Development of a Sequence Based Method for the Prediction of Protein Function

Mark Nicholas Wass

September 2008

A dissertation submitted for the degree of Doctor of Philosophy of the University of London and for the diploma of Imperial College of Science, Technology and Medicine

Structural Bioinformatics Group
Division of Molecular Biosciences
Faculty of Natural Sciences
Imperial College London
South Kensington
London, SW7 2AZ

Abstract

The success of genome sequencing has resulted in millions of proteins that are functionally uncharacterised. Where protein function is known, the Gene Ontology (GO) is now widely used to provide annotations in a controlled and hierarchical manner. This thesis describes the development and evaluation of a novel approach (ConFunc) to predict protein function from sequence using the vocabulary of GO.

ConFunc infers function by grouping homologues according to their GO functional annotations and identifying conserved residues associated with them, for comparison to proteins of unknown function. Position Specific Scoring Matrices are calculated for the conserved residues to capture the amino acid composition and enable comparison with the query sequence. The use of different settings and approaches was assessed, including: sequence alignment methods, conservation thresholds and expectation value thresholds to discriminate between functions. The agreement of the conserved residues identified by ConFunc with known functional residues was also considered.

ConFunc was thoroughly tested against predictions made using annotation transfer from BLAST, PSI-BLAST and Pfam hits. For a large test set of proteins, ConFunc outperforms all of these methods at high levels of precision (percentage of predictions that are correct), obtaining up to six time greater recall (percentage of annotations that are correctly inferred) at equivalent levels of precision. Analysis in the *twilight zone* of sequence similarity demonstrates the ability of ConFunc to correctly infer function at low levels of sequence similarity; an area where BLAST performance is poor, with ConFunc maximum precision 16% greater than BLAST, at which level ConFunc recall is 3.5 times greater. Practical application of ConFunc is shown by current initiatives to provide annotations for the proteomes of *Campylobacter jejuni* and *Plasmodium falciparum,* in association with experimental groups. The ConFunc web server is also described which makes the method widely available to the scientific community.

Contents

ABSTRACT 2

CONTENTS 3

LIST OF FIGURES 7

LIST OF TABLES 9

LIST OF ABBREVIATIONS 10

1. CHAPTER 1 12

INTRODUCTION 12

1.1. PROTEIN FUNCTION AND STRUCTURE CLASSIFICATIONS 13

1.1.1. What is protein function? *13*

1.1.2. The Enzyme Commission (E.C.) classification *14*

1.1.3. Gene Ontology *15*

1.1.4. Structural Classifications of Proteins - SCOP & CATH *19*

1.1.5. Protein Sequence Families *22*

1.1.6. Protein Motif Methods *23*

1.2. SEQUENCE SIMILARITY SEARCHING AND ALIGNMENT 24

1.2.1. Comparing Sequences – Substitution Matrices *24*

1.2.2. Dynamic Programming – Optimal Alignments *28*

1.2.3. The BLAST Algorithm *28*

1.2.4. Sequence Alignment Statistics *29*

1.2.5. Position Specific Scoring Matrices & PSI-BLAST *31*

1.2.6. Multiple Sequence Alignment *34*

1.3. IDENTIFICATION OF CONSERVED AND FUNCTIONAL RESIDUES 39

1.3.1. Conservation Scores *39*

1.3.2. Identifying Functional Residues *41*

1.4. RELATIONSHIPS BETWEEN SEQUENCE, STRUCTURE AND FUNCTION 45

1.4.1. The Relationship between Protein Sequence and Structure *45*

1.4.2. The Relationship between Protein Sequence and Function *46*

1.4.3. The Relationship between Protein Structure and Function *53*

1.5. PROTEIN FUNCTION PREDICTION APPROACHES 56

1.5.1. Sequence Based Methods *57*

1.5.2. Structure Based Methods *64*

1.5.3. Other Methods *67*

1.5.4. Combined Methods 69
1.6. ASSESSING FUNCTION PREDICTIONS 71
1.6.1. Semantic Similarity Measures 73
1.6.2. Precision & Recall 75
1.6.3. Compatible Terms 77
1.7. COMPARISON OF FUNCTION PREDICTION METHODS 77
1.8. INTERNATIONAL ASSESSMENT OF FUNCTION PREDICTION 81
1.9. SCOPE AND OUTLINE OF THIS THESIS 84

2. CHAPTER 2 86

CONVERGENT EVOLUTION OF ENZYME ACTIVE SITES 86

2.1. OVERVIEW 86
2.2. INTRODUCTION 86
2.2.1. Mechanistic and Transformational Analogues 87
2.2.2. Previous studies of Transformational analogues 88
2.3. METHOD 89
2.3.1. Identification of Mechanistic Analogues 89
2.3.2. Identifying Transformational Analogues 89
2.4. RESULTS 91
2.4.1. Mechanistic Analogues 91
2.4.2. Transformational Analogues 92
2.5. CONCLUSION 93
2.5.1. Limitations 93
2.5.2. Concluding Remarks 94

3. CHAPTER 3 95

CONFUNC METHODS 95

3.1. OVERVIEW 95
3.2. AIMS AND OBJECTIVES 95
3.3. AN OVERVIEW OF THE CONFUNC APPROACH 97
3.3.1. Identifying Homologues 99
3.3.2. Aligning Homologues 100
3.3.3. Generating GO Term Subalignments 101
3.3.4. Identifying Conserved Residues 102
3.3.5. Calculating Conservation Z Scores 103
3.3.6. Generating Position Specific Scoring Matrices (PSSMs) 103
3.3.7. Scoring Query Sequences against PSSMs 103
3.3.8. Calculating GO Term E-values 104

3.3.9. Inferring GO Function from E-values *107*
3.4. THE COMPARISON SCHEME USED HERE 112
3.4.1. Generating a Test Set *115*
3.4.2. Comparison with Other Methods *116*
4. CHAPTER 4 **118**
DEVELOPMENT OF CONFUNC **118**
4.1. INTRODUCTION 118
4.2. INFERRING GO FUNCTIONS FROM GO TERM E-VALUES 119
4.2.1. A Simple E-value Threshold *119*
4.2.2. Ratio Thresholds *122*
4.3. ASSESSING THE RESIDUE CONSERVATION Z SCORE 125
4.4. ASSESSING THE MULTIPLE SEQUENCE ALIGNMENT METHOD 127
4.5. SUMMARY 128
5. CHAPTER 5 **130**
BENCHMARKING CONFUNC **130**
5.1. OVERVIEW 130
5.2. INTRODUCTION 130
5.3. COMPARISON WITH BLAST & PSI-BLAST 132
5.3.1. Assessment with Non-Electronic Annotations *132*
5.3.2. Assessment with Electronic Annotations *137*
5.3.3. Differences between Non-Electronic and Electronic GO annotations *138*
5.4. FUNCTIONAL ANNOTATION IN THE TWILIGHT ZONE 141
5.5. COMPARISON WITH PFAM 142
5.6. PREDICTING DIFFERENT FUNCTIONAL TYPES 145
5.7. PREDICTIONS AT DIFFERENT LEVELS OF GENE ONTOLOGY 149
5.8. SUMMARY 157
6. CHAPTER 6 **159**
THE CONFUNC SERVER **159**
6.1. CONFUNC SUBMISSION 159
6.2. CONFUNC RESULTS 159
6.3. FUTURE DEVELOPMENTS 160
6.3.1. Submission Options *162*
6.3.2. Updating the Display of Results *162*
7. CHAPTER 7 **164**

DISCUSSION **164**

7.1. IS CONFUNC A SUCCESS? 164
7.2. CONFUNC AND PHUNCTIONER 166
7.3. ASSESSMENT OF CONFUNC CONSERVED RESIDUES 168
7.4. PRACTICAL APPLICATION OF CONFUNC 176
7.4.1. Functional Annotation of Campylobacter jejuni *176*

