example, in the context of homology recognition the concept is that by exposing the system to examples of protein homologues and non-homologues the 'machine' can learn the rules of homology and therefore predict homology for unknown examples. Examples of machine learning methods that utilise supervised learning are decision trees, artificial neural networks and support vector machines.

By contrast, in unsupervised learning no predefined class labels are associated with the entities. Here, the goal is to explore the data and discover similarities between the entities and then these similarities can then be used to define clusters of similar entities. Unsupervised learning is often utilised in clustering analysis and has found a niche in microarray gene expression analysis (Tarca et al. 2007).

Biological research and the development of the machine learning field have been intricately linked. Many early machine learning techniques were modelled on biological phenomena, particularly the activity of a neuron. In 1957, Rosenblatt (1957) developed the perceptron, which was a simple model of neuronal activity and this itself spawned the field of artificial neural networks. The use of machine learning systems to attempt to answer biological questions was first pioneered with the use of the perceptron to recognise features associated with translation initiation sequences in *E. coli* (Stormo et al. 1982).

In the past 10 years there has been a significant boom in the use of machine learning approaches to a wide range of bioinformatics applications. These include the use of artificial neural networks as well as support vector machines. Of particular interest to this thesis is the use of neural networks to classify homology based on the SCOP protein database (Dietmann, Holm 2001). This approach used neural networks to validate groupings within fold space and the clustering within these fold groups. A forward-feeding neural network was trained on data based on the connectivity of these clusters consisting of 11,907 unrelated pairs (same SCOP fold but different superfamily) and 3,635 related pairs (same SCOP superfamily) from a representative set of single domain PDB structures. The trained classifier was then used to predict 77% of the homologues in the SCOP database with 85% accuracy. A wide variety of other problems have also been tackled through machine learning approaches, including the detection of homology based on local structure information by support vector machines (Hou Y et al. 2003) and the prediction of beta-turns in proteins (Shepherd et al. 1999). Neural networks have also been used to combine the different outputs from threading algorithms to increase performance in fold recognition (McGuffin LJ, Jones DT 2002) and furthermore used to evaluate such predictions (Juan D et al. 2003). One of the most common application in bioinformatics has been the prediction of secondary structure using neural networks (Meiler J, Baker D 2003;Cai YD et al. 2002) and support vector machines (Kim H, Park H 2003;Ward et al. 2003).

Although there are many types of machine learning algorithms the next section describes in detail the two systems used in this thesis: artificial neural networks and support vector machines.

### ***1.5.1 Artificial Neural Networks***

Artificial neural networks were originally developed with the goal of modelling information processing and learning in the brain (Rumelhart et al. 1986). The networks used in machine learning approaches today are quite distinct from the biological networks

of the brain but have proved to be useful applications in a number of fields. In a mathematical sense, neural networks can be viewed as a broad class of parameterised graphical models consisting of networks with interconnected units evolving in time (Baldi, Brunak 2001), each interconnected unit representing the neurones of the brain and connectivity is described by updating weights. Networks can be of varied architecture such as recurrent, feed-forward, and layered. Recurrent networks suggesting the presence of directed loops, feed-forward without loops. Layered architectures are those where units are separated into classes and the connectivity defined between the classes. Most current applications, especially those in the bioinformatics field, use Feed-forward layered neural networks in which the units are often partitioned into visible and hidden units. Visible units being those in contact with the external world, for example input units and output units, such units are often clustered into layers. Hidden layers contain units that do not access the external environment and therefore reside between the input and output units (Figure 1.8.).

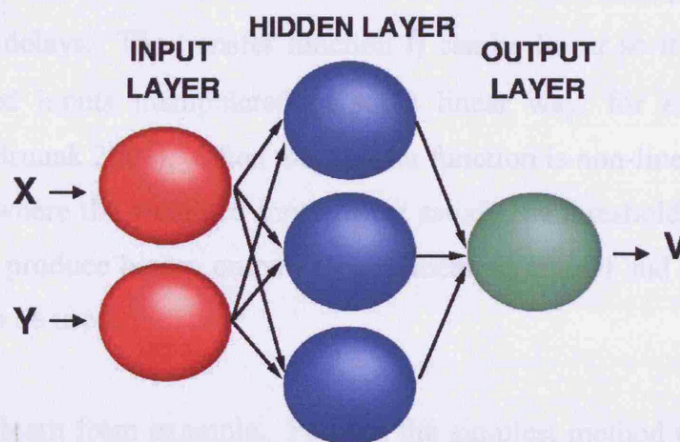


Figure 1.8. Schematic diagram showing an example of the architecture of a feed-forwards layered neural network.  $X$  and  $Y$  represent the input features fed into the network.  $V$  represents the output classification or prediction. Note the connectivity flows in a unidirectional manner from input layer to hidden layer to output layer.

The behaviour of each unit is described by a function, which can be discrete or differentiable. In general a unit  $i$  receives a total input  $x_i$  from the units connected to it which in a feed-forward layer come from the connected units proceeding it in the

network. It then produces an output  $y_i = f_i(x_i)$  where  $f_i$  is the defined transfer function of the unit. Units in the same layer will have the same transfer function and therefore the total input is a weighted sum of all incoming outputs from the previous layer and can be described as;

$$x_i = \sum_{j \in N-(i)} w_{ij} y_j + w_i$$

where the output is;

$$y_i = f_i(x_i) = f_i \left( \sum_{j \in N-(i)} w_{ij} y_j + w_i \right)$$

$w_i$  represents the bias or threshold of a unit so it can be viewed as a connection with weight  $w_i$  to an additional unit. The weights are the adjustable parameters of a neural network which evolve during the learning process. Other adjustable parameters are available depending on the update function chosen, for example time constants, momentums and delays. The transfer function  $f_i$  can be linear so it passes forward the summed weighted inputs manipulated in some linear way, for example an identity function (Baldi, Brunak 2001). Often the transfer function is non-linear and may involve a threshold gate where the weighted inputs must satisfy the threshold set, such functions are often used to produce binary outputs. Non-linear sigmoidal and normalised transfer functions can also be used.

Neural networks learn from example. Perhaps the simplest method to organise data for training the classifier is the “split-sample” method. Using the split-sample method, the data the neural network learns from is split into three equally sized, non-biased partitions in the format of a training set, a test set and a validation set. This data gives the neural network the input values and the associated, appropriate output values. The neural network then processes this data in an iterative fashion, adjusting the weights of the units to minimise the summed distance of the networks output from the output values given in the data sets. A training set and test set are used so that the network learns only from the training set but checks its performance against the test set. This aims to prevent the

network ‘over-fitting’ to the training set, which is important as the network should learn the global rules of the data and not just ‘memorise’ the input values and their corresponding output. Once learning is completed, the validation set is used to estimate the generalisation error of the classifier on a completely unseen data-set. The generalisation error (the average distance of prediction by the classifier compared to the correct answer) provides a measure of the performance of the predictor when evaluating the predictive power of the classifier.

When the data sets are small it is not appropriate to train on just one third of the data set as in the split-sample method as this may lead to sub-optimal learning and overfitting. In these cases ‘resampling’ methods are often used (Weiss 1991). The two most popular resampling methods used to estimate the generalisation error of a classifier is cross-validation and bootstrapping. Cross-validation, often termed **k-fold cross-validation**, divides the data into **k** subsets of equal size. The classifier is trained **k** times, each time leaving out one of the subsets from training and using that subset to compute the error. ‘Leave-one-out’ cross validation is often superior to the ‘split-sample’ method when the data-sets are small because it trains and validates on the whole data-set (Goutte C. 1997).

An alternative resampling method to leave-one-out cross validation is bootstrapping. In its simplest form instead of repeatedly analysing subsets of the data, random sub-samples are used. Each sub-sample is a random sample with replacement from the full dataset. At least 200 iterations for bootstrap estimates are usually necessary to obtain a good estimate of generalisation. Bootstrap has the advantage of being more robust on small data sizes than leave-one-out cross validation but is considerably more computationally expensive (Efron 1982).

Various algorithms and functions exist that are used to update the weights for each iteration through the training data. Such functions are often called ‘learning functions’. Possibly, the most well-used learning algorithm is called backpropagation. With this algorithm an input pattern is presented to the network and is propagated through to the output layer. The output is then compared to the desired output (or teaching output). The

difference (or delta) is then used together with the output of the source unit to compute the changes to the weight. To compute the delta values of inner hidden units for which no teaching output is available, the deltas of the following layer, which have already been computed, are used in the formula below. In this way the errors (deltas) are propagated backwards hence ‘backpropagation’.

The backpropagation weight rule is given below:

$$\Delta w_{ij} = \eta \delta_j o_i$$

$$\delta_j = \begin{cases} f'_j(net_j)(t_j - o_j) & \text{if unit } j \text{ is a output-unit} \\ f'_j(net_j) \sum_k \delta_k w_{jk} & \text{if unit } j \text{ is a hidden-unit} \end{cases}$$

where:

- $\eta$  learning factor eta (a constant)
- $\delta_j$  error (difference between the real output and the teaching input) of unit  $j$
- $t_j$  teaching input of unit  $j$
- $o_i$  output of the preceding unit  $i$
- $i$  index of a predecessor to the current unit  $j$  with link  $w_{ij}$  from  $i$  to  $j$
- $j$  index of the current unit
- $k$  index of a successor to the current unit  $j$  with link  $w_{jk}$  from  $j$  to  $k$

There are different approaches to learning. With online learning the weight changes are applied to the network after each learning pattern, alternatively with batch learning the weight changes are accumulated for all the patterns in the training set and applied after one full cycle of the training set. Simple backpropagation is just one example of a learning function. There are further, more sophisticated, versions of backpropagation that have extended parameters including momentum, time delays, weight decays for example that change the emphasis of how the network learns.

An important aspect of machine learning approaches is feature selection. Features can be described as the individual measurable heuristic variables that provide the inputs to the machine learning algorithm. The selection of optimal features is important for a variety of reasons (Guyon 2003). Most importantly having discriminatory features that aid in the separation of classes in a classification is essential if a successful machine learning

system is going to be created. Furthermore the presence of non-discriminatory features may lead to a poorer learning capacity (Milne 1995). Also by removing features that are non-discriminatory for the classifier the computational cost of learning is reduced. Finally meaningful discriminating features allow a better understanding of the underlying processes that determine the classification.

Many methods can be used to aid feature selection. Ideally an exhaustive search of feature space is applied whereby all combinations of features are used to train the predictor and the best combination identified. However, if your potential feature size is very large, for example the genes of interest in a microarray experiment, this is often not computationally feasible. Feature selection methods often use a ranking approach whereby 'good' informative features will be ranked highly. Ranking methods often presume the data to be normally distributed which is often not the case for bioinformatic data (Al-Shahib et al. 2005).

### ***1.5.2 Support Vector Machines***

As with artificial neural networks, Support Vector Machines (SVMs) are a class of classifier that attempt to distinguish between two classes of entity based on the values of common features. SVMs are underpinned by a statistical learning theory which facilitates the separation of two classes of entities by placing a division (or hyperplane) between them (Vladimir N.Vapnik 1995).

If you consider Figure 1.9 it is apparent that there are a number of different decisions boundaries that can separate the two classes of data. SVMs attempt to define the decision boundary or hyperplane that achieves maximum separation, or "margin" between the two classes. The margin is defined as the distance between the planar decision surface that separates two classes and the closest training samples to the decision surface. To account for the fact that many data-sets can not be completely resolved a user-defined parameter



'C' determines how many data points are allowed to be misclassified without effecting the position of the hyperplane thus accounting for outliers in the data-sets.

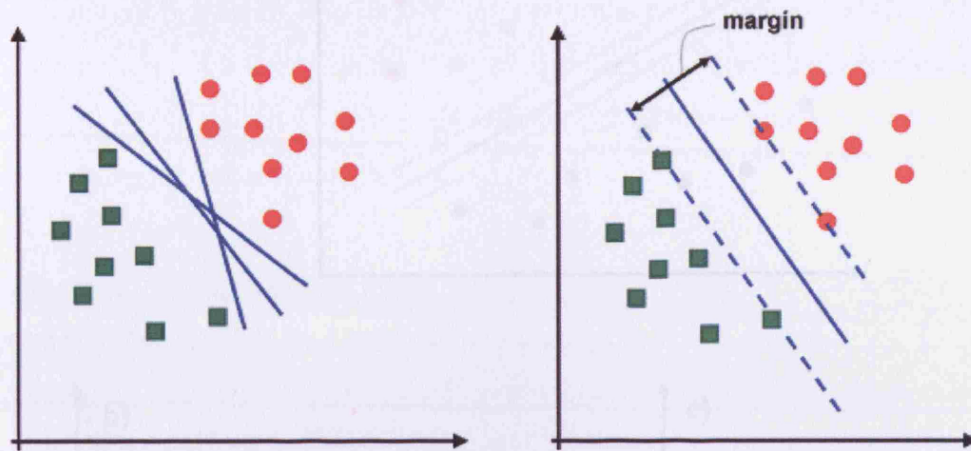


Figure 1.9. Support Vector Machines Class Boundaries. Two-dimensional data points belonging to two different classes (circles and squares) are shown in the left panel. The right panel shows the maximum-margin decision boundary implemented by the SVMs.

Figure 1.9 shows a linear SVM but non-linear SVMs can also be applied should the problem require it. This is achieved by performing a kernel transformation that may allow better classification of the entities in a non-linear higher dimensional feature space. Figure 1.10(b) shows two classes described by two features, one of which does not vary. However, by squaring the variable feature (effectively placing the solution in higher dimensions), it is possible to separate the classes using a linear hyperplane. This approach is referred to as the 'kernel trick'. Figure 1.10d shows a more realistic example where the data points have been transformed into 4 dimensions by the kernel function, producing a non-linear solution in 2 dimensions. Non-linear kernels transform the data into higher dimensions to allow a linear hyperplane to separate the data classes. Two commonly used non-linear kernels include the polynomial kernel and the radial basis function.







## ***1.6 Aims***

The aim of this thesis is to explore the performance of different approaches to homologue recognition, with the ultimate goal of developing a machine learning system that can combine information from the most successful of these methods to predict homology in an automated and accurate fashion.

Machine learning gains its power from the ability to learn from known examples. Within the CATH classification there is vast knowledge base available for a machine learning system to learn the rules of homology. A machine learning approach can, once validated and optimised, provide an automatic classification system. This would be hugely advantageous for the CATH protein database, where regular updating of the resource currently includes a large degree of manual classification which is becoming increasingly difficult due to the high numbers of novel structures being produced by the structural genomic initiatives. Furthermore because of the observed variability of sequence, structure and function conservation within and between superfamilies, a dynamic machine learning system will be able to characterise these patterns more successfully than using empirical thresholds. The development and utilisation of superfamily specific measures of variation and the use of a machine learning approach should hopefully improve the recognition of remote homologues.

Chapter 2 presents an analysis to inform the design and implementation of the homology recognition pipeline, showing how different homologous superfamilies evolve in sequence, structure and function and characterising the mechanisms by which this happens. Furthermore, new information on the variability observed is presented in an established web resource, the Dictionary of Homologous Superfamilies, which has been expanded and improved in a number of ways.

Chapter 3 presents a new structural comparison algorithm, CATHEDRAL, which combines both secondary structure matching and accurate residue alignment in an

iterative protocol for determining the location of previously observed folds in novel multi-domain structures. A rigorous benchmarking protocol is also described that assesses the performance of CATHEDRAL against other leading structural comparison methods.

In Chapter 4 several methods for detecting homology are optimised and benchmarked. These included methods that compare the sequence similarity of proteins, the structural similarity and methods that attempt to assess functional similarity.

Finally Chapter 5 details the implementation, optimisation and benchmarking of a neural network classifier to provide an automated method of homologue recognition.

## **2 Analysis of Sequence, Structure and Functional Variability between Evolutionary Relatives in CATH Superfamilies**

### ***2.1 Background***

In the “post genomic” era a major bioinformatics challenge is the assignment of structural and functional information to the millions of protein sequences determined by the international genomics initiatives. As it is not feasible to determine the structures of all proteins by experimental methods, *in silico* structure and function prediction methods must be improved and this requires a greater understanding of protein evolution.

Proteins evolve through changes in the DNA of the genes encoding them, giving rise to families of homologous relatives. These changes may take the form of single point mutations, residue insertions/deletions or gross, large scale duplications or rearrangements (Heringa, Taylor 1997; Vogel et al. 2004; Vogel et al. 2005). Homologues can be orthologues or paralogues, orthologues having diverged after a speciation event and paralogues arising from a gene duplication event. Mutations in the genes manifest themselves as changes in the amino acid sequence they encode. These changes in protein sequence are accumulated over time and in paralogous proteins these changes may be tolerated if they give rise to new functions beneficial to the organism.

Similarity in sequence, structure and function can give clues for detecting evolutionary relationships between proteins. It has been shown that if the sequence identity is greater than 35% two proteins adopt a similar structure (Chothia, Lesk 1986; Flores et al. 1993). Distant relatives may have diverged from the ancestral sequence to such an extent that there is no longer any detectable sequence similarity. Homologues that have low sequence identity can still have very similar structures, particularly orthologues sharing

similar functions, since structure is frequently more highly conserved than sequence in protein evolution (Chothia, Lesk 1986) (see Figure 2.1).

Function is conserved to various degrees between protein homologues. Studies have shown that proteins that share 60% sequence identity are highly likely to share similar functions whereas for more distantly related proteins sharing less than 30% sequence identity, functional variation is significant (Todd et al. 2001). More recent studies revealed that if the bias in the PDB is taken into account only 50% sequence identity is required (Tian, Skolnick 2003; Rost 2002). Several studies in enzyme families have shown that although functions may change between paralogous relatives, this is usually associated with a change in the substrate on which the enzyme acts and the chemistry performed by the enzyme is often quite well conserved (Todd et al. 2001). Furthermore paralogous proteins are often recruited by different metabolic pathways for their reaction chemistry (Rison, Thornton 2002)

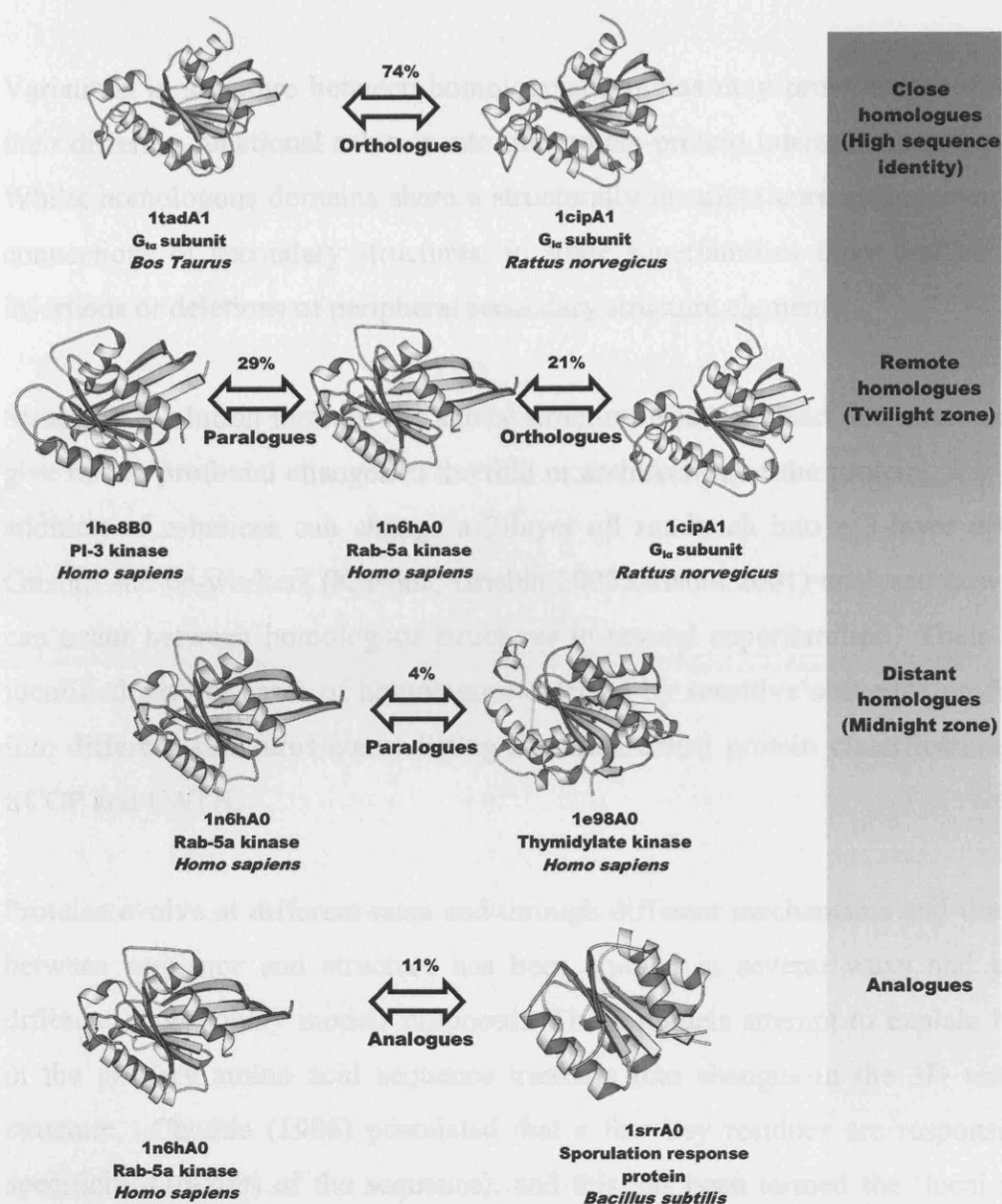


Figure 2.1. Schematic representation of the progression from close homologues, through more remote (twilight zone) and very remote (midnight zone) homologues and finally analogous/homologous structural relatives. The percentages represent the sequence identity between the two protein domains.

Variations in structure between homologous proteins may provide a useful insight into their differing functional roles or into the protein-protein interactions they participate in. Whilst homologous domains share a structurally invariant core arrangement and similar connections of secondary structures, in some superfamilies there can be considerable insertions or deletions of peripheral secondary structure elements.

Structural evolution through secondary structure insertions and deletions can sometimes give rise to profound changes in the fold or architecture of the protein. For example the addition of  $\alpha$ -helices can change a 2-layer  $\alpha\beta$  sandwich into a 3-layer  $\alpha\beta\alpha$  sandwich. Grishin and co-workers (Krishna, Grishin 2005; Grishin 2001) analysed how fold change can occur between homologous structures in several superfamilies. Their studies have identified several cases of homologues detected by sensitive sequence profiles that fold into different structures contradicting the hierarchical protein classification schemes of SCOP and CATH.

Proteins evolve at different rates and through different mechanisms and the relationship between sequence and structure has been studied in several ways and a number of different evolutionary models proposed. These models attempt to explain how changes in the primary amino acid sequence translate into changes in the 3D tertiary protein structure. Chothia (1986) postulated that a few key residues are responsible for fold specificity (10-20% of the sequence), and this has been termed the 'local model'. An alternative 'global model', first proposed by Lattman and Rose (1993), suggests that changes occurring over the entire sequence contribute to changes in the 3D structure of the protein. This was later supported by Wood & Pearson (1999) who concluded that, on average, sequence change correlates with structural change in protein families.

Extensive evidence supports the local model postulating a logarithmic relationship between sequence similarity and structural similarity (Flores et al. 1993). Several studies by Ptitsyn and co-workers (1998; Ptitsyn, Ting 1999) identified highly conserved residues that were observed to be important in protein folding. Specifically in the cytochrome

subfamilies (Ptitsyn 1998) four completely conserved residues were shown to be important in forming a network of conserved contacts connecting the N and C terminal helices. The importance of these contacts in folding had been confirmed by their presence in molten globule-like folding intermediates. Furthermore these residues had no apparent functional importance, therefore suggesting their role to be purely in protein folding. Subsequent studies on the globin family (Ptitsyn, Ting 1999) found a similar cluster of non-functional conserved residues. These were shown to exist in the interface between helices which are known to fold in the early stages of the formation of the native state and remain relatively stable in the equilibrium molten globule state.

Structural changes occurring between relatives in a protein family can be analysed in a number of ways, for example, structural diversity can be viewed in terms of the number, length and structural location of residue insertions and deletions (indels) across a homologous superfamily. Pascarella & Argos (1992) analysed the occurrence of indels across 32 structural families and revealed that indels are usually one to five residues in length, with very few occurrences of indels greater than 10 residues (1-2%). This was subsequently supported by an analysis by Flores and co-workers (1993) on homologous protein pairs with between 0 and 100% sequence identity, who found that indels tend to be no more than 6 residues in length.

As well as residue indels, structural diversity can also be measured by secondary structure composition. Flores *et al* also studied the conservation of secondary structures and showed on a dataset comprising 90 homologous proteins with a range of sequence identities, that the proportion of residues in the same secondary environment decreased linearly with sequence identity. Similar studies by Russell & Barton (1994) showed that in remote homologues, secondary structure similarity can fall as low as 41% which is equivalent to what you might expect to find through chance. Mizuguchi and Blundell (2000) constructed a secondary structure substitution table where secondary structure elements (SSEs) were classified according to their length and solvent exposure. Analysis of substitutions from SSEs to coil, or complete deletions of SSEs showed that length was the biggest factor for SSE deletion, i.e. short secondary structures were more likely to be



inserted or deleted during evolution. Also short and medium buried strands were much less likely to be substituted by coiled regions than strands on the surface of proteins.

These studies highlighted the fact that although some folds remain highly conserved during evolution others are tolerant to structural variation and secondary structure embellishments. Understanding the tolerance to and impact of evolutionary changes in particular superfamilies is important in informing the classification of new domains as well as aiding comparative modelling and prediction of functional modifications.

The degree of variability in sequence, structure and function between homologues can be highlighted by looking at specific examples of homologous superfamilies in the CATH database (Orengo et al. 2001). It is clear that different superfamilies show different relationships in the conservation of each of the specific features. In some cases the functional properties of relatives impacts on the tolerance to structural change. For example in the globin superfamily the requirement for haem binding results in high structural conservation even at low sequence identity. Conversely in the mainly- $\beta$  immunoglobulin superfamily, function is determined by the loop regions and this may account for the structural variability observed (Lesk, Chothia 1982; Lesk, Chothia 1980). However, in the functionally diverse glycosidase family structure is highly conserved down to low sequence identity even for functionally unconserved proteins (Orengo et al. 2001).

This chapter explores how different homologous superfamilies of proteins evolve in sequence, structure and function and characterises the mechanisms by which this occurs. Differing evolutionary constraints give varying tolerance to change in sequence, structure and function in protein superfamilies and this chapter presents several ways to measure this variability and tries to draw some conclusions about protein evolution from these observations.

Various methods for measuring and analysing the different types of variability (sequence/structure/function) observed in homologous superfamilies are presented. This

involved the development of a new algorithm EquivSEC that measures the variability in secondary structure packing across a family of proteins. New information on the variability observed is presented in an established web resource the Dictionary of Homologous Superfamilies, which has been expanded and improved in a number of ways. Specific examples of superfamilies that illustrate how changes in structure can manifest changes in function are also presented.

Knowledge of variability across homologous superfamilies is important for classifying new relatives and the information captured in the DHS and presented here is later exploited in novel machine learning methods for homologue recognition presented in chapter 5.

## **2.2 Methods**

Protein family resources such as the CATH database can be used to assign structural and functional properties to uncharacterised sequences through homology. To improve the performance of these resources, more sensitive methods are needed for recognising very remote homologues. In this chapter methods will be presented to capture information on similarity between relatives in a homologous superfamily to aid the recognition of remote homologues.

The information captured is presented as a significant update to the Dictionary of Homologous Superfamilies (DHS) web-resource (Bray et al. 2000) ([www.biochem.ulc.ac.uk/bsm/dhs](http://www.biochem.ulc.ac.uk/bsm/dhs)). The DHS provides both structural and functional annotations of domains within each H-level (superfamily) in CATH v2.5.1, extending from 362 families in the original release to 1459 families in CATH v2.5.1.

The methods section first describes how information on the sequence, structure and functional relationships is compiled for all relatives in each CATH superfamily. Subsequently, a similar dataset of highly populated superfamilies is described which was used to explore structural variability in CATH superfamilies in more detail.

### ***2.2.1 Data Sets for Measuring Sequence, Structural and Functional Variability in CATH Superfamilies***

The domain structure dataset used for analysing structural variability was based on version 2.5.1 of the CATH database. The dataset contained only well-resolved structures determined by X-ray crystallography ( $\leq 3.0$  Å). To compile information on the level of sequence similarity and structural between all homologous relatives, 1459 CATH families from version 2.5.1 were used. To examine trends in structural variation on the basis of variability in secondary structure composition and orientation only well populated superfamilies were used to ensure any trends identified were based on

superfamilies that had been well sampled. Well populated superfamilies contained at least three non-redundant representatives (at 35% sequence identity (S35 Reps)). This gave a total of 294 well-populated superfamilies. To explore in more detail the mechanism by which structural change can modulate the functions of relatives within a superfamily, a set of 74 very highly populated ( $\geq 9$  S35Reps) superfamilies was selected. See Table 1 for a summary of all the datasets.

Dataset	Used For	Number of Superfamilies
All CATH Superfamilies	<ul style="list-style-type: none"> <li>• Compiling pairwise similarity on sequence similarity, structural similarity for the DHS</li> <li>• Extracting functional information for the DHS</li> <li>• Identifying sequence relatives in UNIPROT for the DHS</li> <li>• Correlating sequence similarity, structural similarity and functional similarity.</li> </ul>	1459
Well Populated Superfamilies ( $\geq$ 35 Reps)	<ul style="list-style-type: none"> <li>• Identifying N-fold variation in domain size</li> <li>• Analysing variation in secondary structure composition</li> <li>• Analysing variation in secondary structure orientation</li> </ul>	294
Highly Populated Superfamilies ( $\geq$ 35Reps)	<ul style="list-style-type: none"> <li>• Identifying trends in the manner by which structural changes modify functions</li> </ul>	74

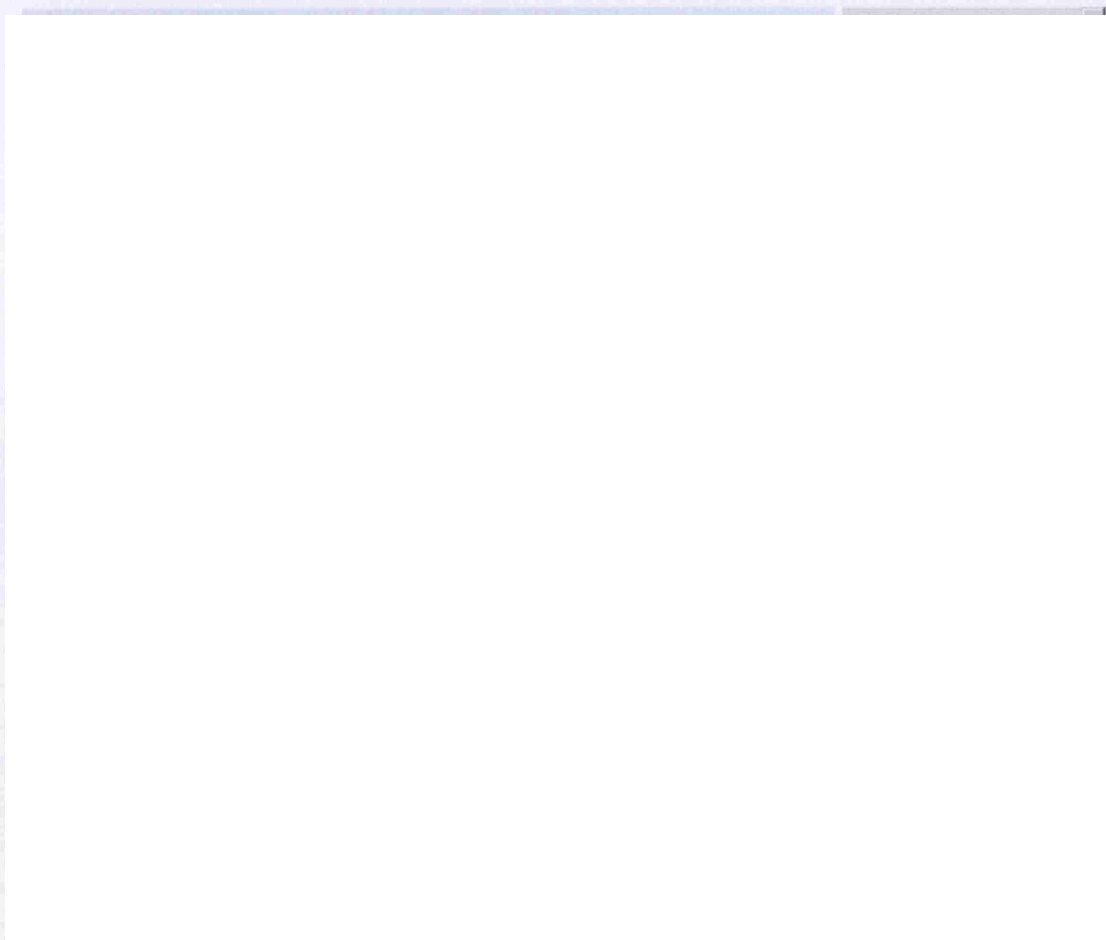
**Table 2.1. Table summarising the 3 datasets used for measuring sequence, structural and functional similarity in this chapter.**

### ***2.2.2 Methods for Measuring Structural Similarity between Relatives***

For each superfamily, pairwise structural similarity scores between all non-identical relatives were measured. These were calculated by performing structural alignments using SSAP (see Section 1.3.2.3 in the Introduction). This uses vector views from all C $\beta$  atoms to determine equivalent residues in the two structures and to populate a score matrix from which an optimal alignment can be found using double dynamic programming. The average and associated standard deviation of SSAP scores were also calculated for each superfamily.

Multiple structural alignments of all S35Reps from a superfamily were also performed using the residue-based CORA algorithm (Orengo 1999). CORA first uses the SSAP global structural alignment algorithm (Taylor, Orengo 1989) to perform pairwise structural comparisons between all the structures being compared. CORA then creates a multiple structural alignment by successively aligning proteins to an evolving consensus 3D template, which encodes the average structural properties of the aligned domains. Proteins are aligned in order of decreasing structural similarity (as measured by the pairwise SSAP score). After each protein is aligned the consensus template is recalculated to take account of any additional structural features of the newly aligned proteins and to recalculate conservation and variability at different positions in the alignment.

Multiple alignments are presented in the DHS both as CORAPLOTS (Bray et al. 2000) (see Figure 2.2) and in the form of a 2DSEC (Reeves et al. 2006) diagram (see Figure 2.3) alongside co-ordinate data of the superposed structures in PDB format. Sequence representations of the alignments are available to download in FASTA format. In the CORAPLOT images of the multiple alignments, residues in each domain are coloured according to residue type and ligand binding where possible.



**Figure 2.2.** Screenshot from DHS website showing a CORAPLOT multiple alignment of the S35Reps of the DNA helicase RuvA, C-terminal domain superfamily (1.10.8.10). Also shown are the pairwise structural comparison SSAP outputs.

2DSEC uses a multiple structural alignment to create a summary of the secondary structures present in each structure. Equivalent (consensus) secondary structures are identified together with those secondary structures present in only one or a few relatives in the alignment. These are described as secondary structure embellishments to the structural core (consensus) for the superfamily. 2DSEC can be used to identify those superfamilies with large secondary structure embellishments. In order to capture this information a simple measure of structural variability is used that calculates the difference in the number of secondary structures in the smallest and largest relatives normalised by the number of secondary structures in the larger protein (Equation 2.1).

$$\% \text{ Variability} = \frac{\text{SSE}_{\text{MAX}} - \text{SSE}_{\text{MIN}}}{\text{SSE}_{\text{MAX}}} \times 100$$

Equation 2.1 The 2DSEC percentage variability score. SSE standing for Secondary Structure Elements.

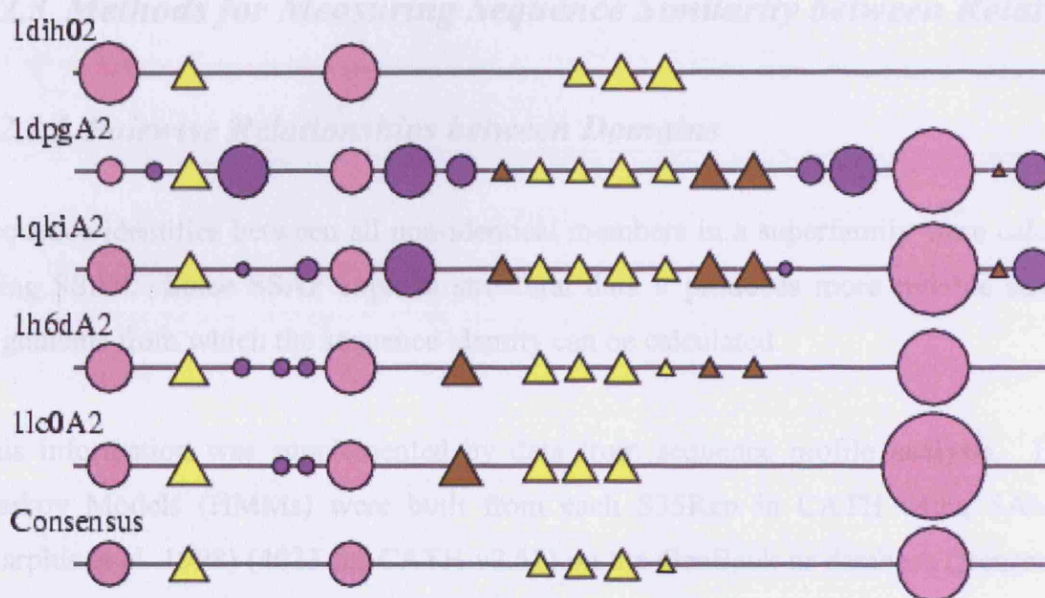


Figure 2.3. A 2DSEC plot of the ATP-grasp large domain. Pink circles represent consensus helices ( $\geq 5\%$  of the aligned domains) and purple circles represent embellished helices. Yellow triangles represent consensus strands and brown triangles represent strand embellishments. The size of the symbol is in proportion to the length (number of residues) in the secondary structure.

In some highly structurally variable superfamilies, automatic multiple structural alignments of all relatives across the whole superfamily can be problematic, resulting in misalignments of some equivalent secondary structures. Therefore the DHS also provides alignments for structurally similar subgroups (SSGs) within a superfamily. To



identify these subgroups all S35Reps are structurally compared pairwise using SSAP and then clustered into structurally coherent subgroups using multi-linkage clustering with a threshold on the SSAP similarity score of 80 (out of 100) and an overlap of 60% of the residues. These thresholds have been selected as previous empirical studies showed that they corresponded to a high degree of structural similarity between relatives, ensuring that reliable multiple structural alignments could be constructed (Orengo 1999). Domains in each SSG cluster are then multiply aligned using CORA.

## ***2.2.3 Methods for Measuring Sequence Similarity between Relatives***

### ***2.2.3.1 Pairwise Relationships between Domains***

Sequence identities between all non-identical members in a superfamily were calculated using SSAP. Since SSAP exploits structural data it produces more reliable structural alignments from which the sequence identity can be calculated.

This information was supplemented by data from sequence profile analysis. Hidden Markov Models (HMMs) were built from each S35Rep in CATH using SAM-T2K (Karplus et al. 1998) (4023 for CATH v2.51) on the GenBank nr database (Benson et al. 2006). All non-identical sequences in the CATH database were then scanned against these models and the E-values measured. This gives a measure of the pairwise probabilities of a significant relationship between a pair of protein domains.

### ***2.2.4 Predicting CATH relatives in UniProt***

In order to detect sequence relatives for CATH structural superfamilies sequences from UniProt (Wu et al. 2006) were scanned against the CATH HMM library described in section 2.2.3.1 and homologous sequences were identified as those hitting CATH HMM models with an E-value < 0.001 and a 60% residue overlap. The residue overlap is the percentage ratio of the number of residues aligned divided by the length of the CATH

HMM. These thresholds were taken from previous studies on strategies to identify superfamily relatives in the genomes (Sillitoe et al. 2005).

The harvested sequences were then compared with all their domain relatives by BLAST (Altschul et al. 1997), to determine the pairwise sequence identity between relatives within each CATH superfamily. Directed multi-linkage clustering was then used to group sequences into appropriate sequence bins i.e. 30, 40, 50, 60, 70, 80, 90, 95 and 100%.

### ***2.2.5 Extracting Functional Information from Public Resources for Sequences in the CATH-DHS.***

To increase the functional information associated with each superfamily CATH domains and their associated sequence relatives were annotated with information from a variety of functional databases (ENZYME (Bairoch 2000), GO (Gene Ontology Consortium, (Ashburner et al. 2000), KEGG (Kanehisa, Goto 2000), COG (Tatusov et al. 2003), SWISSPROT (Boeckmann et al. 2003)) (see Figure 2.4). This is achieved by BLASTING sequences from the CATH-DHS against these resources. Only 95% sequence identity hits, with an 80% residue overlap, were taken as genuine matches inline with the PFScape protocol (Lee et al. 2005).



Figure 2.4. Screenshot of the DHS website showing the functional annotations available. Functional annotations assigned to CATH domains and harvested sequence relatives include GO (gene ontology) annotations, SWISSPROT, EC (enzyme classification), KEGG (Kyoto Encyclopedia of Genes and Genomes) and COG (cluster of orthologous groups of proteins).

### ***2.2.6 Measuring the Variability in Secondary Structure Orientations – The EquivSEC Program***

EquivSEC is a new algorithm written to provide a measure of structural variability in a homologous superfamily. The algorithm is used to observe the variability in angles of secondary structure packing within a superfamily.

For a given superfamily, domains were structurally aligned using the CORA algorithm (Orengo, 99) and the 2DSEC algorithm (Reeves et al. 2006) used to determine equivalent, consensus secondary structures shared by at least 75% of the domains

aligned. This information is then input into the EquivSEC program. The algorithm determines the structural variability across a homologous superfamily by calculating the variability of angles and distances between the equivalent, consensus secondary structure pairs.

Axial vectors through the secondary structures are derived using the Richards and Kundrot algorithm (1988) implemented by the ProSEC suite of programs (Slidel 1996). Various geometric relationships between pairs of vectors including the dot-product angle and dihedral angle, the distance and chirality are calculated. EquivSEC assigns equivalent secondary structure vector pairs across a superfamily of structures in CATH and calculates the variability in angles and distances between them. For each vector pair the mean, minimum, maximum and standard deviation of both angles and distances are recorded.

EquivSEC looks at the variability in secondary structure packing, across a homologous superfamily, from a number of different perspectives. Firstly deviations in packing of secondary structures for all possible pairs of equivalent secondary structures are examined, termed 'global packing'. Subsequently only those secondary structures that are defined as being in contact (closest approach vector distance  $\leq 12\text{\AA}$ ) are considered, termed 'local packing'.

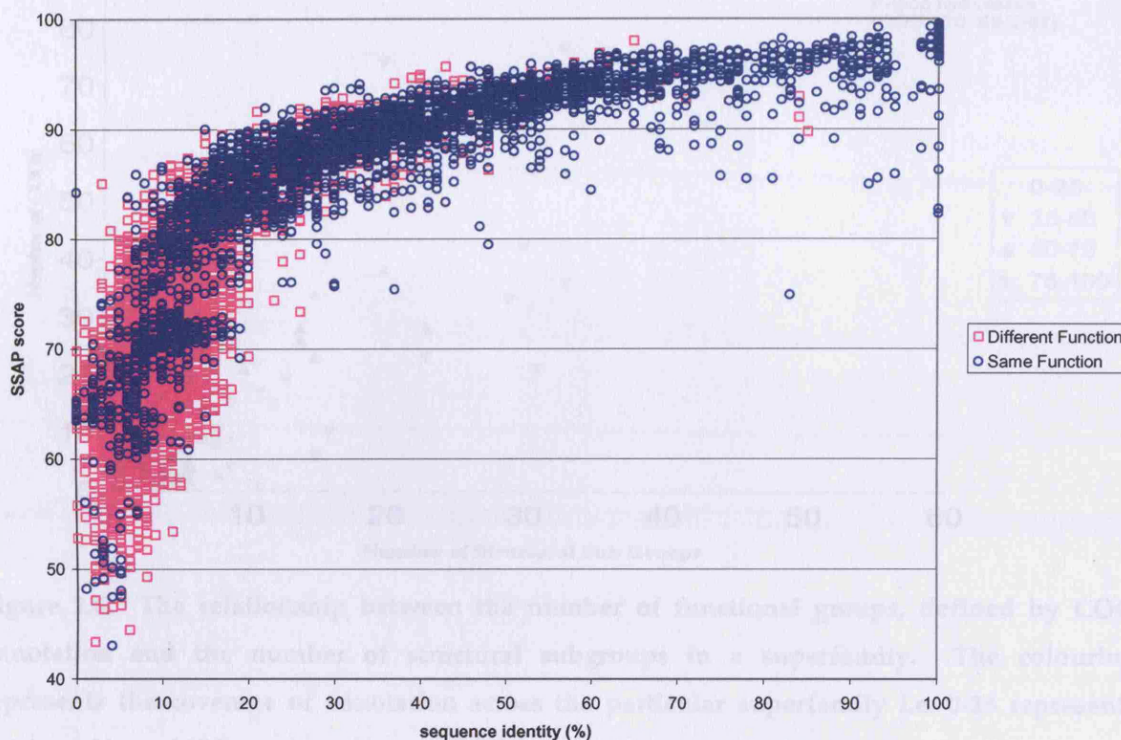
## **2.3 Results**

### ***2.3.1 The Extent of Structural Change in Domain Superfamilies and the Correlation between Sequence, Structural and Functional Similarity***

The information held in the DHS allows various analyses of the correlations between sequence, structural and functional similarity for homologous proteins. The Appendix presents a summary table of information for each CATH superfamily which has more than one diverse sequence family (S35 family). This information is also available on-line in a searchable format (<http://www.cathdb.info/cgi-bin/cath/DhsSummaryTable.pl>). For each superfamily, information on structural variability in the form of the minimum observed SSAP score, the 2DSEC percentage variation score, the EquivSEC average deviation in contacting secondary structures and the number of distinct structural sub-groups is presented. This is supplemented with data on the number of distinct sequence groups in the superfamily as well as the prevalence of the superfamily in the genomes. Finally the number of distinct COG annotations associated with the domains of the superfamily is also listed.