8. CHAPTER 8 **178**

CONCLUSIONS **178**

8.1. FUTURE WORK 178
8.2. OUTLOOK 180

ACKNOWLEDGEMENTS **182**

REFERENCES **183**

List of Figures

1.1	The Gene Ontology.	17
1.2	The SCOP hierarchy.	21
1.3	A ProSite signature.	23
1.4	BLOSUM62 amino acid substitution matrix.	27
1.5	Evolutionary Trace.	43
1.6	The relationship between protein sequence and structure.	45
1.7	The relationship between protein function and sequence similarity e-values.	47
1.8	The relationship between protein function and sequence identity.	49
1.9	Overprediction and underprediction of protein function.	72
1.10	Calculating semantic similarity	74
1.11	Hierarchical precision and recall.	76
2.1	Identifying Mechanistic Analogues	90
2.2	Overall results of convergent evolution study	91
3.1	An outline of the steps in the ConFunc algorithm.	98
3.2	The ConFunc algorithm.	99
3.3	Generating GO subalignments.	101
3.4	Fitting the random scores to an extreme value distribution	106
3.5	The e-value and ratio thresholds.	108
3.6	The frequency threshold.	111
3.7	Example of precision and recall calculations.	114
4.1	ConFunc performance using a simple e-value threshold.	120
4.2	Distribution of GO term e-values.	121
4.3	The distribution of lowest GO term e-values.	122
4.4	ConFunc performance using ratio and frequency thresholds.	124
4.5	Assessing the residue Z score threshold.	126
4.6	ConFunc performance using different sequence alignment methods.	128
5.1	ConFunc benchmarking with BLAST and PSI-BLAST.	134

5.2	Assessment for reduced test set.	136
5.3	PSI-BLAST performance.	136
5.4	Precision-Recall analysis using electronic annotations.	137
5.5	Prediction of GTPase functions.	139
5.6	Functional annotation in the twilight zone.	141
5.7	Comparison of ConFunc with Pfam.	143
5.8	Pfam performance for individual GO levels.	144
5.9	Prediction of different functional types.	147
5.10	The function prediction domain problem.	148
5.11	Function prediction at different GO levels.	151
5.12	Function prediction at different GO levels.	153
5.13	ConFunc performance at different GO levels.	155
6.1	The ConFunc server submission page.	160
6.2	ConFunc server results.	161
6.3	Subalignments on the ConFunc server.	162
7.1	Conserved residues for the RAS protein sequence P01112 mapped onto structure 1aa9.	172
7.2	Conserved residues from P49789 mapped onto structure 2fit.	174

List of Tables

1.1	The E.C. classification of enzymes.	15
1.2	Gene Ontology evidence codes.	18
1.3	Accuracy of multiple sequence alignment tools.	38
1.4	The transitive annotation-based scale.	72
2.1	Definitions of Convergence – Mechanistic and Transformational analogues	87
4.1	ConFunc coverage for different Z score thresholds.	127
5.1	ConFunc, BLAST & PSI-BLAST Coverage.	134
5.2	Distribution of annotations in the test set.	146
7.1.	Comparison of conserved residues with functional residues.	170
7.2	Comparison of conserved residues with known functional residues at different residue Z score thresholds.	175

List of Abbreviations

AFP	Automated Function Prediction.
BLAST	Basic Local Alignment Search Tool.
BLOCKS	A database of sequence alignment blocks (not an abbreviation).
BLOSUM	BLOCKS SUbstitution Matrix.
CATH	Class, Architecture, Topology and Homologous superfamily.
CSA	Catalytic Site Atlas.
DAG	Directed Acyclic Graph
DSSP	Definition of Secondary Structure of Proteins.
EC	Enzyme Commission – a classification of enzyme function.
FASTA	Fast Alignment Search Tool.
FFT	Fast Fourier Transform.
FN	False Negative.
FP	False Positive.
FSSP	Families of Structurally Similar Proteins.
GO	Gene Ontology.
HMM	Hidden Markov Model.
HSP	High-scoring Segment Pair.
HSSP	Homology derived Secondary Structure of Proteins.
IMPALA	Integrating Matrix Profiles And Local Alignments.
InterPro	Combined source of Protein functional sites, families and domains (not an abbreviation).

MSA	Multiple sequence alignment.
PAM	Point Accepted Mutation.
PDB	Protein Data Bank.
PSI-BLAST	Position Specific Iterated BLAST.
PSSM	Position Specific Scoring Matrix.
RMSD	Root Mean Squared Deviation.
ROC	Receiver Operating Characteristic.
SCOP	Structural Classification Of Proteins.
SCOPEC	A database that combines SCOP and EC (not an abbreviation).
SMART	Simple Modular Architecture Research Tool.
SSM	Secondary Structure Matching.
SVM	Support Vector Machine.
TN	True Negative.
TP	True Positive.

Chapter 1

Introduction

Protein functional annotation is an important task of the genomics era. The ability to obtain rapidly protein and genome sequences has resulted in many proteins whose function has not been experimentally characterised. Further this characterisation process is slow compared to sequencing itself, resulting in a need for approaches to predict accurately protein function in a fast and automated way. There are currently 650 complete genomes and more than a further 2000 being sequenced (numbers from http://www.ncbi.nlm.nih.gov/genomes/static/gpstat.html, correct as of 31st January 2008). These additional genomes coupled with sequences that will become available from metagenomics projects such as the Sorcerer II Global Ocean Sampling project (Yooseph, et al., 2007), is likely to ensure that the difference between the number of protein sequences and those that are experimentally characterised will continue to increase. The volume of sequences thus requiring annotation makes it important that such methods are automated, enabling them to annotate whole genomes without human intervention.

This work describes the development of ConFunc, an automated sequence based protein function prediction method, which could be scaled to run in a genomics functional annotation pipeline. Much of the work described in this thesis has been published in *Bioinformatics* (Wass and Sternberg, 2008). ConFunc uses the Gene Ontology as a source of functional annotations, identifying conserved residues and creating profiles for them, which are then used to infer function.

This chapter introduces the field of protein function. It considers the relationships between protein sequence, structure and function and then describes existing approaches that have been used to infer protein function. Chapter 2 reports on a study of the convergent evolution of enzyme active sites, which was published in the *Journal of Molecular Biology* (Gherardini, et al., 2007). Chapter 3 details the steps in the ConFunc predictive process, and describes the different approaches and settings that were

investigated during its development. Chapter 4 reports on how the various settings considered during development affected ConFunc performance. Chapter 5 presents the results of benchmarking ConFunc against BLAST, PSI-BLAST and Pfam. Chapter 6 describes the ConFunc web server. Chapter 7 discusses the factors that affect ConFunc performance and the comparison to other methods from Chapter 5. Chapter 8 presents the conclusions that can be drawn from this research and the potential to extend the research with further work.

1.1. Protein Function and Structure Classifications

1.1.1. What is protein function?

It has been suggested that protein function can be considered to be '*everything that happens to or through a protein*' (Rost, et al., 2003). This description demonstrates the complexity of defining protein function because function can be considered at many different levels, all of which are organisationally related to one another. These levels range from biochemical molecular function, for example the reaction an enzyme catalyses, to the cellular function, such as the larger biological process that the enzyme is part of. Higher still, one can consider function at the level of a eukaryotic tissue, organ or whole organism. Different approaches for effectively describing these types of function have been proposed. Bork, et al. (1998) suggested three hierarchical levels to describe function:

Molecular function – describes the biochemical functions that proteins perform individually such as enzyme catalysis and ligand binding.

Cellular function – many individual proteins act together to perform complex cellular processes, such as signal transduction and metabolic pathways. As function is dependent upon the location of protein expression, subcellular localisation is incorporated in this level.

Phenotypic function - The combination of proteins involved in the different cellular processes result in a phenotype. This level would describe the morphology, physiological properties and further the phenotype dysfunction of the protein.

Similarly the Gene Ontology (GO, see 1.1.3) provides three different categories to describe function, two of which, molecular function and biological process are similar to the molecular function and cellular function proposed above.