The DHS provides a rich source of information on structural variability for CATH superfamilies and is publicly available through the web for biologists wishing to study particular superfamilies. Aspects of this data were analysed to study the correlation between sequence, structure and function variability and are presented below.

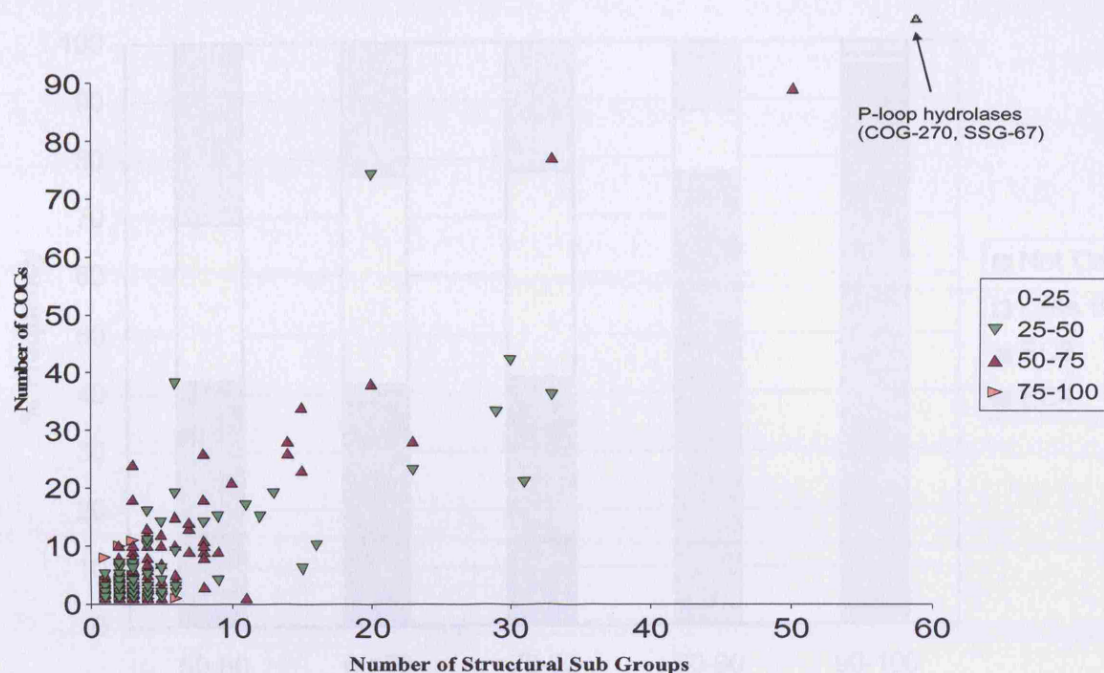
Figure 2.5 plots the sequence identity versus the structural similarity (defined by SSAP) for all non-identical relatives in each CATH superfamily. There is a gradual, linear decrease in structural similarity with decreasing sequence identity down to ~30% which agrees with previous observations of Pearson & Wood (1999). Below 30% significant structural changes are observed in all superfamilies as observed by Chothia & Lesk (1986).



**Figure 2.5.** Scatter plot showing the relationship between sequence, structure and function of all homologues in enzyme superfamilies. Relatives having the same EC classification number are shown in blue. Those with different EC numbers are shown in pink.

At low sequence identity a significant number of homologues are expected to be paralogues that is relatives that have arisen through a gene duplication event within a genome and which are therefore able to acquire new functions. This is supported by the fact that very remote homologues (<20% identity) are more likely to have diverse structures and differing functions as shown in Figure 2.5. This is further supported by Figure 2.6 which shows some correlation between structural diversity as measured by the number of structural subgroups and the number of functions exhibited by sequence relatives as annotated by the COG database.





**Figure 2.6.** The relationship between the number of functional groups, defined by COG annotation and the number of structural subgroups in a superfamily. The colouring represents the coverage of annotation across the particular superfamily i.e. 0-25 represents between 0% and 25% members have a COG annotation.

In order to investigate further the correlation between structural and functional similarity, the proportion of relatives sharing similar functions was analysed at different structural similarity scores. Figure 2.7 plots the correlation between structural similarity and EC conservation for all non-identical homologous pairs. When the SSAP score is above 90 greater than 95% of relatives have similar functions (sharing 3 or more EC numbers). However, when the structural similarity drops below a SSAP score of 70 the function is conserved in only ~40% of relatives. This highlights the fact that remote homologues which have diverged in structure may also have diverged in function.

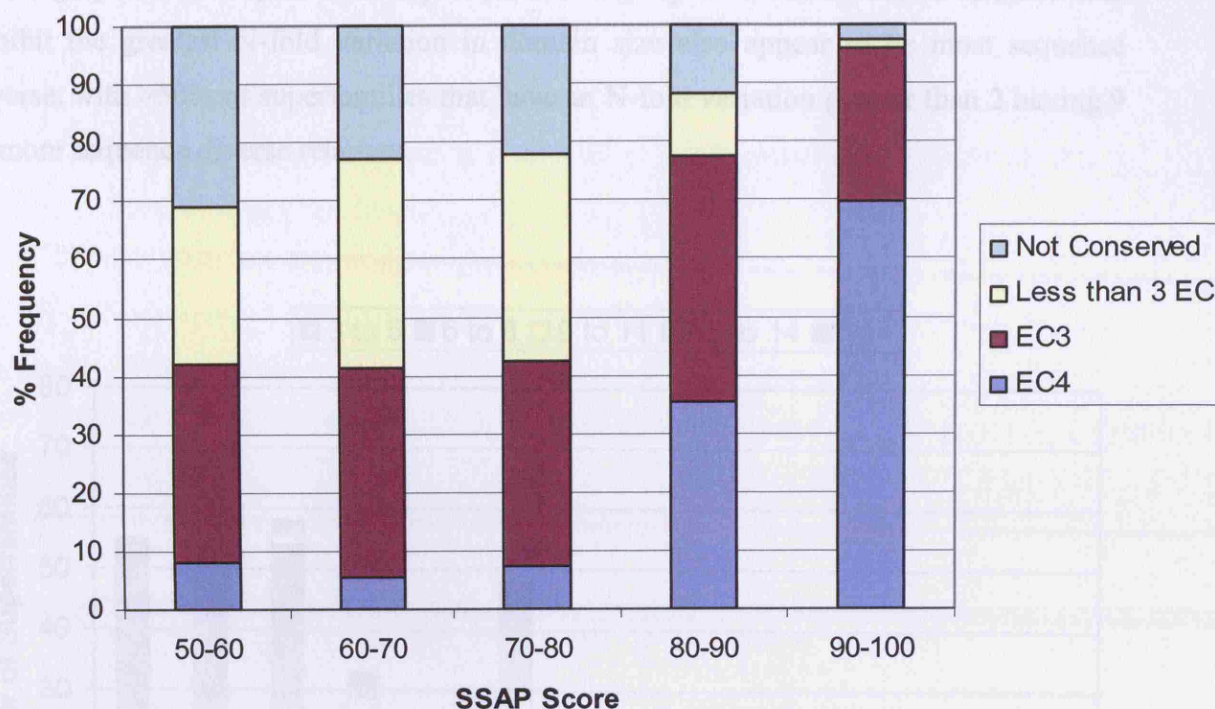


Figure 2.7. Structural similarity measured by SSAP versus EC conservation for all homologous pairs in CATH with EC classifications. EC4 indicates that all 4 levels in the EC classification are the same, EC3 indicates the first 3 levels are the same, less than 3 EC indicates that one or two EC levels are the same.

Different homologous superfamilies appear to have different tolerances to structural change (see Appendix). Some well populated families exhibit a high degree of tolerance to structural change with remote relatives showing considerable structural variation. Whereas, relatives in other families remain highly conserved structurally even as their sequence similarity falls.

Figure 2.8 shows the variation in the domain size (in terms of the number of secondary structure elements) of relatives in each of the 294 well populated homologous superfamilies (3 or more S35Reps) (see Table 2.1). Approximately 60% of superfamilies show N-fold variation in domain size less than 1.6 and this includes some superfamilies that are very sequence diverse (15 or more diverse relatives). However, in 67 superfamilies a 2 fold or more variation in size of relatives is observed and relatives in



two highly structurally diverse superfamilies vary up to 9-fold. Those families that exhibit the greatest N-fold variation in domain size also appear to be most sequence diverse, with >50% of superfamilies that have an N-fold variation greater than 2 having 9 or more sequence diverse relatives.

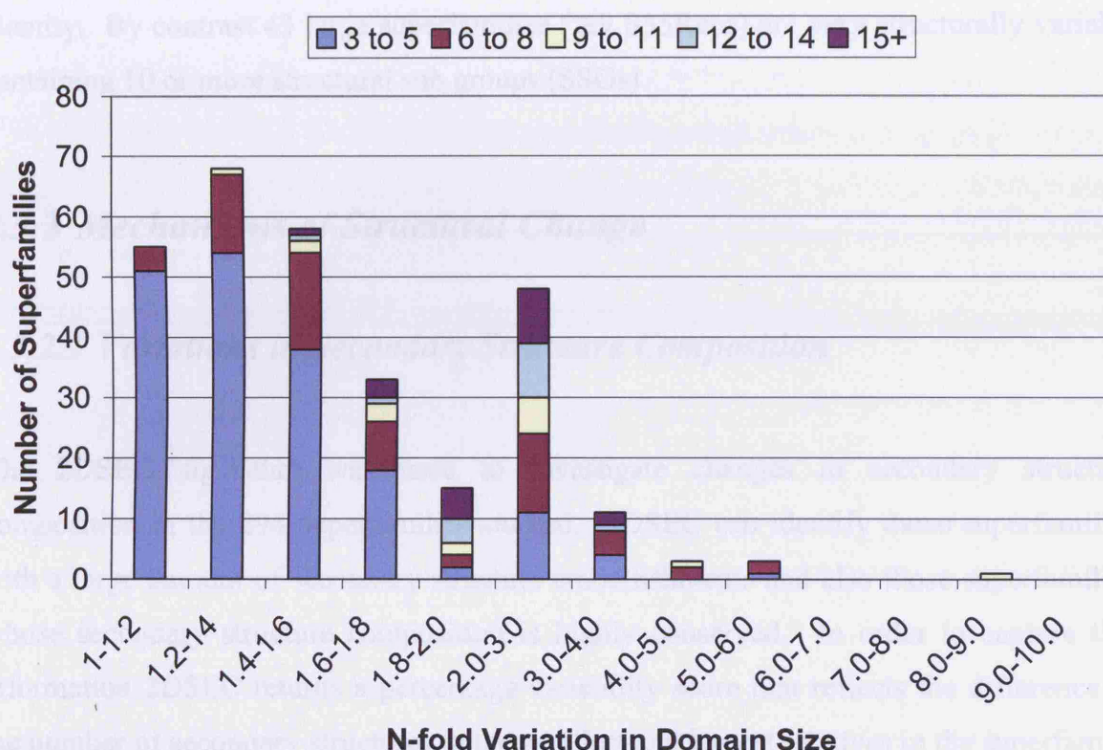


Figure 2.8. Percentage variability in domain size of relatives in relatives of the same superfamily. Colours indicate the number of diverse relatives (blue 3-5, red 6-8, yellow 9-11, turquoise 12-14 and purple 15+).

In superfamilies that are not well populated at present, the apparent existence of structural conservation (i.e. high structural similarity between relatives) may be a consequence of the fact that sufficient relatives have not yet been structurally determined for any significant variations in structure to be detected. Therefore to identify significant trends, the 74 most highly populated superfamilies having nine or more diverse relatives (i.e. sharing  $\leq 5\%$  sequence identity) were considered.

There are only 25 large superfamilies ( $\geq 9$  sequence diverse representatives, S35Reps), which are structurally well conserved, with a high mean SSAP score ( $>85$ ) and low standard deviation ( $<4$ ). One example of this is the Globin family (1.10.490.10) that exhibits an average SSAP score of greater than 85 between its 16 diverse sequence relatives. These domains all have an obligate requirement to bind haem and this poses physical constraints on the domain preventing structural drift even at low sequence identity. By contrast 43 large superfamilies ( $\geq 9$  S35Reps) are very structurally variable containing 10 or more structural sub groups (SSGs).

## ***2.3.2 Mechanisms of Structural Change***

### ***2.3.2.1 Variations in Secondary Structure Composition***

The 2DSEC algorithm was used to investigate changes in secondary structure composition in the 294 superfamilies studied. 2DSEC can identify those superfamilies with a large amount of secondary structure embellishments and also those superfamilies whose secondary structure composition is highly conserved. In order to capture this information 2DSEC returns a percentage variability score that reflects the difference in the number of secondary structures in the smallest and largest relatives in the superfamily (see Equation 2.1 in Section 2.2.2).

Previous analysis of CATH enzyme superfamilies has suggested secondary structural embellishments may have an impact on function by modifying the geometry of the active site or by disrupting the surface topology impacting on substrate specificity and protein-protein interactions (Todd et al. 1999). With this in mind the dataset of 74 highly populated superfamilies ( $\geq 9$  sequence diverse relatives) were analysed and investigated to determine what impact structural embellishments have on the function.

Some families are very structurally conserved in terms of secondary structure embellishments. Many of these families appear to have roles in cell signalling (e.g.

kinases, PH and PTB domains) where they would be expected to take part in highly specific protein-protein interactions, this may restrict these domains tolerance to embellishments. Families such as the 2Fe-2S ferredoxin related family are on average only composed of 6 secondary structures and embellishments may not be tolerated because they would impact on the multiple complexes formed to mediate electron transport and other redox catalysis reactions.

In the cytochrome *P450* family the need to conserve co-factor binding geometry may limit structural variation. The orthogonal bundle of four  $\beta$ -sheets and up to 13  $\alpha$ -helices is highly conserved even in very sequence diverse relatives. This domain is responsible for the binding of the haem co-factor and the conserved beta-sheets form a highly conserved hydrophobic channel for the binding of substrates for redox catalysis. It is feasible that both the binding of the co-factor and the formation of the specific channel reduce the possibility of structural embellishments as they would disrupt either one of these functions.

Analysis showed that for greater than 50% of the highly populated CATH superfamilies ( $\geq 9$  S35Reps), considerable structural change can occur, with some relatives varying in the number of secondary structures twofold or more (see Figure 2.8). By analysing the structural variability score for superfamilies in different architectures (see Appendix) it can be seen that four architectures more frequently comprise families with structurally embellished relatives,  $\alpha$ -orthogonal (1.10), the two-layer  $\beta$ -sandwich (2.60), two layer ( $\alpha\beta$ ) sandwiches (3.30) and the three-layer ( $\alpha\beta$ ) sandwiches (3.40) (Figure 2.10). These architectures are very highly populated in CATH and the Protein Data Bank (PDB), comprising nearly 60% of all structural families. Structural annotation of completed genomes suggest that the high populations of these architectures in the PDB is not simply due to over-sampling but genuinely reflects high occurrence in the genomes (Orengo, Thornton 2005). These architectures also contain some of the most structurally variable superfamilies in terms of number of structural sub-groups (SSGs). For example the  $\alpha$ -orthogonal EF-hand superfamily has 39 SSGs, the two-layer  $\beta$ -sandwich Immunoglobulins has 56 SSGs, the two layer ( $\alpha\beta$ ) sandwich ATP-grasp fold B domain

superfamily has 16 SSGs and the three-layer ( $\alpha\beta$ ) sandwich NAD(P) binding Rossmann like domain has 50 SSGs.

Figure 2.9 shows that the majority of insertions (>85%) comprise only one or two secondary structures at most.

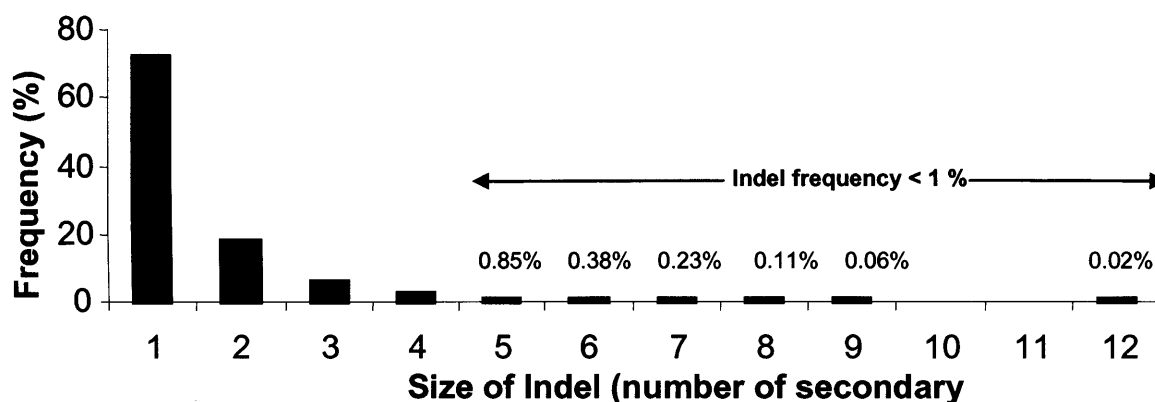


Figure 2.9. Percentage frequency of insertions comprising one or more secondary structures.

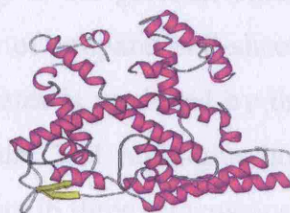
$\beta$ -Strand embellishments in these structures frequently occur as additions or extensions to existing  $\beta$ -sheets or form external  $\beta$ -hairpins. Helices are usually inserted as single elements on the periphery of domains. Detailed observations on selected superfamilies by Gabrielle Reeves, with whom I collaborated on this work, showed that although secondary structure insertions often occur in different places along the peptide chain, they tend to be co-located three dimensionally (Reeves et al. 2006).

## 1.10

1ois0

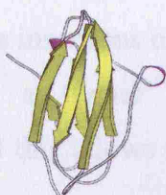


1s4bP

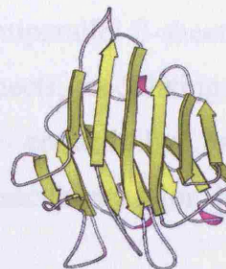


## 2.60

1k5nA

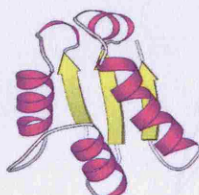


3enrB

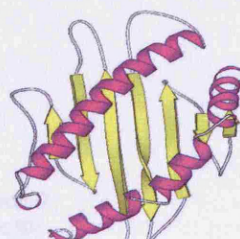


## 3.30

1ay7B

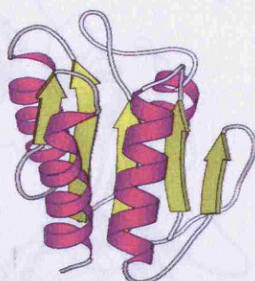


1k5nA



## 3.40

4fxn0



1qmuA

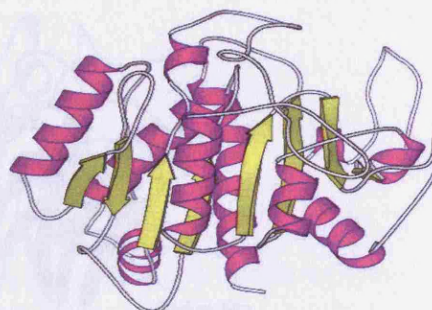
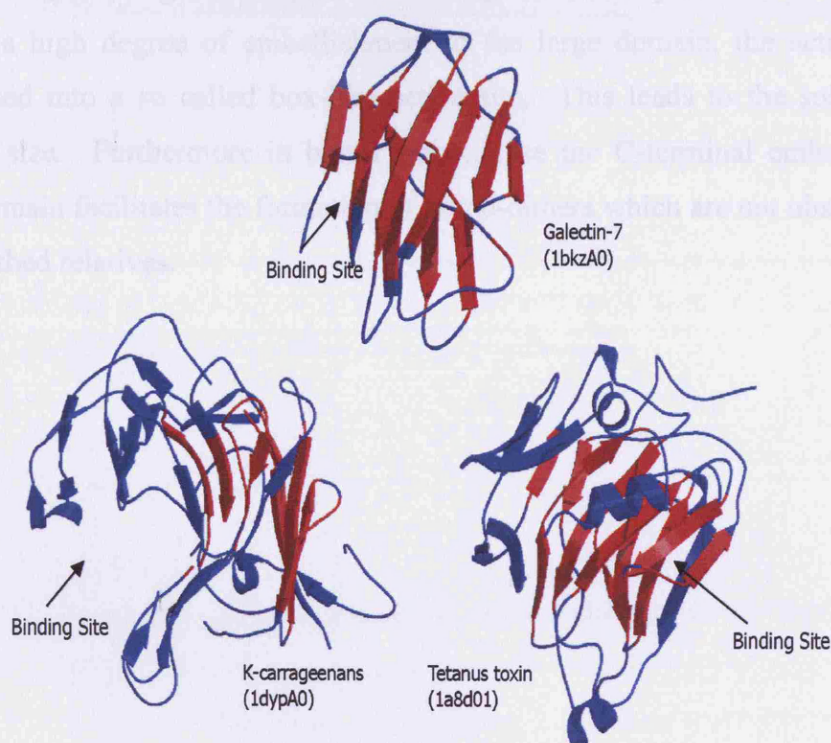


Figure 2.10. MOLSCRIPT representations of smallest and largest (in terms of secondary structures) for selected superfamilies from the most frequently embellished architectures, the orthogonal (1.10) architecture, the two-layer  $\beta$  sandwiches (2.60), the two-layer  $\alpha\beta$  sandwiches (3.30) and the three-layer  $\alpha\beta$  sandwiches (3.40).



An example of a particularly variable superfamily is the galectin binding superfamily. Domains in this superfamily have a conserved pair of antiparallel  $\beta$ -sheets each with five  $\beta$ -strands. The binding of a variety of carbohydrates is mediated by the loops on both ends of the  $\beta$ -sheets (Leonidas et al. 1998). Figure 2.11 highlights how the extensive embellishments on the edge of both sheets aggregate in three dimensions to modulate the environment of the carbohydrate binding site. Galectin-7 is the smallest member of the superfamily just containing the conserved pair of antiparallel  $\beta$ -sheets. By contrast the Tetanus toxin shows insertions of  $\beta$ -strands to both sheets, modulating the active site. K-carrageenans has a tunnel shaped active site created by extensive structural embellishments and this allows it to bind large polysaccharides for degradation (Michel et al. 2001).



**Figure 2.11.** Three domains from the galectin-type carbohydrate recognition domain superfamily. Secondary structures that are conserved in 75% of members in the superfamily are coloured red, regions of the structure that are coloured blue are secondary structure embellishments or coil.

One of the most structurally embellished superfamilies is the Large Domain of the ATP dependent amine/thiol ligase superfamily also known as the ATP-Grasp superfamily. Nearly all relatives in the superfamily share three common domains (see Figure 2.12). ATP is bound in the cleft between two of the domains, which are referred to as the small and the large ATP binding domains. Members of this protein superfamily typically catalyse ATP-dependent ligation of a carboxylate substrate to an amine or thiol group of a second substrate (Todd et al. 1999). The size of the  $\beta$ -sheet in the Large domain varies between 5 and 11 strands and the total number of secondary structures can vary 2.5 fold. The less embellished members of the superfamily (such as D-alanine-D-alanine ligase and glutathione synthase) form an L-shaped configuration and have a very accessible active site. This allows them to bind large substrates such as peptidoglycans in D-alanine-D-alanine ligase and glutathione in glutathione synthase. In the relatives that exhibit a high degree of embellishment to the large domain, the active site becomes condensed into a so called box-like active site. This leads to the substrates being of smaller size. Furthermore in biotin carboxylase the C-terminal embellishment of the large domain facilitates the formation of homo-dimers which are not observed in the non-embellished relatives.

### Distributions of Angles

For each superfamily the overall variation in secondary structure orientation was calculated for both the global and local orientations. Local orientations are secondary structure vectors and global orientations are vectors calculated from all secondary structure pairs within a domain. The analysis was performed locally considering only those angles that are present in 27.5% of the superfamily structures including those embeddings.

Figure 2.12 shows the three domains of the ATP-Grasp Superfamily.

Table 2.1 shows the distribution of angles in the ATP-Grasp Superfamily. The majority of the pairwise angles are in the range of approximately 50° to 100°. Approximately 50% of angles between subunits are in the range of approximately 50° to 100° and approximately 50% are in the range of approximately 100° to 150°. The angles in the range of approximately 100° to 150° are in the range of approximately 100° to 150°.

Figure 2.12. Three domains of the ATP-Grasp Superfamily. In red the large domain, in blue the small domain and in light blue the B domain. Residues shown in the yellow are involved in ATP binding and residues in green are involved in substrate binding.

### 2.3.2.2 Variation in Secondary Structure Orientations

EquivSEC (see Section 2.2.6) was used to analyse the degree of variation in secondary structure packing between homologues in 294 well populated superfamilies CATH superfamilies. The dataset described in Section 2.2.1 (also see Table 2.1) was used and further pruned to include only those domains which have at least 3 equivalent secondary structures to other S35 domains in the superfamily. The aim of the analysis was to try and identify those families that were particularly tolerant to changes in secondary structure packing and those where the orientations were highly conserved and also to identify global trends in secondary structure packing.



## Distributions of Angles

For each superfamily the average variation in secondary structure orientation was calculated for both the global and local models, local refers to contacting secondary structures ( $<12\text{\AA}$  between secondary structure vectors), global refers to all secondary structure pairs regardless of proximity. Global and local analysis was performed firstly considering only consensus secondary structures (i.e. secondary structure elements present in  $\geq 75\%$  of relatives) and secondly considering all secondary structures including those embellishments present in only a few relatives.

Table 2.2 shows that for the local analysis, when you consider contacting consensus secondary structures that are present in 75% of relatives in a superfamily, the majority of the pairwise angles show little tolerance to variability. Approximately 50% of angles between equivalent, contacting secondary structures vary less than  $8^\circ$  and  $\sim 80\%$  vary less than  $15^\circ$ . A minority of pairs,  $\sim 5\%$ , show variations in angle of greater than  $25^\circ$ . When you include those structural embellishments that are not common to a majority of relatives the variation in angles between contacting secondary structure pairs increases, with  $\sim 60\%$  of angles varying between 8 and  $25^\circ$ .

Variation in Packing (degrees)	% Superfamilies			
	Local / Consensus	Local / All	Global / Consensus	Global / All
>25	5.08	5.19	9.67	5.86
16-25	15.63	19.26	19.70	30.77
8-15	33.20	37.41	40.89	38.46
<8	46.09	38.15	29.74	24.91

**Table 2.2.** Table showing the % of superfamilies which have an average variation in secondary structure orientation, greater than  $25^\circ$ , between 16 and  $25^\circ$ , 8 and  $15^\circ$  and less than  $8^\circ$ . Local refers to contacting secondary structures ( $<12\text{\AA}$  between secondary structure vectors). Global refers to all secondary structure pairs regardless of proximity. Conserved refers to secondary structure pairs present in at least 75% relatives. All refers to all secondary structure pairs regardless of how many relatives possess them.

These observations suggest that on average consensus secondary pairs are more conserved in their secondary structure orientations than those secondary structure pairs that are only present in a minority of relatives in a superfamily. It is likely that consensus secondary structure elements are more buried and in the core and therefore less likely to change in orientation because they need to maintain the robust structural framework which can support changes in the peripheral elements. Varying orientations in peripheral secondary structures may be more tolerated because they bring about structural changes on the surface of the protein (e.g. in the region of the active site / protein-protein interfaces) which promote changes in substrate specificity i.e. modifying the functions of a paralogous domain in a manner that is beneficial to the organism.

There is also clearly a difference in observed tolerance to change in secondary structure orientations when you compare global with local pairs. Secondary structure pairs in close proximity appear on average more conserved in orientation than those pairs that are distant in the proteins structure. When you consider global consensus pairs the variability in orientation increases with 30% of pairs showing variability greater than  $16^\circ$  compared to 20% for contacting consensus pairs. In many of the structures analysed a large proportion of contacting pairs are adjacent beta strands hydrogen bonded to each other within a beta sheet and therefore these bonds would act as constraints on significant shifts in orientation. In other cases salt bridges also act to mediate contacts between secondary structures, again constraining secondary structure shifts.

Lesk & Chothia (1980;1982) reported shifts of up to  $30^\circ$  in secondary structure orientations in two large superfamilies, the mainly alpha globlins and the mainly beta immunoglobulins. Here on a much larger dataset of 294 well populated superfamilies it is shown that the majority (85%) of consensus, secondary structure pairs exhibit variation in orientation below  $20^\circ$  though rare shifts of up to  $70^\circ$  are observed (see Figure 2.13).

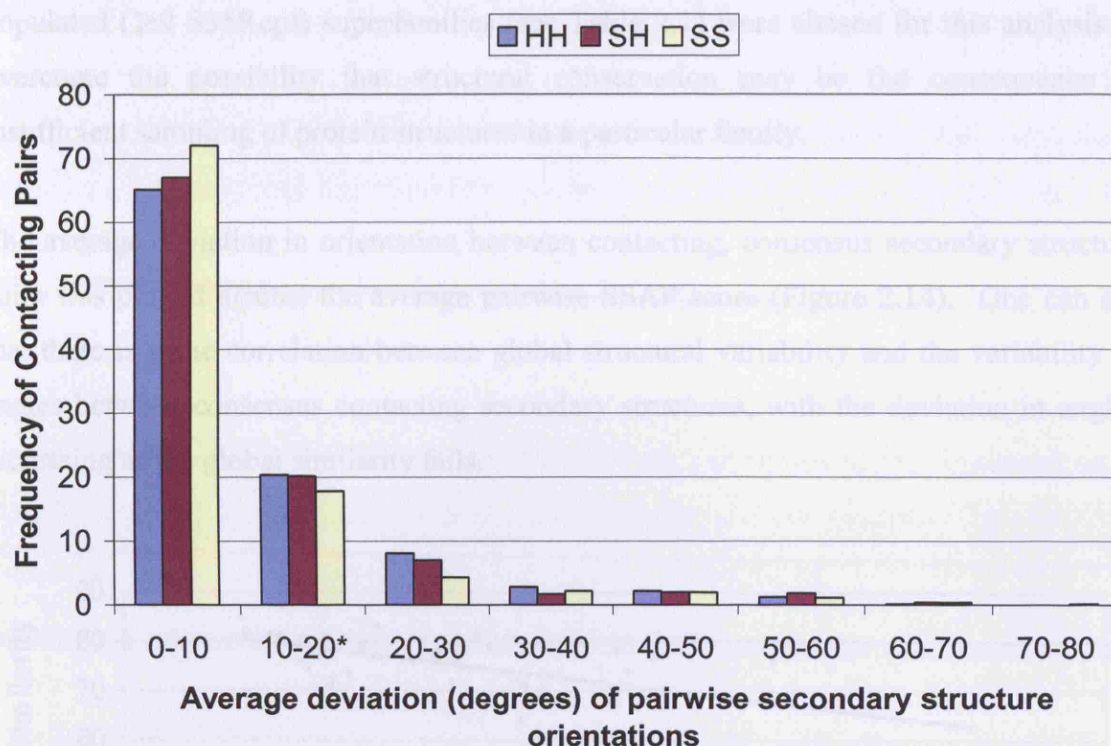


Figure 2.13. The observed frequency of changes in angle for each type of consensus, contacting secondary structure pairs. HH representing helix-helix pairs, SH representing strand-helix pairs and SS representing strand-strand pairs.

It can be seen that helix-helix pairs contribute most to the variability, followed by helix-strand pairs and finally strand-strand pairs exhibit the least variability. Beta strands are likely to be more constrained in their orientation due to the presence of hydrogen bonds between adjacent strands, whereas helix-helix, and helix-strand pairs have no such constraints and therefore have more freedom in their orientations.

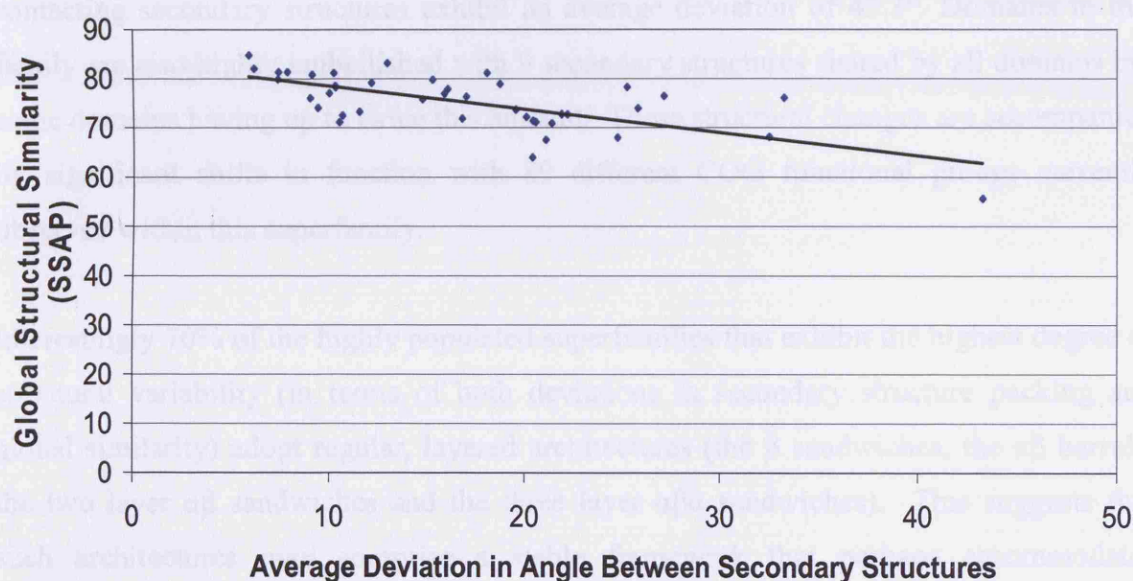
#### 2.3.2.2.1 Correlation between Global Structural Similarity and Conservation of Secondary Structure Orientations

The relationship between a superfamily's tolerance to changes in secondary structure orientation and the average pairwise SSAP score (a measure of global structural similarity) between domains in a superfamily was also investigated. Only the 74 highly



populated ( $\geq 9$  S35Reps) superfamilies (see Table 2.1) were chosen for this analysis to overcome the possibility that structural conservation may be the consequence of insufficient sampling of protein structures in a particular family.

The average deviation in orientation between contacting, consensus secondary structure pairs was plotted against the average pairwise SSAP score (Figure 2.14). One can see that there is some correlation between global structural variability and the variability of angles between consensus contacting secondary structures, with the deviation in angles increasing as the global similarity falls.



**Figure 2.14.** Average SSAP score plotted against the average deviation in contacting, consensus secondary structure orientation for each superfamily with  $\geq 9$  S35Reps.

Superfamilies that have high average SSAP scores also have highly conserved orientations of contacting secondary structures in the core. The Glutathione S-transferase superfamily (1.20.1050.10) has an average SSAP score of 84.7 and exhibits a deviation in conserved secondary structure packing of only  $5.9^\circ$ . The Transferase (phosphotransferase) domain 1 superfamily of kinases (1.10.510.10) are also highly conserved structurally with an average SSAP score of 82.0 and an average deviation in

conserved secondary structure packing of 6.1°. Proteins in both of these superfamilies are small and have functions related to cell-signalling and form multiple and varied protein-protein interactions. Despite this conservation in structure and low variation in secondary structure packing, considerable divergence in sequence is observed between relatives in this family. This may cause changes in the surface features of the proteins promoting diverse partnerships without causing significant changes to the conserved structural cores of the relatives.

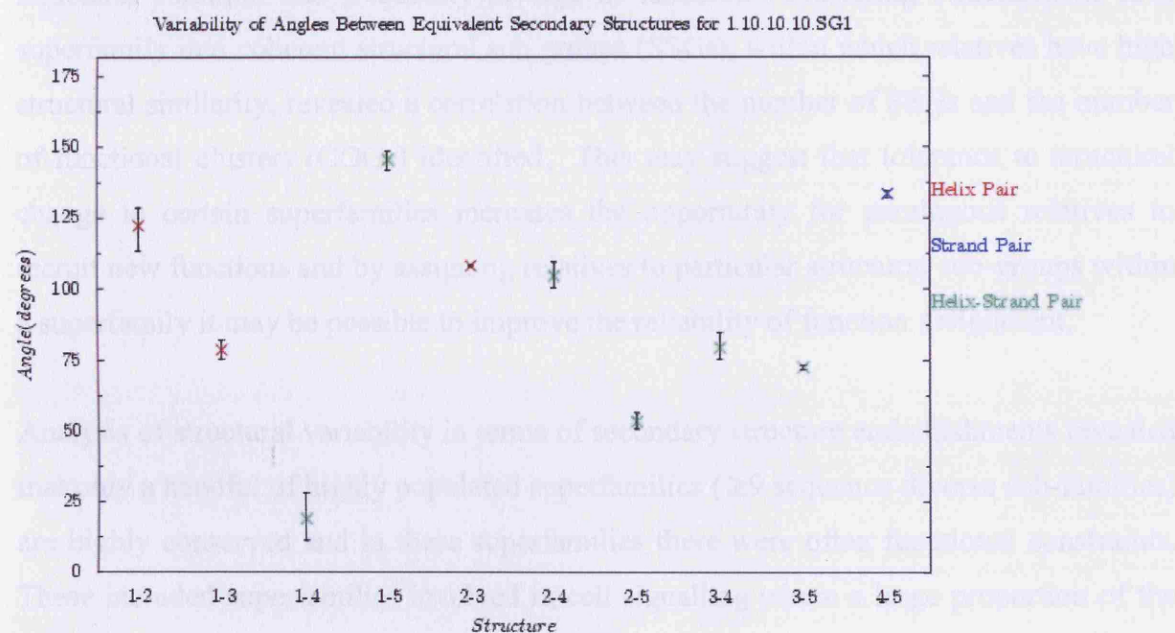
The NAD(P) Rossmann like superfamily (3.40.50.720) has the lowest average SSAP score (55.6) and is highly structurally variable with 50 structural sub-groups. The consensus contacting secondary structures exhibit an average deviation of 43.3°. Domains in this family are also highly embellished with 9 secondary structures shared by all domains but some domains having up to twice this amount. These structural changes are accompanied by significant shifts in function with 89 different COG functional groups currently observed within this superfamily.

Interestingly 70% of the highly populated superfamilies that exhibit the highest degree of structural variability (in terms of both deviations in secondary structure packing and global similarity) adopt regular, layered architectures (the  $\beta$  sandwiches, the  $\alpha\beta$  barrels, the two layer  $\alpha\beta$  sandwiches and the three layer  $\alpha\beta\alpha$  sandwiches). This suggests that such architectures may comprise a stable framework that perhaps accommodates significant sequence change by adjusting the orientation of the secondary structure layers. Furthermore, the robust structural framework in the core of these proteins also appears to support the diverse secondary structure embellishments observed in some relatives (average 2DSEC Percentage Variability score  $\sim$  67%). As a consequence relatives are highly varied in function.

## ***EquivSEC as a Diagnostic Tool***

EquivSEC can produce plots showing the variability in angle between contacting secondary structures for a particular superfamily (see Figure 2.15). These are presented in the DHS resource for each superfamily as well as for structural subgroups and sequence sub-families (S35 & S95). These plots provide a useful tool for identifying contacting secondary structures that are constrained in their packing and those that are more flexible.

EquivSEC plots may also be useful for homology modelling as the information on the angles adopted by pairs of contacting secondary structures in a particular superfamily can be used to constrain the modelling of relatives. EquivSEC plots have also proved useful in the classification of new protein domain structures in CATH.



**Figure 2.15.** An example of the EquivSEC output for a particular domain superfamily (1.10.10.10, “winged helix” DNA binding domain). The mean angle between equivalent secondary structure pairs is plotted as well as the range of observed angles.

## 2.4 Discussion

Analysis of structural variation in domain families can reveal constraints on protein evolution, which can aid structure prediction and classification. Despite the exponential increase in the PDB the data in this chapter reveals similar trends in the relationship between structural variability and sequence divergence as those detected 20 years ago with smaller datasets. It is apparent that some superfamilies remain structurally well conserved even when sequences diverge considerably, whilst others can exhibit extensive structural changes.

Analyses presented in this chapter explored the correlation between sequence, structural and functional variability. Results showed that greater than half of the highly populated superfamilies (comprising  $\geq 9$  sequence diverse sub-families) also show a high degree of structural variation and frequently diverge in function. Clustering structures in each superfamily into coherent structural sub groups (SSGs), within which relatives have high structural similarity, revealed a correlation between the number of SSGs and the number of functional clusters (COGs) identified. This may suggest that tolerance to structural change in certain superfamilies increases the opportunity for paralogous relatives to recruit new functions and by assigning relatives to particular structural sub-groups within a superfamily it may be possible to improve the reliability of function assignment.

Analysis of structural variability in terms of secondary structure embellishments revealed that only a handful of highly populated superfamilies ( $\geq 9$  sequence diverse sub-families) are highly conserved and in these superfamilies there were often functional constraints. These included superfamilies involved in cell signalling where a large proportion of the exposed structure is likely to be involved in ligand binding and protein-protein interactions. In other highly conserved superfamilies (e.g. the globins) structural constraints are observed that are likely to promote the conservation of the geometry of co-factor binding sites.



The analysis of highly variable superfamilies allowed insights to be gleaned on the impact of structural embellishments on function. Large structural embellishments often arise through the accretion of several individual secondary structure insertions which are distributed throughout the polypeptide chain but aggregate in 3D. In some cases these embellishments are located around the active site and modulate the geometry and substrate accessibility e.g. in the galectin binding domains. In other relatives embellishments promote the formation of alternative oligomeric states (e.g. the ATP Grasp superfamily) or create additional interfaces for interactions with other proteins.

The greater tolerance of regular layered architectures such as the two-layer  $\beta$  sandwiches and the two and three layer  $\alpha\beta$  sandwiches to both secondary structural embellishments and shifts in secondary structure orientations suggest that these layered arrangements may promote greater stability and tolerance to structural change. For example layers may be able to accommodate residue indels more easily as they can be extended easily, particularly  $\beta$ -sheets thereby accommodating the addition of secondary structures without significantly disrupting the packing of the layers.

Information on structural and function variability in all well populated CATH domain superfamilies is presented in the Dictionary of Homologous Superfamilies (DHS) and can be used in gauging whether structurally similar domains are likely to be homologous.

In summary sequence, structure and function can sometimes vary extensively between homologous domains in superfamilies and ways of measuring this variability in superfamilies will significantly aid the classification process. Furthermore the general correlations seen between sequence, structure and function variation suggest that methods for detecting remote homologues in these more variable superfamilies will need to include multiple measures of similarity between relatives i.e. based on sequence, structure and function similarity. Also these studies confirm that global thresholds are inappropriate at defining homologous relationships in terms of sequence, structure and functional similarity because superfamilies behave so differently. This is discussed in more detail in chapters 5 where machine learning approaches are applied to combine



different types of information and improve the recognition of remote homologues and functionally related protein domains.

# 3 Optimisation of the CATHEDRAL Algorithm to Classify Domains in CATH

## *3.1 Background and Aims*

Protein structures are composed of individual folding units called domains and genome analysis suggests that up to 80% of proteins in eukaryotic organisms (60% in prokaryotes) are multi-domain (Apic et al. 2001). Each domain adopts a specific fold in 3D space and it has been estimated that there are several thousand possible folds in nature (Chothia 1992; Orengo et al. 1994; Coulson, Moult 2002; Grant et al. 2004). The protein domain can be considered an evolutionary unit and as such, many groups have sought to construct classifications of domains at the structural level. The two most comprehensive of these are SCOP (Murzin et al. 1995) and CATH (Orengo et al. 1997).

In 2005, over 7000 new protein structures were deposited in the PDB and according to version 2.6 of the CATH database nearly 50% of these were multi-domain. The first step in the classification of a new protein structure is to identify its composite domains. Traditionally this has been a two stage process whereby the domain boundaries are resolved, followed by the recognition of the individual domain folds. Although many new structures comprise domains with high sequence similarity to previously classified structures, a significant proportion are the result of Structural Genomic Initiatives (SGIs), which specifically target novel genes and families. Despite this aim, automatic classification remains a reasonable goal as recent analysis (Todd et al. 2005) has shown that 90% of non-redundant SGI structures (i.e.  $\leq 30\%$  sequence identity to a previously classified structure) have an analogous or homologous structure in CATH. In this

situation, structural comparison algorithms are essential to facilitate the automatic classification of domains.

There are many methods in the community for comparing protein structures. These range from secondary structure based methods such as GRATH (Harrison et al. 2003) and SSM (Krissinel, Henrick 2004), through to residue distance and contact based methods such as DALI (Holm, Sander 1993), CE (Shindyalov, Bourne 1998), SSAP (Taylor, Orengo 1989) and LSQMAN (Kleywegt 1996).

It is worthwhile to note that domain boundary recognition, in itself, is difficult, but discontinuous domains make it even more complicated. Domains are not always arranged linearly along the polypeptide chain and can be formed from disconnected regions of the sequence. Jones and co workers (Jones et al. 1998) observed that approximately 20% of domains from multi-domain proteins in the PDB are discontinuous.

Several methods have been developed to automatically detect domain boundaries in protein structures through *ab initio* knowledge of domain structure and interactions. Approaches such as DOMAK (Siddiqui, Barton 1995) rely on the hypothesis that a domain makes more internal contacts (intra-domain) than external contacts (residue contacts to the remainder of the structure). PUU (Holm, Sander 1994) uses a harmonic model to describe inter-domain dynamics, and is used to define domain units in the FSSP database (Holm, Sander 1998). By contrast, the DETECTIVE method (Swindells 1995) attempts to determine the hydrophobic core at the heart of each compact domain structure. These three methods were integrated to provide a consensus approach to domain boundary assignment in the original CATH classification protocol by combining the results from these independent methods (Jones et al. 1998).

Most automatic domain boundary recognition methods described above, all report between 70%-80% accuracy in domain boundary assignment based on their own benchmarking tests. However in practice it has been observed that the methods often contradict each other in their results (Frances Pearl, personal communication).