The complexity involved with accurately describing protein function has resulted in many different classification systems. Protein function is often described using free text, for example Swiss-Prot (Wu, et al., 2006) keywords. Such approaches do not have a formal structure and generally have no way of describing the relationships between different keywords. As a result free text annotations are often inconsistent between sequences, with the possibility of two sequences with the same function being assigned different keywords. This problem is greatest for automated programs and highlights the need for structured classifications. This work will predominantly use the Gene Ontology (Ashburner, et al., 2000) and Enzyme Commission Classification (Webb, 1992), which are explained in detail in the following sections. Other classification systems provide controlled structures for capturing the function of proteins, often in a species specific manner, and are reviewed and compared by Ouzounis et al. (2003) and Rison et al. (2000).

1.1.2. The Enzyme Commission (E.C.) classification

The Enzyme Commission classification was the first system to describe protein function in a controlled numerical fashion. The enzyme commission scheme is a classification for enzymes, with each E.C. number consisting of 4 digits, describing the overall chemical transformation catalysed by an enzyme. The first digit describes the general functional class of the enzyme, of which there are six. The information represented in the subsequent digits is then dependent upon the first digit. For example the second digit in the oxidoreductase class describes the enzyme substrate, while for hydrolases the second digit describes the type of bond that is hydrolysed (Table 1.1). In general each digit further specifies the reaction performed. The fourth digit (serial number) generally describes specific reaction substrates/products or cofactors used. As E.C. only captures details of the chemical reaction catalysed, further systems such as MACiE (Mechanism, Annotation and Classification in Enzymes; Holliday, et al., 2005) have been developed to store enzyme mechanism data to supplement E.C.

First Digit (class)	*Second Digit (sub-class)*	Third Digit (sub-subclass)
1. Oxidoreductases	Enzyme substrate	Acceptor type
2. Transferases	Group transferred	Information on group transferred
3. Hydrolases	Type of bond cleaved	Substrate type
4. Lyases	Type of bond formed	Information of group eliminated
5. Isomerases	Type of reorganisation	Substrate type
6. Ligases	Type of bond formed	Type of compound formed

Table 1.1 The E.C. classification of enzymes. The first E.C. digit describes the general functional class, and the subsequent digits further characterise the overall chemical reaction performed. The meaning of each digit is dependent upon the first E.C. digit as shown (adapted from Todd, et al. 2001).

1.1.3. Gene Ontology

The Gene Ontology (GO; Ashburner, et al., 2000) provides a structured and controlled vocabulary for functional annotation to ensure that annotations are accurate and can easily be compared between different gene products. It builds upon concepts from earlier single organism classifications but is general, enabling the classification of functions present in different organisms. GO is effectively three different ontologies, one each for: molecular function, biological process and cellular component. Molecular function terms describe biochemical function while biological process terms describe the role that proteins play in a system. Finally, cellular component annotations capture details about the cellular location where a gene product is active. GO has wider application than the E.C. classification as it is possible to annotate both enzymes and non-enzymes and the three ontology categories provide a wealth of varied annotation information that is not captured by E.C. However the catalytic function category within the molecular function ontology is largely similar to the E.C. classification.

GO uses a controlled vocabulary and every function is associated with a unique identifier and text describing the function. GO has a Directed Acyclic Graph (DAG) structure (Figure 1.1), with terms related by two relationships: *is:a* and *part:of*. The *is:a* relationship within the DAG structure enables a hierarchical structure to be generated, with general functional terms present near the root of the graph, progressing to more specific terms as shown in Figure 1.1. The presence of annotations in each of the GO

ontologies fully captures protein function. Proteins may have more than a single annotation in each of the ontologies. For example, the protein described in Figure 1.1 is annotated with two molecular functions, a GTPase catalytic activity and a GTP binding function. The molecular functions are involved in larger processes, which are represented by the biological process annotation. In this case they are organ morphogenesis and cell surface receptor linked signal transduction. Finally, the protein is involved in these processes at the plasma membrane and in the cytoplasm, as documented by its cellular component annotations.

The Gene Ontology provides both general and specific functional terms, but unlike the four levels in E.C. classification the differences in specificity between different levels are not set. Different areas of the graph have different depths such that a specific function may be eight levels from the root in one area of GO, while another equally specific function may only be three levels from the root. This makes it difficult to identify and compare functions at equivalent levels of specificity when they are in different parts of GO. It is possible to assign levels to each GO node but this simply describes the number of nodes from the root and can only loosely be used to compare functions between areas of the graph.

Each GO annotation is associated with an evidence code. They explain how the annotation was identified and give an indication of how confident it is. The different evidence types are shown in Table 1.2. The experimental evidence codes (IDA, IPI, IMP, IGI and IEP) are used when a paper shows results from a physical experiment that indicates association of the gene/gene product with the GO term. The pubmed id of the paper is included as part of the annotation, enabling users to access the source of the annotation. Author statement evidence codes (TAS and NAS) are used when annotations are based on the statements of an author. As with experimental evidence codes the pubmed id also forms part of the annotation. Computational analysis evidence codes (ISS, IGC, RCA) are used when the evidence for an annotation is obtained largely from computational analyses which have been analysed by a curator. Such annotations that have not been reviewed manually are given IEA evidence codes. Finally the Inferred by