Furthermore Holland and co workers (Holland et al. 2006) report that all the approaches struggle to correctly assign boundaries for domain architectures that do not form compact structures, for example the alpha horseshoe domains.

Although these algorithms effectively delineate domains for a large percentage of protein chains in the PDB (even those which contain novel folds), they provide no indication as to whether each individual domain is structurally similar to folds already classified within the CATH database. Therefore, it is still necessary to compare the excised domain against a library of CATH domains in order to classify the fold. Since structure-based database scans and manual validation of domain boundaries are both slow, this has remained one of the major bottlenecks in the CATH classification process.

As discussed above it is likely that there are only a limited number of protein folds, and a newly determined multi-domain structure could well contain folds which have already been classified in CATH. Exploiting the concept of domain recurrence to detect known folds in multi-domain structures is a sensible strategy that should allow the classification procedure to be much more efficient by enabling both the identification of domain boundaries and the subsequent assignment of the delineated domain into the correct fold group. The concept of recurrence is not new and has been successfully exploited by other structural classifications. For example, the DALI algorithm is employed to detect recurrent folds for classification in the DALI Domain Database (Holm, Sander 1998), whilst the SCOP database employs manual inspection to locate known recurring folds.

There are several powerful algorithms for structure comparison that could be used to compare a given protein chain against a library of known domain units. Most rely on a two stage process of initially measuring the similarity of residues and/or secondary structures, followed by a subsequent alignment stage that maximises the score of aligned positions and optimises the superposition. The similarity of residues and/or secondary structures is generally measured by comparing the geometric properties of C $\alpha$  atoms, C $\beta$  atoms or secondary structures such as distances and vector properties (angles or chirality for example).

The different protein comparison methods produce an array of different scoring functions to help assess the similarity of the two proteins aligned; these may take the form of statistical and geometric measures.

Most structure comparison methods produce a raw score which can be modelled statistically against a benchmarked dataset or a random sample to produce a measure of significance e.g. Z-Scores (DALI) or p-values (STRUCTAL). The most common geometrical, quantitative measure of similarity used is the Root Mean Square Deviation (RMSD). This is simply the square root of the average squared distance between equivalent atoms in the alignment (see Section 1.3.2.1 in the Introduction). The RMSD is a useful gauge of structural similarity but it is directly linked to the number of equivalent positions. Many methods 'prune' their alignments to reduce the number of equivalent residues to those that are the most geometrically similar, artificially reducing the overall RMSD, whereas other methods attempt to ensure the best global alignment by including all equivalent residues leading to an artificially increased RMSD. Despite its limitation RMSD is the most widely used geometrical measure.

Most amino acid mutations occur in the loop regions of proteins so a fast and effective way of comparing protein structures is to only consider similarities between the secondary structures. Secondary structure matching methods are extremely fast at searching databases of folds and often used to identify likely fold matches that can be more accurately aligned using residue based methods.

A common way of representing the secondary structures of a protein is to use Graph theory (Harrison et al. 2002; Grindley et al. 1993). A graph is a two dimensional set of objects, termed nodes, connected by edges that describe the relationship between them. The use of graph theory to represent protein structures was first developed by Grindley and co-workers (1993). For a detailed description of two established secondary structure based algorithms GRATH and SSM see Section (1.3.2.2) in the Introduction.

Further granularity can be achieved by methods that align individual residues. The common goal of these residue based alignment methods is to identify a set of residue pairs from each protein that are structurally similar. There are two general strategies for finding such alignments: (1) search for transformations that optimally position the two structures with respect to one another, and then use the transformation to find the best alignment, and (2) directly search for a good alignment using optimisation strategies such as dynamic programming or simulated annealing. STRUCTAL and LSQMAN belong to the first group whereas SSAP, DALI and CE belong to the second. For a detailed description of these residue based alignment methods see Section (1.3.2.3) in the Introduction.

The performance of an automatic structural alignment method should be assessed both on its ability to generate biologically-meaningful alignments and its capacity to accurately detect similar folds and homologous protein structures. As Kolodny and co-workers highlight (Kolodny et al. 2005), not all structural comparison methods are as good at scoring their alignments as they are at producing them. An RMSD value, or in fact any linear transformation of this, remains dependent on the number of aligned residues. Some algorithms are optimised to find highly conserved regions between two protein structures. This can be useful in detecting similarities within extremely diverse superfamilies and fold groups. However, these methods do not necessarily give a globally optimal alignment and can assign high significance to the chance similarity of matching small structural motifs that may not be in equivalent positions in the two structures being compared. Hence, for the purpose of domain boundary recognition it is also vital to consider the number of aligned residues as a proportion of those residues in the larger of the two structures, as well as the RMSD of the superposed residues.

The aim of this chapter was to optimise a novel algorithm, CATHEDRAL, for assigning domain boundaries and folds to multi-domain protein structures. CATHEDRAL combines secondary structure matching (GRATH) and residue alignment (SSAP) algorithms and was designed and implemented in collaboration with Oliver Redfern. The

extensive benchmarking described in this chapter was performed solely by the author of this thesis.

The fidelity of domain boundaries assigned using this approach is highly dependent on the quality of the structural alignment produced by SSAP. A comprehensive analysis of the ability of SSAP to generate accurate alignments and score structural similarity was undertaken and placed in context with other publicly available methods.

## **3.2 Methods**

### **3.2.1 The CATHEDRAL Protocol**

Secondary structure based alignment methods (e.g. GRATH) are an order of magnitude faster than residue based methods (e.g. SSAP), but limited by the fact they match core secondary structure motifs, rather than generating globally optimal residue alignments. As distant relatives in many folds and homologous superfamilies exhibit a large degree of structural embellishments around a common core (see Chapter 2), determining overall structural similarity with only secondary structure matching remains problematic. However, the CATHEDRAL protocol was designed to utilise the secondary structure matching GRATH algorithm as a fast pre-filter to select putative fold matches from the CATH library to be further aligned at the residue level by SSAP.

CATHEDRAL was designed to aid the classification of new proteins structures into the CATH database by identifying component domain folds and using structural alignments to classified relatives to predict domain boundaries.

#### **3.2.1.1 CATHEDRAL – Recognising Domains in Multi-Domain Chains**

CATHEDRAL operates in an iterative fashion, whereby once a domain is assigned, the remainder of the query chain is then rescanned against the fold library for each assignment.

The performance of the GRATH algorithm developed by Harrison *et al.* (2003) was shown to perform well for identifying similar folds to a given query structure. However, significant E-values can sometimes represent small common motifs occurring in different folds. For example, a small domain consisting mainly of a  $\beta\alpha\beta$  motif may match a region of a larger domain, however this similarity might not represent a genuine fold similarity.



In the CATHEDRAL algorithm this limitation is overcome by searching for large domain matches first (Figure 3.1). The fold library is split into a 'large library' (containing folds with 5 or more secondary structures) and a 'small library' (containing folds with less than five secondary structures). To limit the effects of matching small motifs incorrectly, all domains are assigned using the large library before the small library is queried.

Each query chain is first scanned against the fold library and the hits ranked by the GRATH E-value. To increase the chance of finding the closest structural match and hence the best domain boundary assignment, representatives from the top 10 fold groups are recompared using SSAP. Preliminary optimisation studies showed that the correct fold was ranked in the top 10 for ~95% of domains.

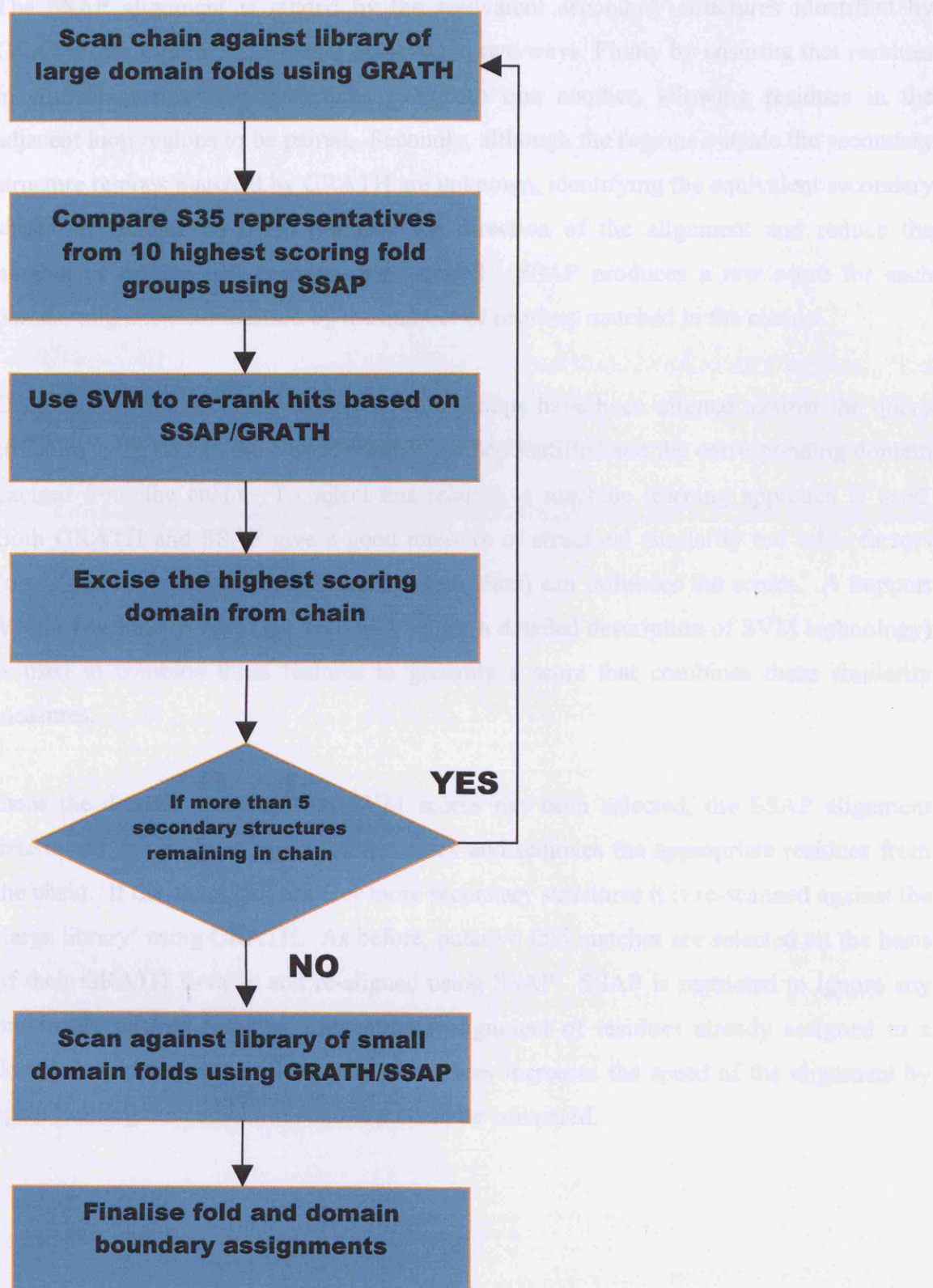


Figure 3.1. Flow chart of CATHEDRAL algorithm for assigning folds and domain boundaries to protein chains.

The SSAP alignment is guided by the equivalent secondary structures identified by GRATH (see Figure 3.2). This is achieved in two ways. Firstly by ensuring that residues in equivalent secondary structures pair with one another, allowing residues in the adjacent loop regions to be paired. Secondly, although the regions outside the secondary structure regions matched by GRATH are unknown, identifying the equivalent secondary structures enables SSAP to orientate the direction of the alignment and reduce the number of residue pair comparisons required. SSAP produces a raw score for each domain alignment normalised by the number of residues matched in the chain.

Once representatives from the top 10 fold groups have been aligned against the query structure using SSAP, the closest relative can be identified and the corresponding domain excised from the chain. To select this relative, a machine learning approach is used. Both GRATH and SSAP give a good measure of structural similarity but other factors (e.g. alignment overlap, domain size, protein class) can influence the scores. A Support Vector Machine (SVM) (see Section 1.5.2 for a detailed description of SVM technology) is used to combine these features to generate a score that combines these similarity measures.

Once the domain with the best SVM scores has been selected, the SSAP alignment determines the assigned region of the query and removes the appropriate residues from the chain. If the chain still has 5 or more secondary structures it is re-scanned against the 'large library' using GRATH. As before, putative fold matches are selected on the basis of their GRATH E-value and re-aligned using SSAP. SSAP is restricted to ignore any previously aligned residues, preventing realignment of residues already assigned to a domain. Furthermore, excluding these residues increases the speed of the alignment by again limiting the number of residue pairs to be compared.

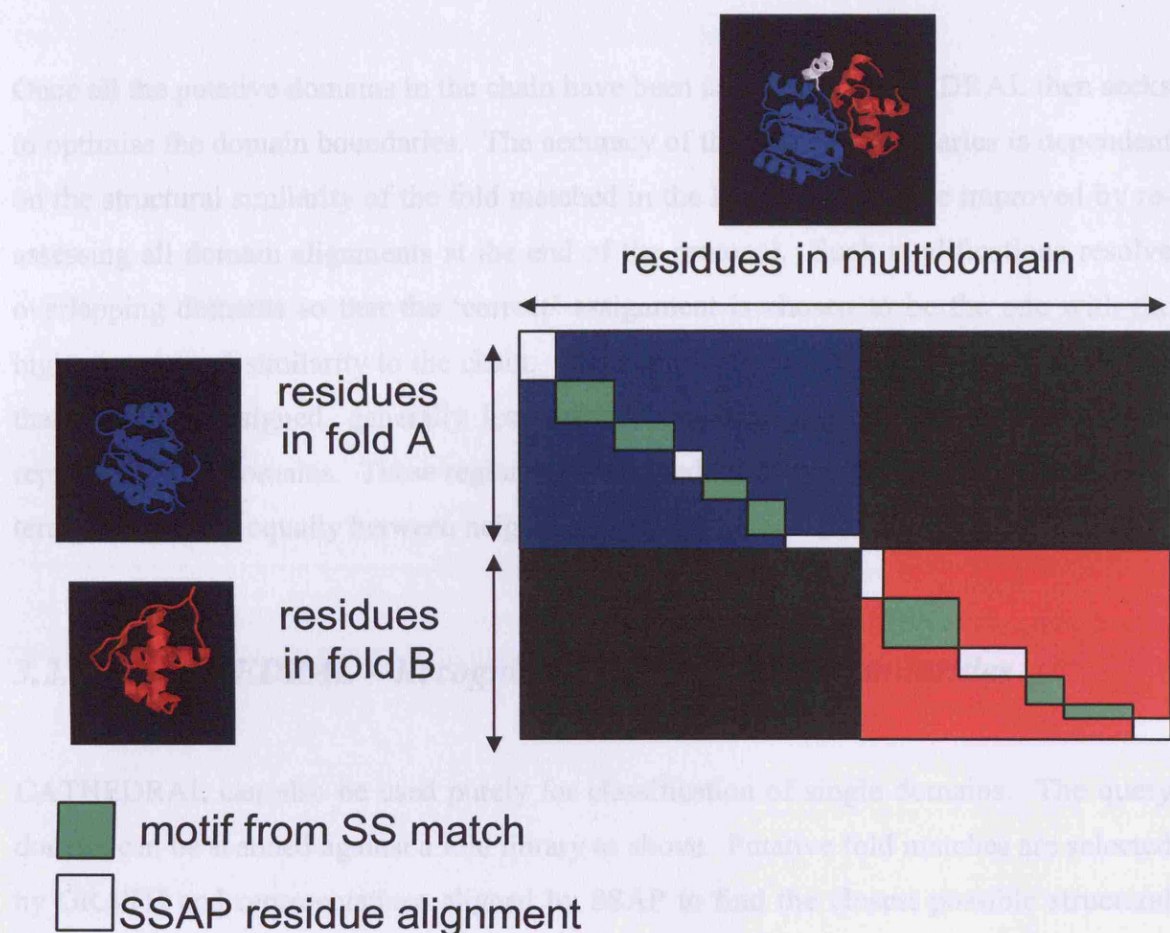


Figure 3.2. Diagram showing how the CATHEDRAL algorithm uses the information from the secondary structure matching step to guide the dynamic programming to find the optimal alignment.

SSAP can occasionally fail to identify discontinuous domains. CATHEDRAL overcomes this limitation by excising and assigning contiguous domains first. SSAP is then far more likely to correctly identify the discontinuous domain correctly once the inserted domain(s) are removed.

CATHEDRAL continues to iteratively assign and excise domains from the chain using the 'large' library with GRATH and SSAP for up to 10 iterations or until there are less than 5 secondary structures left to be assigned. At this point, the remainder of the chain is scanned against the 'small' library using GRATH and then as before the top 10 folds are passed to SSAP to generate residue alignments.

Once all the putative domains in the chain have been assigned, CATHEDRAL then seeks to optimise the domain boundaries. The accuracy of the domain boundaries is dependent on the structural similarity of the fold matched in the library and can be improved by re-assessing all domain alignments at the end of the protocol. Such modifications resolve overlapping domains so that the ‘correct’ assignment is chosen to be the one with the highest structural similarity to the chain. There are also often small regions of the chain that remain unassigned, generally less than 20 residues and so they are unlikely to represent whole domains. These regions are assigned to the nearest domain at the C or N termini, or shared equally between neighbouring domains.

### ***3.2.1.2 CATHEDRAL – Recognising Single Domain Similarities***

CATHEDRAL can also be used purely for classification of single domains. The query domain can be scanned against a fold library as above. Putative fold matches are selected by GRATH and representatives aligned by SSAP to find the closest possible structural match. In benchmarking CATHEDRAL to assess its performance in fold recognition this mode was used. In this mode CATHEDRAL is run with just one iteration and the native SSAP scoring scheme is used instead of the SVM score.

### ***3.2.2 Data Sets Used for Optimising and Benchmarking CATHEDRAL***

A benchmarking protocol was developed to optimise CATHEDRAL and assess its performance in aligning single domain structures against other publicly available structure comparison algorithms. The dataset for the benchmarking of the different structural algorithms encompassed 6003 domains from different sequence families (S35Reps) in CATH v2.6.0, with all domains sharing less than 35% sequence identity from any other so that structural alignment was not trivial. These domains included

representatives from 907 folds from all the four classes of the CATH classification, resulting in over 18 million individual comparisons.

To minimise any bias in the CATH dataset a second dataset that was a subset of CATH v2.6.0 and SCOP v1.65 was also constructed. Each of the 6003 CATH (S35Rep) domains was checked to see if it had an equivalent SCOP domain with at least 80% residue overlap and was in the same SCOP superfamily sharing 80% of the members. This restricted the CATH-SCOP dataset to 1779 sequence diverse domains encompassing 406 folds.

### ***3.2.3 Comparing the Performance of CATHEDRAL in Aligning Single Domain Structures against Other Publicly Available Methods***

CATHEDRAL was benchmarked against a number of other structural comparison methods, SSAP, GRATH, STRUCTAL, DALI, LSQMAN and CE in a number of different ways (see Introduction Section 1.3.2.3 for a description of these methods).

An ‘all against all’ structural comparison was performed of the 6003 unique, sequence diverse CATH domains from v2.6, culminating in over 18 million individual comparisons. The analysis was repeated on the CATH-SCOP dataset thereby removing any bias or circularity in using the CATH database as a golden standard when benchmarking the ‘in-house’ methods, SSAP and GRATH.

All programs took the PDB coordinates of the CATH domains as input and were run in their ‘default’ mode. Each produced a native score, a geometrical measure (e.g. RMSD) and an alignment for the given pair of structures. If a method generated more than one score, the one which produced the optimal performance was chosen for use in the



analysis. Where a method produced more than one alignment for a given pair of protein domains, the alignment with the greatest number of equivalent residues was taken.

### ***3.2.4 Assessing the Performance of Fold Recognition Methods***

#### ***3.2.4.1.1 Comparison of methods using ROC curves***

Structure comparison and fold recognition methods can be analysed using Receiver Operating Characteristic (ROC) curves, which depict the discriminatory power of a scoring scheme with respect to a gold standard classification. A ROC curve is a plot of the true positive rate (sensitivity) against the false positive rate (1-specificity), over a range of possible score cut-offs. To assess the performance of the structure comparison methods, a true match is defined as two domains which share the same CATH topology (fold) assignment. The alignments for each method are ordered by their respective scoring scheme and at varying thresholds the number of true positives and false positives are calculated for the ROC curve.

Kolodny *et al.* (2005) previously showed that a new geometric score (SAS see Section 3.2.4.1.3) was more effective at distinguishing between all true and false positives than most of the native scores calculated by each algorithm. Therefore, ROC curves were also plotted using the geometric scores (e.g. SAS) to assess whether this was also true for this data set.

ROC curves were calculated for all methods using the whole CATH dataset and the CATH-SCOP subset to assess the performance of each method for detecting all true fold relationships.

#### ***3.2.4.1.2 Assessing the Performance of the Structure Comparison Methods in Ranking the Correct Fold Matches***

For the purpose of domain boundary recognition it is more important to recognise the closest structural match rather than identifying all true relationships as with the ROC curve analysis. Therefore this approach best represents how structure comparison methods would be used in a classification protocol. For example a query domain is scanned against a library of classified domains and the best matches provide the curator with evidence of the domain fold.

Indeed, another way of measuring the performance of the structure comparison methods is to calculate how well they identify the correct fold match as the best hit. For each query domain, the matches are sorted based on the appropriate score and the rank of the correct fold is calculated. A graph was plotted showing the cumulative percentage of correctly assigned folds at each descending rank. These graphs were generated for both the CATH and CATH-SCOP dataset.

#### ***3.2.4.1.3 Assessing the Performance of the Structure Comparison Methods by Measuring the Geometric Quality of the Alignments Using Common Geometric Scoring Schemes***

In addition to assessing the ability of each structure comparison method to recognise correct fold similarities, it is also informative to consider the geometric quality of the alignment itself.

All the methods produce a geometric measure (RMSD) of how well two given structures can be superposed following alignment and related domains should have a low RMSD. As stated in the Section 3.1, RMSD does not take into account the size of the proteins aligned, or the number of residues matched in the alignment. Therefore, alignments that only include highly structurally equivalent residues will produce a small RMSD on superposition even if they only align a small domain to a motif within a much larger



domain. In this study the quality of the alignments are explored in terms of properties that are associated with a good global alignment. Such properties include a low RMSD, the number of aligned residues and the fraction of the smallest or largest protein included in the alignment.

For each correct fold match, three common geometric scoring schemes were evaluated to compare the quality of the alignment produced by each method. The methods used included SAS (Equation 3.1),  $SI_{MAX}$  (Equation 3.2) and  $SI_{MIN}$  (Equation 3.3). The different measures attempt to capture a ‘good’ global alignment by normalising the RMSD by the length of the alignment as a fraction of the size of the proteins aligned. All the measurements are in Å and the percentage of alignments within a particular threshold was plotted for each measure.

$$SAS = RMSD \times 100 / N$$

**Equation 3.1. SAS (Structural Alignment Score). N represents the number of aligned residues.**

$$SI_{MAX} = RMSD \times \max(L_1, L_2) / N$$

**Equation 3.2.  $SI_{MAX}$ , N represents the number of aligned residues, and  $L_1, L_2$  the number of residues in the respective domains.**

$$SI_{MIN} = RMSD \times \min(L_1, L_2) / N$$

**Equation 3.3.  $SI_{MIN}$ , N represents the number of aligned residues, and  $L_1, L_2$  the number of residues in the respective domains.**

#### ***3.2.4.1.4 Assessing Alignment Quality***

A set of manually curated alignments (BALiBASE) was also used to validate the quality of the structural alignments produced by each of the structure comparison methods. BALiBASE (Thompson et al. 1999) is a database of manually-refined multiple alignments specifically designed for the evaluation and comparison of multiple sequence alignment programs. The sequences included in the database are selected from alignments in either the FSSP (Holm et al. 1992) or HOMSTRAD (Mizuguchi et al. 1998) structural

databases, or from manually constructed structural alignments taken from the literature. When sufficient structures are not available, additional sequences are included from the HSSP database (Schneider et al. 1997). The VAST Web server (Madej et al. 1995) is used to confirm that the sequences in each alignment are structural neighbours and can be structurally superimposed. Functional sites are identified using the PDBsum database (Laskowski et al. 1997) and the alignments are manually verified to ensure that conserved residues and secondary structures are correctly aligned.

Fourteen BaliBase multiple alignments were chosen, comprising 108 individual pairwise alignments. The alignments were restricted to protein chains sharing less than 25% sequence identity making alignment non-trivial and the dataset covered all three major structural classes (mainly  $\alpha$ , mainly  $\beta$  and  $\alpha\beta$ ). The alignments produced by the structural comparison algorithms can be compared against the manually curated BaliBase alignments and the quality of the alignments generated by the different methods measured by the score,  $f_m$  (Sauder et al. 2000).  $f_m$  is defined as the number of amino acids correctly aligned in the structural alignment divided by the total number of aligned residues in the BaliBase alignment.

#### ***3.2.4.2 Assessing the performance of CATHEDRAL for assigning domains to Multi-Domain protein chains***

The previous sections dealt with the benchmarking of the CATHEDRAL protocol as regards its ability to recognise the correct fold of a query domain in a library of domain folds. Here, the ability of CATHEDRAL algorithm to recognise individual domains in a multi-domain structure is assessed and the protocol for optimising this algorithm is described.

This protocol required the optimisation of (1) the alignment using an SVM, (2) a score threshold for accurately identifying domains and (3) the accuracy of domain boundary assignment.

The dataset used to benchmark CATHEDRAL for recognising domain boundaries was created from a set of 1071 non-redundant (at 35% sequence identity) representatives (S35Reps) from multi-domain sequence families. From this set, those chains containing domains from folds with less than 2 S35Reps were removed leaving a final dataset of 680 multi-domain chains, containing 1593 domains, 245 unique folds and 462 unique superfamilies from all 4 classes of the CATH hierarchy.

No other publicly available structure comparison algorithm has been developed for explicitly recognising domains within multi-domain proteins by fold recurrence. However, there are several sequence based approaches that perform this task. Therefore, to place the performance of CATHEDRAL's domain boundary recognition in context the 680 chains were also scanned against Hidden Markov Models built from each structure in the CATH dataset. The HMMer suite (Eddy 1996) was used to build the models from each sequence in the CATH dataset. These models were subsequently scanned against the protein chains and domain boundaries assigned based on the top ranking HMM match defined by the HMMer E-value.

## **3.3 Results**

### **3.3.1 Structure Comparison Methods – Assessing the Performance of CATHEDRAL in Recognising the Correct Fold**

There are a range of different criteria to take into account when assessing the performance of a structural comparison method. It is important that the method can accurately score an alignment so that similar (e.g. related at the fold level) structures have higher scores than dissimilar proteins. Another important criterion to assess is the ability of the algorithm to produce biologically meaningful alignments.

#### **3.3.1.1 Benchmarking the Structure Comparison Methods for Fold Recognition**

The performance of CATHEDRAL was benchmarked against other widely used structural comparisons methods namely, SSAP, DALI, STRUCTAL, LSQMAN, CE and GRATH. The results were analysed in several ways to gauge the ability of the methods to identify fold similarities and produce high quality structural alignments.

Firstly, the ability of the methods to recognise fold similarities was assessed based on their scoring schemes. Some methods produce both a raw score of similarity and also a statistical score. In these cases the score that produced the best ROC curves was used. For SSAP this was the native score, for GRATH the E-value, for CE the Z-score, for DALI the Z-score, for LSQMAN its native score and STRUCTAL its native score.

Figure 3.3 shows the ROC curve derived for each method against both the CATH (Figure 3.3(a)) and CATH-SCOP (Figure 3.3(b)) datasets. The uppermost ROC curve represents the method whose scoring scheme best reproduces the true fold matches presented in CATH and therefore best discriminates between false and true positives. It is imperative

to minimise the number of errors introduced into a classification therefore for the purpose of this benchmark we are especially interested in the methods performance at low error rates. Therefore each method shall be assessed on the basis of its coverage achieved at a 5% error rate rather than the area under the whole ROC curve. For all methods except STRUCTAL, a similar performance is observed on both the CATH and CATH-SCOP datasets, though in some cases better performances are observed on the CATH-SCOP dataset. As the CATH-SCOP dataset represents the agreement between the two classifications, it is reasonable to expect that the dataset will contain fewer errors than the larger CATH dataset and will also contain more easily classified structures, this may account for the increase in performance for all the methods.

The method that shows the greatest increase in performance when using the CATH-SCOP dataset is SSAP. This is surprising as it is the main structural comparison algorithm used in the classification protocol of the CATH database. However, this does suggest that there is no bias towards SSAP or CATHEDRAL when using CATH as the gold standard for benchmarking. The CATHEDRAL algorithm appears to accumulate false positives 'earlier' with the CATH-SCOP dataset. This may be due to the increase in small proteins in this dataset which the filtering step of CATHEDRAL removes.

The top performing methods on both datasets, for a 5% error rate, are CATHEDRAL and DALI, both achieving coverage of nearly 80%. STRUCTAL achieves the second highest coverage of 72% for a 5% error. SSAP and LSQMAN are the next best performing methods, with LSQMAN outperforming SSAP on the CATH dataset. CE is the worst performing method on both datasets.

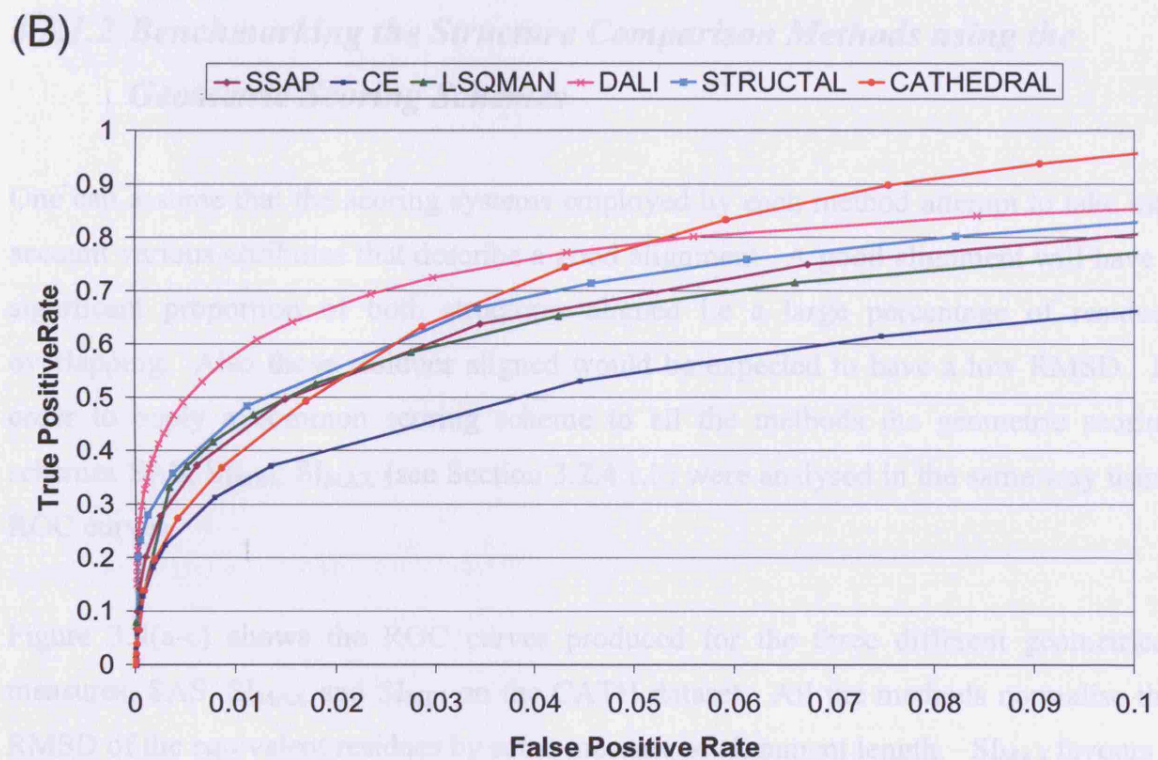
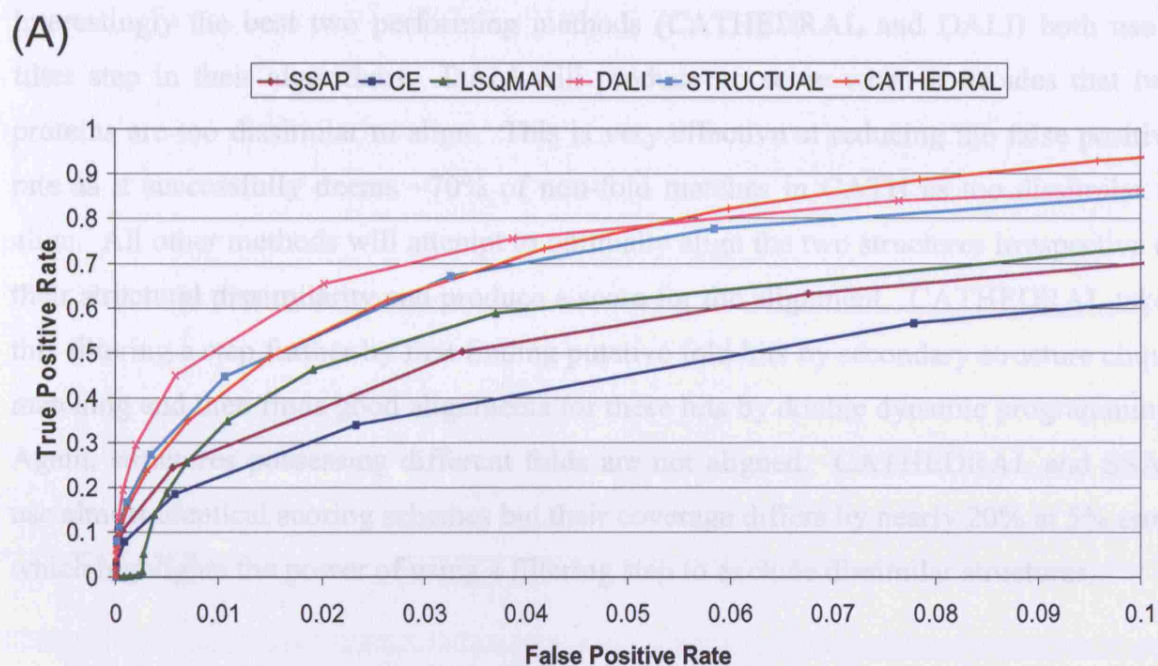


Figure 3.3(a-b). ROC curves showing the performance of the 6 structural comparisons methods in their ability to identify correct fold matches. (a) is the performance on the CATH dataset, (b) is the performance on the CATH-SCOP dataset (see methods).

Interestingly the best two performing methods (CATHEDRAL and DALI) both use a filter step in their algorithms. DALI will produce no score when it decides that two proteins are too dissimilar to align. This is very effective at reducing the false positive rate as it successfully deems ~70% of non-fold matches in CATH as too dissimilar to align. All other methods will attempt to optimally align the two structures irrespective of their structural dissimilarity and produce a score for the alignment. CATHEDRAL takes this filtering a step further by first finding putative fold hits by secondary structure clique matching and then finds good alignments for these hits by double dynamic programming. Again, structures possessing different folds are not aligned. CATHEDRAL and SSAP use almost identical scoring schemes but their coverage differs by nearly 20% at 5% error which highlights the power of using a filtering step to exclude dissimilar structures.

### ***3.3.1.2 Benchmarking the Structure Comparison Methods using the Geometric Scoring Schemes***

One can assume that the scoring systems employed by each method attempt to take into account various attributes that describe a good alignment. A good alignment will have a significant proportion of both structures aligned i.e a large percentage of residues overlapping. Also those residues aligned would be expected to have a low RMSD. In order to apply a common scoring scheme to all the methods the geometric scoring schemes SAS,  $SI_{MIN}$ ,  $SI_{MAX}$  (see Section 3.2.4.1.3) were analysed in the same way using ROC curves.

Figure 3.4(a-c) shows the ROC curves produced for the three different geometrical measures, SAS,  $SI_{MAX}$  and  $SI_{MIN}$  on the CATH dataset. All the methods normalise the RMSD of the equivalent residues by some measure of alignment length.  $SI_{MAX}$  favours a global alignment between two structures whilst  $SI_{MIN}$  will also give good scores for matches that represent a more local alignment.



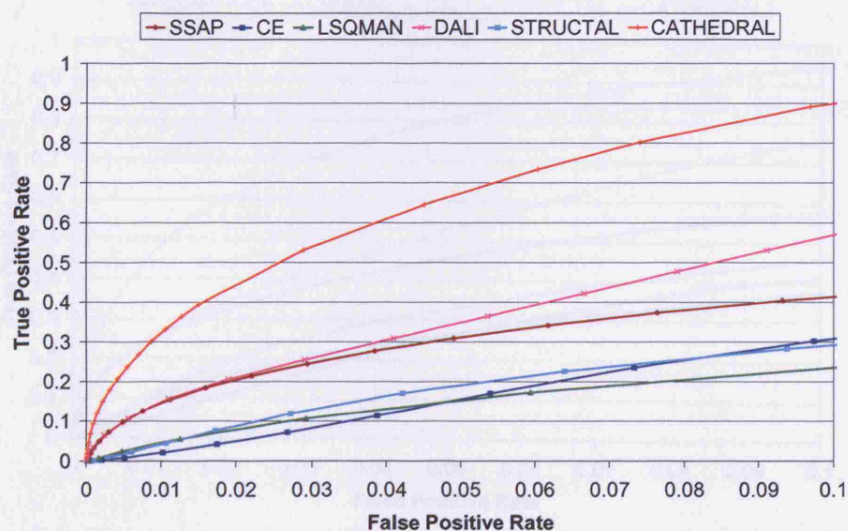
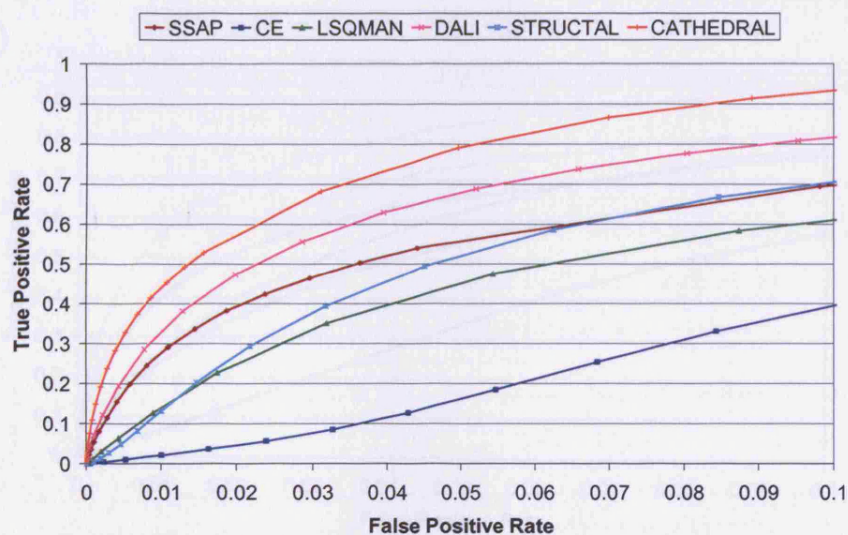
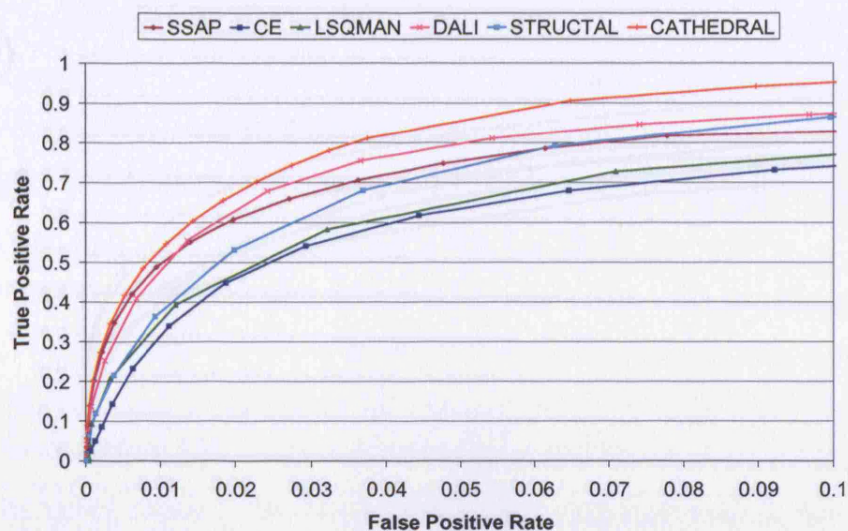


Figure 3.4(a), (b), (c). ROC curves plotted for different structural comparison methods based on geometric scores where a positive match represents a true fold match. Plot (a) is based on the  $R_{\text{MSD}}$  score, (b) is based on the  $R_{\text{TM}}$  score, and (c) is based on the  $R_{\text{TM}}$  score.

(A)

(B)

(C)



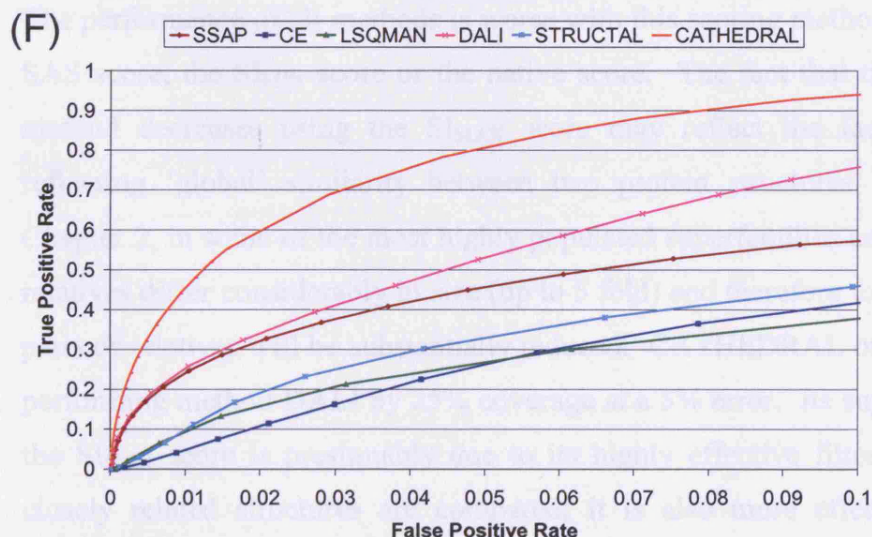
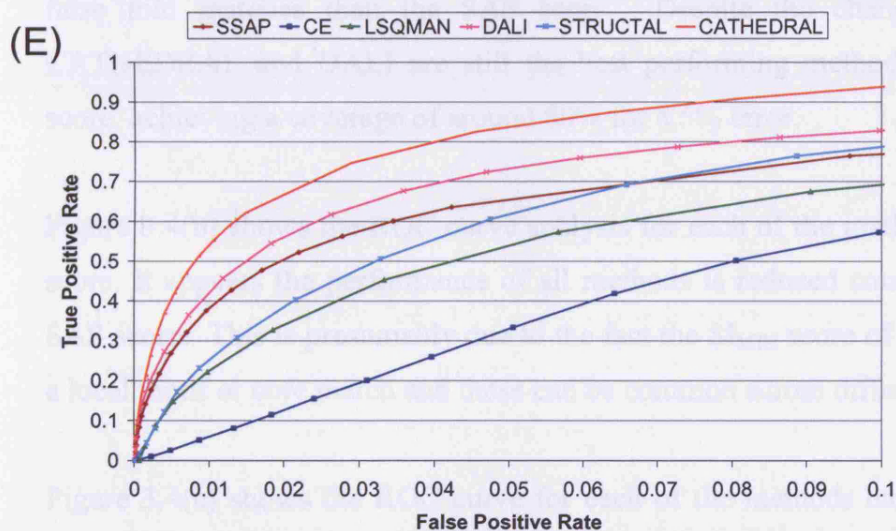
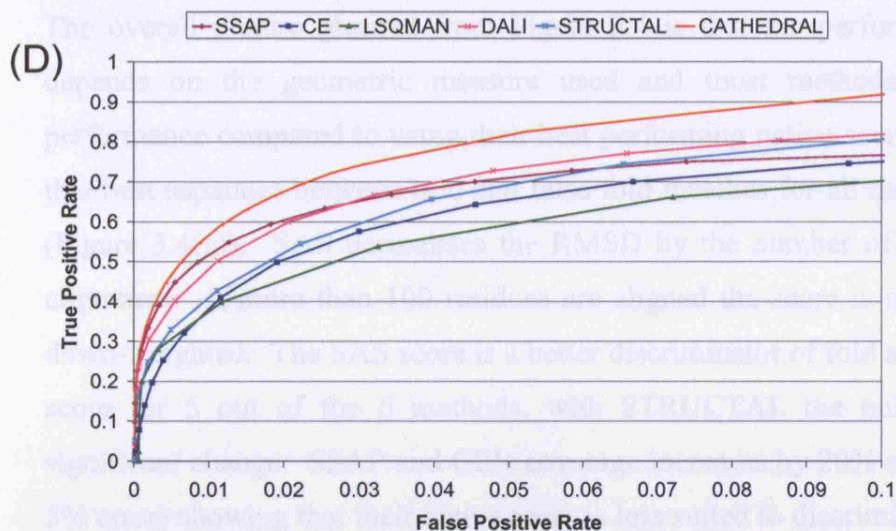


Figure 3.4(a,b,c,d,e,f). ROC curves plotted for different structural comparison methods based on geometric scores where a positive match represents a true fold match. Plot 2a is based on the SAS score, 2b is based on the  $SI_{MIN}$  score and 2c is based on the  $SI_{MAX}$  score for the CATH dataset. Plot 2d is based on the SAS score, 2e is based on the  $SI_{MIN}$  score and 2f is based on the  $SI_{MAX}$  score for the CATH-SCOP dataset

The overall picture gleaned from Figure 3.4 is that the performance of each method depends on the geometric measure used and most methods show an increase in performance compared to using their best performing native score. The geometric score that best separates between true and false fold matches for all methods is the SAS score (Figure 3.4(a)). SAS normalises the RMSD by the number of aligned residues in the alignment. If more than 100 residues are aligned the score is up-weighted, if less it is down-weighted. The SAS score is a better discriminator of fold similarity than the native score for 5 out of the 6 methods, with STRUCTAL the only method showing no significant change. SSAP and CE's coverage increases by 20% and 17% respectively (at 5% error) showing that their native score is less suited to discriminating between true and false fold matches than the SAS score. Despite the change in scoring scheme, CATHEDRAL and DALI are still the best performing methods when using the SAS score, achieving a coverage of around 80% for a 5% error.