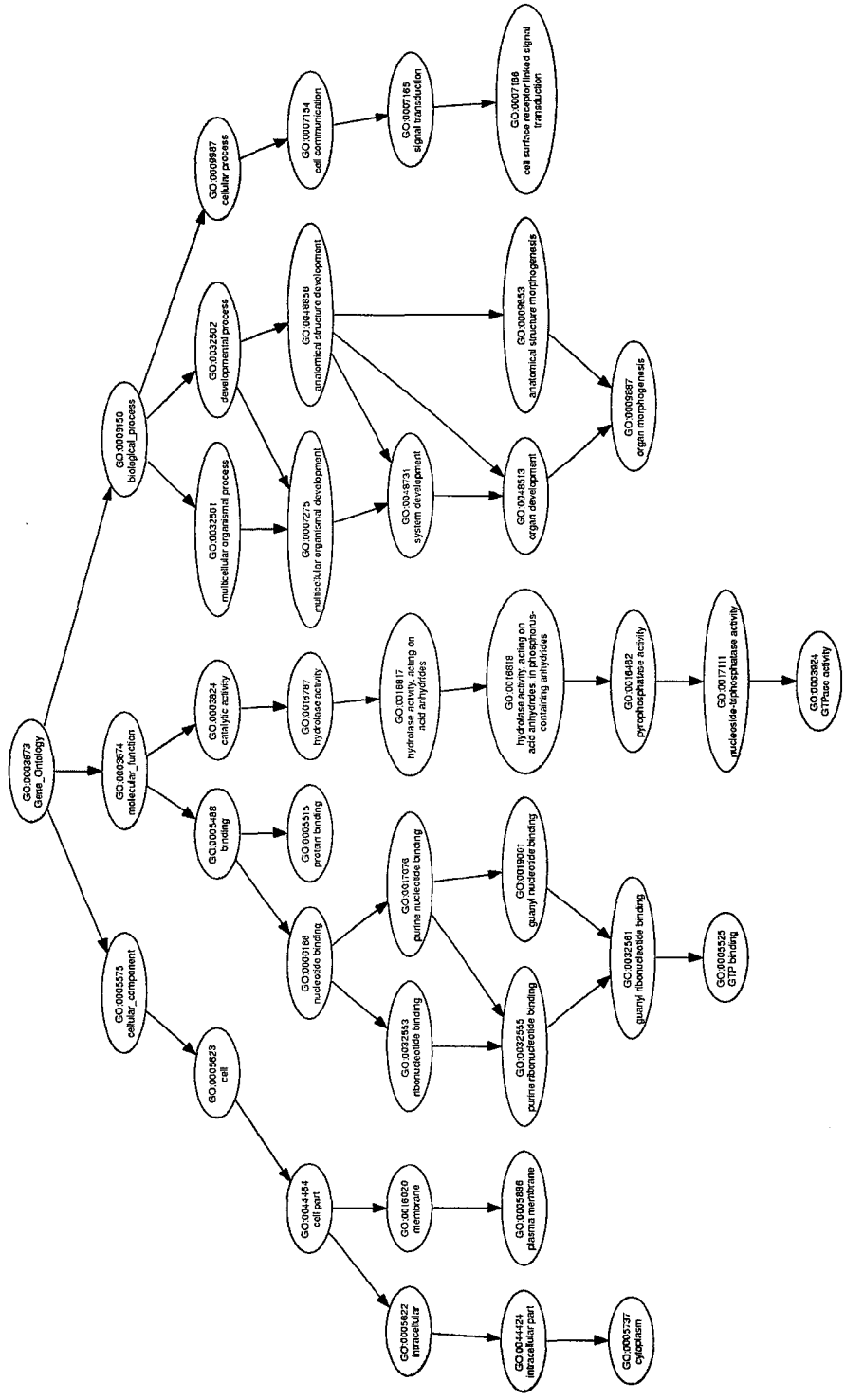


Figure 1.1 The Gene Ontology. The GO annotation of the Human Ras protein (Swiss-Prot sequence P01112).

Evidence Code	*Description*	*Number of Annotations*
IC	Inferred by curator	759
IDA	Inferred from direct assay	17074
IEA	Inferred from electronic annotation	1288029
IEP	Inferred from expression pattern	1080
IGI	Inferred from genetic interaction	1983
IMP	Inferred from mutant phenotype	8203
IPI	Inferred from physical interaction	5408
ISS	Inferred from sequence or structural similarity	3278
NAS	Non-traceable author statement	6235
ND	No biological data available	2382
TAS	Traceable author statement	21195
NR	Not recorded	1473
RCA	Inferred from reviewed computational analysis	NA
IGC	Inferred from genomic context	NA

Table 1.2 ***Gene Ontology evidence codes.*** The evidence codes assigned to Gene Ontology annotations indicate the type of information that was used to infer them. The number of molecular function annotations with each evidence code in the EBI Gene Ontology annotations (GOA) version 28 is also shown. Evidence codes RCA and IGC were introduced more recently than GOA version 28 and so they do not have an entry for the number of annotations.

Curator (IC) and No biological data available (ND) evidence codes are used by curators when none of the other evidence codes are appropriate.

The Gene Ontology annotations (GOA) produced by the EBI are used in this thesis. The EBI has a group of annotators who assign manual annotations from literature and experiment. The EBI GOA also contains electronic annotations, which are generated using mappings from other databases to GO. Mappings from E.C., Swiss-Prot keywords, Interpro and HAMAP are predominantly used. These annotations are automatically created by applying the mappings and they are not manually reviewed. There is less confidence in these annotations because of this automated process. There are two unknown sources of error as the GO annotations do not record or consider the evidence

for the annotation from its original source. The mapping itself is the second source of potential error.

To date there has not been a systematic study of the error rate associated with GO annotations and each of the different evidence codes. Research by Jones et al. (2007) compared the annotations of similar GO annotated sequences. For a set of sequences they used BLAST to identify homologues and then identified the BLAST hit with the most similar GO annotation. They calculated a measure of precision for the annotations of these two sequences and compared the precision to a reference set where errors had been introduced at a known rate. They estimated annotation error rates for curated evidence codes (IC, ID, IEP, IGI, IMP, IPI, NAS, ND, RCA, TAS and NR) to be between 13-18%. They separately estimated the error rate of annotations with the ISS (inferred from sequence or structural similarity) evidence code to have a 49% error rate. It is not clear that the approach taken in this study truly calculates the GO annotation error rate. They chose to compare the GO terms between different sequences, this appears to use an assumption that two similar sequences should share exactly the same function, which numerous studies have shown is not necessarily the case (see section 0). This is particularly important as a $1x10^{-10}$ e-value threshold was used, making it possible that while the sequences compared are homologous, they are far from identical. Further their results do not appear to consider that BLAST may only identify regions (such as individual domains) that are similar between the sequences used. If this is the case then their error rate calculation would be affected by comparing GO terms that represent functions associated with domains not shared by both sequences and incorrectly increase their error rate calculation.

1.1.4. Structural Classifications of Proteins - SCOP & CATH

The Structural Classification of Proteins (SCOP; Murzin, et al., 1995) predominantly uses protein structural features along with sequence, functional and evolutionary properties to hierarchically classify protein domains. They define a protein domain as an evolutionary unit, such that it exists in isolation in nature, or is observed in more than one context in multi-domain proteins (Andreeva, et al., 2004). Protein domains are grouped by species

and hierarchically into families, superfamilies, folds and classes (Figure 1.2). The classification process makes use of automated tools but there is an emphasis on manual interpretation of data to make classifications. The main SCOP levels of classification are described below.

Family – Proteins belonging to the same family either share sequence identities of 30% or greater, or they have lower levels of sequence identity but their functions and structures are very similar. There is a clear common evolutionary origin between the proteins present in the same family.

Superfamily – Families are grouped into superfamilies. Families in the same superfamily may have low sequence identities but they share common structural and possibly functional features that suggest a common evolutionary origin.

Fold – Superfamilies are grouped into folds if they have common major secondary structure arrangements with similar topological connections. Proteins in the same fold may have peripheral secondary structures in addition to their central secondary structure, which can form up to half of the structure. Members belonging to the same fold (but different superfamilies) are likely to have different evolutionary origins and are often analogues.

Class – The highest level of the SCOP hierarchy groups folds according to five structural classes based upon the composition of their secondary structure. The five classes are: *all alpha*, *all beta*, *alpha and beta* (α-helix and β-sheet interspersed), *alpha plus beta* (separated α-helix and β-sheet) and *small proteins*. There are two further classes; one is *multi-domain,* which contains proteins with different folds that do not have a known homologue, and finally *membrane and cell surface proteins.*

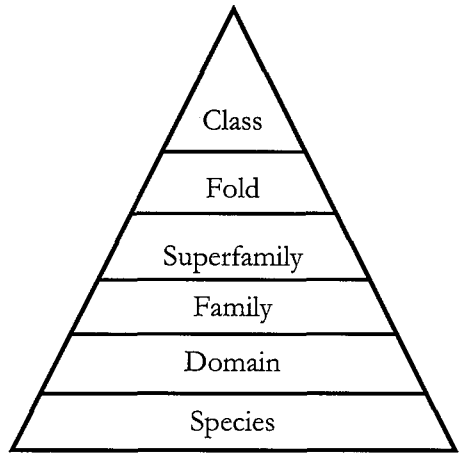


Figure 1.2 The SCOP hierarchy. Adapted from Murzin et al. (1995) Figure 1.

CATH (Class, Architecture, Topology, and Homologous superfamily; Orengo, et al., 1997) provides another classification of protein structure. In comparison with SCOP, it is largely automated, with manual classification steps increasingly being automated (Greene, et al., 2007). The four major levels of the CATH hierarchical classification are described below:

Homologous Superfamily – Domains grouped at this level are considered to have a common evolutionary origin and are therefore homologues. Domains grouped into the same homologous superfamily either share high levels of sequence or structural similarity.

Topology – Homologous superfamilies are grouped according to their overall secondary structure and the connectivity of these secondary structure elements.

Architecture – Describes the overall domain structure based upon the orientation of the secondary structure elements.

Class – Describes domain secondary structure composition and its packing in the structure. There are three main classes: *mainly-alpha*, *mainly-beta* and *alpha-beta*. They are similar to the SCOP classes with the exception that the CATH *alpha-beta* class is the equivalent of two SCOP classes (*alpha+beta* and *alpha-beta*). There is a fourth class for proteins with little secondary structure.