Figure 3.4(b) shows the ROC curve analysis for each of the methods based on the  $SI_{MIN}$  score. It appears the performance of all methods is reduced compared to the use of the SAS score. This is presumably due to the fact the  $SI_{MIN}$  score of an alignment represents a local motif or core match and these can be common across different folds.

Figure 3.4(c) shows the ROC curve for each of the methods based on the  $SI_{MAX}$  score. The performance of all methods is worse with this scoring method when compared to the SAS score, the  $SI_{MIN}$  score or the native score. The fact that the performance of each method decreases using the  $SI_{MAX}$  score may reflect the fact that this measure is reflecting 'global' similarity between two protein structures. As was discussed in Chapter 2, in some of the most highly populated superfamilies and fold groups in CATH relatives differ considerably in size (up to 5 fold) and therefore the  $SI_{MAX}$  scores for these pairs of relatives will be substantially reduced. CATHEDRAL outperforms the next best performing method DALI by 25% coverage at a 5% error. Its superior performance with the  $SI_{MAX}$  score is presumably due to its highly effective filter step that ensures only closely related structures are compared; it is also more effective at matching large alignments, and the  $SI_{MAX}$  score is designed to favour this characteristic.

Figure 3.4(d,e,f) shows the performance of each of the methods using the geometric scores on the CATH-SCOP dataset. The ranking of the methods is the same for the CATH dataset, with CATHEDRAL and DALI again appearing as the best performing methods when using the SAS,  $SI_{MIN}$  or  $SI_{MAX}$  scores. All the methods show a significant increase in coverage when using the  $SI_{MIN}$  or  $SI_{MAX}$  scores on the CATH-SCOP dataset compared to the CATH dataset. A possible explanation for this is the fact that the union of the two databases encourages consistent classifications. Since SCOP is devised using largely manual validation some very diverse relatives (i.e. differing greatly in size) may have been missed and hence there may be less deviation in sizes of relatives in the CATH-SCOP dataset.

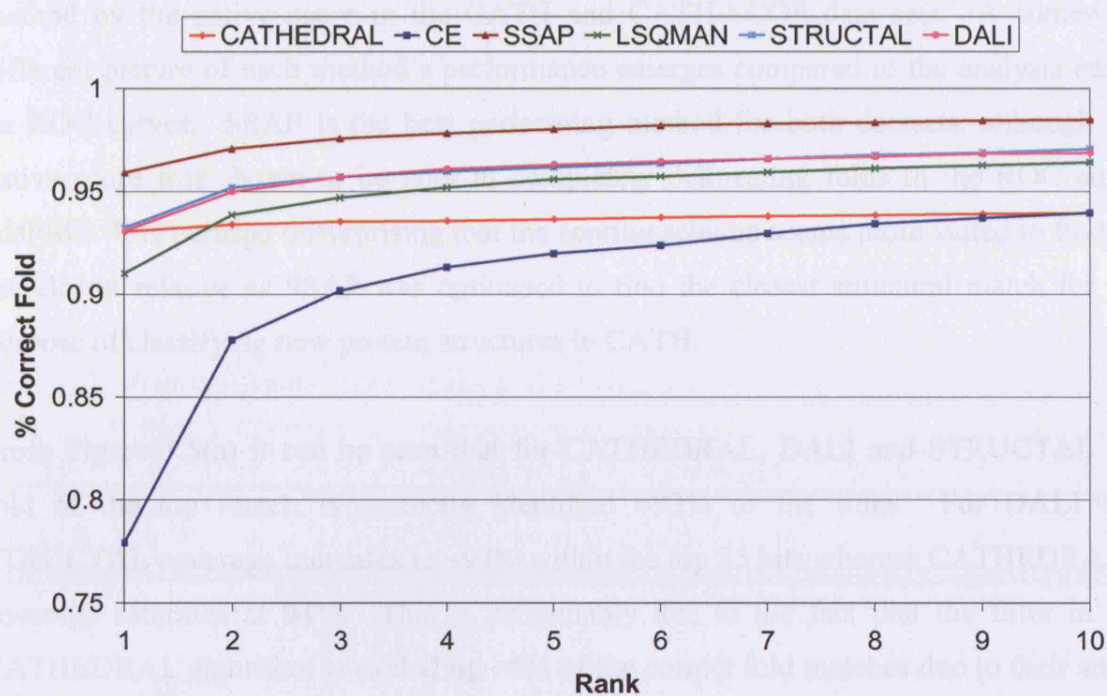
For nearly all methods the SAS score best discriminates between true and false fold matches giving 85% coverage for CATHEDRAL at a 5% error rate on the CATH dataset.

### ***3.3.1.3 Performance of CATHEDRAL in Ranking Correct Fold Matches.***

Another way to evaluate the ability of the methods is to identify the correct fold as the top match when comparing a structure against a library of all possible folds. That is, how effective are the scores at recognising the most closely related structure? In general ROC curves show how closely a particular scoring scheme can replicate an existing classification, whereas considering the rank shows how effective a method or scoring scheme is at correctly classifying a domain using a nearest neighbour approach.

Each domain in the data-set was compared against the fold library, the results were sorted by each score and the frequency of producing a correct match at a particular rank is calculated over all the query domains. The cumulative percentage of correctly assigned folds can then be plotted at each descending rank.

(A)



(B)

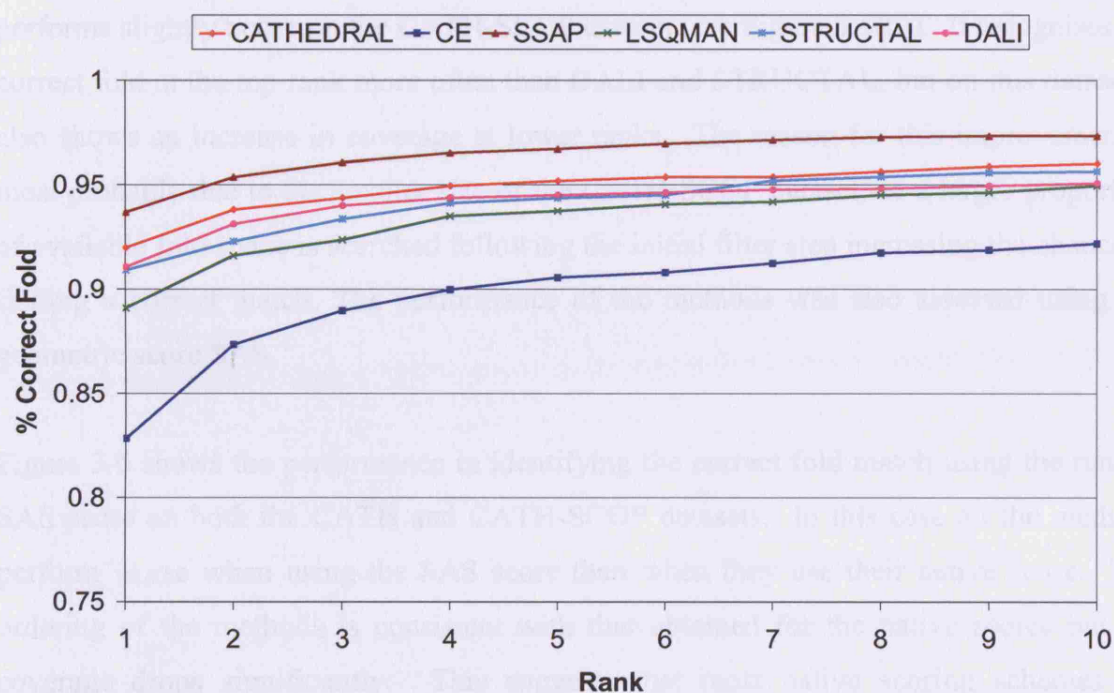


Figure 3.5(a,b). Plot of the percentage of correct folds matched against the ranked native score for the (a) CATH and (b) CATH-SCOP dataset.

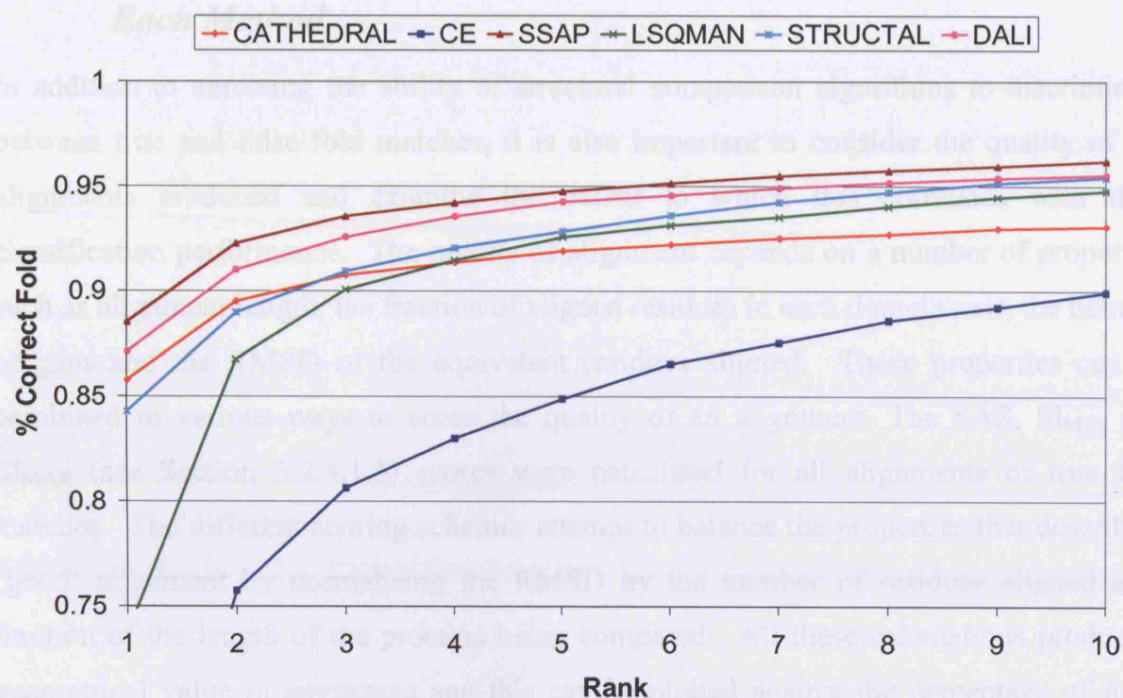
Figure 3.5 shows the position of the correct folds when ranking the results for each method by the native score in the CATH and CATH-SCOP data sets. A somewhat different picture of each method's performance emerges compared to the analysis based on ROC curves. SSAP is the best performing method for both datasets, although the native score was shown to be poor at completely delineating folds in the ROC curve analysis. It is perhaps unsurprising that the scoring scheme seems more suited to finding the closest relative as SSAP was optimised to find the closest structural match for the purpose of classifying new protein structures in CATH.

From Figure 3.5(a) it can be seen that for CATHEDRAL, DALI and STRUCTAL the fold of the top match is correctly identified ~93% of the time. For DALI and STRUCTAL coverage increases to ~97% within the top 25 hits whereas CATHEDRAL's coverage saturates at 94%. This is presumably due to the fact that the filter in the CATHEDRAL algorithm is excluding ~6% of the correct fold matches due to their small size. Only a minor improvement is seen past the top rank, which implies that if the correct fold passes the initial filter it will be placed at the top rank. CATHEDRAL performs slightly better on the CATH-SCOP dataset (see Figure 3.5(b)). It recognises the correct fold at the top rank more often than DALI and STRUCTAL, but on this dataset it also shows an increase in coverage at lower ranks. The reason for this improvement is most probably due to the smaller size of the CATH-SCOP dataset, so a larger proportion of available fold space is searched following the initial filter step increasing the chance of finding a correct match. The performance of the methods was also assessed using the geometric score SAS.

Figure 3.6 shows the performance in identifying the correct fold match using the ranked SAS score on both the CATH and CATH-SCOP datasets. In this case all the methods perform worse when using the SAS score than when they use their native score. The ordering of the methods is consistent with that obtained for the native scores but the coverage drops significantly. This suggests that most native scoring schemes are optimised to recognise the closest relative rather than to optimally distinguish members of different folds.



(A)



(B)

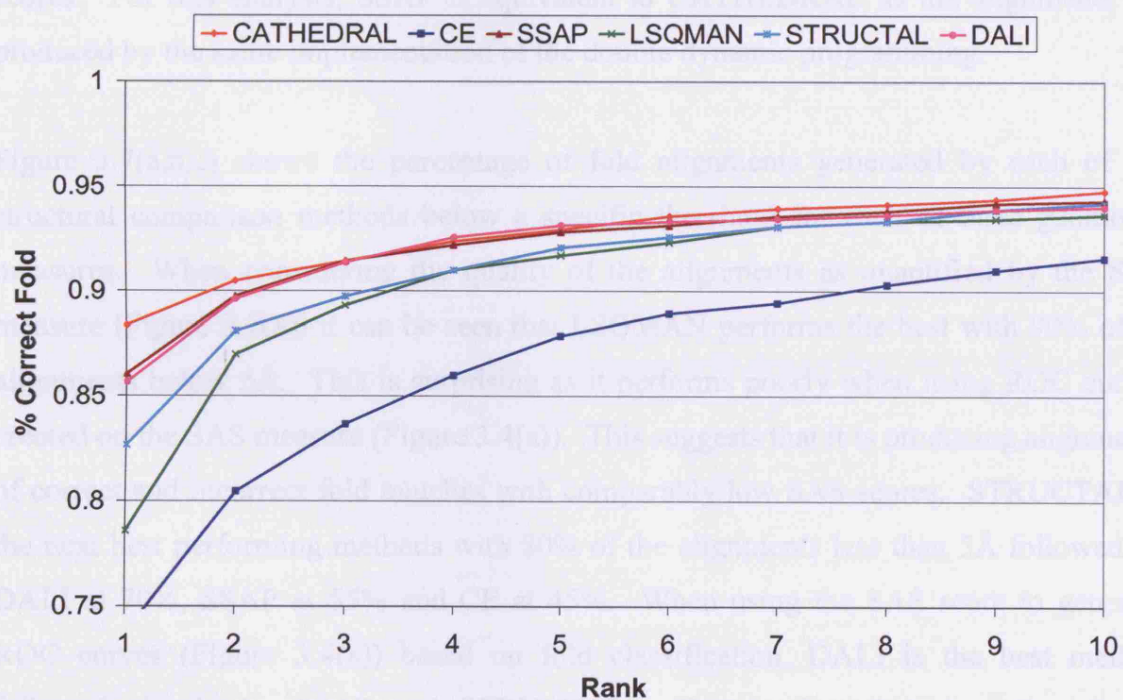


Figure 3.6(a,b). Plot of the percentage of correct folds matched against the SAS score for the (a) CATH and (b) CATH-SCOP dataset.

### ***3.3.1.4 Assessing the Quality of the Structural Alignments Produced by Each Method***

In addition to assessing the ability of structural comparison algorithms to discriminate between true and false fold matches, it is also important to consider the quality of the alignments produced and examine the extent to which this correlates with their classification performance. The quality of alignment depends on a number of properties such as alignment length, the fraction of aligned residues in each domain pair, the number of gaps and the RMSD of the equivalent residues aligned. These properties can be combined in various ways to score the quality of an alignment. The SAS,  $SI_{MIN}$  and  $SI_{MAX}$  (see Section 3.2.4.1.3) scores were calculated for all alignments of true fold matches. The different scoring schemes attempt to balance the properties that describe a ‘good’ alignment by normalising the RMSD by the number of residues aligned as a fraction of the length of the proteins being compared. All these calculations produce a geometrical value in angstroms and this can be plotted against the percentage of folds with alignments at a particular threshold. Higher quality alignments should have lower scores. For this analysis, SSAP is equivalent to CATHEDRAL as the alignments are produced by the same implementation of the double dynamic programming.

Figure 3.7(a,b,c) shows the percentage of fold alignments generated by each of the structural comparison methods below a specific threshold for each of three geometric measures. When considering the quality of the alignments as quantified by the SAS measure (Figure 3.7(a)) it can be seen that LSQMAN performs the best with 90% of its alignments below 5Å. This is surprising as it performs poorly when using ROC curves created on the SAS measure (Figure 3.4(a)). This suggests that it is producing alignments of correct and incorrect fold matches with comparably low SAS scores. STRUCTAL is the next best performing methods with 80% of the alignments less than 5Å followed by DALI at 70%, SSAP at 55% and CE at 45%. When using the SAS score to generate ROC curves (Figure 3.4(a)) based on fold classification, DALI is the best method followed closely by SSAP and STRUCTAL. This implies that even though the alignments produced by SSAP and DALI give higher SAS scores than LSQMAN and STRUCTAL they are still better able to discriminate between true and false fold matches.



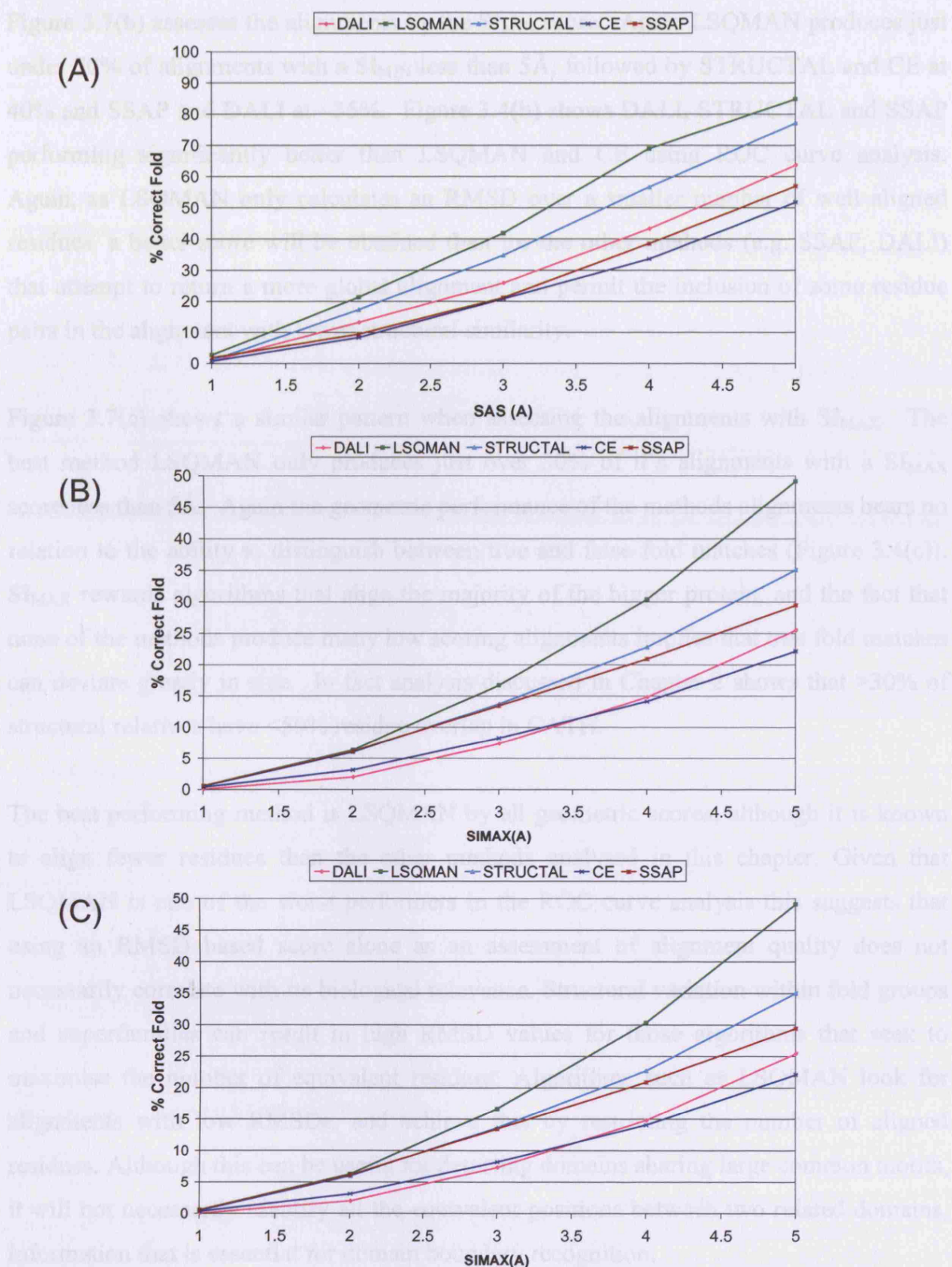


Figure 3.7. Plot showing the percentage of fold alignments within a particular threshold for the SAS(a), SI<sub>MIN</sub>(b) and SI<sub>MAX</sub>(c) measures

Figure 3.7(b) assesses the alignments by the  $SI_{MIN}$  score. Again LSQMAN produces just under 70% of alignments with a  $SI_{MIN}$  less than 5Å, followed by STRUCTAL and CE at 40% and SSAP and DALI at ~35%. Figure 3.4(b) shows DALI, STRUCTAL and SSAP performing significantly better than LSQMAN and CE using ROC curve analysis. Again, as LSQMAN only calculates an RMSD over a smaller number of well aligned residues, a better score will be obtained than for the other methods (e.g. SSAP, DALI) that attempt to return a more global alignment and permit the inclusion of some residue pairs in the alignment with lower structural similarity.

Figure 3.7(c) shows a similar pattern when assessing the alignments with  $SI_{MAX}$ . The best method LSQMAN only produces just over 30% of its alignments with a  $SI_{MAX}$  score less than 5Å. Again the geometric performance of the methods alignments bears no relation to the ability to distinguish between true and false fold matches (Figure 3.4(c)).  $SI_{MAX}$  rewards algorithms that align the majority of the bigger protein, and the fact that none of the methods produce many low scoring alignments implies that true fold matches can deviate greatly in size. In fact analysis discussed in Chapter 2 shows that >30% of structural relatives have <50% residue overlap in CATH.

The best performing method is LSQMAN by all geometric scores, although it is known to align fewer residues than the other methods analysed in this chapter. Given that LSQMAN is one of the worst performers in the ROC curve analysis this suggests that using an RMSD based score alone as an assessment of alignment quality does not necessarily correlate with its biological relevance. Structural variation within fold groups and superfamilies can result in high RMSD values for those algorithms that seek to maximise the number of equivalent residues. Algorithms such as LSQMAN look for alignments with low RMSDs, and achieve this by restricting the number of aligned residues. Although this can be useful for detecting domains sharing large common motifs, it will not necessarily identify all the equivalent positions between two related domains, information that is essential for domain boundary recognition.

In some domain architectures (e.g. 3 layer  $\alpha\beta$  sandwiches) it is clear that large structural motifs (e.g.  $\beta\alpha\beta\alpha$ ) can overlap between domains but that these do not always coincide with equivalent secondary structures. Furthermore, any similarity score based on RMSD will be dependent on the number of superposed residues and hence aligning more residues over more variable parts of two structures, can give a disproportionately high RMSD value, even if the alignment is actually more biologically valid. For the purposes of domain boundary recognition, it is clearly important to identify an alignment between two domains that maximises the number of equivalent residues. Consequently, for a given pair of fold relatives the average number of residues aligned by each method was also considered. Table 3.1 shows this calculation relative to SSAP, as SSAP aligns more residues than all other methods. There appears to be some correlation between this average value and the SAS ROC curves shown in Figures 3.4 and 3.5. More specifically, DALI, STRUCTAL and SSAP align the most residues and also perform best by the ROC curve analysis.

	SSAP	LSQMAN	DALI	STRUCTAL	CE
<b>Percentage of aligned residues with respect to SSAP</b>	N/A	50	75	76	57
<b>Best Performing Method at 5% Error Using SAS Score (Figure 3.4(a))</b>	2	4	1	3	5

**Table 3.1** The percentage of residues aligned by each method relative to SSAP for all genuine fold matches. Defined by the formula below:

$$\frac{\text{Number of residues aligned by method}}{\text{Number of Residues Aligned by SSAP}} \times 100\%$$



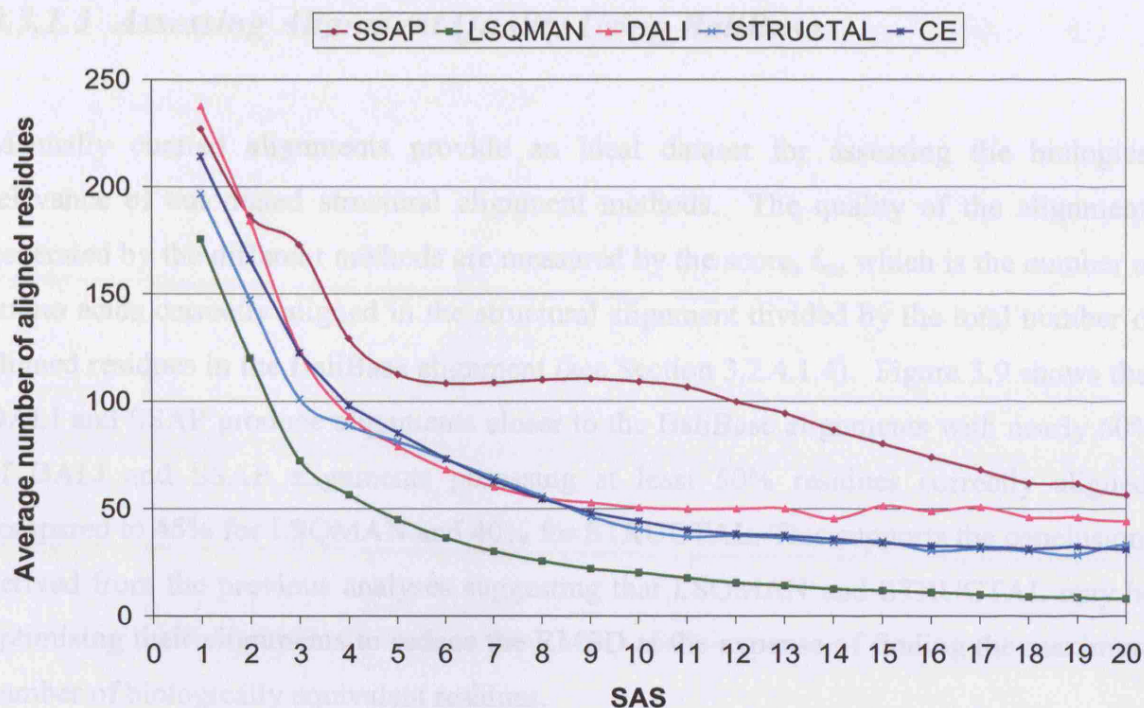


Figure 3.8. Average number of aligned residues for a given SAS score.

The size of the domain fragments aligned for a given SAS score was also determined. Figure 3.8 shows that although LSQMAN and STRUCTAL are returning a higher proportion of scores below a given geometric threshold, they are recognising and aligning fewer residues. This may be valuable for finding the most superposable motifs between two domains; however, it is less useful for assigning domain boundaries. Taken together the ROC curve analysis (Figure 3.4), ranking analysis and the studies on the geometrical quality of the alignments (Figure 3.7 and Figure 3.8) suggests that SSAP/CATHEDRAL is a very appropriate method to use for domain boundary recognition, as it searches for largest fragments whilst ensuring these can be superimposed with reasonable scores.

### 3.3.1.5 Assessing Alignment Quality Using BaliBase

Manually curated alignments provide an ideal dataset for assessing the biological relevance of automated structural alignment methods. The quality of the alignments generated by the different methods are measured by the score,  $f_m$ , which is the number of amino acids correctly aligned in the structural alignment divided by the total number of aligned residues in the BaliBase alignment (see Section 3.2.4.1.4). Figure 3.9 shows that DALI and SSAP produce alignments closer to the BaliBase alignments with nearly 60% of DALI and SSAP alignments possessing at least 50% residues correctly aligned, compared to 45% for LSQMAN and 40% for STRUCTAL. This supports the conclusions derived from the previous analyses suggesting that LSQMAN and STRUCTAL may be optimising their alignments to reduce the RMSD at the expense of finding the maximum number of biologically equivalent residues.

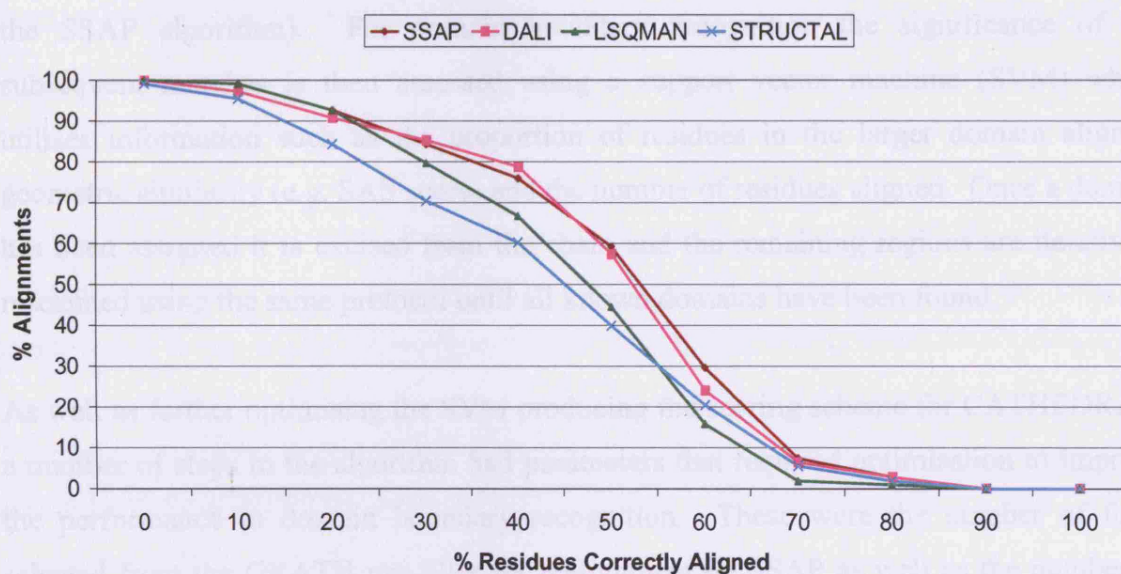


Figure 3.9. Graph showing how the alignments of each method compared to manually validated BaliBase alignments. The curve with the greatest area underneath represents the method that most agrees with the manually curated BaliBase alignments.

### ***3.3.2 Optimising CATHEDRAL to identify domains within Multi-domain Protein Structures***

A vital part of protein structure classification protocols is the recognition of domains within protein chains. The benchmarking analysis presented in this chapter has shown that CATHEDRAL is a fast and reliable algorithm for comparing protein domains and identifying the correct fold. This section describes the optimisation and performance of CATHEDRAL for domain boundary recognition.

As described in the Section 3.2, CATHEDRAL uses a graph theoretical approach (GRATH) as a pre-filter to find putative folds within the protein chain. Representatives from each superfamily in the putative folds are then aligned against the query chain using the slow, but more accurate, double dynamic programming algorithm (DDP) (as used by the SSAP algorithm). For domain boundary recognition the significance of the subsequent matches is then assessed using a support vector machine (SVM) which utilises information such as the proportion of residues in the larger domain aligned, geometric similarity (e.g. SAS score) and the number of residues aligned. Once a domain has been assigned it is excised from the chain and the remaining regions are iteratively rescanned using the same protocol until all known domains have been found.

As well as further optimising the SVM producing the scoring scheme for CATHEDRAL, a number of steps in the algorithm had parameters that required optimisation to improve the performance in domain boundary recognition. These were the number of folds selected from the GRATH pre-filter for realignment by SSAP as well as the number of representatives from each fold group that should be aligned by SSAP in order to find the closest structural relative.

### ***3.3.2.1 Optimising the CATHEDRAL Scoring Scheme Using an SVM***

As described above many factors are important for recognising global similarity between two domains. It is important to consider both geometric properties of the alignment such as the RMSD as well as to the percentage of aligned residues in the largest protein. This problem becomes more complex when detecting individual domains in a multi-domain chain, as the “real” length of the largest protein domain may be unknown. This could lead to the problem of misclassification caused by high scores being returned by small domains simply matching with secondary structure motifs in a larger domain.

The highest scoring match identified by CATHEDRAL is used to determine the boundaries of the domain. Residues associated with that domain are then removed from the protein chain which is then rescanned against the fold library. It is therefore imperative that the top match represents the best global match. An SVM provides a convenient method to combine useful independent indicators of alignment quality to more accurately rank potential fold matches to a query chain. The features that were used as inputs to the SVM were the GRATH score, the GRATH clique size, the SSAP score, the proportion of residues in the larger domain that has been aligned, the RMSD, the number of aligned residues and the SAS score. The SAS score was used rather than the  $SI_{MAX}$  because the length of the larger domain in the multidomain structure is unknown.

Five fold cross validation was used to ensure un-biased training of the SVM leading to a generalised classifier. This process involves splitting the dataset into 5 sets, and each set is successively taken as the test set while the remaining 4 sets represent the training set. The performance is then calculated on the average over the 5 test sets.

Figure 3.10 shows the performance of the SVM score for assigning domains within multi domain structures, compared to the other scoring schemes. The ROC curve benchmark shows an average value for the SVM performance using 5-fold cross validation. It can be seen that the SVM score significantly outperforms all other scoring schemes achieving 75% coverage for a 5% error.



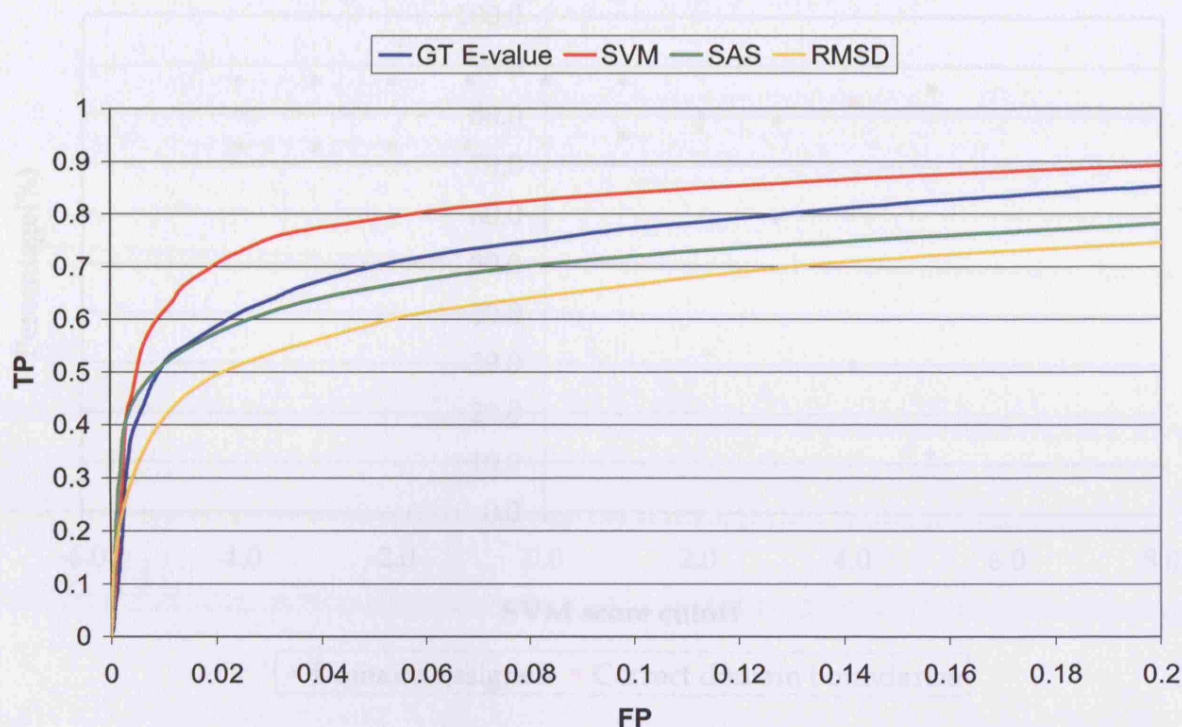
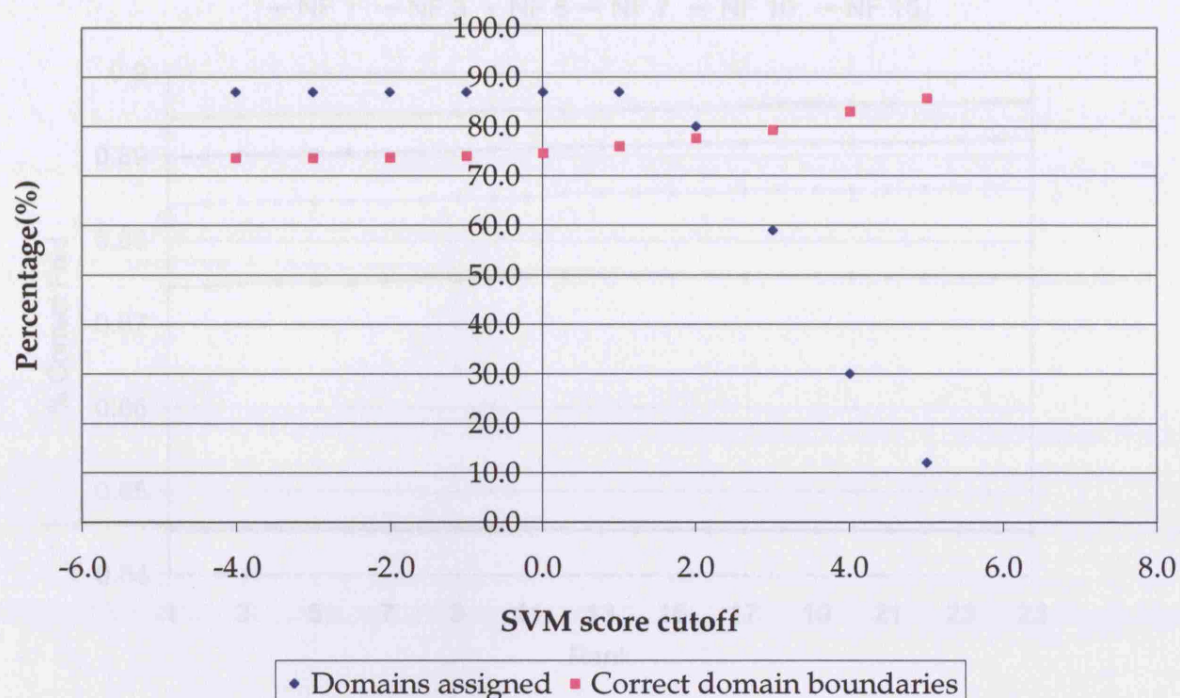


Figure 3.10. Comparison of the performance of the GRATH (GT), SAS, RMSD and SVM scores for assigning domains within multi-domain chains.

Furthermore Figure 3.11 shows that by using the SVM score to rank the hits, CATHEDRAL was able to assign 90% of domains in the query data set to the correct fold group, with 76% of domain boundaries within 10 residues of the actual boundary. When the threshold is increased to within 15 residues of the actual boundary the coverage increases to 85%. Also worth noting is that once the SVM score-cut-off is increased above 1.5, the coverage drops dramatically, but the accuracy of the domain boundaries does not increase significantly suggesting this is an appropriate threshold to use in CATHEDRAL.



**Figure 3.11.** Percentage of domain assigned (blue) and the percentage of domain boundaries within 10 residues of verified boundaries (pink) at a range of SVM score cut-offs.

### 3.3.2.2 *Optimising the Fidelity of the GRATH-Filter*

In benchmarking CATHEDRAL for fold recognition the top 10 folds identified by GRATH were realigned using SSAP. This value was adjusted to see if CATHEDRAL's performance in domain boundary recognition could be improved. Figure 3.12 shows the percentage of correct fold matches obtained in the v2.6 CATH dataset by varying the number of putative folds selected for realignment by SSAP. It can be seen that no significant increase is observed if the number of folds selected is increased from 10 to 15 and therefore to preserve the speed of the algorithm 10 was selected as an optimum value.

To further increase the speed of the algorithm it was hypothesised that a limited number of relatives from each fold group could be taken without compromising the fidelity of the domain boundaries.



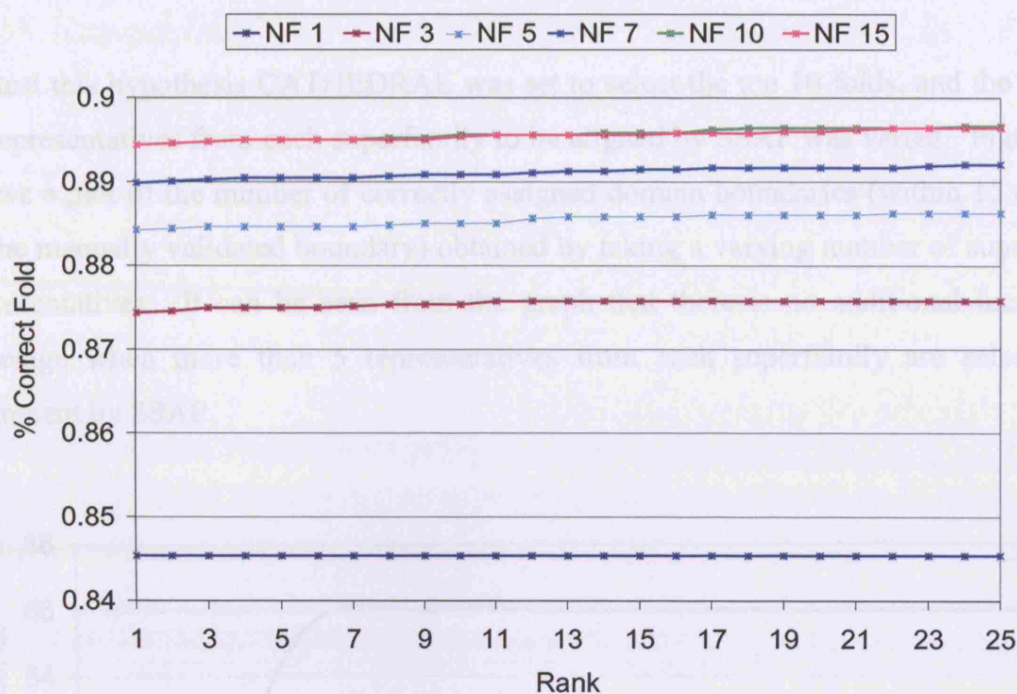


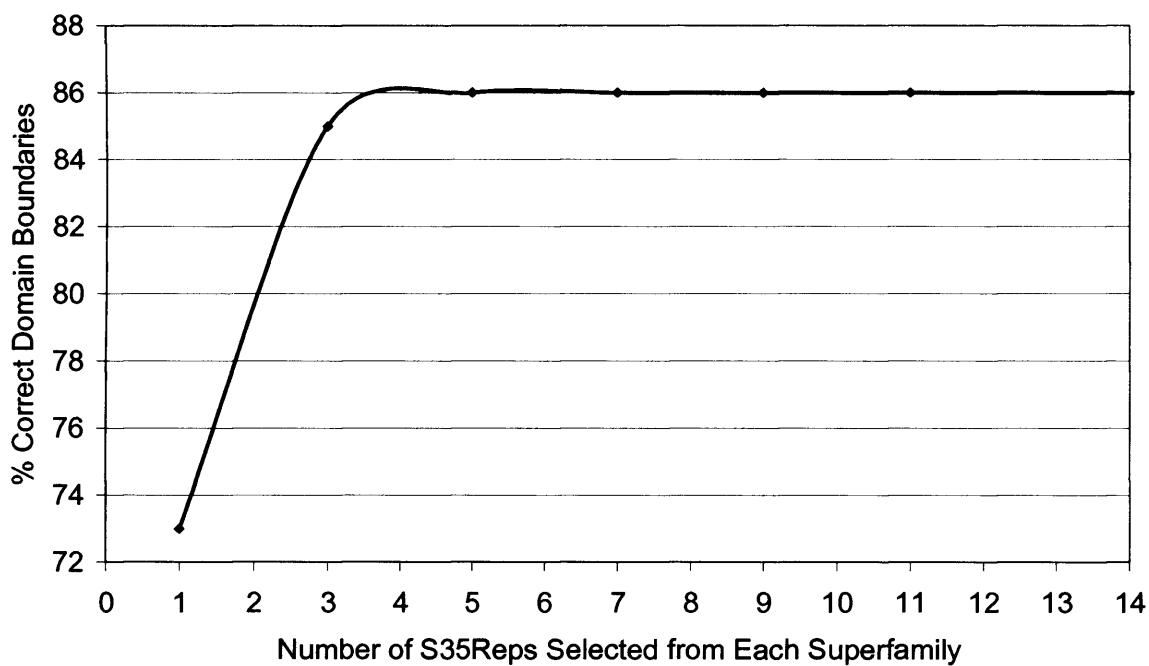
Figure 3.12. Percentage of correct folds identified at a particular rank for varying numbers of putative folds (NF) selected by the GRATH.

### 3.3.2.3 Optimising the number of representatives from each fold group to compare by SSAP

Once the GRATH-Filter has identified the fold groups closest to the putative domain, representatives from each fold are aligned to the query chain by the SSAP algorithm. By default CATHEDRAL takes all sequence diverse relatives (having  $\leq 35\%$  sequence identity to each other) from each fold group, however this can result in many thousands of comparisons especially if the fold group is highly populated, such as the Rossmann fold. However it is important to find the closest structural relative for each assignment to ensure the accuracy of the domain boundaries.

To further increase the speed of the algorithm it was hypothesised that a limited number of relatives from each fold group could be taken without compromising the fidelity of the domain boundaries.

To test this hypothesis CATHEDRAL was set to select the top 10 folds, and the number of representatives from each superfamily to be aligned by SSAP was varied. Figure 3.13 shows a plot of the number of correctly assigned domain boundaries (within 15 residues of the manually validated boundary) obtained by taking a varying number of superfamily representatives. It can be seen from the graph that there is no additional increase in coverage when more than 5 representatives from each superfamily are selected for alignment by SSAP.