1.1.5. Protein Sequence Families

SCOP and CATH, described in the previous section, classify protein structures into domains. Other resources group protein sequences into families. Such families are generally represented by sequence alignments of homologous proteins. While SCOP and CATH have a modular approach by considering protein domains individually, protein sequences often contain multiple domains and as a result sequence families rarely represent a single domain. It is more common for a protein sequence family to contain multiple domains that are present in most or all of the proteins belonging to that family. Pfam, ProDom and SMART are widely used databases of protein sequence families and they are described below.

1.1.5.1. Pfam

Pfam (Finn, et al., 2006) is a widely used database of protein domain families, which can be used to identify protein domain architecture. Unlike SCOP and CATH, it uses protein sequence and domains. Families are represented by multiple sequence alignments and Hidden Markov models (HMMs). Pfam has two components: Pfam-A and Pfam-B. Pfam-A is manually curated and therefore the alignments are of high quality. Pfam-A families are built using seed alignments (alignments of a small number of representative sequences), which are used to generate HMMs for the family. HMMer (Eddy, 1998) is used to search UniProt with the HMMs to identify members of the families, with all hits above a family specific cut off added to the full alignment for that family. Pfam-B provides greater coverage of sequence space by using alignments from ProDom (Servant,

et al., 2002) and removing any residues that are already classed as part of a domain by Pfam-A. The Pfam-B alignments are of lower quality but they are useful when sequences do not have Pfam-A hits. Recent versions of Pfam have introduced clans, which are groups of families sharing a common evolutionary origin.

1.1.6. Protein Motif Methods

A number of approaches have identified motifs associated with functions which can then be used to search through target sequences to infer function. PROSITE (Hulo, et al., 2006) is a database of protein signatures that relate to protein families, domains or functional information. The signatures are stored as patterns and profiles which are generated from alignments of homologous sequences. PROSITE patterns are regular expressions that capture the conservation of residues and regions that are important for biological function (Figure 1.3), whereas PROSITE profiles can capture features throughout a whole domain (Sigrist, et al., 2002). PROSITE is calibrated against Swiss-Prot, which enables estimates of the expected accuracy of each pattern and profile to be calculated.

PRINTS (Attwood, et al., 2002) is a database of sequence fingerprints. Proteins can often be represented by multiple conserved motifs, so a fingerprint is a group of motifs found in a multiple sequence alignment. They can provide a better way of capturing conservation than individual motifs. The BLOCKS (Henikoff and Henikoff, 1992) database consists of ungapped blocks of local alignments representing protein families or conserved regions and can also be used in a similar way.

[STAIV] – September - [LIVMF] - [LIVM] - D - [DSTA] - G - [LIVMFC] - x(2,3) - [DNH]

Figure 1.3 A ProSite signature. The ProSite signature for the active site of enzymes in the subtilase family. Square brackets indicate that one of the enclosed residues is found at that position. Curly brackets indicate that any but the enclosed residues can occur at that position. X indicates that any residue can occur at that position.

1.1.6.1. Interpro

Interpro (Mulder, et al., 2005) combines the various protein domain/family and motif databases. These include: PROSITE, PRINTS, ProDom, Pfam, SMART, TIGRFAMs (Haft, et al., 2003), PIRSF (Wu, et al., 2004), SUPERFAMILY(Gough, et al., 2001), Gene3D (Yeats, et al., 2007) and PANTHER (Mi, et al., 2005). The integration of these methods into Interpro allows the use of evidence from different sources to be combined and provide increased confidence in the data obtained. The signatures for the same domain or family from each of the member databases are grouped into a single Interpro entry.

Interpro is arranged into families, domains, repeat, post-translational modifications and sites (sites are further split into active and binding sites). Entries can also be related by two relationships: parent/child and contains/found. In the parent/child relationship the child is a subset of the parent and is therefore more specific than the parent entry. The contains/found relationship is used for domain composition as domains can be found in multiple families but are not subtypes.

1.2. Sequence Similarity Searching and Alignment

Sequence searching and the alignment of homologous sequences is a crucial component of bioinformatics, as seen by the number of methods described in section 1.5.1 that use BLAST, PSI-BLAST and other alignment tools. In this section these methods will be described, with focus on those methods used in this research.

1.2.1. Comparing Sequences – Substitution Matrices

A simple approach for assessing the similarity of two sequences is to align them and calculate the percentage of amino acid positions that are identical between them. However, such a simple approach is not adequate because the twenty amino acids have physico-chemical similarities and differences, which are not accounted for by sequence identity. For example, a substitution from aspartic acid to glutamic acid is very minor compared to a change to tyrosine, as aspartic and glutamic acid have the same charge and

near identical side chains. In contrast the tyrosine side chain is considerably different from aspartic acid and contains benzene. Substitution matrices have been developed to represent how favourable changes between amino acids are. In the previous example, the aspartic acid to glutamic acid substitution would have a high substitution score, whereas the unfavourable aspartic acid to tyrosine substitution would have a much lower score.

Substitution matrices have the dimensions of 20x20 (for proteins), with each cell representing the likelihood of substitution of one amino acid for another. Substitution matrices are used extensively for the scoring of sequence alignments (see below) and therefore an optimal substitution matrix will score biologically relevant alignments positively and give negative results for random matches. A standard way of generating a substitution matrix is to calculate the probability of an amino acid substitution by calculating the frequency of such a substitution in alignments between homologous sequences. It is important that substitution matrices have a negative expectation score (equation 1.1) for a pair of randomly picked residues

$$\sum_{i,j} p_i p_j S(i,j) < 0 \qquad \textbf{1.1}$$

where $S(i,j)$ is the substitution score for amino acid i with amino acid j and p_i and p_j are the background frequencies of amino acids i and j respectively. The expectation score for a random substitution needs to be negative because otherwise long alignments could have high scores as a result of their length, without a significant biological relationship (Altschul and Gish, 1996).

A general formula exists for the generation of a substitution matrix from observed rates of substitution (Altschul and Gish, 1996)

$$S(i,j) = \frac{\ln\left(q_{i,j} / p_i p_j\right)}{\lambda} \qquad \textbf{1.2}$$

where $q_{i,j}$ is the target substitution frequency (the observed frequency of substitution from amino acid i to j). The product of the background frequencies ($p_i p_j$) represents the probability of chance substitutions of i for j. The target substitution frequency is divided

by this product to account for the different amino acid frequencies, ensuring that conservative changes of rare amino acids are considered favourably. The score is then divided by a scaling factor λ (specific to the scoring system) and rounded to the nearest integer. The resulting score is referred to as a log odd score. The widely used point accepted mutation (PAM) and BLOCKS substitution matrices (BLOSUM) are described in the following sections.

1.2.1.1. PAM Matrices

The PAM matrices (Dayhoff, et al., 1978) consider substitutions between amino acids over a period of evolutionary time, which is known as a point accepted mutation. One PAM represents the probability that during an evolutionary time a single point mutation occurs over 100 residues, equivalent to a one percent mutation rate. By extension 50 PAMS represents the evolutionary period during which 50 point mutations have occurred in 100 residues (i.e. a 50% mutation rate). Such matrices over longer evolutionary times (i.e. n PAMS) can be obtained by taking PAM1 to the power of n. PAM matrices perform best when used with closely related sequences, because closely related sequences (alignments of sequences with greater than 85% identity) were used for the generation of the matrices. The conversion to longer evolutionary times from the PAM 1 matrix does not consider the potential conservation of structurally and or functionally important residues, which can result in mutation rates being overestimated.

1.2.1.2. BLOSUM Matrices

The BLOCKS database (Henikoff and Henikoff, 1996) was used to generate the BLOSUM matrices (Henikoff and Henikoff, 1992). The BLOCKS database contains ungapped blocks of local alignments representing protein families or conserved regions. The blocks are used to calculate target and background frequency probabilities for amino acid substitutions and result in the BLOSUM matrices.