**Figure 3.13.** The percentage of correctly assigned domain boundaries (within 15 residues of the manually validated boundary) against varying number of superfamily representatives.

### ***3.3.3 Comparison of CATHEDRAL performance in domain boundary assignment using Sequence Based HMMs***

The ability of CATHEDRAL to accurately delineate protein chains into their composite domains was compared to the performance achieved using HMMs. The test dataset was scanned against a library of HMMs built from each S35 sequence in CATH.

Domain boundaries were assigned to the query chains in the same way as CATHEDRAL, but using the HMM E-value instead of the CATHEDRAL SVM score to rank hits. The HMM method was only able to discover 65% of domain folds within the dataset chains. One of the main reasons for this low coverage was that 11% of the chains did not match any CATH domains HMMs, using an E-value threshold of 0.001. This E-value threshold has been benchmarked to give a 5% error rate in previous analyses (Sillitoe et al. 2005). Of the domain boundaries assigned, only 33% were within 10 residues compared with 76% for CATHEDRAL. The number of assigned domains would have been increased by using a less conservative E-value threshold. However, this would be accompanied by a decrease in the quality of the domain boundaries. The domain recognition performance is on a par with the method of Nagarajan and Yona (Nagarajan, Yona 2004). This method predicted the correct domain architecture for 35% of a data set of multi-domain PDB chains. However by incorporating structural information they were able to increase the percentage of boundaries, within 10 residues, to 63%.

### 3.4 Discussion

The first step in the classification of proteins domains into structural families is the delineation of multi-domain chains into their composite domains. This work presented the optimisation and benchmarking of a protocol for domain boundary assignment in multi-domain proteins (CATHEDRAL), which exploits the recurrence of folds in different multi-domain contexts.

CATHEDRAL combines two established structural comparison algorithms in order to develop a fast and accurate protocol for domain assignment and ultimately homologue recognition. For 76% of the test set, all domain boundaries within the multi-domain chains were correctly assigned within 10 residues compared to 33% with HMMer. This shows considerable improvement over a previous consensus protocol of automatic domain boundary assignment. For this only 10-20% of domains on average could be identified as having reliable boundary assignments from agreement between 3 independent methods (Jones et al. 1998).

CATHEDRAL misses 10% of the domains in its target dataset of multi-domain chains. 30% of these are missed because they are too small (<3 secondary structures) and are not recognised by the CATHEDRAL protocol due to an inability to describe a domain as a graph when there are less than three edges (Harrison et al. 2003). 20% are distorted or irregular structures that give poorly defined graphs. The remaining 50% are missed because they do not pass the GRATH score similarity cut-offs, as the relatives are too distant and related structural motifs in neighbouring fold groups are better matched.

The domains that CATHEDRAL misses highlight the fact that although CATH classification of protein folds gives a discrete description of fold space some highly populated regions of fold space would perhaps be better represented as a continuum (Orengo et al. 1993). Koppensteiner *et al* (2000) noted that it is possible to “walk” from one  $\alpha / \beta$  sandwich fold to another, through the extension of  $\alpha / \beta$  motifs. Furthermore

Harrison (2002) showed that extensive overlap between fold groups are observed due to large common structural motifs.

An important insight gleaned in this chapter is the importance of the measure used to score structural similarity. Kolodny and co-workers (2005) showed how geometric scores based on the RMSD are often better discriminators of fold space than the native scores employed by many algorithms which often perform better at detecting the closest structural neighbour. Also shown was the importance of achieving a global alignment in terms of domain boundary assignment. The development of an optimal SVM approach in the CATHEDRAL protocol which combined different types of scores helped significantly in identifying domain boundaries with multidomain structures.



# **4 Benchmarking Methods for Detecting Homologous Relationships between Proteins**

## ***4.1 Background and Aims***

Homologues can be described as proteins that share a common, evolutionary ancestor. The motivation for the development of reliable systems to recognise protein homology is to further the understanding of how proteins evolve, particularly the evolution of new functions. This will assist in improving the accuracy of functional annotation, through homology. In the absence of other biological information the detection of homology is a prerequisite step towards understanding the function of a new protein. Evidence for an evolutionary relationship between two proteins can be extracted from their sequences, their structures and also the function they perform. In this chapter, several methods for detecting homology are optimised and benchmarked.

### ***4.1.1 Using Sequence Similarity to Assess Homology***

Sequence provides the most direct and unambiguous form of evidence for homology, as homologous proteins will have descended from a common ancestor with a common sequence. Over evolutionary time the sequences diverge and become more dissimilar through random mutations, insertions and deletions of amino acids.

Brenner (1998) showed that pairwise methods of sequence comparison such as BLAST (Altschul et al. 1990) can detect homologous relationships when the sequence identity is

above 30%. However for more distant homologues (sequence identity below 30%) only half of the relationships can be detected by pairwise sequence comparisons. More sophisticated methods scan query sequences against a library of sequence profiles representing different evolutionary families. Sequence profiles are essentially patterns derived from a multiple sequence alignment where each position in the sequence has been assigned a probability value for each amino acid residue type based on its observed frequency. Such methods include HMMer (Eddy 1996), SAM (Karplus et al. 1998) and PSI-BLAST (Altschul et al. 1997).

Benchmarking of sequence methods to detect remote homology requires a dataset of known evolutionary relationships between protein domains. As 3D structure comparison has previously been shown to detect more distant evolutionary relationships than sequence comparison methods, classifications of domain structure have been exploited in benchmarking sequence based remote homology detection methods (Sadreyev, Grishin 2003; Park et al. 1998; Sillitoe et al. 2005). Park *et al* (1998) used a dataset of SCOP structural homologues to show that profile-sequence methods can identify three times as many homologues as sequence-sequence methods at sequence identities below 30%. A more recent development is the advent of profile-profile methods, which scan a sequence profile against a library of profiles. Profile-profile methods include COMPASS (Sadreyev, Grishin 2003), prof\_sim (Yona, Levitt 2002), LAMA (Petrokovski 1996), PRC (Madera 2006) and HHSearch (Soding 2005).

Section 1.3.1 in Chapter 1 describes the most common types of sequence comparison methods and how their algorithms attempt to recognise and define sequence similarity. The sequence methods analysed in this chapter include BLAST, PSI-BLAST, the HMM based methods SAM and HMMer and the HMM-HMM profile method PRC. These methods were compared with the goal of determining the method that best recognises homologous relationships particularly remote relationships that are expensive in terms of curation time when classifying new proteins into protein family resources.

### ***4.1.2 Using Structural Similarity to Assess Homology***

Chothia and Lesk (1986) demonstrated that structure is more highly conserved than sequence during evolution. Therefore structural comparison methods provide a useful tool for gleaning evidence of homologous relationships even when there is no longer any detectable sequence similarity. However structural similarity should not be used alone to gauge homology since it has been hypothesised that evolutionary unrelated proteins may have similar folds because there are limited number of ways of packing  $\alpha$ -helices and  $\beta$ -sheets in three dimensions (Chothia 1992).

A number of structure comparison algorithms were assessed to determine their performance in homologue recognition. These included in-house methods such as CATHEDRAL and SSAP and a number of other publicly available methods STRUCTAL, DALI, LSQMAN and CE. See Section (1.3.2.3) in the Introduction for detailed descriptions of SSAP, STRUCTAL, DALI, LSQMAN and CE and Section 3.2 for a detailed description of CATHEDRAL.

In Chapter 2 various data was presented that highlights the high degree of structural variability observed between homologous protein domains in some domain families. Methods that capture this intra-superfamily variability and use this information to guide the classification of new domains into the family may therefore be useful. As mentioned in Section 2.2.2 the CORA algorithm can multiply align structural relatives from a domain superfamily to identify the consensus positions and capture their most conserved structural characteristics (Orengo 1999). Furthermore information is encoded in a consensus 3D template that new relatives can be scanned against using the CORALIGN program. Orengo showed on four large CATH superfamilies that the CORALIGN contact score was better at separating analogous from homologues than the pairwise SSAP score (Orengo 1999). In this chapter structural profiles were benchmarked to see if they gave a more sensitive signal in the detection of remote homologues than using pairwise comparisons.

### ***4.1.3 Using Functional Similarity to Assess Homology***

Functional conservation between two protein domains is also an indicator of homology. Studies of enzyme families have shown that even remote paralogues can perform similar reaction chemistry as evidenced by the detection of common intermediates along the reaction pathway even when they are acting on different substrates (Teichmann et al. 2001; Todd et al. 2001). Many bioinformatic resources now hold data in the form of annotations describing the functional properties and biological roles of proteins. For example ENZYME (Webb 1965) provides a four digit classification of all known enzymes where the first three describe the catalytic action of the enzyme and fourth digit denotes the substrate specificity. The KEGG resource (Kanehisa, Goto 2000) is a collection of manually drawn pathway maps presenting knowledge on molecular interaction and pathway networks. The database COG (Tatusov et al. 2003) presents Clusters of Orthologous Proteins phylogenetically classified from the completed genomes. These ‘COGs’ are often associated with a specific functional annotation.

The majority of annotation data is written as scientific natural language, which is suitable for human digestion but poses problems for machine processing. Ontologies provide a way of organising data in a manner that is still accessible to humans but at the same time can be exploited computationally. They provide a set of vocabulary terms that have well defined relationships between them, namely the ‘is-a’ relationship between parent and child term and ‘part of’ between a part and the whole.

The Gene Ontology (Ashburner et al. 2000) or GO is an important bioinformatic ontology aiming to provide consistent descriptors of proteins in every species. The resource is comprised of three orthogonal taxonomies which describe a proteins molecular function, its role in biological processes and its association with other cellular components. GO contains about 23,000 ‘phrases’ (as of May 2007) held in a Directed Acyclic Graph (DAG), where each term may have multiple parents. For example, an ATP-dependent DNA helicase is a child of ‘DNA binding’, ‘DNA helicase’ and ‘ATP-binding’.

Using functional data to infer evolutionary relationships poses two major challenges. Firstly how do we annotate the protein domains with functional descriptors and secondly how do we compare these descriptors in a meaningful way?

The most reliable and specific method of functionally annotating proteins is by manual analysis. Manual annotation involves skilled biologists exploiting a plethora of information from many resources to make an informed description of the protein's function. Such processes are obviously slow and so cannot be applied to provide functional descriptors for all proteins. Electronic annotation methods aim to provide a fast and efficient way of associating functional descriptors to a large number of proteins. The Gene Ontology Annotation (GOA) database (Camon et al. 2004) provides high quality electronic and manual annotations to the UniProt Knowledgebase. GOA takes advantage of existing references, resources and publications to assign GO terms to UniProt entries. For example a UniProt entry may have an existing EC annotation in its descriptor field and therefore using an existing mapping of EC numbers to GO terms the entry can be assigned a GO term.

As mentioned in Section 2.1 of Chapter 2 studies have shown that at a conservative 50% sequence identity homologous domains are very likely to have related functions (Todd et al. 2001;Rost 2002;Tian, Skolnick 2003). Furthermore distant paralogues that have been recruited into a new pathway to perform a new function often still have a conserved reaction chemistry even if the substrate has changed (Todd et al. 2001). Therefore safe thresholds of global sequence similarity can be used to infer functional annotation from one protein to a close homologue.

As annotations can be assigned through different approaches that have implications on the confidence of the annotations, it is important to capture information on the method used to assign the annotation. GO annotations have an associated field attributed to the source of the annotation. A controlled vocabulary allows the traceability of an annotation to be accountable. The annotation must indicate what kind of evidence was used to support the association between protein and GO term. The gene ontology provides 12

possible categories that can be used to assign a source to a particular annotation. Such sources can be from the literature as is the case with Traceable Author Statements (TAS), direct from experimental evidence as in IDA (Inferred from direct assay), IEP (Inferred from Expression Pattern). They can be inferred from sequence or structural similarity (ISS) or from electronic annotation (IEA) also.

In order to determine the functional similarity between two protein, Lord *et al* (2003) developed a method that measured the semantic similarity between entities in the Gene Ontology, this is described here as the GOSIM method. Semantic similarity relies on extracting the knowledge content of the associated annotations, with more specific terms (for example 'transmembrane receptor') being semantically more similar than less specific terms (for example 'receptor'). There are various approaches to measure semantic similarity using the structure of the ontology but the vagaries of the GO hierarchy limit the suitability of many of them. For example measuring path distances between terms is inappropriate because the assumption that all links between terms are of equal weight is not strictly true for GO. Another hypothesis often applied is that the greater the distance from the root of the graph, the more specific the term. However GO varies widely in the distance of the nodes from the root so this approach is also problematic.

Instead of relying on the structure of GO, Lord and co-workers used a corpus of annotations to examine the usage of the terms as proposed by Resnik (1999). This relies on the premise that terms that are found frequently in the corpus have less information content than terms that are rare. In the calculation of the information content for each term in the corpus it is important to acknowledge that the observation of a term implicitly leads to the observation of all terms in the pathway from that term back to the root of the ontology. For example if the term 'receptor' occurs then the terms 'signal transducer' and 'molecular function' must also have occurred i.e. a term occurs if that term, or any of its children terms occur. Lord calculates the probability for each term as the terms occurrence in the corpus divided by the number of times any term occurs. To measure the semantic similarity between two terms the probability of the 'minimum subsumer' is



calculated. This score is the negative natural log of the information content (or probability) of the shared parent of the two terms. Using a corpus of human proteins in SWISS-PROT, Lord and co-workers showed that there was correlation between sequence similarity and semantic similarity for these proteins and concluded this served as a good validation of the semantic similarity measure. In this chapter, the Lord method for comparing the functional similarity of two proteins was used to improve the recognition of homologous relationships and its performance in this context was assessed.

Other approaches to comparing the functional properties of proteins include methods that seek similarity in the functional keywords with which the proteins are annotated. The SAWTED (Structure Assignment with Text Description) method of MacCallum *et al* (2000) was developed to compare SWISSPROT annotations with the goal of providing an automated function filter for PSI-BLAST scans. The authors showed that by comparing SWISSPROT terms they could increase the number of remote homologous identified by PSI-BLAST for a given error rate. The method implements the vector-cosine model of text retrieval described by Wilbur and Yang (1996). As with the GOSIM method SAWTED relies on calculating the information content of a word based upon its frequency of use in the corpus. The methodology of the SAWTED algorithm can be applied to any corpus of biological data to assess the similarity of the associated annotations. In this chapter the method is applied to text information in the PDB file of each protein domain and optimised to detect homologous relationships.

This chapter describes the optimisation and benchmarking of various algorithms that detect the similarity of proteins sequences, structures and functions and assesses their performance in identifying homologous relationships. The suitability of methods to provide evolutionary evidence of homology in each category was assessed independently and therefore the analysis is presented in 3 sections; sequence similarity methods, structural similarity methods and functional similarity methods. In the next chapter a machine learning method is described which attempts to combine all three types of measures to increase performance in homologue detection.

## **4.2 Methods**

### **4.2.1 Benchmarking Sequence Comparison Methods**

Pairwise (BLAST), profile and HMM based methods (PSI-BLAST, SAM and HMMer) and the HMM-HMM profile method PRC were benchmarked using a number of sequence data sets from version 3 of CATH. To gauge the ability of the different methods to detect homologous relationships, three separate sets that varied in difficulty were constructed. The 'S35' set contains domains with 35% or less sequence identity to any relative in the set, the 'S20' set contains domains with 20% or less sequence identity to any relative and finally the 'S10' set contained domains that had only very remote homologous relationships no greater than 10% sequence identity. The S35 set contained 6570 domains from 937 superfamilies. The S20 set contained 3144 domains from 482 superfamilies and the S10 set 1434 domains from 288 superfamilies.

#### **4.2.1.1 Benchmarking Procedure**

HMMer, SAM and PRC require HMM models of sequence families as part of their protocol. Models were built using the SAM-T2K protocol, with each sequence in the S35 dataset as a seed (7841 for CATH v3.0) on the GenBank nr database (Benson et al. 2006). The alignments generated in the iterative HMM procedure were used to generate the PSSM used for the benchmarking of PSI-BLAST. HMMer and PRC models were generated by converting SAM models using Martin Madera's `convert.pl` script (<http://www.mrc-lmb.cam.ac.uk/genomes/julian/convert/convert.html>) and calibrated with 1000 random sequences.

Each dataset (S35, S20 and S10) was scanned all-against-all using each method. For BLAST, this is sequences against sequences, for SAM/HMMer sequences against HMMs and for PRC this is HMMs against HMMs. For PSI-BLAST, profiles were scanned

against CATH sequences (as well as GenBank to provide a sufficient background), allowing up to 20 iterations. Each method was executed using default parameters.

For each method and dataset all the hits from the all against all scan are ranked by E-value and at 10 fold E-value cut-offs the coverage is plotted against the error rate. The error rate or Error Per Query (EPQ) can be defined as the number of false positives observed at a particular E-value threshold divided by the number of true positives and false positives observed up to this point. The coverage is simply the percentage of true positives observed. These are similar to the ROC curves described in Section 3.2.4.1.1 where the proportion of true positives and false positives is plotted.

EPQ provides a more intuitive reading of the results enabling the coverage at a given number of false positives to be attained. This is particularly useful when the benchmarking dataset has a large disparity in the number of true and false positives as is the case here, or when the method does not give you a true all against all comparison.

#### ***4.2.1.2 Exceptions and Rules***

When benchmarking methods it is important to use clear definitions of true and false positives. Here a true positive is defined as a non-trivial true superfamily match and a false positive as a non-ambiguous false superfamily match.

There is no value in allowing a true positive to be counted when a sequence hits itself as this is a trivial detection. Similarly in the case of the HMM methods a trivial match also includes a match between a model and sequence that is contained in that model and this is therefore ignored.

At the topology level of the CATH hierarchy all domains share the same core fold, and within each fold domains are clustered into superfamilies where a clear evolutionary relationship exists. Two domains which share the same fold but are in different

superfamilies may be homologues but there may not currently be enough evidence of an evolutionary link (through sequence, structure and function). The relationship may have been missed by existing homologue detection methods. The relationship can therefore be defined as ambiguous and several groups have therefore opted to discount matches involving proteins from the same fold group, but different superfamilies (Soding 2005). These ambiguous matches are therefore simply ignored in the benchmarking calculations. For the purpose of benchmarking any hit between two sequences/models that belong to different superfamilies but the same fold are neither counted as true positives or false positives.

Previous benchmarks based on structural classifications have yielded unexpectedly poor performance for profile-profile methods (Soding 2005). This was shown to be caused by the fact that very remote homologues had been classified as unrelated in the gold standard structural classification (SCOP in the case of Soding's study). Such exceptions occur due to the top down hierarchical nature of such databases and due to the duplication of repeating structural motifs (for example the beta propellers) that are evolutionary related but have different 3D structures (in the case of the beta propellers differing numbers of beta sheets or "blades") (Jawad, Paoli 2002).

Matches between different structural families in CATH by sensitive profile-profile based methods have been manually analysed by the CATH curation team and where validated as homologues used to provide information on acceptable 'crosshits' between families. These 'crosshits' form a list of exceptions which when observed in the benchmarking protocol are also counted as neither a true positive nor a false positive. Recent studies by Reid (2007, In Press) showed that an automated procedure of selecting potential crosshits performed with comparable accuracy when compared to manual curation. This method assigned allowed crosshits to domain pairs from different CATH fold groups where the E-value for any sequence method was less than 0.01 and the local structural similarity as defined by SSAP yielded a SAS score less than 8Å. This approach reduces the number of false positives for those matches which have a high degree of local structural similarity and which therefore might reasonably be assumed to be homologues. The automated

crosshit detection method found ~87% of those crosshits defined manually by experts (Table 4.1). In the work presented in this chapter the automated crosshit exception procedure was also used.

	<b>Curated Exceptions (2100)</b>	<b>SAS8 Exceptions(2051)</b>
<b>FAD/NAD-binding domain (3.50.50) vs. Rossmann fold (3.40.50)</b>	75.1% (1674)	66.9% (1371)
<b>Neuraminidase (2.120.10) vs. Methylamine Dehydrogenase (2.130.10)</b>	12.5% (279)	13.6% (279)
<b>Methanol Dehydrogenase (2.140.10) vs. Methylamine Dehydrogenase (2.130.10)</b>	4.3% (96)	4.7% (96)
<b>PCNA (3.70.10) vs. Leucine-rich repeat (3.80.10)</b>	1.3% (30)	1.5% (30)
<b>Neuraminidase (2.120.10) vs. Methanol Dehydrogenase (2.140.10)</b>	0.8% (17)	0.8% (17)
<b>Tachylectin-2 (2.115.10) vs. Neuraminidase (2.12.10)</b>	0.2% (4)	0.2% (4)
	<b>100% (2100)</b>	<b>87.6% (1797)</b>

**Table 4.1.** Curated exceptions for PRC on the ‘S35’ dataset at an E-value cut-off of 0.01, compared with those produced in the same conditions using the SAS-8 heuristic. The numbers in the column headers show the total number of exceptions for each class. The CATH codes of each type of exception are shown in brackets.

## ***4.2.2 Benchmarking Structural Comparison Methods***

The dataset for the benchmarking of the different structural algorithms was the same set as described in Section 3.2.2 encompassing 6003 sequence diverse domains from different sequence families (S35Reps) in CATH v2.6.0. These domains included representatives from 907 folds and 1572 superfamilies from all the four classes of the CATH classification (mainly alpha, mainly beta, mixed alpha/beta, and those domains of few secondary structures).

Again as described in Section 3.2.2 a second dataset that represented a subset of CATH v2.6.0 and SCOP v1.65 was also used in this analysis. The CATH-SCOP dataset contained 1779 sequence diverse domains encompassing 406 folds and 709 superfamilies.

#### ***4.2.2.1 Pairwise Domain Structure Comparison***

The combinatorial method CATHEDRAL and the residue based SSAP were benchmarked against other publicly available structural comparison methods, STRUCTIONAL, DALI, LSQMAN and CE to assess their abilities to identify homologous relationships.

The benchmarking protocol was identical to that described in Section 3.2.3. An all against all structural comparison of the 6003 sequence diverse CATH domains from v2.6 was performed and the analysis was repeated on the CATH-SCOP dataset.

ROC curves (see Section 3.2.4.1.1) were calculated for all methods on their ability to detect homologous relationships for both the CATH dataset and the CATH-SCOP subset. All hits from the all against all comparison on each dataset were ranked based upon the methods native score and the error rate plotted against the coverage.

As shown in previously in Section 3.3.1.2 the ability of the structural comparison methods to detect correct fold relationships varied with the scoring schemes used. The geometric SAS score (see Section 3.2.4.1.3) was shown to be a better discriminator of fold space than the native score for nearly all methods. Therefore ROC curves were also plotted using the SAS score to determine how well all the methods can recognise homologous relationships using this scoring scheme.

For all these scenarios true positives were defined as matches between homologous domains and false positives defined as non-ambiguous non homologous matches. Non-

ambiguous is defined as described above (see Section 4.2.1.2) i.e. where hits between protein domains do not show significant structural similarity to place them in the same fold group.

As shown previously in Section 3.3.1.3 of Chapter 3 the ability of a method to completely delineate fold space in a ROC curve analysis does not necessarily correlate with its ability to recognise the closest fold match for a particular domain. Therefore the ranking of the correct superfamily in the list of matches was also investigated.

For each domain scanned against both the CATH and CATH-SCOP dataset, the matches are sorted based on the appropriate score. The frequency of producing a correct match at a particular rank was calculated and a graph plotted showing the cumulative percentage of correctly assigned homologous superfamilies at each descending rank. Plots were made based both on the structure comparison methods native score and the geometric score SAS.

#### ***4.2.2.2 CORALIGN – Profile Based Structural Alignment***

As mentioned in Section 4.1.2 CORALIGN is a structural alignment method that scores the comparison of a query structure against a 3D template on the basis of shared contacts. Because some of the superfamilies in CATH are very structurally diverse (e.g. the P-loop Hydrolases) Structural Sub-Groups (SSGs) were first identified within each superfamily in v2.6 of CATH in order to improve the quality of the multiple structural alignments used to generate the 3D templates for each superfamily. An SSG can be defined as a cluster of non-identical homologous domains that have a structural similarity defined by a SSAP score greater than or equal to 85 and a residue overlap of greater than 60%. The clusters are created by directed multi-linkage clustering and a total of 1885 clusters with greater than one member were created from the 1572 superfamilies in CATH v2.6. The domains in each SSG were then aligned using the CORA algorithm.



CORA generates consensus 3D information in the form of contact plot. A contact plot being defined as a 2D matrix, labelled horizontally and vertically with the protein's residues, with the cells of the matrix labelled depending on whether pairs of residues are in contact in the structure. Contacts are identified if the  $C_{\beta}$  atoms from each residue pair are within 8Å separation. To create the consensus plot CORA generates conformant contact plots using the conserved positions in the multiple alignment to account for insertions and deletions. Finally these conformant plots are overlayed for each protein in the alignment to detect highly conserved residue contacts.

The program CORALIGN can then be used to compare a new protein structure against the 3D consensus template for a particular domain superfamily. CORALIGN exploits the same double dynamic programming algorithm of SSAP (Taylor, Orengo 1989) to compare consensus structural environments (i.e. average vectors) between the template and target. A contact based score is produced which is the percentage of completely conserved contacts in the template structure that are also present in the target structure.

The same 6003 sequence diverse dataset detailed above was scanned against a library of CORA templates built from all the 3D SSGs and ROC curves produced to determine the performance of CORALIGN in homologue recognition.

#### ***4.2.3 Implementing and Benchmarking Function Comparison Methods***

Two methods, GOSIM and SAWTED, for comparing functional annotations were benchmarked for their ability to both recognise homologous protein domains and also homologous domains with the same function. For the purpose of benchmarking, CATH was used as the standard for homology and the Enzyme Classification (EC) was used for validating functionally related homologues. The PDBSprotEC (Martin 2004) database, linking PDB chains to EC numbers via SwissProt, was used to assign the EC mapping of

a PDB chain to all CATH v2.6 S35Reps within that chain. This gave 2823 S35Rep domains with an associated EC annotation from the 6003 S35 Rep domains in CATH.

#### ***4.2.3.1 GO Mapping to CATH Domains***

All 6003 sequence diverse (S35Reps) domains from CATH v2.6 were annotated, where possible, with GO terms using the GOA multispecies UniProt to PDB electronically inferred mapping. The GOA mappings are to the protein chain and the associated GO terms are then inferred to all domains within the chain. This mapping resulted in 3250 S35Reps with at least one GO term assigned. Direct functional annotation achieved just over 54% coverage of CATH domains in the dataset. Further annotations were obtained using homology to transfer functional information.

As described in Section 2.3.4 an Intermediate Sequence Library (ISL) was constructed by scanning UniProt against the CATH HMM library described in section 2.2.3.1. Homologous sequences were identified as those hits with an E-value  $< 0.001$  and greater than 60% sequence overlap. These homologous sequences were then clustered into sequence families using directed multi-linkage clustering at sequence identity thresholds of 95%, 60% and 35%. By inferring GO annotations from clustered genomic sequences to CATH domains that are in the same 95% sequence identity cluster the coverage of functional annotations increases to 57% of CATH domains. Inferring at 60% sequence identity increased coverage to 60% and inferring at 35% sequence identity increased coverage to 64%. This gave four datasets termed GOA\_ID, GOA\_S95, GOA\_S60 and GOA\_S35.

The GOSIM method requires a corpus of annotations to calculate the probability of observing each GO term and to calculate its subsequent information content. The GOA multispecies UniProt mapping was used as a corpus and the frequency of each GO term in the corpus calculated. To compare the semantic similarity of each term the Lord method described in Section 4.1.3 was used (Lord et al. 2003).

For each dataset the ability to recognise homologous and functional relationships was explored using ROC curves.

#### ***4.2.3.2 Using SAWTED to Compare PDB Functional Information***

Many protein structures have no associated functional annotation in terms of GO or EC. Inferring annotations from homologous sequences can increase the coverage but this has a trade-off in the accuracy of the inferred annotation. Each CATH domain has an associated PDB file providing the 3D coordinates of the structure. As well as the structural information each PDB file has a series of fields with text information about the protein, including a 'header', a 'title', a 'compound' field and a 'keywords' field. For example the PDB file for the structure 1cuk contains the header 'Helicase', the title 'Escherichia Coli RUVA Protein', and has associated keywords 'DNA Repair, SOS Response, DNA-Binding, DNA Recombination and Helicase'. By extracting this information from the PDB file for each CATH domain we can achieve 100% coverage of CATH domains having associated functional text information.

The SAWTED method of MacCallum (2000) as described in Section 4.1.3 can be implemented to score vectors of text from a corpus of knowledge. The text information in the header, title, compound and keywords fields was extracted from each of the PDB files associated with the 6003 S35Reps from version 2.6 of CATH to create the corpus. To reduce the noise unique words and words that were just numbers were removed. Vectors for all PDBs were calculated and scored, all against all, according to the vector-cosine model of text retrieval (Wilbur, Yang 1996). The similarity scores derived were modelled on a normal distribution and converted to Z-Scores.

The corpus described above contains many non-informative terms, such as conjunctions and many non-specific descriptions, 'Protein' for example. Although they appear with high frequency in the corpus they should be assigned a low information content and

matching them should not yield a significant similarity score. In order to see the effect of limiting the corpus of text, used for comparing these terms, only text extracted from the PDB file that was also a SWISSPROT keyword was used.

For both the full and restricted corpuses the ability of the methods to identify homologous relationships and functional relationships was investigated.

## **4.3 Results**

### **4.3.1 Performance of Sequence Comparison Methods in Recognising Homology**

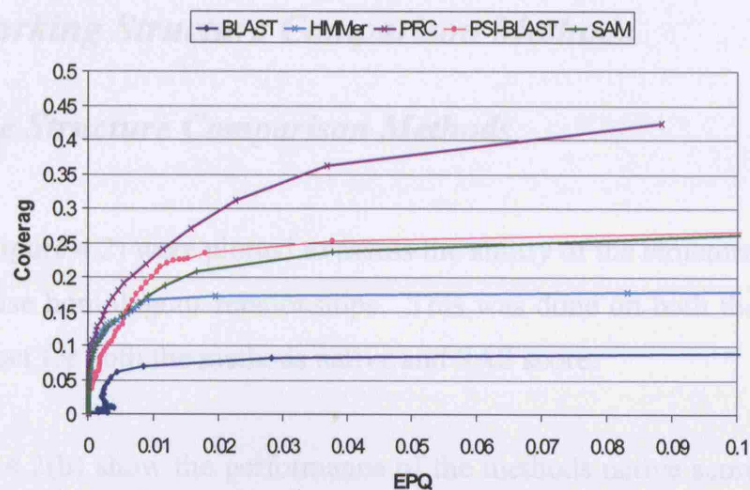
Five different sequence comparison methods were benchmarked against three different sequence datasets from CATH v3.0 (S35, S20 and S10 set) with the S35 set containing no sequences with greater than 35% sequence identity and so on. ROC curves which plotted the error per query against the percentage of true positives were produced for each data set. One can see from Figure 4.1(a-c) that the HMM-HMM method PRC outperforms all methods on all datasets. At a 5% error rate PRC achieves coverage of 37% on the S35 dataset, 31% on the S20 dataset and 22% on the S10 dataset.

Interestingly PSI-BLAST is the next best performing method on the S35 dataset, outperforming the HMM based methods SAM and HMMer by achieving 25% coverage at a 5% error rate compared to 24% for SAM and only 17% for HMMer. This is perhaps surprising as previous analyses (Park et al. 1998;Madera, Gough 2002) had demonstrated that HMM based methods were significantly more sensitive than PSI-BLAST. Generally PSI-BLAST is used to build a profile with the query sequence, in this case however the profile was built with target2k. This means that PSI-BLAST had the benefit of using HMM technology in the construction of its profile. This may account for the increase in PSI-BLAST performance. Another factor may be the significant expansion of the sequence databases since the previous studies were done so that PSI-BLAST profiles may contain more information on remote homologues. The unusual curves associated with BLAST are due to the fact that the axes of EPQ plots are not independent and therefore there may be multiple values of y for a single value of x (i.e. the curve goes backwards).

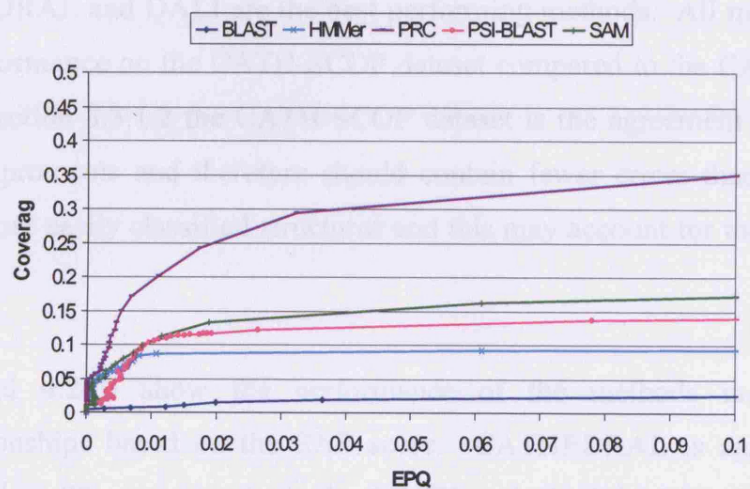
Considering the more remote datasets, SAM performs equally as well as PSI-BLAST on the S20 dataset and significantly outperforms PSI-BLAST on the S10 dataset. Overall

PRC is the best performing method outperforming all profile-sequence and sequence-sequence methods in remote homology detection.

(A)



(B)



(C)

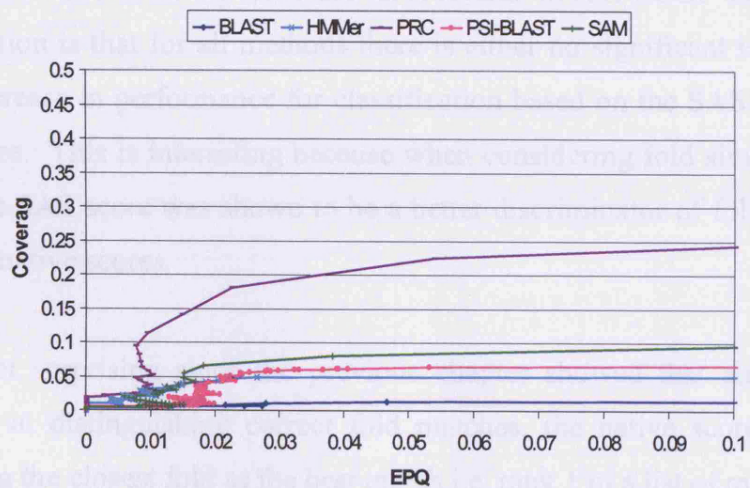


Figure 4.1(a-c). ROC curves plotting coverage against error per query for BLAST, PSI-BLAST, SAM, HMMer and PRC against three different datasets. (a) An S35 set where no homologues shared greater than 35% sequence identity, (b) a S20 set and (c) a S10 set.



## ***4.3.2 Benchmarking Structure Comparison Methods***

### ***4.3.2.1 Pairwise Structure Comparison Methods***

ROC curves (see Figure 4.2) were plotted to assess the ability of the structure comparison methods to recognise homologous relationships. This was done on both the CATH and CATH-SCOP dataset for both the methods native and SAS score.

Figures 4.2(a) and 4.2(b) show the performance of the methods native scores. At a 2% error rate CATHEDRAL and DALI are the best performing methods. All methods show an increase in performance on the CATH-SCOP dataset compared to the CATH dataset. As stipulated in Section 3.3.1.2 the CATH-SCOP dataset is the agreement between the two classification protocols and therefore should contain fewer errors than the CATH dataset and also more easily classified structures and this may account for the increase in performance.

Figures 4.2(c) and 4.2(d) show the performance of the methods in identifying homologous relationships based on the SAS score. CATHEDRAL is again the best performing method at 2% error on both the CATH and CATH-SCOP datasets. An interesting observation is that for all methods there is either no significant improvement or a significant decrease in performance for classification based on the SAS score rather than the native score. This is interesting because when considering fold similarities (see Section 3.3.1.2) the SAS score was shown to be a better discriminator of fold space than nearly all methods native scores.

This is perhaps not surprising since the previous chapter showed that although SAS scores were better at distinguishing correct fold matches, the native score performed better at recognising the closest fold as the best match i.e. rank 1 in a list of matches.

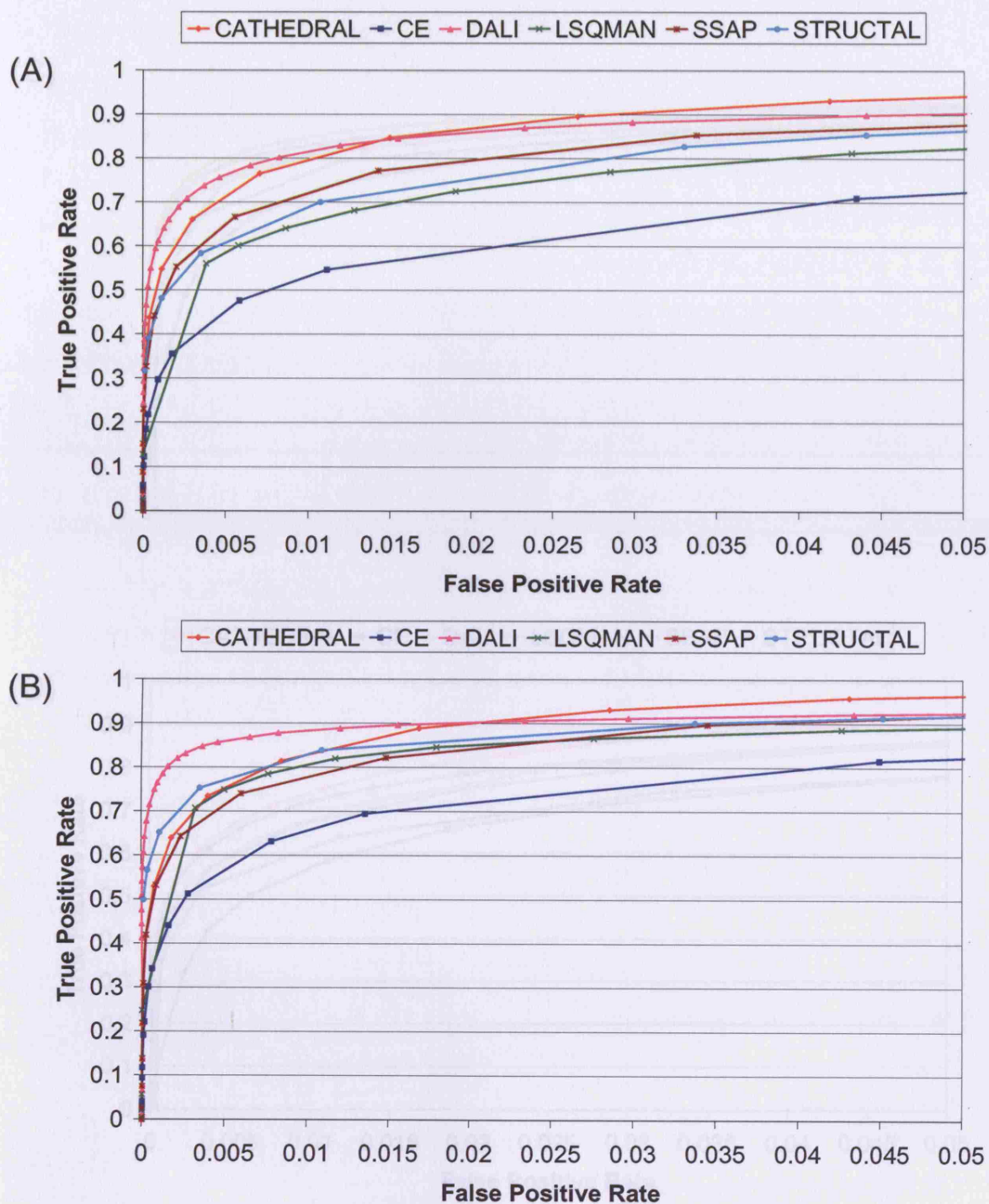


Figure 4.2(a,b,c,d). ROC curves plotted for different alignment comparison methods where a positive match represents a true superfamily match. Plot 2(a) is based on the native scores for the CATH dataset. 2(b) is based on the native scores for the CATH-SCOP dataset. 2(c) is based on the SAS scores for the CATH dataset. Plot 2(d) is based on the SAS scores for the CATH-SCOP dataset.

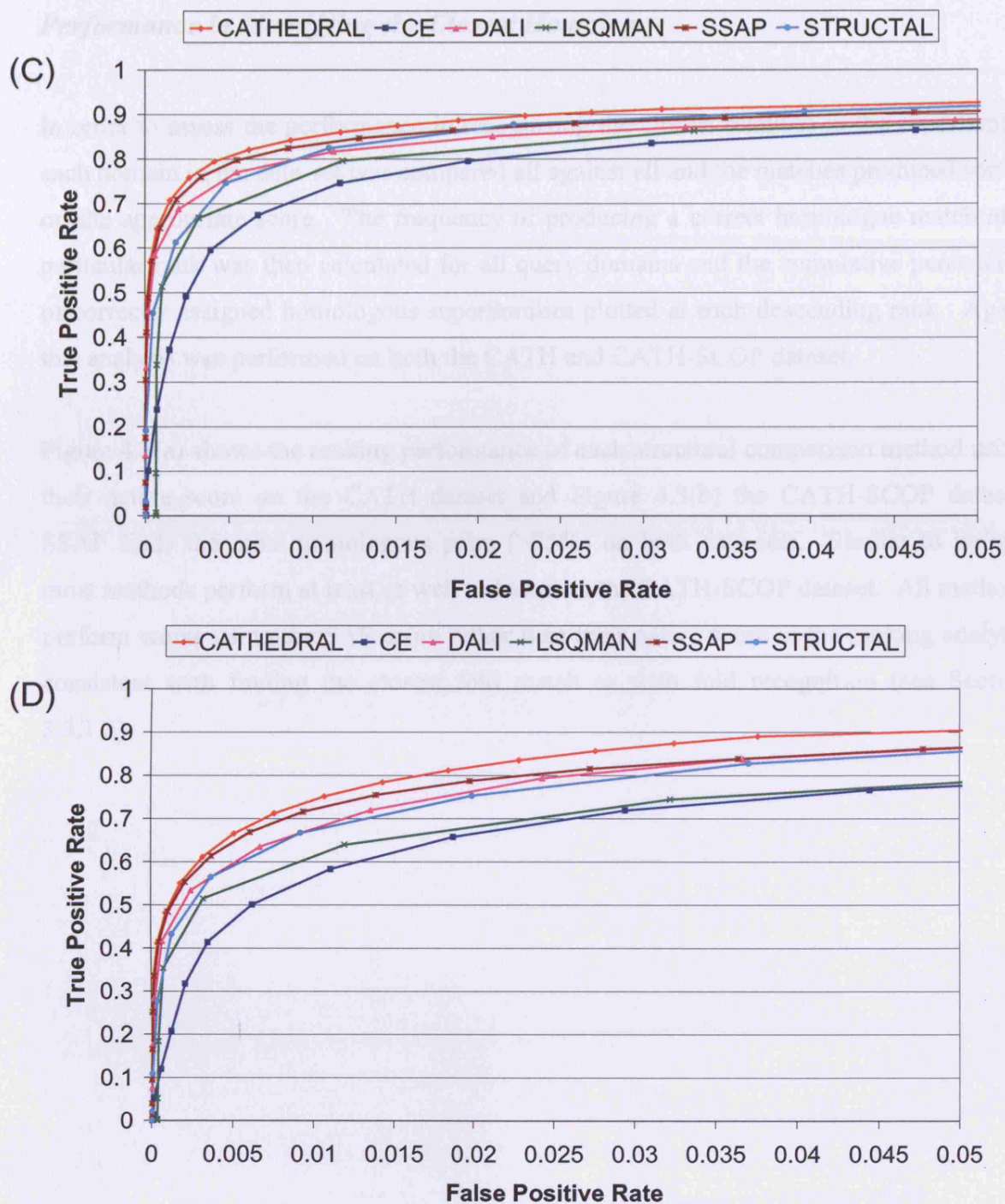


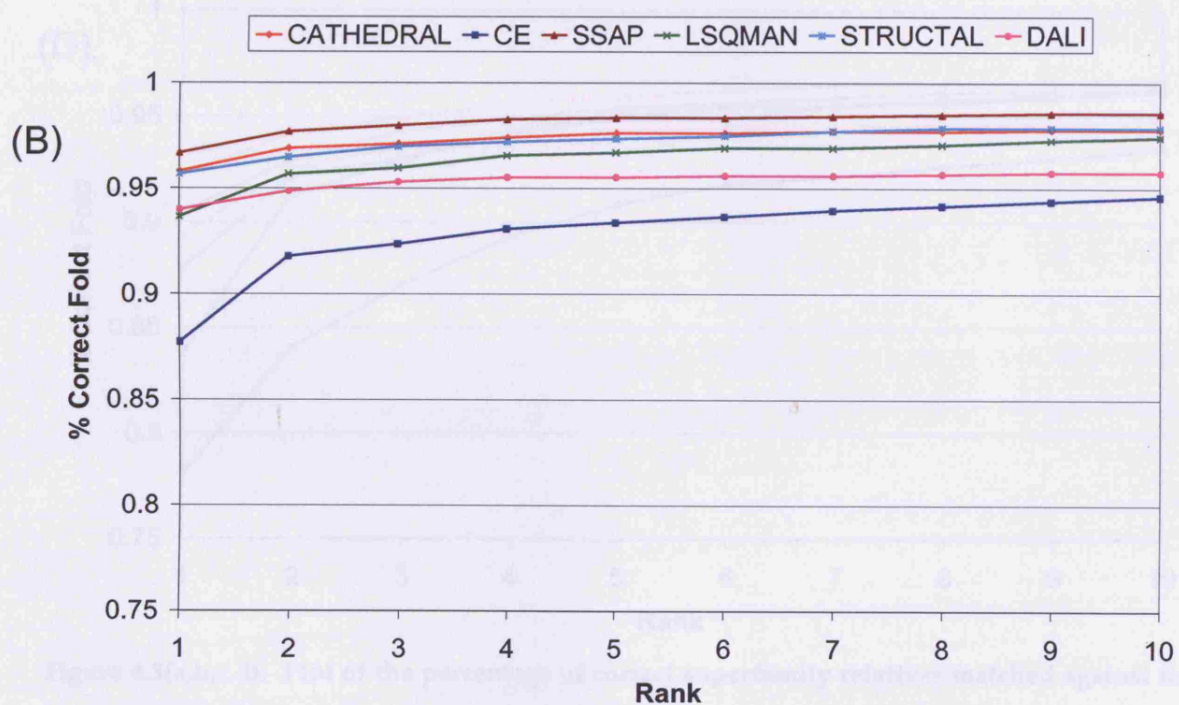
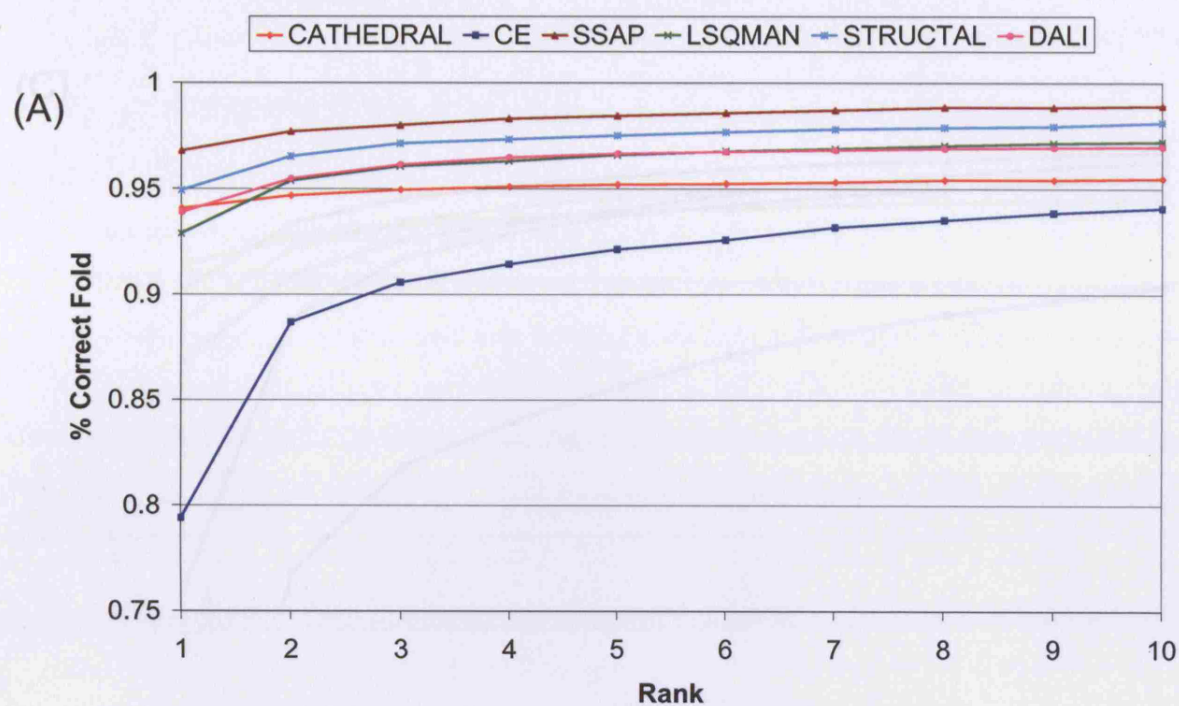
Figure 4.2(a,b,c,d). ROC curves plotted for different structural comparison methods where a positive match represents a true superfamily match. Plot 2(a) is based on the native score on the CATH dataset, 2(b) is based on the native score for the CATH-SCOP dataset. 2(c) is based on the SAS score for the CATH dataset. Plot 2(d) is based on the SAS score for the CATH-SCOP dataset.

### ***Performance in Identifying the Closest Homologue***

In order to assess the performance in recognising the closest relatives in the superfamily each domain in the data-set was compared all against all and the matches produced sorted on the appropriate score. The frequency of producing a correct homologue match at a particular rank was then calculated for all query domains and the cumulative percentage of correctly assigned homologous superfamilies plotted at each descending rank. Again this analysis was performed on both the CATH and CATH-SCOP dataset.

Figure 4.3(a) shows the ranking performance of each structural comparison method using their native score on the CATH dataset and Figure 4.3(b) the CATH-SCOP dataset. SSAP finds the most homologous pairs (>95%) on both data-sets. Similar to before, most methods perform at least as well or better on the CATH-SCOP dataset. All methods perform worse using the SAS score rather than their native score in the ranking analysis consistent with finding the closest fold match as with fold recognition (see Section 3.3.1.3).





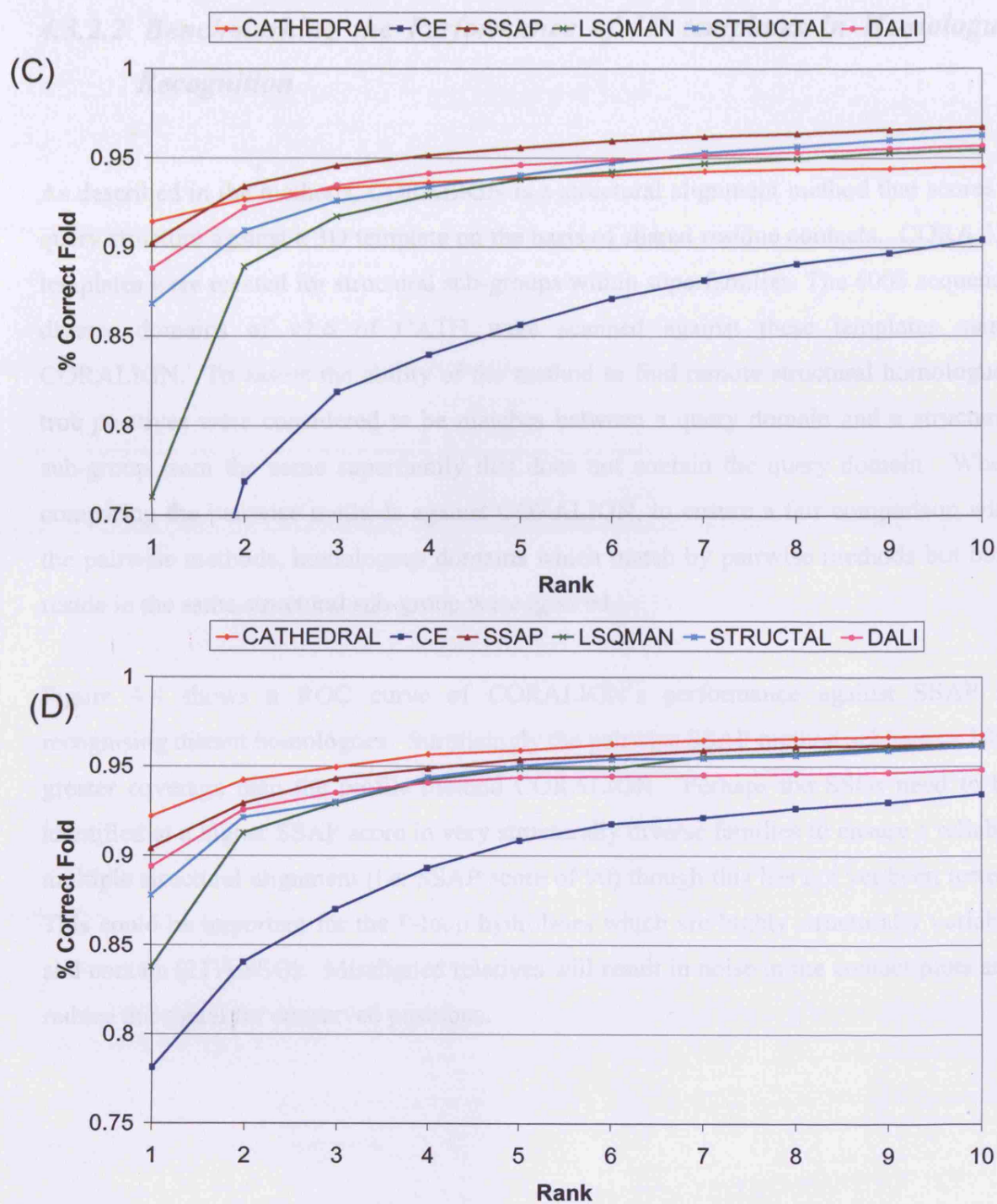


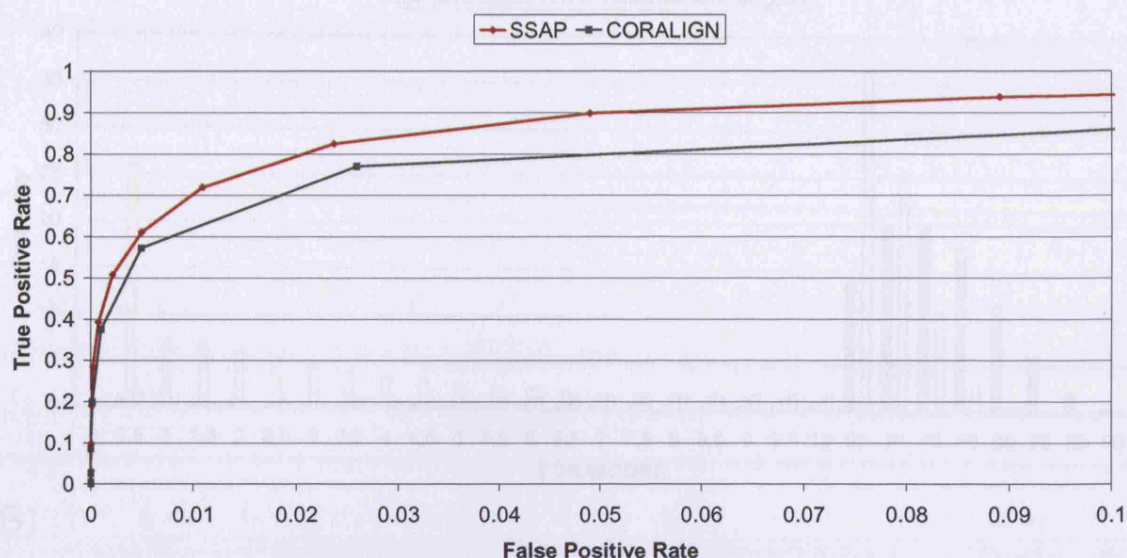
Figure 4.3(a,b,c,d). Plot of the percentage of correct superfamily relatives matched against the ranked native score for the (a) CATH and (b) CATH-SCOP dataset and the ranked SAS score for (c) CATH and (d) CATH-SCOP dataset.

#### ***4.3.2.2 Benchmarking the Performance of 3D templates in Homologue Recognition***

As described in the methods, CORALIGN is a structural alignment method that scores a query structure against a 3D template on the basis of shared residue contacts. CORA 3D templates were created for structural sub-groups within superfamilies. The 6003 sequence diverse domains of v2.6 of CATH were scanned against these templates using CORALIGN. To assess the ability of the method to find remote structural homologues true positives were considered to be matches between a query domain and a structural sub-group from the same superfamily that does not contain the query domain. When comparing the pairwise methods against CORALIGN, to ensure a fair comparison with the pairwise methods, homologous domains which match by pairwise methods but both reside in the same structural sub-group were ignored.

Figure 4.4 shows a ROC curve of CORALIGN's performance against SSAP in recognising distant homologues. Surprisingly the pairwise SSAP method achieves a 10% greater coverage than the profile method CORALIGN. Perhaps the SSGs need to be identified at a higher SSAP score in very structurally diverse families to ensure a reliable multiple structural alignment (i.e. SSAP score of 90) though this has not yet been tested. This could be important for the P-loop hydrolases which are highly structurally variable and contain (211) SSGs. Misaligned relatives will result in noise in the contact plots and reduce the signal for conserved positions.

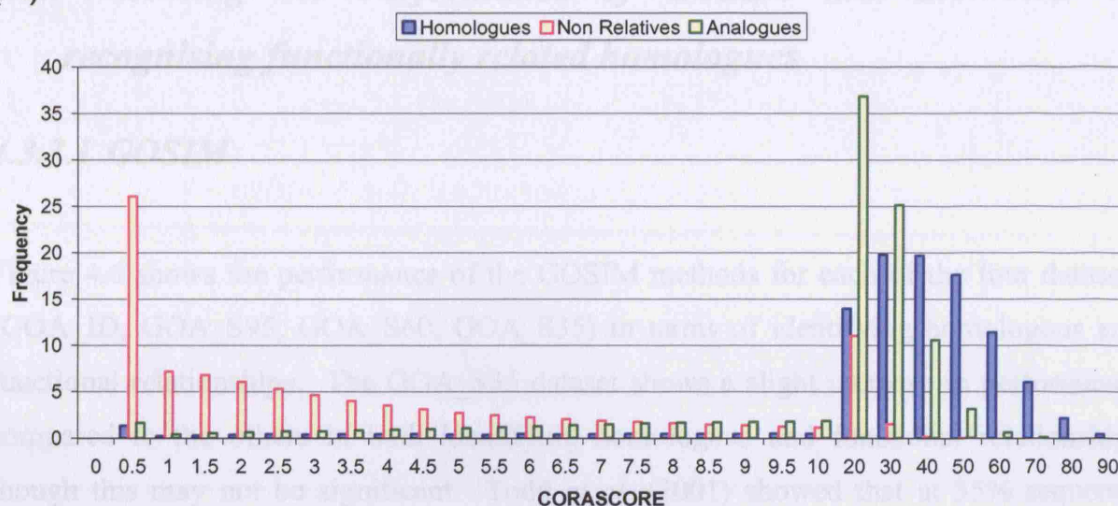




**Figure 4.4.** ROC curve showing the performance of the profile based CORALIGN method against SSAP in the recognition of remote protein homologues.

Figure 4.5(a-b) shows the performance of CORALIGN & SSAP in distinguishing non-related proteins, analogous proteins (those in the same fold but different superfamilies) and homologous proteins. To test which method best discriminates between the homologous and analogous populations a heteroscedastic two-sample unequal variance TTEST was calculated, with the hypothesis that the two distributions are in fact one i.e. the scores cannot discriminate between analogous and homologues. Using the SSAP scores the probability of the two distributions being the same is 0.007 compared to CORALIGN which is 0.06. This shows that SSAP better discriminates between analogous and homologous proteins and would be a better approach to use in a combined homologue detection protocol.

(A)



(B)

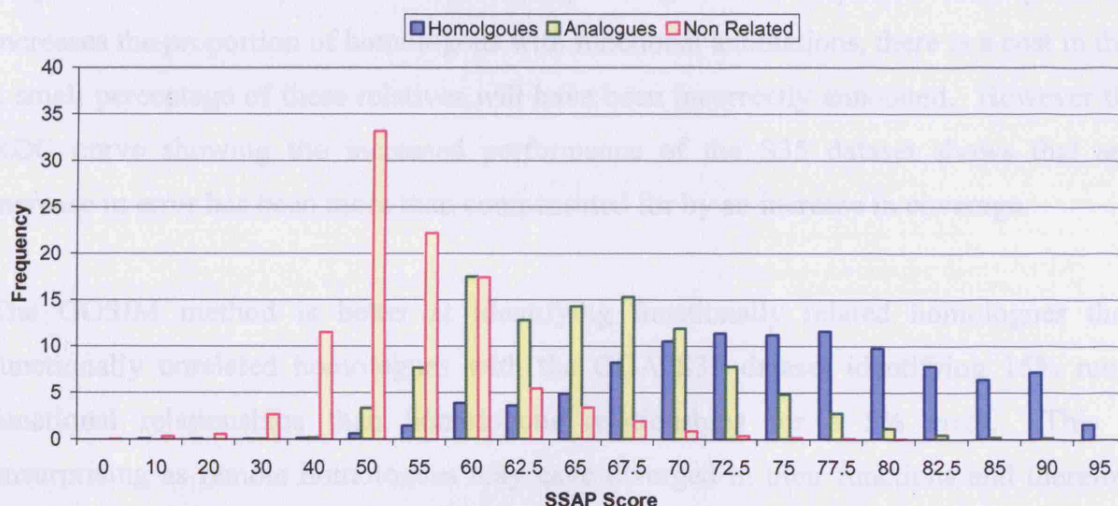


Figure 4.5. Graphs showing the distribution of scores from CORALIGN(a) and SSAP(b) on non-related proteins (pink), analogous proteins (green) and homologous proteins (blue).

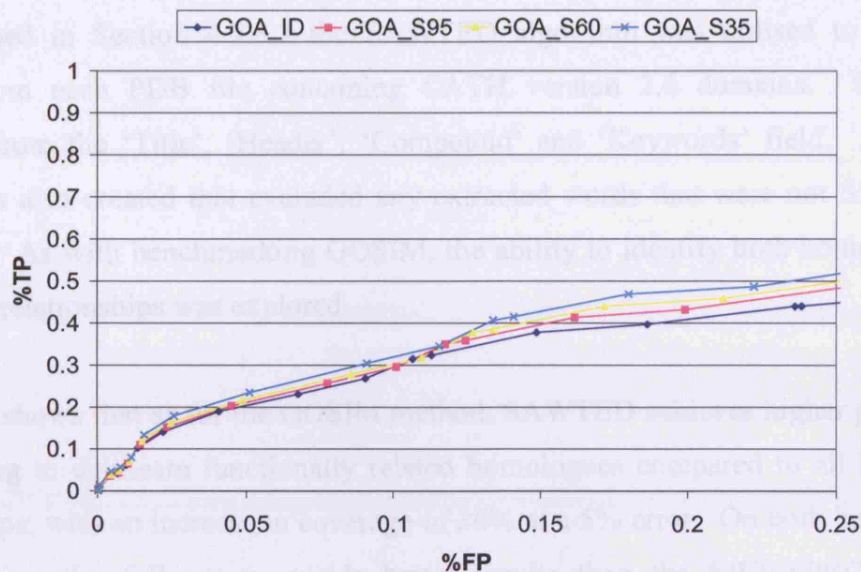
### ***4.3.3 Assessing the Performance of GOSIM and SAWTED in recognising functionally related homologues***

#### ***4.3.3.1 GOSIM***

Figure 4.6 shows the performance of the GOSIM methods for each of the four datasets (GOA\_ID, GOA\_S95, GOA\_S60, GOA\_S35) in terms of identifying homologous and functional relationships. The GOA\_S35 dataset shows a slight increase in performance compared to the others in both identifying homologous and functional relationships though this may not be significant. Todd *et al.* (2001) showed that at 35% sequence identity 90% of homologous domain pairs shared related functions (to three levels of the enzyme classification). Therefore although using a 35% sequence identity cut-off increases the proportion of homologous with functional annotations, there is a cost in that a small percentage of these relatives will have been incorrectly annotated. However the ROC curve showing the increased performance of the S35 dataset shows that any increase in error has been more than compensated for by an increase in coverage.

The GOSIM method is better at identifying functionally related homologues than functionally unrelated homologues with the GOA\_S35 dataset identifying 15% more functional relationships than homologous relationships for a 5% error. This is unsurprising as remote homologues may have diverged in their functions and therefore would be annotated with unrelated GO terms.

(A)



(B)

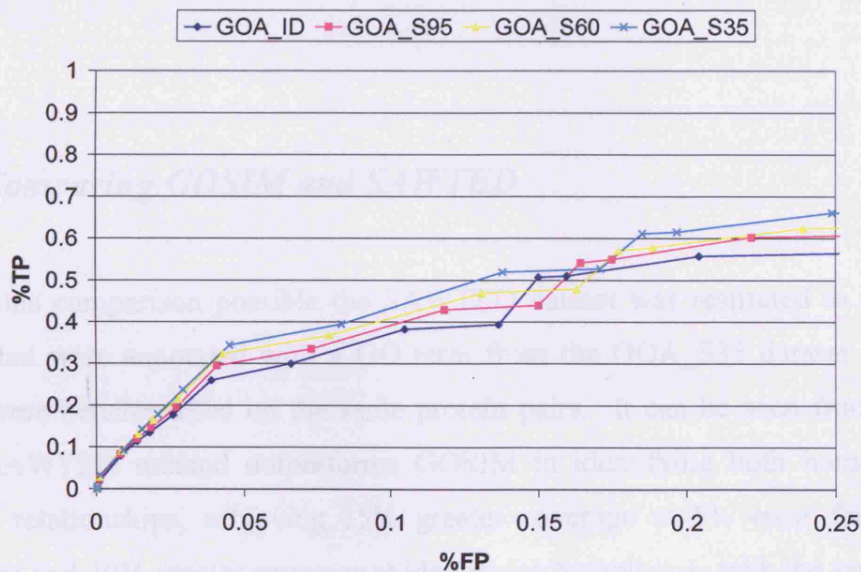


Figure 4.6. ROC curves showing the performance of the GOSIM method of comparing the semantic similarity of GO terms on four differently annotated data-sets. Figure (a) shows the performance based on homology and Figure (b) show the performance based on identifying functionally related homologues as defined by their enzyme classification.



#### **4.3.3.2 SAWTED**

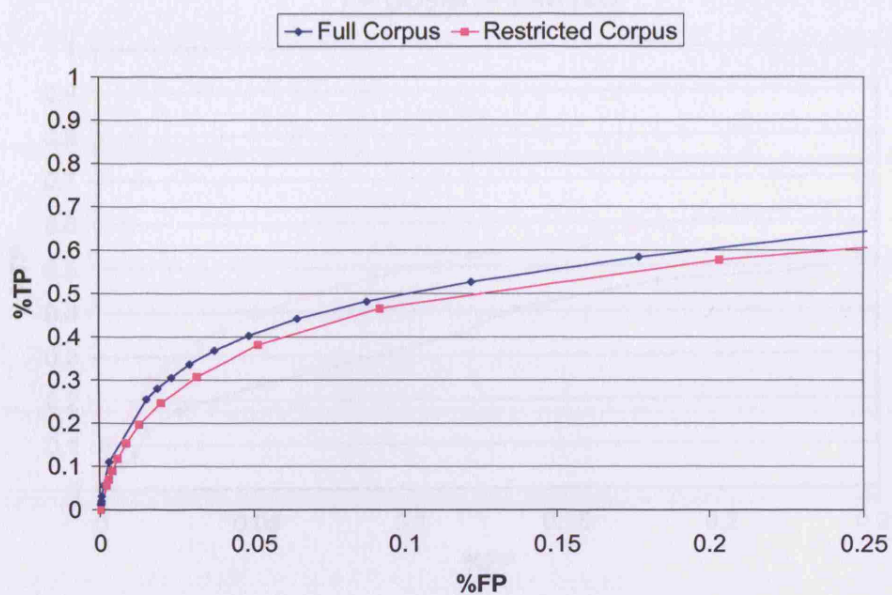
As described in Section 4.2.3.2 the SAWTED algorithm was utilised to create text vectors from each PDB file containing CATH version 2.6 domains. Words were extracted from the 'Title', 'Header', 'Compound' and 'Keywords' field. A restricted corpus was also created that excluded any extracted words that were not SWISSPROT key terms. As with benchmarking GOSIM, the ability to identify both homologous and functional relationships was explored.

Figure 4.7 shows that as for the GOSIM method, SAWTED achieves higher performance when trying to delineate functionally related homologues compared to all homologous relationships, with an increase in coverage of 30% at a 5% error. On both benchmarking criteria using the full corpus yields better results than the SWISSPROT keyword restricted corpus. This suggests that the method can handle noise in the full corpus as low information content and that some of the additional information in the full corpus is valuable.

#### **4.3.3.3 Comparing GOSIM and SAWTED**

To make this comparison possible the SAWTED dataset was restricted to only protein domains that were annotated with a GO term from the GOA\_S35 dataset so that both methods were benchmarked on the same protein pairs. It can be seen from Figure 4.8 that the SAWTED method outperforms GOSIM in identifying both homologous and functional relationships, achieving 15% greater coverage at 5% error for identifying homologues and 40% greater coverage at identifying homologues with the same function.

(A)



(B)

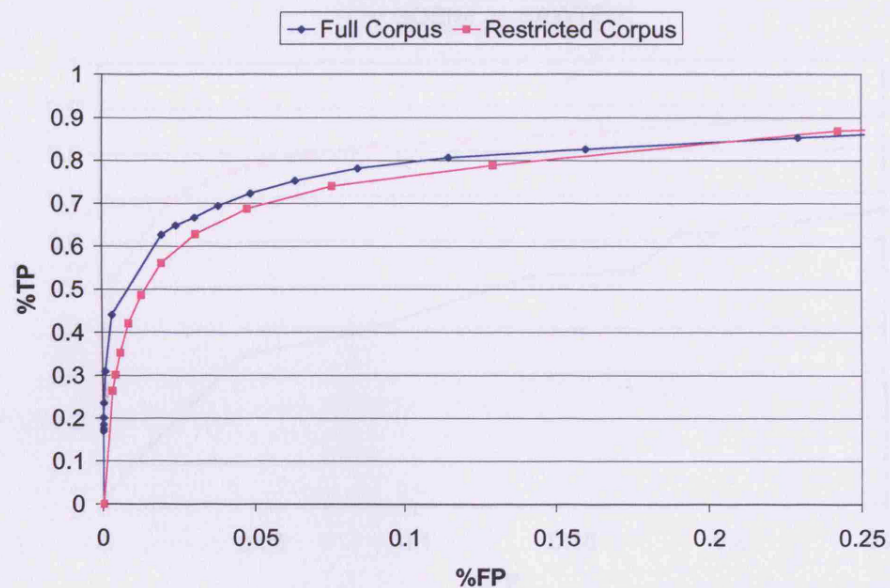
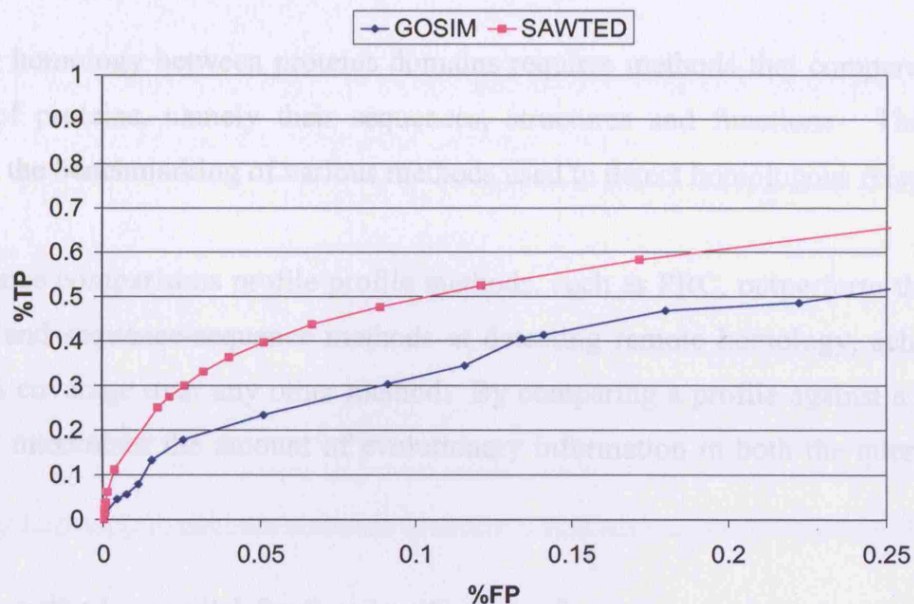


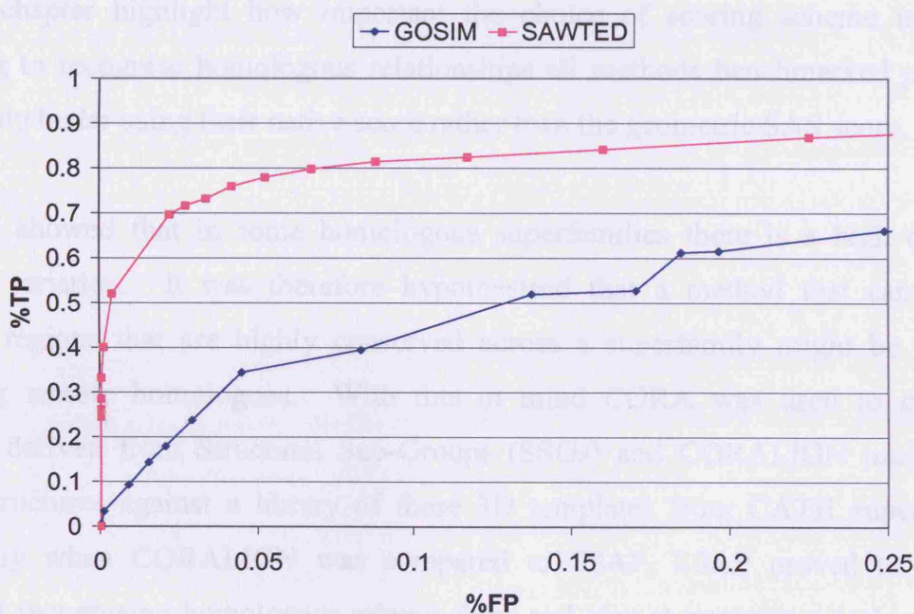
Figure 4.7. ROC curves showing the performance of the SAWTED method of scoring text vectors extracted from the PDB on both the full corpus and a restricted corpus only containing SWISSPROT keywords. Figure (a) shows the performance based on homology and Figure(b) show the performance based on identifying functionally related homologues as defined by their enzyme classification.

## 4.4 Discussion

(A)



(B)



**Figure 4.8.** ROC curves showing the performance of the SAWTED (Full Corpus) and GOSIM (GOA\_S35 dataset) methods. Figure (a) shows the performance based on homology and Figure (b) show the performance based on identifying functionally related proteins as defined by their enzyme classification.



## ***4.4 Discussion***

Detecting homology between proteins domains requires methods that compare different features of proteins, namely their sequences, structures and functions. This chapter presented the benchmarking of various methods used to detect homologous relationships.

For sequence comparisons profile-profile methods, such as PRC, outperform the profile-sequence and sequence-sequence methods at detecting remote homology, achieving an extra 10% coverage over any other method. By comparing a profile against a library of profiles it maximises the amount of evolutionary information in both the query and the target.

Structural methods are vital for the classification of very remote homologues when the sequence signal has faded. The analyses presented in this chapter coupled with the previous chapter highlight how important the choice of scoring scheme is. When attempting to recognise homologous relationships all methods benchmarked performed significantly better using their native score rather than the geometric SAS score.

Chapter 2 showed that in some homologous superfamilies there is a high degree of structural variation. It was therefore hypothesised that a method that captured the structural regions that are highly conserved across a superfamily might be useful in classifying remote homologues. With this in mind CORA was used to create 3D templates derived from Structural Sub-Groups (SSGs) and CORALIGN used to scan domain structures against a library of these 3D templates from CATH superfamilies. Surprisingly when CORALIGN was compared to SSAP, SSAP proved to be more accurate at recognising homologous relationships and also at separating analogues from homologues. A possible explanation for this may be that the conserved contacts currently identified by the CORA method are not in fact indicative of homology but simply represent the core of the fold. Alternatively, higher thresholds on structural similarity

may be needed for clustering the SSGs to ensure more accurate alignments and less noise in the contact plots.

Finally different ways of comparing functional information was explored. Using the GOSIM method to compare the semantic similarity of GO terms it was shown that there was significant gain in performance achieved by extending functional annotations through inheritance. The performance on all data-sets was rather disappointing. This may be partly caused by the fact that some terms are not annotated fully, and have been given very general annotations, e.g. kinase, that result in low semantic similarity scores when compared against more comprehensively annotated terms e.g. serine-threonine kinase. A possible improvement in performance could be achieved by excluding general annotations, though such an approach would have a cost in decreasing the overall annotation coverage.

In contrast to the GO based methods, complete functional annotation of all domains in the dataset could be achieved by using the text information from the PDB files. The SAWTED method which compares vectors of text extracted from the PDB, outperformed the GOSIM method by as much as 40% when identifying functionally related homologues.

In summary, when classifying remote homologues, individual features may have diverged sufficiently between structures to make classification difficult if only one measure of similarity is used. In this chapter different methods of measuring protein similarity have been assessed for their performance in identifying homologous relationships. It is clear that for remote homologues the combination of different signals, sequence, structure and function, may give a more powerful signal for accurate classification into evolutionary families.

In the next chapter the outputs of the methods benchmarked here will be used as features for the construction and benchmarking of a machine learning classifier to predict protein homology.

# 5 A Machine Learning Approach to Homologue Recognition

## *5.1 Background and Aims*

Recent high throughput approaches to analyse biological systems provide vast quantities of data. The international genomics initiatives have resulted in the sequence and structure databases expanding exponentially over time. However the cost of such approaches is that large numbers of the newly deposited sequences and structures have no information about their function. To improve our understanding of the biological context of such entities, biological information associated with evolutionary related proteins can be used. Since only a small percentage of sequences are experimentally characterised (<10%) methods to quantify relatedness, that is to identify remote homology, are in great demand to bridge this knowledge gap.

Traditionally the classification of protein structure data, for example in CATH or SCOP, in terms of evolutionary relationships has been a largely subjective activity, guided by expert knowledge. The maturation of structural classification resources such as SCOP (Andreeva et al. 2004) has required a high degree of manual assessment but how long can these resources keep pace with high throughput structure determination?

CATH (Orengo et al. 1997) employs a semi-automated approach to protein structure classification. The structural comparison algorithms SSAP (Orengo, Taylor 1996) and CATHEDRAL (see Chapter 3) are used to define domain boundaries and assign fold groups based on the three-dimensional environment of residues. Classifying domains further into homologous superfamilies still requires manual analysis, particularly for remote homologues where sequence signals are weak.

The FSSP resource of Holm and co-workers (1997) uses a fully automated approach to determine pairwise measures of homology between protein structures. The DALI algorithm (Holm, Sander 1998) is used to measure structural similarity in terms of a Z-score and the authors state that when the Z-score is greater than 6 and the sequence identity greater than 25%, there is a high probability that an evolutionary relationship exists. However, Hadley and Jones (1999) showed that the agreement with homologues identified by manual inspection in SCOP falls to below half when the sequence identity is below 20% and much less if the DALI Z-score is below 6. This highlights the problems of automated assignment of homology using fixed thresholds.

A principal difficulty in automated homology detection is the classification of proteins with the same fold into appropriate homologous superfamilies. This is difficult for two reasons. Firstly, as demonstrated in Chapter 2 different superfamilies show different tolerances to variability in sequence, structure and function, with some families seemingly well conserved in all attributes, while other families have diverged to such an extent that even the global structural similarity between remote homologues is negligible. Therefore, if two proteins have the same fold but are in different homologous superfamilies they are not necessarily unrelated by evolution, but may simply mean that there is not enough empirical evidence currently to be certain a relationship exists. Furthermore, due to a limited number of ways that  $\alpha$ -helices and  $\beta$ -sheets can pack three dimensionally there is the possibility of convergence of evolutionary unrelated proteins to adopt similar folds (Chothia 1992). These scenarios result in a very broad range of values in sequence, structural and functional similarity observed between relatives in different protein fold groups and superfamilies. Therefore finding an automated approach towards classification which yields both a high coverage and a low error rate is problematic.

Another problem is that fold space is not uniformly populated, four architectures,  $\alpha$ -orthogonal (1.10), the two-layer  $\beta$ -sandwich (2.60), two layer ( $\alpha\beta$ ) sandwiches (3.30) and the three-layer ( $\alpha\beta$ ) sandwiches (3.40) are very highly populated in CATH and the Protein Data Bank (PDB), comprising nearly 60% of all structural families having at least

3 diverse relatives ( $\leq 35\%$  sequence identity). Structural annotation of completed genomes suggest that the high populations of these architectures in the PDB is not simply due to over-sampling but genuinely reflects high occurrence in the genomes (Orengo, Thornton 2005). Many of these highly populated superfamilies are very structurally diverse with some relatives varying in size by fourfold or more. This all contributes to the difficulty of separating homologues from convergently folded analogues and delineating fold space in general.

Dietmann and Holm (2001) produced the most sophisticated and successful approach to automated classification of structural homologues to date. They aimed to replicate the SCOP protein structure classification with a method based on the premise that natural selection preserves structural and functional continuity within a diverging protein family. To overcome the difficulties of different protein families showing different rates of structural divergence, structural similarity was first used to cluster proteins into a discrete fold 'dendrogram'. A neural network was trained against the fold-to-superfamily transition in SCOP (Murzin et al. 1995) using various features including DALI Z-score, sequence identity, conserved ligand contacts and 'functional preference'. The neural network was then used to subdivide the identified fold groups into superfamilies based on these similarity measures. In a validation test against the SCOP classification 77% of homologous pairs were identified with 92% specificity and 85% accuracy (see Equations 3.1-3.3).

Machine learning can be described as the ability of a system to improve its performance based on its past performance. In the context of homology recognition the concept is that by exposing the system to examples of protein homologues and non-homologues the 'machine' can learn the rules of homology and therefore predict such relationships in unknown examples. Machine learning systems generally use non-linear classification to combine information about the training examples to produce some kind of prediction or classification. As detailed in Section 1.5 of the Introduction common examples of machine learning systems to tackle classification problems include artificial neural networks and support vector machines. Since the work of Holm *et al.* over 6 years ago

showed encouraging results the use of machine learning techniques to combine different aspects of protein similarity to recognise homology seems a reasonable approach for optimally combining different evolutionary signals. Furthermore the sequence and structure databases have expanded considerably since then and the increase in information relating to homologues together with improvements in methods of structure comparison (see Chapter 3) and sequence profiles (Reid A et al. 2007) suggests that better performances in these machine learning approaches might be expected. In this chapter an artificial neural network has been used to combine sequence, structure and functional similarity measures. Further information on the construction and training of artificial neural networks can be found in the Section 1.5.1 of the Introduction.

This chapter shows the construction and benchmarking of an artificial neural network classifier to predict protein homology. This included the process of feature selection, architecture optimisation and finally optimising and benchmarking the final classifier. The results from algorithms quantifying the sequence, structure and functional similarity between protein domains were used to generate the features to train the classifier. This yielded a significant improvement in performance of the classifier when compared to the performance of the composite methods alone and other comparable techniques (e.g. of Holm *et al.*).

## ***5.2 Methods***

### ***5.2.1 Data Sets***

Similar datasets to those employed in benchmarking the sequence methods (see Section 4.2.1) were used to train, test and validate artificial neural networks designed to recognise homologous relationships. The benchmarking datasets comprised three separate sets containing increasingly remote homologues. These were constructed from version 2.6 of CATH.

An ‘S35’ set contains domains with 35% or less sequence identity to any relative in the set, an ‘S20’ set contains domains with 20% or less sequence identity to any relative and finally an ‘S10’ set contained domains that had only very remote homologous relationships no greater than 10% sequence identity. The S35 set contained 5234 domains from across 805 superfamilies. The S20 set contained 2934 domains from 549 superfamilies and the S10 set 1495 domains from 362 superfamilies.

### ***5.2.2 Data Generation & Feature Selection***

The best performing methods in terms of homology recognition from Chapter 4 were used to generate the features for the construction of the neural network. These included the profile-profile sequence comparison method PRC (Madera 2006), the structural comparison method, CATHEDRAL (see Chapter 3), the SAWTED method (MacCallum et al. 2000) to compare text vectors, the GOSIM method (Lord et al. 2003) to compare the semantic similarity of associated GO terms and a further method for comparing EC classification terms.

PRC Models were built using SAM-T2K (Karplus et al. 1998), with each sequence in the S35 dataset as a seed on the GenBank nr database. Each dataset (S35, S20, and S10) was scanned all against all HMM against HMM.



An all against all CATHEDRAL scan was performed for all 5234 domains in the S35 dataset as described in Section 3.2.3.

GOSIM was used to compare the associated GO term domain annotations of all domains in each dataset. All domains were annotated with GO terms, where possible, using the GOA multispecies UniProt to PDB electronically inferred mapping, culminating in 54% of the CATH domains having at least one assigned GO term. To increase the annotation coverage further inference of GO terms from homologous genomic sequences was used. The procedure for inference is described in Section 4.2.3.1 of Chapter 4. Inferring GO annotations to homologous domains that share greater than 35% sequence identity increased the domain coverage by GO terms to 64%. Furthermore in the benchmarking procedure detailed in Section 4.3.3.1 this dataset outperformed the 60%, 95% and identical inferred datasets at recognising both homologous and functional relationships.

The SAWTED method, as detailed in Section 4.2.3.2 of Chapter 4, was re-implemented to compare the text information extracted from the PDB files of each CATH domain. Text from the 'Title', 'Header', 'Compound' and 'Keywords' fields of each PDB file was extracted and formed the knowledge corpus. Vectors of text from each PDB file were calculated and scored, all against all, using the vector-cosine model of text retrieval (Wilbur, Yang 1996) as described in Section 4.2.3.2.

As described in Section 4.2.3 in Chapter 4 the PDBSprotEC (Martin 2004) database, linking PDB chains to EC numbers via SwissProt, was used to obtain the EC mapping of a PDB chain for all CATH domains within that chain. This led to 47% of the S35 CATH domains having an associated EC annotation.

From these methods the possible input features of the neural network are listed below and where appropriate the program used to obtain the information is shown in brackets;

- Domain 1 length (number of residues)
- Domain 2 length (number of residues)
- Number of structurally equivalent residues (CATHEDRAL)
- Number of positions in total structural alignment (including gaps) (CATHEDRAL)
- Structural Alignment Score (CATHEDRAL)
- RMSD (CATHEDRAL)
- SAS score (CATHEDRAL)
- E-Value for sequence similarity (PRC)
- Functional similarity raw score (GO-SIM)
- Functional similarity Z-Score (SAWTED)
- EC conservation

A semi-exhaustive feature selection protocol was constructed. The generalisation error was first estimated using only the features extracted from CATHEDRAL, then the PRC features were added and finally all combinations of the functional features (GO-SIM, SAWTED and EC) were evaluated. This culminated in 9 datasets with between 7 and 11 features.

### ***5.2.3 Optimisation and Benchmarking Procedure***

The artificial neural network package used was SNNS (Stuttgart Neural Network Simulator). The neural network was trained on pairwise sequence, structure and functional comparisons of protein domains and asked to produce a prediction of homology between 0 and 1 (0 for non-relative, 1 for homologue).

Layered feed-forward neural networks were constructed with various architectures. One input layer comprising ten input units was connected to a hidden layer, connected to the one output unit. The number of hidden units was initially set to  $2 \times (\text{number of input units}) + 1$  as used by Dietmann and Holm (2001) and varied until optimised. All weights in the neural network were initialised to a random value  $\pm 1$  prior to training.

For each dataset (S35, S20, S10) the input features were normalised to be between 0 and 1, and the datasets randomised and split into three partitions to form the training set, test set and validation set. The number of homologue and non-homologue examples in each set was balanced. This led to the S35 training, test and validation sets each containing 27936 patterns, the S20 sets each containing 14364 patterns and the S10 sets each containing 3442 patterns.

A neural network was trained for each dataset (S35, S20 and S10) using the standard back-propagation algorithm and the free parameters optimised. A binary activation function was used to provide a threshold type response on homologue recognition rather than a linear likelihood response. The early stopping technique was used to prevent the overfitting of the free parameters of the network. The error function (summed difference between desired and actual output) of the training set is monitored during learning and it can be expected to decrease over time until it converges on some value when the network has optimally learned the training set. The error function of the test set will fall and then rise once the neural network is overfitting towards the training set. Early stopping stops the training at the minimum error function of the test set, to ensure that the classifier is not biased towards the training set. Finally a validation set is passed through the trained neural network to assess the performance on unseen data.

For each dataset (S35, S20, S10) the final validation set prediction result was measured using the Mathews Correlation Coefficient (MCC) (see Equation 5.4) and also plotted against the composite methods as a ROC curve. Accuracy is also used as a statistical measure of how well a binary predictor predicts each class based on the number of true positives and false positives observed in the validation set and is defined in Equation 5.1.

Conversely sensitivity is the statistical measure of how well a binary predictor correctly identifies one class, in the context of this chapter homologous pairs for example (see Equation 5.2). Finally specificity is a statistical measure of how well the classifier identifies negative examples (see Equation 5.3).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

**Equation 5.1. Accuracy equation.**

$$sensitivity = \frac{TP}{TP + FN}$$

**Equation 5.2. Sensitivity equation.**

$$specificity = \frac{TN}{TN + FP}$$

**Equation 5.3 Specificity equation.**

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

**Equation 5.4 Matthew's Correlation Coefficient Equation.**

TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

## 5.2.4 Superfold Classifier

As discussed previously certain folds and architectures are far more prevalent in the protein domain databases (e.g. CATH, SCOP) than others. These highly recurrent folds have been termed 'Superfolds' (Orengo et al. 1994). If a general classifier is built using data from these protein structure databases it will contain the same biases. This will lead to the classifier seeing more examples of relationships between these types of domains than between domains of rarer folds. Analyses of completed genomes (Marsden et al. 2007) have shown that sequences adopting these Superfolds are found as frequently in the genomes as in the PDB and therefore an automated classifier with bias to frequently observed relationships may actually be of some benefit. However it was decided to investigate whether creating a separate classifier for each Superfold as well as a 'generic'

classifier for other relationships provides more discriminatory power than a single classifier.

Folds that had greater than 5000 analogous relationships were selected as Superfolds. These included the Orthogonal Bundle Arc Repressor Fold (1.10.10), the Jelly Rolls (2.60.120), the Immunoglobulin-like fold (2.60.40), the TIM Barrels (3.20.20), the Alpha-Beta Plaits (3.30.70) and the Rossmann fold (3.40.50). Homologous relationships in the S35 dataset were partitioned into one of the Superfolds or, if not a Superfold, to a 'generic' bin. Neural networks were constructed, trained and validated as described in Section 5.2.3.

## 5.3 Results

### 5.3.1 Feature Selection

A semi-exhaustive feature selection protocol was constructed using the features described in section 5.2.2. Using the S35 dataset a neural network was trained using the standard back-propagation algorithm with default parameters ( $\eta = 0.1$  &  $d_{max} = 0.2$ ) and the number of nodes in the hidden layer set to  $2 \times (\text{number of input units}) + 1$  for each feature set.