To reduce the overrepresentation of very similar sequences BLOSUM matrices are generated at different levels of sequence identity. For example BLOSUM80 clusters all sequences within the blocks that are 80% or more identical, with each cluster treated as a

single sequence. The BLOSUM matrices have been shown to perform better than equivalent PAM matrices for use in both sequence alignment and homology searching (Henikoff and Henikoff, 1992). These improvements are greater for more distantly related sequences, which is likely to be a consequence of the PAM matrices being generated from only very closely related sequences and the extrapolation of this PAM to consider greater evolutionary distances.

The BLOSUM50, BLOSUM62 and BLOSUM80 matrices are most widely used and BLOSUM62 (

Figure 1.4) is generally thought to balance best between correctly identifying closely related sequences and missing very few distant homologues.

	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Figure 1.4 BLOSUM62 amino acid substitution matrix. A 20x20 matrix, containing the log-odds scores for amino acid substitution.

1.2.2. Dynamic Programming – Optimal Alignments

Many alignment methods use dynamic programming (Bellman, 1957), which is a general algorithm guaranteed to produce the optimal alignment of two sequences. For example the Needleman-Wunsch (1970) algorithm generates a global alignment, aligning two sequences over their whole length allowing gaps. The Smith-Waterman (1981) algorithm is similar to the Needleman-Wunsch algorithm but it generates local alignments, aligning subsequences within two sequences. This is important because similarity often occurs over only sections of protein sequence.

These dynamic programming algorithms provide optimal alignments but they prove time consuming for large sequences particularly when used with the rapidly increasing protein databanks. Therefore sub-optimal but fast algorithms have become essential. Such heuristic approaches can identify significant sequence similarities by considering small regions of proteins. FASTA (Fast Alignment Search Tool; Pearson and Lipman, 1988), BLAST (Altschul, et al., 1990) and PSI-BLAST (Altschul, et al., 1997) are the most commonly used algorithms. BLAST and PSI-BLAST are described in the following sections as they are used extensively in this research.

1.2.3. The BLAST Algorithm

FASTA was the first heuristic algorithm for searching a database for similar sequences. It was closely followed by the Basic Local Alignment Search Tool (BLAST; Altschul, et al., 1990), which although similar to FASTA, is faster. Initially, BLAST generated ungapped alignments by searching for segments of the query sequence in the sequence database. Later releases of BLAST allowed the generation of gapped alignments and also improved sensitivity and speed (Altschul and Koonin, 1998; Altschul, et al., 1997). In outline, BLAST works by searching rapidly for small segments or words that align with the query sequence and when scored using a substitution matrix are above a threshold T. This identifies regions that are likely to be related. These hits are extended to generate a larger alignment with a score greater than a second threshold S. The algorithm is described in detail below:

1. Search for all words (generally 3 residues), that when aligned with the query sequence score greater than a threshold ***T***. The scores are calculated using a substitution matrix such as BLOSUM (see 1.2.1).

2. Many hits are generated, so to reduce the number of alignment generated, all pairs (query sequence and database sequence) with two non-overlapping hits within a threshold distance, ***A***, are identified. Where hits overlap, the most recent one is ignored. Alignments are extended only for those pairs that meet these conditions. BLAST originally required only a single hit to initiate an extension. This two hit requirement reduces the number of extensions made, increasing the speed of the algorithm compared to the original BLAST algorithm, which only required a single hit to initiate alignment extension.

3. The extended alignments generated from the initial hits are ungapped and bidirectional. The extension continues as long as the cumulative alignment score is greater than ***S*** until it cannot be further improved. The alignment extension also stops for other conditions:
    - The cumulative alignment score falls by greater than a threshold ***X***, from its maximum score or to zero or below.
    - The end of one of the sequences is reached.

4. The extended hits are referred to as High Scoring Segment pairs (HSP). The highest HSPs are selected for further bidirectional extension, using a gapped alignment.

5. Finally this gapped alignment is realigned with relaxed parameters, to maximise alignment length.

1.2.4. Sequence Alignment Statistics

The statistics of local alignments are well established and have been mathematically proven for ungapped local alignments (Karlin and Altschul, 1990). As described earlier, it

is important that the expectation score of a pair of randomly chosen amino acids is negative (equation 1.1) to ensure that long alignments are not favourable due to their length. This condition is also necessary for the statistical theory to hold. Karlin and Altschul (1990, 1993) have shown that the distribution of optimal ungapped alignment score for two random sequences can be approximated to an extreme value distribution. The fitting results in two statistical parameters λ and K (Altschul and Gish, 1996). These parameters are used to obtain a normalised score S', using equation 1.3

$$S' = \frac{\lambda S - \ln K}{\ln 2} \qquad \textbf{1.3}$$

where S is the raw score. The normalised score is then used to calculate the probability P, that an optimal ungapped subalignment scores at least x as

$$P(S \geq x) = 1 - \exp(-Kmne^{-\lambda x}) \qquad \textbf{1.4}$$

where m and n are the lengths of the two sequences. For a database search m is the length of the query sequence and n the size of the database in residues, which for BLAST is the number of residues in the database. Other methods, such as FASTA, use the number of sequences in the database instead. The probability calculated is a p-value. The p-value is converted to an e-value using equation 1.5.

$$e = \frac{mn}{2^{s'}} \qquad \textbf{1.5}$$

The e-value describes the number of times that an alignment with a score greater than or equal to S' would occur by chance in a database of size n. This measure is used by programs such as BLAST to indicate the significance of the alignments generated.

These statistics have only been proven for ungapped local alignments. However, numerous simulations and computational analyses (Altschul and Gish, 1996; Altschul and Koonin, 1998; Collins, et al., 1988; Karlin and Altschul, 1990; Karlin and Altschul, 1993; Smith, et al., 1985; Waterman and Vingron, 1994) suggest that the extreme value distribution of scores and subsequent statistics hold for ungapped alignments and as such they are used in gapped BLAST (Altschul, et al., 1997) and FASTA for the calculation of

e-values. As BLAST only generates gapped alignments for a small number of sequences with potential similarity, it is unable to estimate the parameters λ and K, so it uses precalculated estimates generated from simulation.

1.2.5. Position Specific Scoring Matrices & PSI-BLAST

Position Specific Scoring Matrices (PSSM) are specialised substitution matrices, which represent the amino acid frequencies in a group (or family) of proteins present in a multiple sequence alignment, whereas a standard substitution matrix represents the average substitution rate across a much wider range of proteins. PSSMs also differ from standard substitution matrices because they capture the positional nature of the amino acid frequencies, with a substitution score for each amino acid at each position of the alignment. A PSSM is therefore an N x 20 matrix, where N is the length of the alignment. As PSSMs capture the positional frequencies of the amino acid residues for an alignment of proteins, they are more powerful at identifying remote relationships with distant homologues than standard substitution score approaches using pairwise alignments. This is demonstrated by PSI-BLAST.

Position-Specific Iterated BLAST (PSI-BLAST) is a development of the original BLAST algorithm that builds PSSMs to perform multiple iterations of a BLAST search. The steps in a PSI-BLAST search are as follows:

1. A standard BLAST search is run first.
2. The sequences identified are filtered removing any above a certain e-value threshold (default 0.01) and sequences identical to the query. A single representative is retained for sequences with very high identity (>98%).
3. A multiple alignment is formed by stacking the remaining sequence segments.
4. The multiple alignment is used to generate a PSSM (this process is described in detail below).
5. The PSSM is used as input for a further BLAST search and the steps 2-5 are repeated adding any new sequences identified to the PSSM. This is done for either a set number of iterations or until the search converges (no further sequences are identified).

The alignment statistics for ungapped alignments that are used for BLAST still approximate for the PSI-BLAST gapped alignments and are used to calculate e-values.

The use of PSSMs in PSI-BLAST to search for homologies is three times more sensitive than BLASTs comparison of single sequences (Park, et al., 1998). However, the iterative process can also result in drift, with the results from subsequent iterations moving away from the original query sequence and the sequences originally identified by BLAST. This is most likely to happen if PSI-BLAST erroneously identifies a sequence as related to the query and then uses it to calculate the PSSM used for the next iteration. Regions of the PSSMs can become very specific, often representing highly conserved functional or structural regions and are more general where there is greater variability between sequences. The PSI-BLAST PSSM calculation consists of three main steps, which are described below.