Table 5.1 shows the feature selection that gives the optimal performance. The feature combinations are ranked by the mean squared error (MSE), the average distance of prediction by the classifier compared to the correct answer. It can be seen that the GOSIM score feature increases the summed squared error and therefore provides no additional discriminatory power when included alongside the SAWTED or EC conservation score. The GOSIM score feature was therefore not used as a feature in the benchmarking protocol to reduce the dimensionality of the classification problem. This left 10 input features to utilise in the classifier.

The process of identifying the optimal features is not independent of the optimisation of the structure of the neural network. With this in mind the exhaustive feature selection was repeated over a range of neural network architectures with the number of hidden nodes varied from 15 to 25. Presented in Table 5.1 are the results of the feature selection for the optimal number of hidden nodes (20).

Rank	Domain 1 Length	Domain 2 Length	No. of Equiv. Residues	Length of Alignment	Structural Alignment Score	RMSD	SAS Score	PRC E-value	GO-SIM Score	SAWTED Z-Score	EC	SSE/N
1	X	X	X	X	X	X	X	X		X	X	0.0536
2	X	X	X	X	X	X	X	X	X	X	X	0.0537
3	X	X	X	X	X	X	X	X	X		X	0.0547
4	X	X	X	X	X	X	X	X		X		0.0551
5	X	X	X	X	X	X	X	X			X	0.0557
6	X	X	X	X	X	X	X	X	X	X		0.0559
7	X	X	X	X	X	X	X	X	X			0.0564
8	X	X	X	X	X	X	X	X				0.0600
9	X	X	X	X	X	X	X					0.0976

**Table 5.1.** Shows the rank based on the normalised sum squared error for each of the feature sets.

### 5.3.2 Optimisation of the Neural Network

Following the feature selection the architecture was re-optimised using the final feature set. Again using the S35 dataset a neural network was trained using the standard back-propagation algorithm with default parameters and the number of nodes in the hidden layer varied. Early stopping was used to prevent overfitting. Figure 5.1 shows that the optimum number of hidden nodes (defined by the lowest sum squared error function) was found to be 20. Therefore the architecture of an input layer containing 10 nodes, a hidden layer containing 20 nodes and an output layer of one node was used for all training. This leads to a final architecture with  $(10 \times 20) + (20 \times 1) = 220$  adjustable weights.



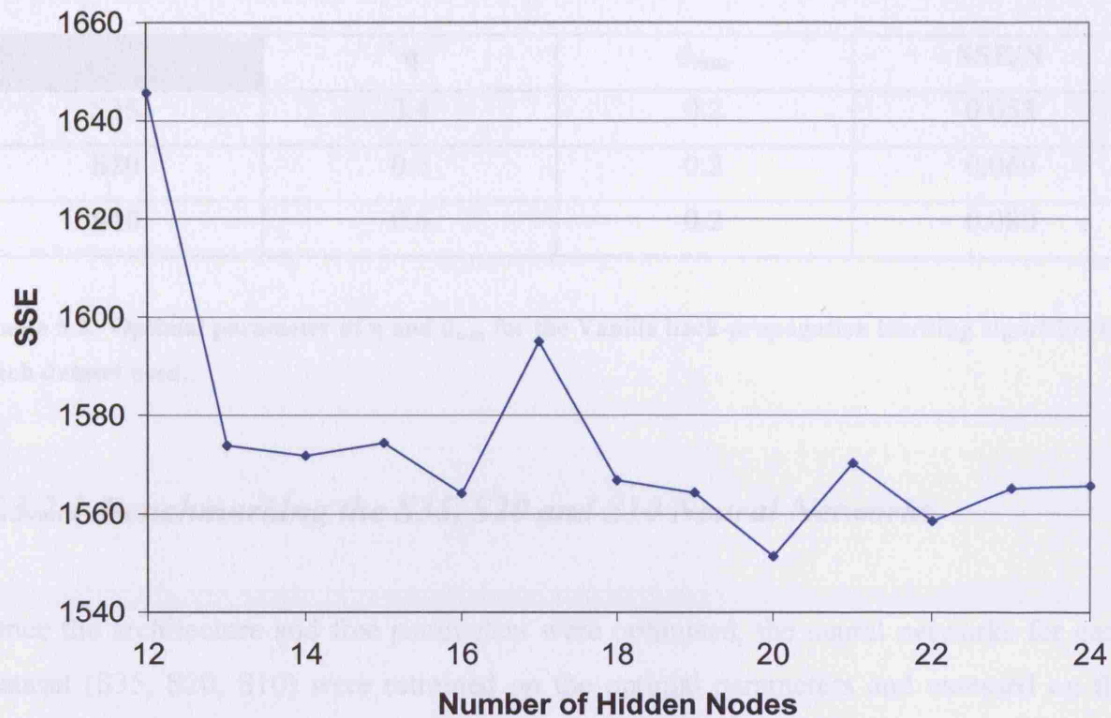


Figure 5.1. Graph showing the optimisation of the architecture of the hidden layer.

Once the architecture was optimised it was necessary to optimise the free parameters associated with the standard back-propagation algorithm for each dataset. The version of the standard back propagation used in the SNNS package is termed “Vanilla” back-propagation and has two free parameters; the learning parameter  $\eta$  and the parameter  $d_{max}$ . The learning parameter specifies the step width of the gradient descent and  $d_{max}$  represents the maximum difference between a teaching value and an acceptable value of an output unit which is tolerated i.e. which is propagated back through the network. If values above 0.9 should be regarded as 1 and values below 0.1 regarded as 0 then  $d_{max}$  should be set to 0.1. This prevents overtraining of the network.

Using each dataset a neural network was trained using the standard back-propagation algorithm with parameters ( $\eta = 0.1-1.0$  &  $d_{max} = 0-0.2$ ) again early stopping was used to prevent overfitting. Table 5.2 shows the optimal values of these parameters for each the S35, S20 and S10 datasets.

	$\eta$	$d_{max}$	SSE/N
<b>S35</b>	0.4	0.2	0.053
<b>S20</b>	0.6	0.2	0.069
<b>S10</b>	0.6	0.2	0.080

**Table 5.2.** Optimal parameter of  $\eta$  and  $d_{max}$  for the Vanilla back-propagation learning algorithm for each dataset used.

### 5.3.2.1 Benchmarking the S35, S20 and S10 Neural Networks

Once the architecture and free parameters were optimised, the neural networks for each dataset (S35, S20, S10) were retrained on the optimal parameters and assessed on the basis of the predictions made for the validation set. Figure 5.2 shows the performance of the neural network predictors against the best performing sequence comparison method, PRC, the best performing structural comparison method, CATHEDRAL (native score) and the best performing functional methods SAWTED and EC-Conservation for each dataset (S35, S20, S10). One can see the neural network predictors for each dataset significantly outperforms all other methods. On the S35 dataset (Figure 5.2(a)) the neural network predictor achieves 97% coverage for a 5% error rate. The sequence profile based PRC method is the next best performing method with coverage of 86% followed by the native CATHEDRAL score at 85%. The SAWTED PDB text comparison method achieves coverage of 45%, whilst using EC conservation yields coverage of 20% for a 5% error.

On the S20 dataset (Figure 5.2(b)) the neural network predictor again achieves 97% coverage for a 5% error. CATHEDRAL is the next best performing method on the S20 dataset achieving 15% greater coverage than PRC at a 5% error. This fall in PRCs performance is to be expected as the sequence signal is reduced. On the S10 dataset the neural network still achieves 95% coverage at 5% error, with CATHEDRAL at 80% and

PRC at 62%. It is highly encouraging that the performance of the neural networks remains relatively consistent on all datasets even when the composite methods are beginning to struggle to recognise the more remote homologous.



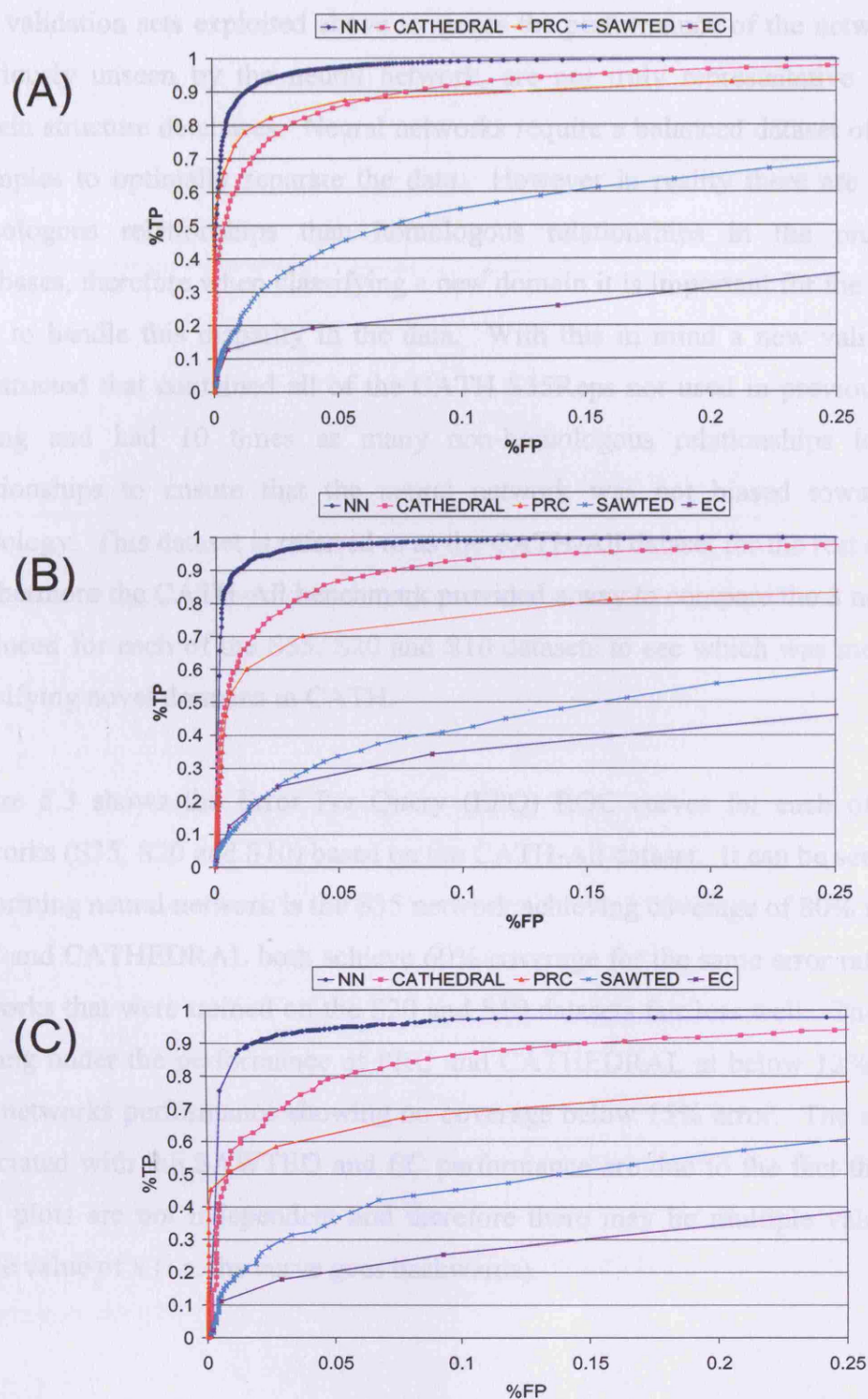


Figure 5.2. ROC curves showing the performance of the neural networks at homology recognition compared to the sequence comparison methods (PRC), the structure comparison method (CATHEDRAL (native score)) and function comparison methods (PDB-SAWTED & EC Conservation) on the S35(a), S20(b) and S10(c) datasets.

The validation sets exploited above to assess the performance of the networks, although previously unseen by the neural network, are not truly representative of the data in protein structure databases. Neural networks require a balanced dataset of true and false examples to optimally separate the data. However in reality there are far more non-homologous relationships than homologous relationships in the protein structure databases, therefore when classifying a new domain it is important for the classifier to be able to handle this disparity in the data. With this in mind a new validation set was constructed that contained all of the CATH S35Reps not used in previous training and testing and had 10 times as many non-homologous relationships to homologous relationships to ensure that the neural network was not biased towards predicting homology. This dataset is referred to as the CATH-All dataset for the rest of this chapter. Furthermore the CATH-All benchmark provided a way to compare the 3 neural networks produced for each of the S35, S20 and S10 datasets to see which was most effective at classifying novel domains in CATH.

Figure 5.3 shows the Error Per Query (EPQ) ROC curves for each of the 3 neural networks (S35, S20 and S10) based on the CATH-All dataset. It can be seen that the best performing neural network is the S35 network achieving coverage of 80% for a 5% error. PRC and CATHEDRAL both achieve 60% coverage for the same error rate. The neural networks that were trained on the S20 and S10 datasets fair less well. The S20 network dipping under the performance of PRC and CATHEDRAL at below 12% error and the S10 networks performance showing no coverage below 15% error. The unusual curves associated with the SAWTED and EC performance are due to the fact that the axes of EPQ plots are not independent and therefore there may be multiple values of y for a single value of x (i.e. the curve goes backwards).

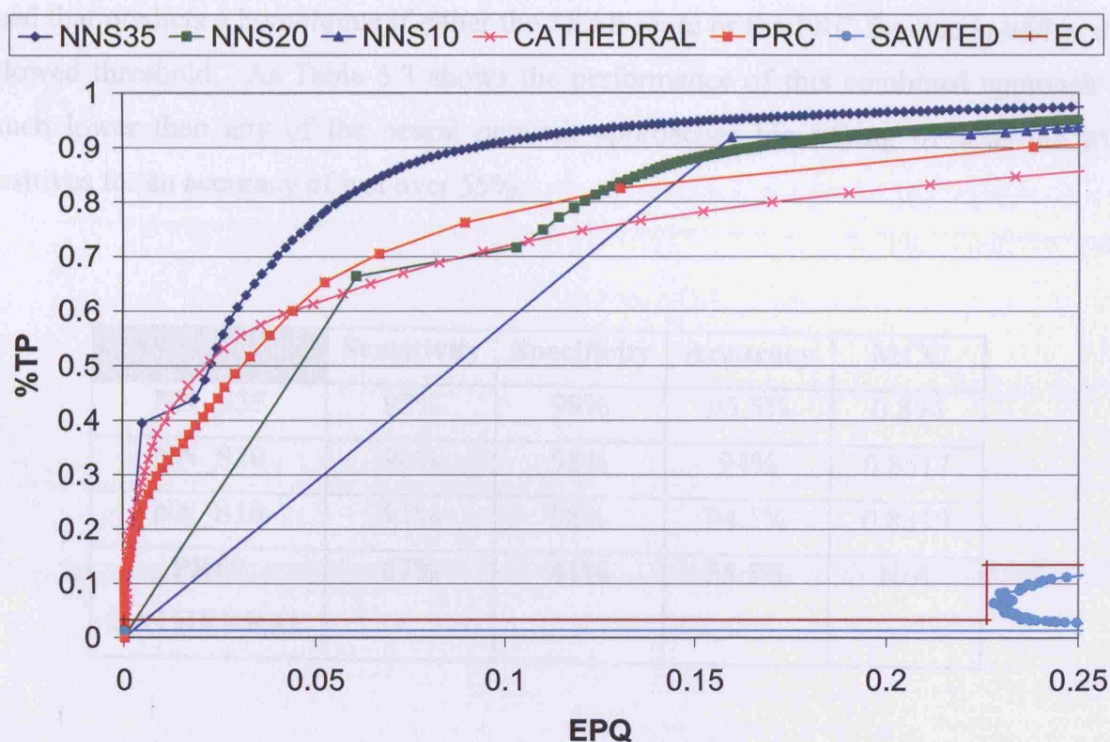


Figure 5.3. EPQ curves showing the performance of the neural networks created on the S35, S20 and S10 datasets using the CATH-All dataset. Also shown is the performance of the individual methods PRC, CATHEDRAL, PDB-SAWTED and EC Conservation.

Table 5.3 shows that when considering the proportion of true positives, false negatives, true negatives and false positives (TP, FP, TN, FP) the S35 dataset performs best at detecting homologues relationships in the CATH-All dataset achieving an accuracy of 95.5% (see Equation 5.1) and a sensitivity of 92% (see Equation 5.2). Table 5.3 shows that both the S20 and S10 accumulate more false positives and therefore it can be assumed that these networks are over sensitive presumably because they were trained on very remote homologues.

Neural networks combine features in a non-linear way. Therefore the effect of combining the PRC and CATHEDRAL methods in a linear, threshold based way was also assessed. Suitable thresholds were calculated that produced a 5% error rate in the methods individual benchmarks as presented in Chapter 4. This corresponded to a



CATHEDRAL score of 80 and a PRC E-value of  $1e-196$ . An OR logic relationship was used that predicts a homologue if either the SSAP score or the PRC E-value is within the allowed threshold. As Table 5.3 shows the performance of this combined approach is much lower than any of the neural network approaches identifying 67% of the true positives for an accuracy of just over 55%.

	Sensitivity	Specificity	Accuracy	MCC
NN_S35	92%	99%	95.5%	0.894
NN_S20	90%	98%	94%	0.8517
NN_S10	91%	98%	94.5%	0.8319
PRC/ CATHEDRAL	67%	41%	55.5%	N/A

**Table 5.3.** The performance of the three neural networks created on the S35, S20 and S10 dataset in homology recognition on the CATH-All dataset in terms of Accuracy ( $TP+TN/TP+FP+TN+FN$ ) and Mathews Correlation Coefficient (MCC). Also shown is the performance of combining PRC and CATHEDRAL scores.

### 5.3.2.2 Benchmarking the SuperFold Neural Networks

The S35 network benchmarked above was not built to take into account of the fact that some areas of fold space are densely populated while others are more sparse. As the classifier will ‘see’ more relationships including domains from highly populated areas of fold space it may be biased towards ‘learning’ characteristics of homology associated with those types of domain. Four architectures,  $\alpha$ -orthogonal (1.10), the two-layer  $\beta$ -sandwich (2.60), two layer ( $\alpha\beta$ ) sandwiches (3.30) and the three-layer ( $\alpha\beta$ ) sandwiches (3.40) are very highly populated in CATH and the Protein Data Bank (PDB), comprising nearly 60% of all structural families. Six highly populated folds from the highly populated architectures were chosen on the basis of having greater than 5000 analogous



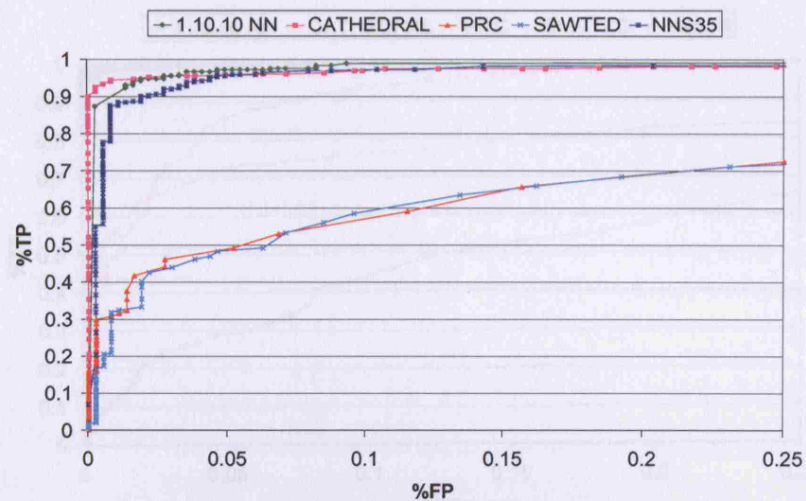
relationships. These folds were the Arc Repressor Fold (1.10.10), the Jelly Rolls (2.60.120), the Immunoglobulin-like fold (2.60.40), the TIM Barrels (3.20.20), the Alpha-Beta Plaits (3.30.70) and the Rossmann fold (3.40.50). Those relationships that did not contain a domain from any of the above folds were termed 'Generic' and designated their own category. For each of these seven categories (6 Superfolds + 1 generic) neural networks were constructed to see if fold specific classifiers could be more discriminatory than one general neural network trained on all the data.

Figure 5.4 shows the validation set ROC curves for neural networks for the 6 highly populated folds and the Generic classifier and compares the performance of them against the S35 network and their individual methods (i.e. CATHEDRAL, PRC, and SAWTED). At a 5% error, the networks trained on the Superfolds perform at least as well as the general S35 network and for some folds a significant improvement is observed.

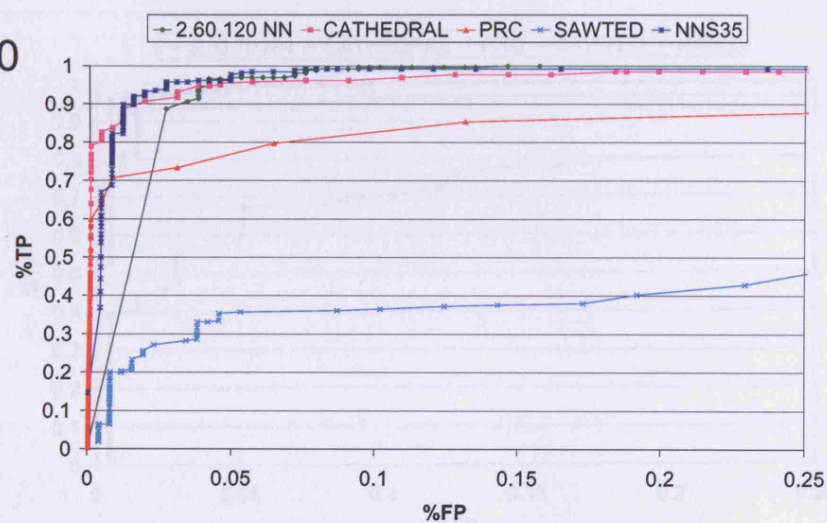
Different Superfolds show markedly different results in terms of how well the classifier and the individual methods perform. The ROC curve associated with the Orthogonal Bundle Arc Repressor Fold (1.10.10) show the classifier achieving coverage of 97% (at 5% error) with CATHEDRAL just behind. However PRC only achieves 50% coverage at the same error rate. This suggests that in this fold group many of the homologous relationships within CATH are very remote with little sequence similarity however the relatives remain structurally similar. For this reason the S35 general classifier also has problems detecting relationships in this fold group.

CATHEDRAL and PRC both perform relatively poorly on the Alpha-Beta Plaits (3.30.70), 85 and 72% coverage for a 5% error respectively. However the classifier stills achieves a coverage of 100% at a 1% error again highlighting the benefits of a combinatorial approach to homologue recognition.

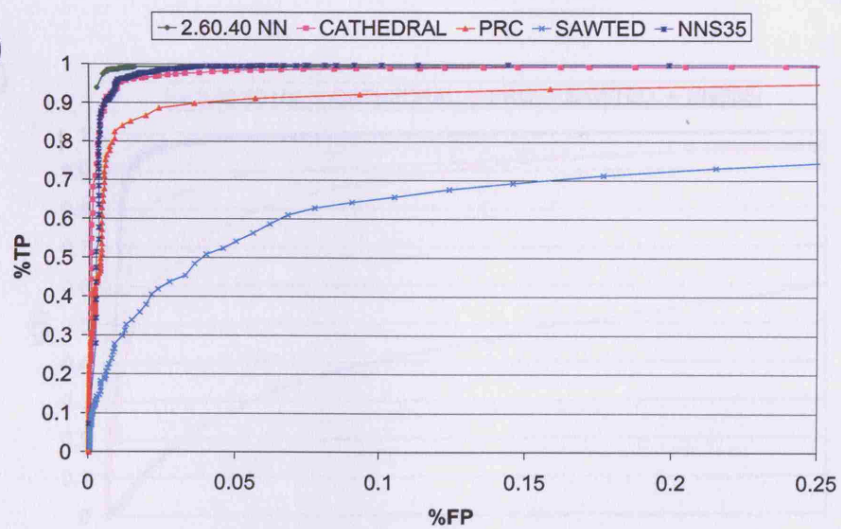
(A) 1.10.10



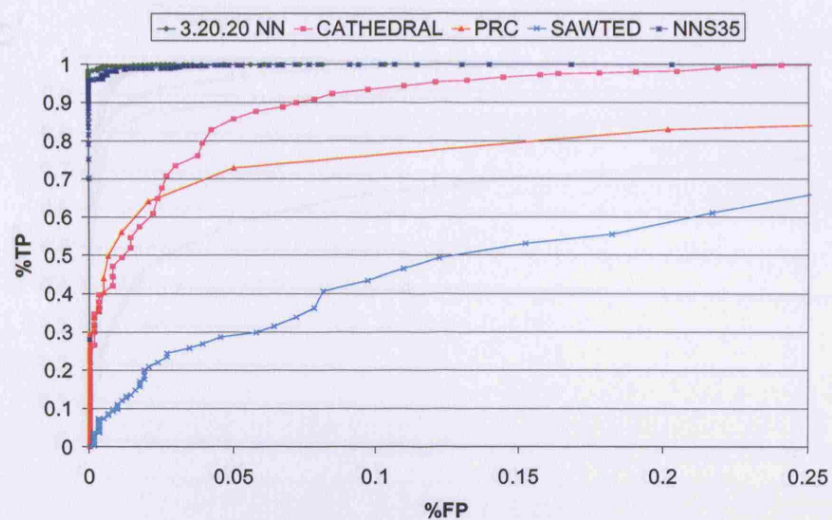
(B) 2.60.120



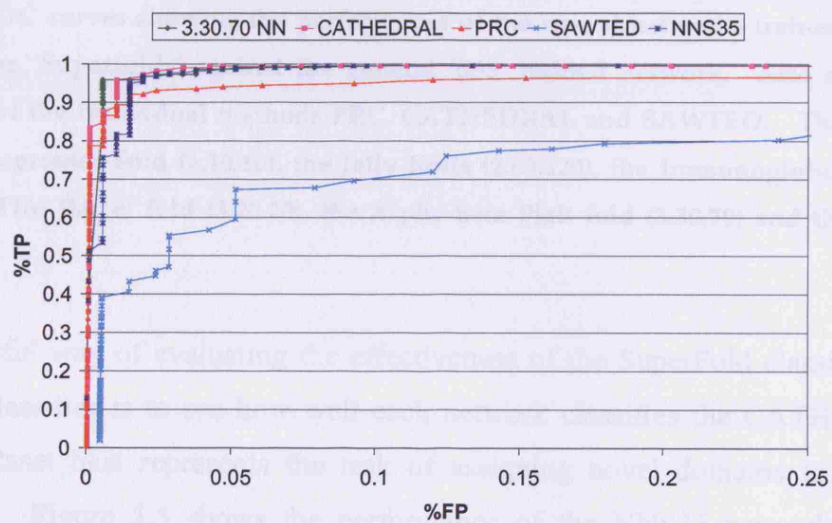
(C) 2.60.40



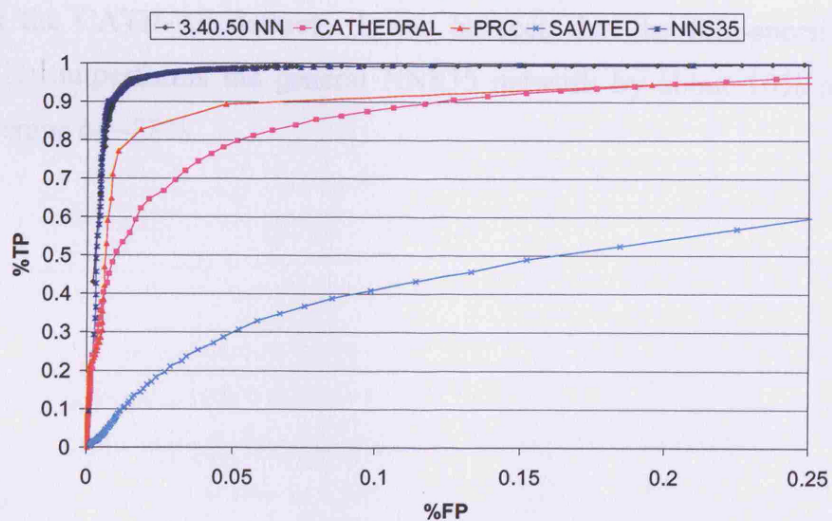
(D) 3.20.20



(E) 3.30.70



(F) 3.40.50





(G) 'Generic'

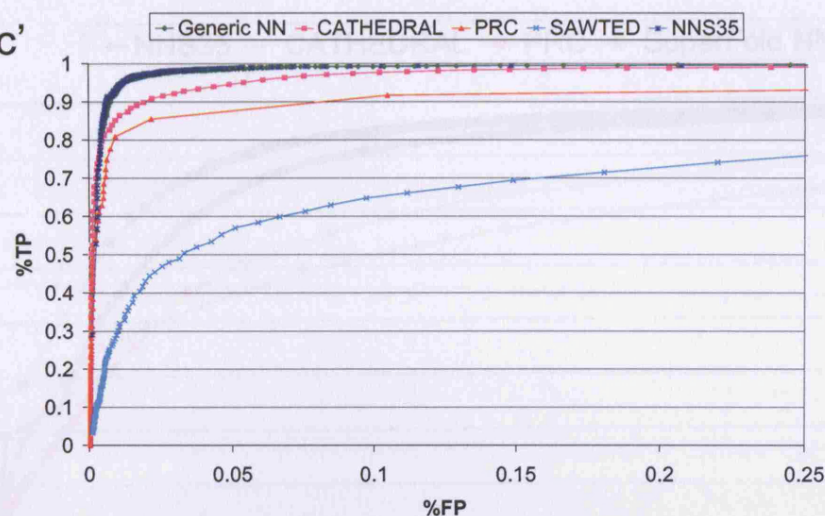


Figure 5.4. ROC curves showing the performance of the neural networks trained specifically on each of the 'Superfolds' against the general 'S35' trained network. Also shown is the performance of the individual methods PRC, CATHEDRAL and SAWTED. The Superfolds are the Arc Repressor Fold (1.10.10), the Jelly Rolls (2.60.120), the Immunoglobulin-like fold (2.60.40), the TIM Barrel fold (3.20.20), the Alpha-Beta Plait fold (3.30.70) and the Rossmann fold (3.40.50).

The most useful way of evaluating the effectiveness of the SuperFold classifiers versus the general classifier is to see how well each network classifies the CATH-All dataset since this dataset best represents the task of assigning novel domains to the CATH classification. Figure 5.5 shows the performance of the NNS35 network versus the SuperFold neural network, which is the combined performance of the fold-specific classifiers, on the CATH-All dataset. It can be seen that the fold-specific classifier (SuperFold NN) outperforms the general NNS35 network by about 10% at 5% error, achieving coverage of ~88%.

	Specificity	Accuracy	MAE
SuperFold NN	93%	73.9%	0.075
NN_S35	83%	64.4%	0.184

Table 5.4. The performance of SuperFold neural networks created on the 330,330 and 310 dataset in homology recognition on the CATH-All dataset in terms of Accuracy (TP+TN/TP+FP+TN+FN) and Matthews Correlation Coefficient (MCC).

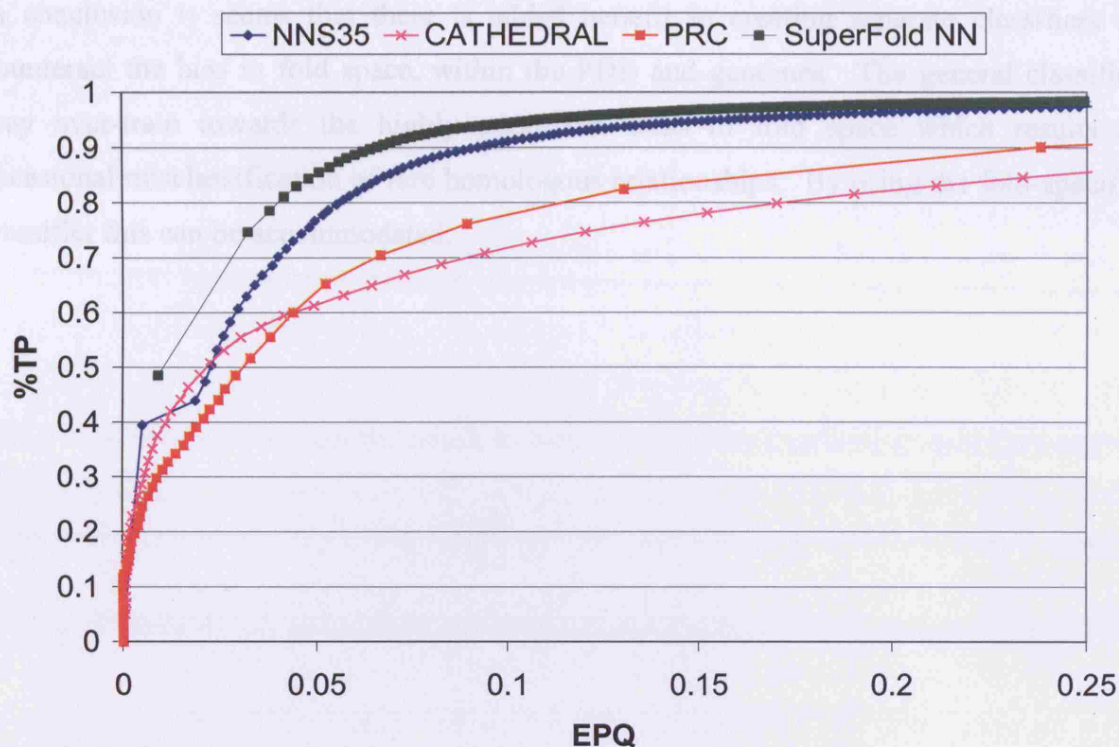


Figure 5.5. ROC curves showing the performance of the NNS35 network, the Superfold neural network and the individual methods for the CATH-All dataset.

Table 5.4 shows that when considering the proportion of true positives, false negatives, true negatives and false positives (TP, FP, TN, FN) the Superfold NN dataset performs best at detecting homologues relationships in the CATH-All dataset achieving an accuracy of 96% (see Equation 5.1) and a sensitivity of 93% (see Equation 5.2).

	Sensitivity	Specificity	Accuracy	MCC
<b>Superfold NN</b>	93%	99%	96%	0.915
<b>NN_S35</b>	92%	99%	95.5%	0.894

Table 5.4. The performance of the three neural networks created on the S35, S20 and S10 dataset in homology recognition on the CATH-All dataset in terms of Accuracy (TP+TN/TP+FP+TN+FN) and Mathews Correlation Coefficient (MCC).

In conclusion it seems that there is added benefit in creating separate classifiers to counteract the bias in fold space, within the PDB and genomes. The general classifier may over-train towards the highly populated areas of fold space which results in occasional misclassification of rare homologous relationships. By using the fold-specific classifier this can be accommodated.

## 5.4 Discussion

Due to high-throughput approaches such as the international genome initiatives, bioinformatic data resources, such as the protein sequence and protein structure databases, have been rewarded by a rapid influx of new data. However many of the proteins have no experimental annotations describing their function. Detecting homologous relationships is a vital tool in transferring information between well characterised and poorly characterised domains, and putting new entities into their correct evolutionary and biological context. As the characterisation of homologous relationships in CATH and SCOP can often be very time consuming because of the manual validation performed, the development of more sensitive automated approaches that achieve high coverage at low error rates is imperative to keep pace with the genome initiatives. This chapter presented a machine learning approach to homology recognition.

The sequence similarity, structural similarity and functional similarity of a pair of protein domains gives three independent signals that can provide evidence for a homologous relationship. During manual classification these signals are used subjectively by the curator to inform their decision. Here an automated approach was presented that uses this information to teach a neural network to recognise the patterns of homology.

Neural networks were trained on 3 balanced datasets containing increasingly remote homologues using the optimal features which included scores extracted from the sequence profile-profile method PRC, the structural comparison method CATHEDRAL, the text vector comparison method SAWTED and also a measure of EC conservation. For all balanced validation datasets the S35 neural network could identify ~97% of the homologous relationships for a 5% error, significantly outperforming the individual methods.

The protein structure databases (e.g. PDB) are not balanced in terms of homologous and non-homologous relationships and therefore a dataset termed CATH-All was constructed



that had 10 times the number of non-homologous pairs to homologues to reflect the typical situation encountered when classifying a new domain in CATH. The neural network trained on the S35 dataset out performed the S20 and S10 datasets significantly with the two datasets trained on more remote homologous relationships proving to be too sensitive and accumulating more false positives. The S35 dataset was able to recognise 92% of the homologous relationships in this dataset at an accuracy of 96%.

Protein structure space is not evenly distributed, with the presence of highly populated architectures accounting for 60% of the PDB and the genomes. Therefore a general classifier will inherently be biased towards learning the relationships between these frequently observed types of domains, neglecting the rarer cases. Therefore separate neural networks were created for the most highly populated folds (Superfolds) within these architectures. The combined performance of these Superfold neural networks achieved a 10% increase in coverage at a 5% error over the general S35 classifier. This suggests that the S35 classifier was over-training on common examples and losing some discriminatory power in delineating the rarer relationships. The SuperFold NN classifier achieves a sensitivity of 93% on the CATH-All dataset with an accuracy of 96%. Previous approaches to using neural networks for detecting homologues (Dietmann, Holm 2001) achieved a sensitivity of 77% with a 85% accuracy on a similar dataset constructed from the SCOP protein classification. Therefore a significant improvement in automated homologue classification has been achieved. Possible reasons for this increase in performance may be due to the use of more sophisticated sequence and structure comparison methods to generate the learning features. Furthermore the protein sequence and structure databases have increased dramatically in size in the past 5 years and therefore more relationships were available to learn from. Finally the development of Superfold classifiers to account for the biases in fold space has provided more specificity allowing the correct classification of rare homologous relationships.

## 6. Conclusions

The aim of this thesis was to investigate different methods of recognising homologous relationships between protein domains with the overall goal of developing an automated protocol for homologue recognition that yielded high coverage and a low error rate. Protein structural family resources such as CATH and SCOP rely on the identification of homologous relationships in the classification of new structures. The identification of these relationships requires a large degree of manual validation and this is becoming increasingly difficult due to the high numbers of novel structures being produced by the structural genomics initiatives. Therefore an accurate and reliable automated homology recognition protocol could make significant strides in relieving this classification burden.

The design and implementation of an automated homologue recognition protocol was informed by an analysis of how different homologous superfamilies of proteins evolve in sequence, structure and function relationships and a characterisation of the mechanisms by which this occurs and this was presented in Chapter 2. It is apparent that some superfamilies remain structurally well conserved even when the sequences diverge significantly whilst others can tolerate extensive structural change. Results showed that greater than half of the highly populated superfamilies (comprising  $\geq 9$  sequence diverse sub-families) also show a high degree of structural variation and frequently diverge in function e.g. in the galectin binding domains the structural embellishments around the active site modulate the geometry and substrate accessibility. Superfamilies that are highly conserved in terms of structure often have functional constraints with many of these families involved in cell signalling where a large proportion of the exposed structure is likely to be involved in ligand binding and protein-protein interactions. Updated information on the variability observed between homologues determined in this analysis was presented in an established web resource the Dictionary of Homologous Superfamilies.

The first step in the classification of protein domains is to delineate the multi-domain protein structure chain into its composite domains followed by the identification of the correct fold. Chapter 3 describes a new structural comparison algorithm, CATHEDRAL, which combines both secondary structure matching and accurate residue alignment in an iterative protocol for determining the location of previously observed folds in novel multi-domain structures. CATHEDRAL was able to assign 76% of domain boundaries within a test set of 680 sequence diverse multi-domain chains correctly (within 10 residues of the manually assigned boundaries) compared to 33% for a sequence based protocol (HMMer). Furthermore CATHEDRAL performed best when benchmarked against other leading structural comparison methods in identifying the correct fold matches between single domains in CATH and a union dataset between SCOP and CATH. An interesting observation in this chapter highlighted the importance of the measure used to score structural similarity. It was shown that geometric scores based on the RMSD (e.g. SAS) are often better discriminators of fold space than the native scores employed by many algorithms which often perform better at detecting the closest structural neighbour. Another key finding was the importance of achieving a global alignment in terms of domain boundary assignment.

Chapter 4 presented the optimisation and benchmarking of several methods for detecting homology, this included methods that compare the structural similarity of proteins and methods that attempt to assess functional similarity. In terms of using sequence similarity as a gauge of homology the profile-profile method PRC outperformed all other sequence similarity methods recognising 10% more homologous relationships. When identifying homologous relationships through structural similarity CATHEDRAL performs better than most other widely used algorithms in recognising global domain structure similarity between homologues. Finally ways of measuring functional similarity to inform the assignment of homology were explored. The text comparison method SAWTED outperformed GOSIM in recognising both functional and homologous relationships, the latter being a method which compares the semantic similarity of assigned GO terms.

The sequence similarity, structural similarity and functional similarity of a pair of protein domains gives three independent signals that can provide evidence for a homologous relationship. In Chapter 5 a neural network was used to combine this information to recognise the patterns of homology. Neural networks were trained on 3 balanced datasets containing increasingly remote homologues using the optimal features which included scores extracted from the sequence profile-profile method PRC, the structural comparison method CATHEDRAL, the text vector comparison method SAWTED and also a measure of EC conservation. On a validation dataset compiled to represent the task of classifying a new domain in CATH the neural network trained on sequence diverse relatives with less than 35% sequence identity recognised over 80% of the homologous relationships for a 5% error significantly outperforming the individual methods.

Some areas of fold space are more populated than others, with the presence of highly populated architectures accounting for 60% of the PDB and the genomes. An analysis was conducted to determine whether neural networks created for the most highly populated folds (superfolds) within these architectures would give rise to more accurate classifiers. The combined performance of the Superfold neural networks achieves a 10% increase in coverage over the general classifier described above recognising 93% of the homologous relationships with an accuracy of 96%. Previous approaches to using neural networks for detecting homologues (Dietmann, Holm 2001) recognised 77% of homologues with 85% accuracy on a similar dataset constructed from the SCOP protein classification.

This thesis presented a new automated approach to recognising homologous relationships between protein domains that should provide great value in the pipeline of the CATH classification system. In the future, similar approaches could be used to identify functionally related homologues.

# References

1. Al-Shahib,A., Breitling,R., and Gilbert,D. 2005. FrankSum: new feature selection method for protein function prediction. *Int. J. Neural Syst.* **15**:259-275.
2. Altschul,S.F., Gish,W., Miller,W., Myers,E.W., and Lipman,D.J. 1990. Basic local alignment search tool. *J. Mol. Biol.* **215**:403-410.
3. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W., and Lipman,D.J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**:3389-3402.
4. Andreeva,A., Howorth,D., Brenner,S.E., Hubbard,T.J., Chothia,C., and Murzin,A.G. 2004. SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res.* **32**:D226-D229.
5. Apic,G., Gough,J., and Teichmann,S.A. 2001. Domain combinations in archaeal, eubacterial and eukaryotic proteomes. *J. Mol. Biol.* **310**:311-325.
6. Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T., Harris,M.A., Hill,D.P., Issel-Tarver,L., Kasarskis,A., Lewis,S., Matese,J.C., Richardson,J.E., Ringwald,M., Rubin,G.M., and Sherlock,G. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**:25-29.
7. Bairoch,A. 2000. The ENZYME database in 2000. *Nucleic Acids Res.* **28**:304-305.
8. Baldi,P. and Brunak,S. 2001. *Bioinformatics: the machine learning approach*. MIT Press, London.
9. Benson,D.A., Karsch-Mizrachi,I., Lipman,D.J., Ostell,J., and Wheeler,D.L. 2006. GenBank. *Nucleic Acids Res.* **34**:D16-D20.
10. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N., and Bourne,P.E. 2000. The Protein Data Bank. *Nucleic Acids Res.* **28**:235-242.
11. Boeckmann,B., Bairoch,A., Apweiler,R., Blatter,M.C., Estreicher,A., Gasteiger,E., Martin,M.J., Michoud,K., O'Donovan,C., Phan,I., Pilbout,S., and Schneider,M. 2003. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.* **31**:365-370.
12. Branden,J. and Tooze 1999. *Introduction to Protein Structure*. Garland.

13. Bray,J.E., Todd,A.E., Pearl,F.M., Thornton,J.M., and Orengo,C.A. 2000. The CATH Dictionary of Homologous Superfamilies (DHS): a consensus approach for identifying distant structural homologues. *Protein Eng* **13**:153-165.
14. Brenner,S.E., Chothia,C., and Hubbard,T.J. 1998. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl. Acad. Sci. U. S. A* **95**:6073-6078.
15. Cai YD, Liu XJ, Xu XB, and Chou KC 2002. Artificial neural network method for predicting protein secondary structure content. *Comput Chem.* **26**:347-350.
16. Camon,E., Magrane,M., Barrell,D., Lee,V., Dimmer,E., Maslen,J., Binns,D., Harte,N., Lopez,R., and Apweiler,R. 2004. The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res.* **32**:D262-D266.
17. Carugo,O. 2003. How root-mean-square distance (r.m.s.d.) values depend on the resolution of protein structures that are compared. *Journal of Applied Crystallography* **36**:125-128.
18. Chothia,C. 1992. Proteins. One thousand families for the molecular biologist. *Nature.* **357**:543-544.
19. Chothia,C. and Lesk,A.M. 1986. The relation between the divergence of sequence and structure in proteins. *EMBO J.* **5**:823-826.
20. Copley,R.R. and Bork,P. 2000. Homology among (betaalpha)(8) barrels: implications for the evolution of metabolic pathways. *Journal of molecular biology* **303**:627-641.
21. Coulson,A.F. and Moult,J. 2002. A unfold, mesofold, and superfold model of protein fold use. *Proteins.* **46**:61-71.
22. Dayhoff,M.O. 1978. Atlas of Protein Sequence and Structure.
23. Dietmann,S. and Holm,L. 2001. Identification of homology in protein structure classification. *Nat. Struct. Biol.* **8**:953-957.
24. Eddy,S.R. 1996. Hidden Markov models. *Curr. Opin. Struct. Biol.* **6**:361-365.
25. Efron,B. 1982. *The Jackknife, the Bootstrap and Other Resampling Plans.*
26. Finn, Mistry J, Schuster-Bockler B, Hollich V, Lassmann T, Moxon S, Marshall M, Khanna A, Durbin R, Eddy SR, Sonnhammer EL, and Bateman A 2007. Pfam: clans, web tools and services. *Nucleic acids research* **34**:D247-D251.