1.2.5.1. PSI-BLAST PSSM Calculation – Reducing the Multiple Alignment

For the calculation of a PSSM, the values for each column (or alignment position) are dependent upon the residues in that column and also the residues in other columns. This is performed by reducing the original multiple sequence alignment *M*. First any gaps added to the query sequence are removed, so that the alignment is the length of the query sequence. The reduced alignment M_C, will contain a subset of the sequences in *M* and will also only consider a subset of the columns in *M*. For column *C*, the set *R*, of sequences that contribute a residue at position *C* (including gaps) are identified and only these sequences are present in M_C. The columns of M_C are only those columns for which all sequences in *R* contribute a residue (including gaps).

1.2.5.2. PSI-BLAST PSSM Calculation - Sequence Weighting

Sequence weights are calculated to account for overrepresentation of very closely related sequences which provide less information than more divergent sequences present in an alignment. The sequence weighting used is similar to the position-based sequence weighting described by Henikoff and Henikoff (1994). For the calculation, gaps are

treated as a distinct 21st amino acid and any columns that contain identical residues are ignored. The weighting considers the number of different residues present and their frequency at each position in M_C. For a position with r different residues, a residue contributes $1/(r \times s)$, where s is the number of sequences that the residue is present in at that position. To calculate the sequence weight, the values from each column of M_C are summed. The sequence weight is then normalised so that the sum of the sequence weights equals one. As M_C may be different for each column C, the sequence weights may also differ for each column. Observed weighted frequencies f_i, for residue type i, are calculated as the sum of the weights for the sequences in M_C with residue i at column C.

It is also necessary to capture the effective number of independent observations that the observed frequencies represent. This is represented as N_C, which is the relative number of independent residue observations. PSI-BLAST calculates N_C as the mean number of different residue types (including gaps) present in the columns of M_C. The maximum value of N_C is 21 (i.e. 20 different amino acids plus the gap character), but for most columns N_C is likely to be considerably smaller and the relative values of N_C between columns are more important.

1.2.5.3. PSI-BLAST PSSM Calculation - Target Frequency Estimation

Equation 1.2 provided a general formula for the calculation of a substitution score based upon the ratio of the observed frequency of residue pairs compared to the background probability of these pairs occurring by chance ($q_{i,j}/p_ip_j$). Due to the weighting described above and the potential for small sample sizes (some positions will only contain a few sequences), it is necessary to modify this calculation to include the target frequency from the original substitution matrix. To resolve this, pseudocount frequencies g_i, are calculated as shown in equation 1.6, where q_{ij} is the target frequency of the standard substitution matrix (e.g. BLOSUM62) for residues i and j and P_j is the background probability for residue j and f_j is the observed weighted frequency of residue j.

$$g_i = \sum_{j \neq i}^{20} \frac{f_j}{P_j} q_{ij} \qquad \textbf{1.6}$$

It is then possible to calculate the target frequency Q_i using the formula

$$Q_i = \frac{\alpha f_i + \beta g_i}{\alpha + \beta} \quad 1.7$$

where α and β are the relative weights for the observed and pseudocount frequencies respectively. β is a pseudocount parameter, the higher its value, the more emphasis is placed upon the prior knowledge present in the original substitution matrix. PSI-BLAST uses an empirically calculated value of 10 for β. α is set at N_C-1 to minimise the scores for columns where no sequences have been aligned to the query sequence.

1.2.6. Multiple Sequence Alignment

The previous methods considered pairwise sequence alignment (i.e. the alignment of only two sequences). However, proteins often have many homologues with which they share similarity, so multiple sequence alignment methods are required. Generating optimal multiple sequence alignments aims to maximise the sum of similarities for all pairs of sequences in the alignment. This is a very time consuming problem, especially when there are many sequences and/or they are long, therefore the time taken to obtain an optimal solution is impractical. Like BLAST for sequence searching, multiple sequence alignment methods need to use heuristic approaches.

The progressive alignment heuristic (Feng and Doolittle, 1987) is widely used by multiple sequence alignment programs. This approach generates a phylogenetic tree (referred to as a guide or merge tree), which is based upon a distance measure between each of the sequences, with the individual sequences forming the leaves of the tree. The internal nodes of the tree represent groups of the input sequences and each group is aligned by either aligning profiles (of the alignments) or the alignments themselves. The algorithm progressively moves up the tree merging the groups, generating larger alignments until an alignment of all the input sequences is obtained at the root. Refinement (often referred to as polishing) of this alignment is a common step and methods often split the alignment into subsets and realign them. Numerous multiple alignment methods have been developed and they generally implement variations of the progressive alignment algorithm to improve both accuracy and speed of the alignments

obtained. Current sequence alignment methods are briefly described below, focusing on the features that differentiate them.

1.2.6.1. Multiple Sequence Alignment Programs

Of the initial multiple alignment methods to use progressive alignment, CLUSTAL W (Thompson, et al., 1994) is the most well known and is still widely used today. It built upon the progressive alignment algorithm by adding sequence weighting to account for redundancy, residue specific gap penalties and varying substitution matrices depending upon the level of divergence.

Other methods for multiple sequence alignment include: T-Coffee (Notredame, et al., 2000), MUSCLE (Edgar, 2004), MAFFT (Katoh, et al., 2005; Katoh, et al., 2002) and ProbCons (Do, et al., 2005). MUSCLE implements an iterative approach to the progressive alignment. It calculates k-mer distances (short exact matches of fixed length k) to generate a rough guide tree, which is used to perform a progressive alignment. This initial multiple sequence alignment is then used as input to generate a more accurate guide tree, which is used for a second progressive alignment. This approach offers increased speed without sacrificing accuracy when compared with other methods (Edgar, 2004).

MAFFT (Katoh, et al., 2002) obtains high levels of accuracy (comparable to or better than other existing methods – see Table 1.3) by using a simplified normalised scoring system and fast Fourier transform (FFT) to identify homologous regions. The FFT approach is based upon the concept that amino acid substitutions with similar physico-chemical properties are less likely to disturb protein structure and are therefore more likely to occur during evolution. Each amino acid is represented by its volume and polarity, which are two important physico-chemical amino acid properties. Sequences are described as vectors of volume and polarity and the FFT of these vectors is used to identify homologous regions between sequences. An alignment of two sequences is obtained by arranging the homologous regions identified between them. Recent improvements to MAFFT include T-Coffee like consistency (see below), which has significantly improved accuracy (Katoh, et al., 2005).

The progressive alignment algorithm initially performs pairwise alignments of the sequences, which are not modified during the later steps of generating the multiple sequence alignment. Any errors in these initial alignments are therefore retained in the multiple sequence alignment. The refinement process mentioned above helps to remove these errors in the final stages of the alignment process. The use of consistency is another approach, which aims to reduce the occurrence of these initial errors by considering the other sequences in the set when generating the initial pairwise alignments. For each pair of aligned residues, consistency considers how much agreement there is for this alignment with the other sequences.

Consistency was introduced in T-Coffee (Tree-based Consistency Objective Function for alignment evaluation). T-Coffee first generates a primary library of both global (generated using CLUSTAL W) and local pairwise alignments of all the sequences that are to be aligned. The alignments in the primary library are weighted according to sequence identity. The primary library is extended by considering each pair of aligned residues with the residues from the remaining sequences. Where there is support for the match from other sequences in the library its weight is increased. The progressive alignment uses the weightings from the extended library to guide the initial pairwise alignment of sequences. This aims to improve the accuracy of the pairwise alignments by incorporating information from the other sequences. Recent improvements have resulted in 3D-Coffee (O'Sullivan, et al., 2004), which enables the use of protein structures to guide the sequence alignment process.