27. Flores,T.P., Orengo,C.A., Moss,D.S., and Thornton,J.M. 1993. Comparison of conformational characteristics in structurally similar protein pairs. *Protein Sci.* **2**:1811-1826.
28. Goutte C. 1997. Note on free lunches and cross-validation. *Neural Computation* **9**:1246-1249.
29. Grant,A., Lee,D., and Orengo,C. 2004. Progress towards mapping the universe of protein folds. *Genome Biol.* **5**:107.
30. Greene,L.H., Lewis,T.E., Addou,S., Cuff,A., Dallman,T., Dibley,M., Redfern,O., Pearl,F., Nambudiry,R., Reid,A., Sillitoe,I., Yeats,C., Thornton,J.M., and Orengo,C.A. 2006. The CATH domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic Acids Res.* **35**:D291-D297.
31. Grindley,H.M., Artymiuk,P.J., Rice,D.W., and Willett,P. 1993. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. Biol.* **229**:707-721.
32. Grishin,N.V. 2001. Fold change in evolution of protein structures. *J. Struct. Biol.* **134**:167-185.
33. Guyon,I. 2003. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* **1**:157-1182.
34. Hadley,C. and Jones,D.T. 1999. A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure.* **7**:1099-1112.
35. Harrison,A., Pearl,F., Mott,R., Thornton,J., and Orengo,C. 2002. Quantifying the similarities within fold space. *J. Mol. Biol.* **323**:909-926.
36. Harrison,A., Pearl,F., Sillitoe,I., Slidel,T., Mott,R., Thornton,J., and Orengo,C. 2003. Recognizing the fold of a protein structure. *Bioinformatics.* **19**:1748-1759.
37. Henikoff, S, and Henikoff JG 1993. Performance evaluation of amino acid substitution matrices. *Proteins* **17**:49-61.
38. Henikoff,S. and Henikoff,J.G. 1991. Automated assembly of protein blocks for database searching. *Nucleic acids research* **19**:6565-6572.
39. Heringa,J. and Taylor,W.R. 1997. Three-dimensional domain duplication, swapping and stealing. *Curr. Opin. Struct. Biol.* **7**:416-421.
40. Holland,T.A., Veretnik,S., Shindyalov,I.N., and Bourne,P.E. 2006. Partitioning protein structures into domains: why is it so difficult? *J. Mol. Biol.* **361**:562-590.

41. Holm,L., Ouzounis,C., Sander,C., Tuparev,G., and Vriend,G. 1992. A database of protein structure families with common folding motifs. *Protein Sci.* **1**:1691-1698.
42. Holm,L. and Sander,C. 1993. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.* **233**:123-138.
43. Holm,L. and Sander,C. 1994. Parser for protein folding units. *Proteins.* **19**:256-268.
44. Holm,L. and Sander,C. 1997. Dali/FSSP classification of three-dimensional protein folds. *Nucleic Acids Res.* **25**:231-234.
45. Holm,L. and Sander,C. 1998. Touring protein fold space with Dali/FSSP. *Nucleic Acids Res.* **26**:316-319.
46. Hou Y, Hsu W, Lee ML, and Bystroff C 2003. Efficient remote homology detection using local structure. *Bioinformatics* **19**:2294-2301.
47. Hubbard,T.J., Murzin,A.G., Brenner,S.E., and Chothia,C. 1997. SCOP: a structural classification of proteins database. *Nucleic Acids Res.* **25**:236-239.
48. Jawad,Z. and Paoli,M. 2002. Novel sequences propel familiar folds. *Structure.* **10**:447-454.
49. Jones,S., Stewart,M., Michie,A., Swindells,M.B., Orengo,C., and Thornton,J.M. 1998. Domain assignment for protein structures using a consensus approach: characterization and analysis. *Protein Sci.* **7**:233-242.
50. Juan D, Grana O, Pazos F, Fariselli P, Casadio R, and Valencia A 2003. A neural network approach to evaluate fold recognition results. *Proteins* **50**:600-608.
51. Kanehisa,M. and Goto,S. 2000. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**:27-30.
52. Karplus,K., Barrett,C., and Hughey,R. 1998. Hidden Markov models for detecting remote protein homologies. *Bioinformatics.* **14**:846-856.
53. Karplus,K., Katzman,S., Shackleford,G., Koeva,M., Draper,J., Barnes,B., Soriano,M., and Hughey,R. 2005. SAM-T04: what is new in protein-structure prediction for CASP6. *Proteins.* **61**:135-142.
54. Kendrew J, Dintzis,H., Parrish,R., Wyckoff H A, and Phillips,D. 1958. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature* **181**:662-666.
55. Kim H and Park H 2003. Protein secondary structure prediction based on an improved support vector machines approach. *Protein Eng.* **16**:553-560.

56. Kleywegt, G.J. 1996. Use of non-crystallographic symmetry in protein structure refinement. *Acta Crystallogr. D. Biol. Crystallogr.* **52**:842-857.
57. Kolodny, R., Koehl, P., and Levitt, M. 2005. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J. Mol. Biol.* **346**:1173-1188.
58. Koppensteiner, W.A., Lackner, P., Wiederstein, M., and Sippl, M.J. 2000. Characterization of novel proteins based on known protein structures. *J. Mol. Biol.* **296**:1139-1152.
59. Krishna, S.S. and Grishin, N.V. 2005. Structural drift: a possible path to protein fold change. *Bioinformatics.* **21**:1308-1310.
60. Krissinel, E. and Henrick, K. 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr. D. Biol. Crystallogr.* **60**:2256-2268.
61. Laskowski, R.A., Hutchinson, E.G., Michie, A.D., Wallace, A.C., Jones, M.L., and Thornton, J.M. 1997. PDBsum: a Web-based database of summaries and analyses of all PDB structures. *Trends Biochem. Sci.* **22**:488-490.
62. Lattman, E.E. and Rose, G.D. 1993. Protein folding--what's the question? *Proc. Natl. Acad. Sci. U. S. A* **90**:439-441.
63. Lee, D., Grant, A., Marsden, R.L., and Orengo, C. 2005. Identification and distribution of protein families in 120 completed genomes using Gene3D. *Proteins.* **59**:603-615.
64. Leonidas, D.D., Vatzaki, E.H., Vorum, H., Celis, J.E., Madsen, P., and Acharya, K.R. 1998. Structural basis for the recognition of carbohydrates by human galectin-7. *Biochemistry* **37**:13930-13940.
65. Lesk, A.M. and 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.
66. Lesk, A.M. and Chothia, C. 1982. Evolution of proteins formed by beta-sheets. II. The core of the immunoglobulin domains. *J. Mol. Biol.* **160**:325-342.
67. Liu, Y. and Eisenberg, D. 2002. 3D domain swapping: as domains continue to swap. *Protein science* **11**:1285-1299.
68. Lord, P.W., Stevens, R.D., Brass, A., and Goble, C.A. 2003. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics.* **19**:1275-1283.

69. Luscombe,N.M., Austin,S.E., Berman,H.M., and Thornton,J.M. 2000. An overview of the structures of protein-DNA complexes. *Genome Biol.* **1**.
70. MacCallum,R.M., Kelley,L.A., and Sternberg,M.J. 2000. SAWTED: structure assignment with text description--enhanced detection of remote homologues with automated SWISS-PROT annotation comparisons. *Bioinformatics.* **16**:125-129.
71. Madej,T., Gibrat,J.F., and Bryant,S.H. 1995. Threading a database of protein cores. *Proteins* **23**:356-369.
72. Madera,M. 2006. PRC - The Profile Comparer. *Unpublished Work*.
73. Madera,M. and Gough,J. 2002. A comparison of profile hidden Markov model procedures for remote homology detection. *Nucleic Acids Res.* **30**:4321-4328.
74. Makarova,K.S. and Grishin,N.V. 1999. Thermolysin and mitochondrial processing peptidase: how far structure-functional convergence goes. *Protein Sci* **8**:2537-2540.
75. Marsden,R., Lewis,T., and Orengo,C.A. 2007. Towards a comprehensive structural coverage of completed genomes; a structural genomics viewpoint. *BMC. Bioinformatics.* **8**:86-88.
76. Martin,A.C. 2004. PDBSprotEC: a Web-accessible database linking PDB chains to EC numbers via SwissProt. *Bioinformatics.* **20**:986-988.
77. McGuffin LJ and Jones DT 2002. C. *Proteins* **48**:44-52.
78. Meiler J and Baker D 2003. Coupled prediction of protein secondary and tertiary structure. *Proc Natl Acad Sci U S A* **100**:12105-12110.
79. Michel,G., Chantalat,L., Duee,E., Barbeyron,T., Henrissat,B., Kloareg,B., and Dideberg,O. 2001. The kappa-carrageenase of *P. carrageenovora* features a tunnel-shaped active site: a novel insight in the evolution of Clan-B glycoside hydrolases. *Structure.* **9**:513-525.
80. Milne,L. 1995. Feature Selection Using Neural Networks with Contribution Measures. *AI'95*.
81. Mizuguchi,K. and Blundell,T. 2000. Analysis of conservation and substitutions of secondary structure elements within protein superfamilies. *Bioinformatics.* **16**:1111-1119.
82. Mizuguchi,K., Deane,C.M., Blundell,T.L., and Overington,J.P. 1998. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.* **7**:2469-2471.

83. Murzin,A.G., Brenner,S.E., Hubbard,T., and Chothia,C. 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **247**:536-540.
84. Nagarajan,N. and Yona,G. 2004. Automatic prediction of protein domains from sequence information using a hybrid learning system. *Bioinformatics.* **20**:1335-1360.
85. Needleman,S.B. and Wunsch,C.D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* **48**:443-453.
86. Orengo,C.A. 1999. CORA--topological fingerprints for protein structural families. *Protein Sci.* **8**:699-715.
87. Orengo,C.A., Flores,T.P., Jones,D.T., Taylor,W.R., and Thornton,J.M. 1993. Recurring structural motifs in proteins with different functions. *Curr. Biol.* **3**:131-139.
88. Orengo,C.A., Jones,D.T., and Thornton,J.M. 1994. Protein superfamilies and domain superfolds. *Nature* **372**:631-634.
89. Orengo,C.A., Michie,A.D., Jones,S., Jones,D.T., Swindells,M.B., and Thornton,J.M. 1997. CATH--a hierarchic classification of protein domain structures. *Structure.* **5**:1093-1108.
90. Orengo,C.A., Sillitoe,I., Reeves,G., and Pearl,F.M. 2001. Review: what can structural classifications reveal about protein evolution? *J. Struct. Biol.* **134**:145-165.
91. Orengo,C.A. and Taylor,W.R. 1996. SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol.* **266**:617-635.
92. Orengo,C.A. and Thornton,J.M. 2005. Protein families and their evolution-a structural perspective. *Annu. Rev. Biochem.* **74**:867-900.
93. Park,J., Karplus,K., Barrett,C., Hughey,R., Haussler,D., Hubbard,T., and Chothia,C. 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.* **284**:1201-1210.
94. Pascarella,S. and Argos,P. 1992. Analysis of insertions/deletions in protein structures. *J. Mol. Biol.* **224**:461-471.
95. Pearson,W.R. 1990. Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol.* **183**:63-98.

96. Pietrokovski,S. 1996. Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nucleic Acids Res.* **24**:3836-3845.
97. Ptitsyn,O.B. 1998. Protein folding and protein evolution: common folding nucleus in different subfamilies of c-type cytochromes. *J. Mol. Biol.* **278**:655-666.
98. Ptitsyn,O.B. and Ting,K.L. 1999. Non-functional conserved residues in globins and their possible role as a folding nucleus. *J. Mol. Biol.* **291**:671-682.
99. Reeves,G.A., Dallman,T.J., Redfern,O.C., Akpor,A., and Orengo,C.A. 2006. Structural diversity of domain superfamilies in the CATH database. *J. Mol. Biol.* **360**:725-741.
100. Reid A, Yeats C, and Orengo,C.A. 2007. Methods of remote homology detection can be combined to increase coverage by 10% in the midnight zone. *Bioinformatics* **23**:2353-2360.
101. Resnik,P. 1999. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research* **11**:95-130.
102. Richards,F.M. and Kundrot,C.E. 1988. Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. *Proteins.* **3**:71-84.
103. Richardson,J.S. 1981. The anatomy and taxonomy of protein structure. *Adv. Protein Chem.* **34**:167-339.
104. Rison,S.C. and Thornton,J.M. 2002. Pathway evolution, structurally speaking. *Curr. Opin. Struct. Biol.* **12**:374-382.
105. Rosenblatt,F.T. 1957. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* **65**:386-408.
106. Rost,B. 2002. Enzyme function less conserved than anticipated. *J. Mol. Biol.* **318**:595-608.
107. Rumelhart,D., Hinton,G.E., and Williams,R.J. 1986. Learning representations by back-propagating errors. *Nature* **332**:553-536.
108. Russell,R.B. and Barton,G.J. 1994. Structural features can be unconserved in proteins with similar folds. An analysis of side-chain to side-chain contacts secondary structure and accessibility. *J. Mol. Biol.* **244**:332-350.
109. Sadreyev,R. and Grishin,N. 2003. COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. *J. Mol. Biol.* **326**:317-336.



110. Sauder,J.M., Arthur,J.W., and Dunbrack,R.L., Jr. 2000. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins* **40**:6-22.
111. Schneider,R., de Daruvar,A., and Sander,C. 1997. The HSSP database of protein structure-sequence alignments. *Nucleic Acids Res.* **25**:226-230.
112. Shepherd,A.J., Gorse,D., and Thornton,J.M. 1999. Prediction of the location and type of beta-turns in proteins using neural networks. *Protein Sci.* **8**:1045-1055.
113. Shindyalov,I.N. and Bourne,P.E. 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng* **11**:739-747.
114. Sibanda,B.L. and Thornton,J.M. 1985. Beta-hairpin families in globular proteins. *Nature* **316**:170-174.
115. Siddiqui,A.S. and Barton,G.J. 1995. Continuous and discontinuous domains: an algorithm for the automatic generation of reliable protein domain definitions. *Protein Sci.* **4**:872-884.
116. Sillitoe,I., Dibley,M., Bray,J., Addou,S., and Orengo,C. 2005. Assessing strategies for improved superfamily recognition. *Protein Sci.* **14**:1800-1810.
117. Slidel,T. 1996. A computational study of chirality in protein structure. *Thesis*.
118. Smith,T.F. and Waterman,M.S. 1981. Identification of common molecular subsequences. *Journal of molecular biology* **147**:195-197.
119. Soding,J. 2005. Protein homology detection by HMM-HMM comparison. *Bioinformatics.* **21**:951-960.
120. Stormo,G.D., Schneider,T.D., Gold,L., and Ehrenfeucht,A. 1982. Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Res* **10**:2997-3011.
121. Subbiah,S., Laurents,D.V., and Levitt,M. 1993. Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. *Curr. Biol.* **3**:141-148.
122. Swindells,M.B. 1995. A procedure for the automatic determination of hydrophobic cores in protein structures. *Protein Sci.* **4**:93-102.
123. Tarca, Carey VJ, Chen XW, Romero R, and Draghici S 2007. Machine Learning and Its Applications to Biology. *PLoS Comput Biol* **3**:e116.
124. Tatusov,R.L., Fedorova,N.D., Jackson,J.D., Jacobs,A.R., Kiryutin,B., Koonin,E.V., Krylov,D.M., Mazumder,R., Mekhedov,S.L., Nikolskaya,A.N.,

- Rao,B.S., Smirnov,S., Sverdlov,A.V., Vasudevan,S., Wolf,Y.I., Yin,J.J., and Natale,D.A. 2003. The COG database: an updated version includes eukaryotes. *BMC. Bioinformatics*. **4**:41-44.
125. Taylor,W.R. 1999. Protein structural domain identification. *Protein Eng.* **12**:203-216.
  126. Taylor,W.R. and Orengo,C.A. 1989. Protein structure alignment. *J. Mol. Biol.* **208**:1-22.
  127. Teichmann,S.A., Rison,S.C., Thornton,J.M., Riley,M., Gough,J., and Chothia,C. 2001. Small-molecule metabolism: an enzyme mosaic. *Trends Biotechnol.* **19**:482-486.
  128. Thompson,J.D., Plewniak,F., and Poch,O. 1999. BALiBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*. **15**:87-88.
  129. Tian,W. and Skolnick,J. 2003. How well is enzyme function conserved as a function of pairwise sequence identity? *J. Mol. Biol.* **333**:863-882.
  130. Todd,A.E., Marsden,R.L., Thornton,J.M., and Orengo,C.A. 2005. Progress of structural genomics initiatives: an analysis of solved target structures. *J. Mol. Biol.* **20**;348:1235-1260.
  131. Todd,A.E., Orengo,C.A., and Thornton,J.M. 1999. Evolution of protein function, from a structural perspective. *Curr. Opin. Chem. Biol.* **3**:548-556.
  132. Todd,A.E., Orengo,C.A., and Thornton,J.M. 2001. Evolution of function in protein superfamilies, from a structural perspective. *J. Mol. Biol.* **307**:1113-1143.
  133. Vladimir N.Vapnik 1995. *The Nature of Statistical Learning Theory*. Springer.
  134. Vogel,C., Berzuini,C., Bashton,M., Gough,J., and Teichmann,S.A. 2004. Supra-domains: evolutionary units larger than single protein domains. *J. Mol. Biol.* **336**:809-823.
  135. Vogel,C., Teichmann,S.A., and Pereira-Leal,J. 2005. The relationship between domain duplication and recombination. *J. Mol. Biol.* **346**:355-365.
  136. Ward,J.J., McGuffin,L.J., Buxton,B.F., and Jones,D.T. 2003. Secondary structure prediction with support vector machines. *Bioinformatics* **19**:1650-1655.
  137. Webb,E.C. 1965. The Nomenclature of multiple enzyme forms. *Enzymol Biol Clin (Basel)* **20**:592.
  138. Weiss,S.M. 1991. *Computer Systems That Learn*.

139. Wilbur, W.J. and Yang, Y. 1996. An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology texts. *Comput. Biol. Med.* **26**:209-222.
140. Wood, T.C. and Pearson, W.R. 1999. Evolution of protein sequences and structures. *J. Mol. Biol.* **291**:977-995.
141. Wu, C.H., Apweiler, R., Bairoch, A., Natale, D.A., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Mazumder, R., O'Donovan, C., Redaschi, N., and Suzek, B. 2006. The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res.* **34**:D187-D191.
142. Yona, G. and Levitt, M. 2002. Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J. Mol. Biol.* **315**:1257-1275.

# Appendix

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
1.10.10.10	47.41	62.5	9.4	50	928	33	77
1.10.10.160	91.28	0.0	7.1	2	9	1	1
1.10.10.60	48.84	25.0	16.8	23	917	23	23
1.10.1060.10	71.83	12.5	14.9	2	85	2	10
1.10.12.10	78.72	0.0	28.5	3	32	2	1
1.10.1200.10	68.15	25.0	10.5	5	162	4	4
1.10.15.10	54.83	40.0	19.6	5	21	4	6
1.10.150.120	83.66	33.3	2.8	2	6	1	2
1.10.150.130	81.87	20.0	13.4	2	52	1	2
1.10.150.20	60.87	20.0	21.4	4	26	4	1
1.10.150.70	44.69	66.7	19.1	11	19	9	4
1.10.20.10	54.69	71.4	10.5	13	277	8	2
1.10.230.10	86.58	0.0	5.4	3	35	1	1
1.10.238.10	43.47	55.6	16.6	28	464	39	3
1.10.275.10	55.35	71.4	12.9	5	27	4	5
1.10.287.110	79.22	25.0	13.8	3	86	3	3
1.10.287.280	61.13	40.0	2.7	2	68	3	1
1.10.287.40	82.68	0.0	5.0	2	14	2	2
1.10.287.60	75.28	0.0	20.8	3	12	2	1
1.10.287.70	35.36	33.3	7.8	4	192	3	2
1.10.287.80	84.25	0.0	4.4	2	24	1	1
1.10.290.10	86.68	0.0	1.4	2	3	1	2
1.10.300.10	89.19	20.0	3.8	2	9	1	1
1.10.340.10	82.64	40.0	10.5	4	49	2	5
1.10.390.10	72.57	50.0	7.7	2	20	2	1
1.10.40.30	67.43	42.9	6.4	4	30	3	4
1.10.405.10	77.50	28.6	22.5	2	3	2	1
1.10.420.10	79.07	40.0	4.7	3	29	2	1
1.10.439.10	85.83	18.2	5.4	2	7	1	1
1.10.443.10	67.90	37.5	15.5	5	315	5	4
1.10.455.10	87.61	0.0	9.3	2	15	1	1

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
1.10.460.10	86.55	9.1	7.1	2	3	1	2
1.10.472.10	68.73	50.0	11.4	14	183	5	1
1.10.486.10	75.20	30.0	11.1	3	17	2	2
1.10.490.10	68.70	44.4	12.6	16	89	13	4
1.10.510.10	75.75	42.9	12.1	9	211	6	2
1.10.520.10	80.27	28.6	6.3	3	18	3	1
1.10.530.10	51.31	53.8	18.0	7	108	6	4
1.10.540.10	89.40	14.3	7.4	3	90	1	1
1.10.560.10	87.59	5.9	8.3	2	19	1	1
1.10.569.10	91.29	16.7	6.3	2	4	1	1
1.10.570.10	91.19	0.0	5.9	2	3	1	2
1.10.580.10	83.48	18.8	8.4	2	7	1	1
1.10.599.10	86.75	0.0	6.1	2	7	1	1
1.10.615.10	56.54	27.8	25.2	5	40	5	1
1.10.620.20	47.50	53.8	14.5	6	36	5	1
1.10.630.10	75.11	18.5	11.6	9	257	6	1
1.10.645.10	87.68	0.0	3.2	2	26	1	4
1.10.720.10	88.10	0.0	7.7	2	6	1	3
1.10.730.10	49.77	41.7	9.2	5	63	4	5
1.10.760.10	47.38	55.6	13.9	23	233	16	10
1.10.8.10	57.16	50.0	12.1	6	15	4	3
1.10.8.60	67.26	50.0	20.3	7	32	4	13
1.10.8.80	71.37	42.9	16.3	2	9	2	2
1.10.800.10	79.11	9.5	4.3	2	10	1	1
1.10.940.10	62.30	0.0	14.1	2	18	2	2
1.20.1010.10	89.91	16.7	7.6	2	4	1	2
1.20.1050.10	73.72	55.6	9.0	10	118	6	2
1.20.1060.10	69.16	28.6	4.4	2	5	3	1
1.20.1070.10	65.06	35.7	10.2	5	520	2	1
1.20.120.10	78.69	33.3	6.8	5	18	2	1
1.20.120.140	85.14	0.0	11.8	3	21	1	2
1.20.120.160	80.01	25.0	5.5	3	54	2	4
1.20.120.190	77.89	16.7	2.4	7	53	2	10
1.20.120.260	86.13	12.5	11.0	2	3	1	1

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
1.20.120.280	78.73	25.0	5.9	2	6	1	1
1.20.120.80	77.37	44.4	4.9	2	33	2	1
1.20.20.10	65.73	33.3	24.3	2	32	2	1
1.20.200.10	67.01	28.6	28.7	4	10	3	5
1.20.210.10	72.65	16.7	4.6	2	33	2	3
1.20.272.10	83.01	0.0	8.9	2	4	2	2
1.20.272.20	82.29	28.6	2.2	2	4	1	1
1.20.58.100	86.23	25.0	3.6	3	5	1	2
1.20.80.10	92.35	0.0	3.1	2	11	1	1
1.20.840.10	70.01	38.9	5.9	2	17	2	2
1.20.910.10	84.09	20.0	6.7	2	13	1	2
1.20.970.10	87.02	33.3	1.5	2	19	1	2
1.20.990.10	84.06	12.5	7.5	2	2	1	1
1.25.40.10	62.06	78.9	12.3	10	973	6	38
1.25.40.120	81.17	0.0	9.4	2	86	2	1
1.25.40.20	66.17	60.0	10.6	4	516	5	1
1.25.40.70	69.09	37.5	10.0	2	18	2	1
1.25.40.80	88.34	0.0	10.7	2	27	1	2
1.50.10.10	75.18	36.8	5.3	2	11	2	1
1.50.10.20	57.94	33.3	8.8	7	84	5	3
1.50.10.30	63.55	15.0	12.2	5	34	5	2
2.10.10.20	75.36	25.0	7.9	3	22	2	1
2.10.109.10	78.84	33.3	13.9	4	111	2	5
2.102.10.10	73.33	40.0	12.0	6	118	4	3
2.120.10.10	71.09	24.1	11.3	7	75	5	4
2.130.10.10	73.73	0.0	11.5	3	376	2	16
2.130.10.30	79.91	27.6	4.5	2	53	1	1
2.140.10.10	83.71	10.6	7.9	2	118	1	3
2.160.10.10	74.42	53.8	12.9	4	138	4	10
2.160.20.10	70.38	24.2	38.3	8	88	7	3
2.170.120.12	81.53	25.0	6.1	2	4	2	1
2.170.130.10	80.97	20.0	9.8	2	210	2	5
2.170.16.10	75.67	29.4	23.3	4	23	3	3
2.20.25.10	60.19	100.0	10.3	6	32	4	2
2.20.25.50	65.99	33.3	20.3	5	17	4	1
2.20.25.60	73.41	28.6	10.4	8	24	2	1
2.20.25.70	76.11	27.3	14.9	6	101	4	11
2.20.28.10	78.48	33.3	28.0	2	13	2	4

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
2.30.100.10	82.39	20.0	8.9	4	20	3	1
2.30.110.20	83.03	7.7	11.6	2	22	1	1
2.30.29.30	60.14	36.4	14.0	23	328	15	1
2.30.30.100	77.36	28.6	17.1	7	52	2	2
2.30.30.40	61.31	28.6	9.7	4	259	7	1
2.30.30.70	71.06	20.0	10.5	4	13	3	1
2.30.30.90	78.74	12.5	13.8	3	4	2	2
2.30.33.40	81.26	14.3	6.6	2	7	2	1
2.30.35.100	79.82	42.9	10.4	4	6	2	1
2.30.35.20	90.26	16.7	2.7	2	7	1	1
2.30.35.30	63.25	55.6	9.6	12	137	5	12
2.30.38.10	90.54	16.7	6.2	2	150	1	5
2.30.39.10	75.80	36.4	14.3	8	96	3	1
2.30.42.10	72.23	33.3	7.8	9	275	5	6
2.40.10.10	0.87	55.6	27.9	23	345	29	5
2.40.100.10	81.11	16.7	5.0	2	56	2	2
2.40.128.20	60.24	28.6	19.7	15	74	12	1
2.40.128.70	81.02	0.0	8.8	3	75	2	4
2.40.160.10	74.94	22.7	8.2	4	41	4	1
2.40.170.10	82.51	13.0	6.2	2	12	2	1
2.40.170.20	74.93	0.0	9.6	2	51	2	5
2.40.240.10	73.33	28.6	24.5	4	16	3	2
2.40.30.10	45.42	45.5	15.7	17	181	9	15
2.40.30.20	71.36	25.0	11.6	3	33	2	1
2.40.37.10	72.07	7.7	22.0	2	40	2	2
2.40.40.20	68.39	30.0	8.1	8	36	3	11
2.40.50.100	55.68	72.7	7.3	8	274	6	10
2.40.50.140	40.72	50.0	25.0	38	332	33	36
2.40.50.150	82.30	25.0	3.5	2	12	1	1
2.40.50.180	79.78	28.6	7.0	2	11	2	6
2.40.70.10	37.94	75.0	24.7	16	184	9	2
2.60.120.10	77.77	14.3	16.0	4	196	3	18
2.60.120.200	44.87	50.0	25.3	19	211	16	3
2.60.120.260	53.39	60.0	27.3	20	126	15	9
2.60.120.320	83.66	25.0	14.1	2	5	1	1
2.60.130.10	72.42	29.4	6.4	3	18	3	1
2.60.15.10	89.79	11.1	18.4	2	14	1	1
2.60.200.20	68.61	22.2	35.9	4	90	3	4



<b>SuperFamily</b>	<b>Min SSAP Score</b>	<b>2DSEC Percentage Variation Score</b>	<b>EquivSEC Score</b>	<b>CATH S35Reps</b>	<b>CATH ISL</b>	<b>SSG Number</b>	<b>COG Number</b>
2.60.40.10	37.21	100.0	14.4	85	1020	56	5
2.60.40.1070	85.84	25.0	3.9	2	14	1	1
2.60.40.1080	85.84	12.5	6.6	2	6	1	3
2.60.40.1090	78.81	11.1	5.9	2	77	2	1
2.60.40.1180	67.37	44.4	11.3	14	66	8	3
2.60.40.200	80.51	18.2	9.6	3	20	2	1
2.60.40.290	73.03	30.0	9.8	3	28	3	2
2.60.40.30	64.79	44.4	12.2	39	371	17	5
2.60.40.320	70.12	42.9	8.6	4	31	3	2
2.60.40.420	43.48	62.5	8.6	22	235	15	6
2.60.40.680	90.56	0.0	5.2	2	6	1	1
2.60.40.710	82.89	20.0	6.4	2	4	2	1
2.60.40.730	82.66	30.0	5.5	2	3	2	1
2.60.40.790	77.06	22.2	5.0	4	108	2	2
2.70.110.10	85.90	3.8	7.0	3	5	1	1
2.70.20.10	84.92	22.2	7.5	2	11	1	2
2.70.40.10	86.07	0.0	10.1	2	40	1	2
2.80.10.50	69.86	35.7	14.7	13	157	8	3
3.10.105.10	88.19	0.0	6.2	2	13	1	4
3.10.120.10	80.73	0.0	5.0	2	36	1	2
3.10.129.10	73.00	10.0	11.4	3	111	3	9
3.10.129.20	79.33	11.1	6.1	2	5	1	1
3.10.150.10	83.79	20.0	14.7	3	20	2	1
3.10.170.10	87.13	0.0	9.3	2	15	1	1
3.10.180.10	62.22	46.7	15.3	13	218	8	9
3.10.20.200	81.01	57.1	15.1	3	24	1	3
3.10.20.30	66.49	28.6	10.2	11	125	7	14
3.10.200.10	86.48	6.7	7.5	3	23	1	1
3.10.28.10	64.98	72.7	11.9	7	87	5	3
3.10.290.10	75.50	12.5	8.7	3	83	3	9
3.10.310.10	79.69	11.1	6.3	2	4	2	2
3.10.330.10	80.77	25.0	21.2	4	3	2	1
3.10.400.10	85.94	7.7	8.3	2	10	1	1
3.10.50.40	64.30	37.5	27.4	4	123	5	4
3.20.10.10	83.54	7.7	9.6	4	15	1	1
3.20.20.10	78.51	5.9	11.5	3	30	3	6
3.20.20.100	77.12	17.4	7.1	3	44	2	4

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
3.20.20.120	70.21	36.8	14.0	6	44	5	3
3.20.20.140	53.73	34.6	18.0	6	71	5	12
3.20.20.150	68.92	33.3	11.6	4	103	4	10
3.20.20.170	64.05	13.6	14.2	4	11	3	3
3.20.20.20	80.86	5.6	7.0	2	16	2	4
3.20.20.210	88.77	4.2	5.7	2	12	1	1
3.20.20.240	82.90	11.5	11.9	2	2	1	1
3.20.20.270	76.87	37.0	11.2	3	14	2	3
3.20.20.280	77.72	20.0	6.4	2	8	2	4
3.20.20.30	69.08	37.5	9.9	5	75	4	1
3.20.20.40	81.70	5.9	9.9	2	20	1	1
3.20.20.60	68.98	28.0	11.7	4	27	4	8
3.20.20.70	64.39	13.6	16.7	6	61	3	3
3.20.20.80	50.29	32.0	13.1	39	436	31	21
3.20.20.90	55.89	45.8	13.3	27	359	20	38
3.20.80.10	77.60	8.3	8.2	2	14	2	4
3.30.160.20	83.47	0.0	4.9	4	50	1	1
3.30.160.70	79.90	0.0	5.1	2	2	2	1
3.30.190.20	88.23	12.5	2.9	2	13	1	1
3.30.200.20	66.64	37.5	11.5	11	1401	10	6
3.30.230.10	71.46	37.5	12.4	5	44	5	10
3.30.230.20	70.77	27.3	5.2	2	16	2	5
3.30.260.10	84.81	16.7	5.4	2	33	1	1
3.30.30.20	88.34	25.0	3.3	2	9	1	2
3.30.300.10	61.92	14.3	9.8	4	12	4	2
3.30.300.20	77.18	28.6	6.3	5	45	3	7
3.30.300.30	87.23	37.5	4.6	2	60	1	5
3.30.310.10	78.03	12.5	5.9	2	28	2	1
3.30.360.10	57.82	68.4	7.7	5	40	4	3
3.30.365.10	64.63	22.2	12.7	7	29	3	2
3.30.365.20	66.20	40.0	9.0	4	25	2	2
3.30.379.10	82.65	12.5	10.5	2	44	2	1
3.30.390.10	72.92	37.5	8.3	6	40	3	2
3.30.390.30	77.07	20.0	9.3	4	12	3	3
3.30.390.50	85.01	0.0	5.6	2	12	1	2
3.30.413.10	78.26	11.1	10.3	2	39	2	4
3.30.420.10	36.93	64.3	39.6	17	714	13	19
3.30.420.110	86.20	11.1	6.4	2	7	1	1
3.30.420.40	51.12	53.8	13.9	13	151	8	14
3.30.428.10	70.66	58.3	8.5	5	82	3	2

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
3.30.43.10	66.11	42.9	12.1	7	104	5	4
3.30.450.20	79.20	20.0	9.1	5	326	4	16
3.30.460.10	71.06	18.2	10.8	4	95	4	5
3.30.465.10	73.94	41.7	9.4	4	17	2	3
3.30.465.20	74.83	33.3	11.5	3	53	3	1
3.30.470.10	76.46	27.3	9.3	4	32	2	1
3.30.470.20	47.65	65.0	10.3	16	136	11	17
3.30.470.30	72.09	22.2	10.6	3	2	3	1
3.30.497.10	77.61	8.3	8.4	5	20	2	1
3.30.499.10	71.08	14.3	33.4	2	7	2	3
3.30.540.10	73.83	28.6	7.0	6	39	3	3
3.30.550.10	80.68	9.1	10.5	3	12	3	2
3.30.559.10	76.61	12.5	12.8	3	16	2	1
3.30.565.10	54.15	50.0	12.5	8	567	6	19
3.30.572.10	76.26	13.3	5.7	2	11	2	1
3.30.69.10	83.48	25.0	5.6	2	8	1	1
3.30.70.100	68.34	28.6	9.6	8	92	4	5
3.30.70.130	72.50	0.0	6.9	2	31	2	2
3.30.70.150	87.28	0.0	4.8	2	31	1	1
3.30.70.160	54.39	50.0	8.7	25	156	15	23
3.30.70.20	65.62	33.3	11.3	5	246	8	26
3.30.70.210	72.61	28.6	9.7	4	73	3	7
3.30.70.220	74.08	25.0	10.6	3	18	3	2
3.30.70.270	29.18	69.2	10.0	7	375	11	3
3.30.70.330	49.85	87.5	23.4	17	586	14	2
3.30.70.350	89.49	0.0	3.4	2	4	1	1
3.30.70.370	70.71	45.5	2.8	3	15	2	3
3.30.70.420	85.89	11.1	12.8	2	28	1	1
3.30.70.430	90.66	0.0	5.4	2	7	1	1
3.30.70.470	78.37	27.3	9.5	2	3	2	2
3.30.70.480	76.60	11.1	13.7	4	25	2	5
3.30.70.520	61.12	16.7	24.6	2	2	2	1
3.30.70.530	78.12	0.0	7.4	3	25	3	2
3.30.70.540	77.24	0.0	34.1	2	7	2	2
3.30.70.60	73.03	0.0	7.1	4	33	3	2
3.30.70.730	82.85	5.6	6.6	2	50	2	3
3.30.70.80	81.81	0.0	18.2	2	41	2	1
3.30.70.810	79.86	22.2	7.4	2	8	1	1
3.30.700.10	66.13	50.0	8.1	3	61	3	3
3.30.750.24	82.29	25.0	6.7	2	90	2	5
3.30.780.10	74.99	0.0	10.5	2	18	2	1

<b>SuperFamily</b>	<b>Min SSAP Score</b>	<b>2DSEC Percentage Variation Score</b>	<b>EquivSEC Score</b>	<b>CATH S35Reps</b>	<b>CATH ISL</b>	<b>SSG Number</b>	<b>COG Number</b>
3.30.830.10	70.54	50.0	9.9	11	103	4	3
3.30.870.10	73.98	25.0	8.3	3	105	2	7
3.30.9.10	43.42	30.8	29.1	5	10	4	4
3.30.920.10	88.67	0.0	6.5	2	9	1	1
3.30.930.10	51.57	56.5	24.9	13	140	12	15
3.40.1050.10	54.76	26.7	21.7	3	32	4	1
3.40.109.10	79.20	13.3	11.9	4	52	2	3
3.40.120.10	90.43	10.0	1.9	2	32	1	2
3.40.140.10	80.21	22.2	5.4	2	15	1	2
3.40.190.10	43.05	50.0	32.4	38	737	23	28
3.40.190.80	73.97	20.0	10.3	6	62	4	3
3.40.192.10	83.16	11.1	8.5	3	11	2	1
3.40.225.10	78.62	40.0	9.0	4	22	2	2
3.40.228.10	81.55	26.3	6.6	3	4	2	4
3.40.250.10	77.86	11.1	11.6	6	153	3	4
3.40.30.10	41.58	46.2	12.2	35	762	29	33
3.40.309.10	81.95	21.4	5.3	3	82	2	2
3.40.33.10	78.47	33.3	4.4	2	51	1	1
3.40.350.10	78.87	14.3	8.3	2	26	2	1
3.40.390.10	48.83	46.7	16.5	12	231	10	7
3.40.430.10	82.30	20.0	9.2	7	39	1	2
3.40.47.10	55.08	50.0	8.8	7	171	5	6
3.40.470.10	69.95	16.7	10.2	2	33	2	3
3.40.50.10	57.59	33.3	20.6	8	48	6	1
3.40.50.1000	76.66	26.7	14.4	5	594	3	24
3.40.50.10090	69.62	30.8	29.5	2	44	2	1
3.40.50.1010	64.52	50.0	16.3	4	22	4	1
3.40.50.1100	63.85	56.3	8.7	7	89	6	5
3.40.50.1110	72.40	13.3	22.4	3	115	3	3
3.40.50.1120	78.59	26.3	11.6	2	22	2	2
3.40.50.1220	70.10	40.0	10.1	8	58	5	10
3.40.50.1240	61.33	43.5	15.6	7	132	6	3

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
3.40.50.1370	66.54	35.7	11.5	3	22	3	2
3.40.50.1380	75.27	35.7	7.9	3	8	2	3
3.40.50.140	83.55	23.1	3.5	2	4	1	3
3.40.50.1400	67.30	18.2	10.6	6	40	4	3
3.40.50.1420	83.02	10.0	15.9	2	104	2	4
3.40.50.1460	79.39	23.1	3.7	3	16	3	1
3.40.50.150	51.57	44.4	11.3	23	1520	20	74
3.40.50.1580	76.51	23.5	11.3	5	46	3	4
3.40.50.170	85.97	20.0	4.8	2	22	1	3
3.40.50.1770	87.56	0.0	5.7	2	8	1	2
3.40.50.1820	39.50	65.6	22.6	43	1580	30	42
3.40.50.1860	71.01	18.2	4.7	2	17	2	3
3.40.50.1890	84.83	0.0	6.8	2	16	1	1
3.40.50.1940	77.42	35.7	10.0	3	86	3	8
3.40.50.1950	87.88	15.4	6.7	2	13	1	3
3.40.50.1980	90.74	0.0	6.1	2	32	1	2
3.40.50.20	61.52	50.0	12.4	10	42	7	9
3.40.50.200	76.66	21.7	9.5	3	118	2	2
3.40.50.2000	69.54	26.7	10.4	6	206	6	9
3.40.50.2010	89.51	0.0	6.1	2	15	1	2
3.40.50.2030	65.88	29.4	15.2	4	6	3	2
3.40.50.2050	83.20	25.0	3.2	2	19	1	1
3.40.50.2300	65.32	42.9	10.3	26	407	11	1
3.40.50.261	81.03	8.3	8.5	2	14	2	3
3.40.50.270	74.34	11.1	9.7	3	14	2	5

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
3.40.50.280	71.67	16.7	8.2	4	18	4	3
3.40.50.360	81.03	10.0	8.3	4	62	2	8
3.40.50.410	83.56	14.3	6.4	4	142	1	10
3.40.50.50	84.23	25.0	4.0	3	19	1	2
3.40.50.610	67.45	50.0	12.3	6	152	6	15
3.40.50.620	61.47	46.7	13.8	11	104	7	13
3.40.50.720	49.48	50.0	20.1	77	2127	50	89
3.40.50.740	76.88	33.3	7.5	3	19	3	5
3.40.50.790	85.68	22.2	9.2	2	9	1	1
3.40.50.80	79.02	38.5	7.6	9	117	4	5
3.40.50.800	83.94	11.1	6.6	7	72	1	5
3.40.50.850	78.76	28.6	6.7	3	42	2	2
3.40.50.880	53.44	44.4	15.7	10	158	7	14
3.40.50.920	74.80	20.0	12.8	3	49	3	10
3.40.50.960	86.59	20.0	5.3	2	4	1	2
3.40.50.9600	78.68	13.3	7.7	5	125	2	2
3.40.50.970	60.67	50.0	7.9	14	147	8	18
3.40.50.980	73.48	30.8	7.1	4	201	3	6
3.40.510.10	35.98	44.0	16.5	8	67	8	10
3.40.605.10	53.34	23.8	7.4	4	26	2	2
3.40.630.10	64.02	25.0	10.4	8	167	4	12
3.40.630.30	48.25	46.2	11.8	12	696	10	21
3.40.640.10	67.65	40.9	14.0	25	479	14	28
3.40.710.10	54.52	45.8	9.9	12	252	9	9
3.40.718.10	76.11	6.9	10.2	2	13	3	3
3.40.720.10	84.69	0.0	9.4	2	14	1	2
3.40.800.10	85.89	5.6	4.3	2	26	1	1
3.40.930.10	85.45	20.0	8.3	2	38	1	3
3.40.950.10	85.32	27.3	2.3	2	5	1	1
3.40.980.10	78.10	7.7	9.6	2	21	2	3
3.50.12.10	80.65	9.1	22.8	4	178	2	4
3.50.30.10	83.79	40.0	21.1	2	18	2	3
3.50.50.60	42.08	23.1	11.9	28	1273	15	34
3.50.6.10	80.41	23.5	17.7	3	104	2	3
3.50.60.10	78.45	17.4	8.1	3	38	3	1
3.50.7.10	78.69	8.3	17.5	2	5	2	1
3.60.10.10	79.20	18.2	10.4	3	179	2	4
3.60.15.10	68.90	15.0	8.8	5	303	4	13
3.60.20.10	51.49	41.7	19.5	17	81	8	8
3.60.21.10	58.03	34.6	16.9	5	310	5	14

SuperFamily	Min SSAP Score	2DSEC Percentage Variation Score	EquivSEC Score	CATH S35Reps	CATH ISL	SSG Number	COG Number
3.65.10.10	79.11	16.7	8.9	4	29	3	2
3.70.10.10	86.95	4.5	3.6	2	35	1	1
3.75.10.10	79.28	29.6	6.5	2	25	1	3
3.80.10.10	52.62	69.6	7.3	12	860	10	5
3.90.10.10	69.22	70.0	6.1	7	57	5	2
3.90.110.10	80.28	27.3	11.6	3	18	3	1
3.90.120.10	68.88	28.6	65.0	3	39	3	1
3.90.149.10	79.99	21.4	4.7	2	18	2	1
3.90.180.10	80.91	25.0	16.6	4	60	2	6
3.90.190.10	59.82	52.4	7.4	7	215	6	4
3.90.190.20	76.82	18.2	9.2	4	22	4	5
3.90.199.10	84.54	5.6	9.3	2	6	1	1
3.90.226.10	68.60	45.0	17.3	9	134	5	7
3.90.230.10	82.48	23.5	11.7	4	63	2	2
3.90.245.10	86.14	4.2	8.6	2	18	1	1
3.90.25.10	66.19	50.0	6.1	5	68	4	3
3.90.280.10	77.69	33.3	13.0	3	29	2	1
3.90.320.20	91.94	0.0	1.1	4	1	1	1
3.90.45.10	84.00	10.0	6.6	2	19	1	1
3.90.470.10	83.62	25.0	2.5	2	12	1	1
3.90.55.10	69.80	11.1	8.8	2	48	2	3
3.90.550.10	55.44	43.5	13.7	14	637	14	26
3.90.640.10	77.51	16.7	11.2	5	42	2	3
3.90.660.10	70.30	7.1	17.9	2	12	2	1
3.90.70.10	55.51	45.0	16.5	6	81	5	2
3.90.700.10	73.91	33.3	9.6	3	5	2	2
3.90.710.10	54.41	12.5	7.2	2	16	3	1
3.90.730.10	83.01	23.1	7.6	4	22	2	1
3.90.740.10	73.26	50.0	8.3	3	6	2	3
3.90.76.10	81.51	28.6	14.7	2	50	1	3
3.90.770.10	82.51	11.8	6.6	2	8	1	1
3.90.78.10	86.90	55.6	4.5	2	3	1	1
3.90.79.10	69.36	30.8	10.3	5	247	4	7
3.90.80.10	79.56	41.2	7.4	2	8	2	1
3.90.800.10	80.82	25.0	6.5	2	10	2	1
3.90.850.10	78.25	31.6	8.8	2	32	2	4
3.90.870.10	67.00	6.7	12.8	4	20	4	3
3.90.930.12	82.22	25.0	5.6	4	17	1	1



<b>SuperFamily</b>	<b>Min SSAP Score</b>	<b>2DSEC Percentage Variation Score</b>	<b>EquivSEC Score</b>	<b>CATH S35Reps</b>	<b>CATH ISL</b>	<b>SSG Number</b>	<b>COG Number</b>
4.10.320.10	77.75	0.0	1.4	2	13	2	1
4.10.640.10	81.47	33.3	8.1	2	15	1	1
4.10.860.10	86.63	33.3	8.5	2	11	1	4