ProbCons uses maximum expected accuracy and probabilistic consistency for the progressive alignment of multiple sequences. A pair-HMM is used for the initial pairwise alignment which uses the maximum expected accuracy (the expected number of correctly aligned pairs of residues). Posterior probabilities are calculated for each pair of residues. They are transformed using probabilistic consistency, which for the pairwise alignment of two sequences considers how other sequences (from the set of sequence for the multiple sequence alignment) align with the two sequences being aligned. The maximum expected accuracies are used in the computation of the guide tree and the subsequent alignments use the probabilistic consistency values for scoring.

Opal (Wheeler and Kececioglu, 2007) obtains high levels of accuracy (see Table 1.3) by implementing two modifications to the typical progressive alignments performed by most methods. The first change uses a normalised alignment cost, to calculate the distances between sequences, which is then used for the construction of the merge tree. This cost simply divides the sum of pairs score for an optimal alignment for two sequences by the number of columns. Standard approaches use sequence identity or compressed identity, which accounts for multiple substitutions at the same position as measures of distance between the sequences.

Secondly, Opal modifies the refinement stage where the initial alignment is normally split into two and the groups realigned. Instead Opal splits the initial alignment into three groups, which are then realigned. In addition this process is run 'on-the-fly' by performing this splitting on the intermediate alignments generated at internal nodes of the merge tree, rather than only once the complete alignment has been generated.

1.2.6.2. Performance of Multiple Sequence Alignment Programs

The performance of multiple sequence alignment programs is generally benchmarked by comparing the alignments with reference sets of structurally aligned proteins. A number of reference sets are available including: BAliBASE (Thompson, et al., 2005), SABmark (Van Walle, et al., 2005) and PALI (Balaji, et al., 2001). BAliBASE is generated by a combination of automated and manual methods, while most other reference sets are generated by predominantly automated methods resulting in varying quality and accuracy in the reference alignments. BALiBASE may therefore be considered as more reliable, although the comparison between different alignment methods over a large set of sequences from the automated reference sets is also valid (Edgar and Batzoglou, 2006). Each of the alignment methods described in the previous section was benchmarked using one or more of these reference sets. In many cases only a subsection of the alignments present in a reference set were used. This may result in apparent differences between the alignment programs due to differences in the reference alignments.

The recent benchmarking of Opal (Wheeler and Kececioglu, 2007) assessed its performance against that of MAFFT, ProbCons, T-Coffee, MUSCLE and CLUSTAL W.

The results of this benchmarking are shown in Table 1.3. For all the reference sets used CLUSTAL W performs worst, followed by MUSCLE and T-Coffee which exhibit very similar performance. The best performers are MAFFT, Opal and ProbCons, which obtain similar scores over the three reference sets, with either MAFFT or Opal obtaining the highest scores, depending upon the reference set and scoring system used. It is not surprising that CLUSTAL W performance is inferior to the other methods as it has changed very little since it was developed in the early nineties. The other approaches have been developed more recently and have introduced important improvements to multiple sequence alignment methods.

The benchmarking results are based on a large number of alignments and while a method such as MAFFT may outperform others overall, there can be individual cases where alternative methods generate more accurate alignments (Edgar and Batzoglou, 2006). However, where an alignment method forms part of an automated program, as for the research described in this thesis, the best overall performing method is likely to be most suitable.

Finally the benchmarking results in Table 1.3 do not account for computation costs, such as the time and memory requirements of the methods. For example MUSCLE and T-Coffee obtain similar performance but MUSCLE is considerably faster

	BALiBASE		*SABmark*	*PALI*		*Average*	
Tool	**SPS**	**TC**	**SPS**	**SPS**	**TC**	**SPS**	**TC**
MAFFT	**85.1**	**60.4**	49.2	84.3	**60.3**	72.9	**60.4**
ProbCons	84.5	57.9	50.1	**84.8**	60.0	**73.1**	59.0
Opal	84.3	58.2	**50.2**	84.6	58.5	**73.1**	58.4
T-Coffee	80.1	54.3	46.7	81.4	55.0	69.4	54.7
Muscle	80.2	52.1	45.6	81.2	55.5	69.0	53.8
CLUSTAL W	73.2	41.6	44.0	74.5	44.3	63.9	43.0

Table 1.3 Accuracy of multiple sequence alignment tools. For each tool and benchmark set the average accuracy across all the alignments used from that benchmark set is shown. The SPS score is the percentage of pairs of aligned position from the reference alignment that are correctly reproduced. The TC score is the percentage of columns from the reference alignment that were completely recovered. This table is adapted from Table 1, Wheeler and Kececioglu (2007).

than T-Coffee (Edgar, 2004). Similarly MAFFT obtains greater accuracy than T-Coffee and is also considerably faster (Katoh, et al., 2002)

1.3. Identification of Conserved and Functional Residues

1.3.1. Conservation Scores

Multiple sequence alignments make it possible to probe the relationships and similarities between the sequences in homologous families of proteins. The sequences are all related but have diverged during evolution. However, even widely divergent protein families exhibit patterns of conservation of amino acid positions in an alignment. It is generally considered that residues important for the structure and/or function of a protein are under greater evolutionary constraints and therefore less likely to mutate. This makes the multiple sequence alignment a very useful resource for identifying such residues, as it identifies equivalent residues throughout a protein family.

Numerous methods exist for the identification of conserved positions within multiple sequence alignments. An overview is provided here and a thorough review has been produced by Valdar (2002). The simplest conservation scores do not consider the composition of the multiple sequence alignment or the substitution of similar amino acids, while more complex scores take these factors and others into account. Simple approaches use the frequency of the most common amino acid at a position to calculate the position's conservation (Wu and Kabat, 1970). Alternatively, other methods use Shannon's information theoretic entropy S, defined as

$$S = -\sum_{i}^{k} p_i \log_2 p_i \qquad 1.8$$

where k is the number of different amino acid types, and p_i is the fractional frequency of type i (i.e. the fraction of the sequences in the alignment that type i occurs in that position). The smaller the entropy, the less diverse a position is, indicating greater conservation.

More recent conservation scores consider the substitution of similar amino acids, by using substitution matrices such as BLOSUM. This is based on the concept that where the property of an amino acid is important, mutation to a different amino acid with similar properties is more favourable than to one with different properties. A further feature is the weighting of sequences in the alignment to account for redundancy. This is important when alignments contain a group of very similar sequences, as if weighting is not used, the conservation scores can be influenced. Weighting the contribution each sequence makes to the conservation scores ensures that such groups of sequences are not overrepresented. Scores that use substitution matrices tend to have a general format as shown in equation 1.9. The conservation C of a position x in an alignment of N sequence is

$$C_x = \lambda \sum_{i}^{N} \sum_{j>i}^{N} m(S_i(x), S_j(x)) \qquad \textbf{1.9}$$

where λ is a normalisation factor and $m(S_i(x),S_j(x))$ is the substitution matrix score for the residue at position x in sequences i and j. If using sequence weighting this equation becomes

$$C_x = \lambda \sum_{i}^{N} \sum_{j>i}^{N} w_i w_j m(S_i(x), S_j(x)) \qquad \textbf{1.10}$$

where w_i and w_j are the weights of sequences i and j respectively. A simple weighting proposed by Vingron and Argos (1989) calculates the weight w of sequence i, in an alignment of N sequences as

$$w_i = \frac{1}{N-1} \sum_{j \neq i}^{N} d(S_i, S_j) \qquad \textbf{1.11}$$

where $d(S_i,S_j)$ is the distance of sequence i from sequence j. The sequence identity is often used as the distance between two sequences but other more complex measures have been derived. For example, Valdar and Thornton (2001) calculate the distance between two sequences using the sum of their scores from a modified substitution matrix over all their aligned positions ($Aligned_{ij}$) as

$$d(S_i, S_j) = 1 - \frac{\sum_{x \in Aligned_{ij}} m(S_i(x), S_j(x))}{n(Aligned_{ij})} \qquad \textbf{1.12}$$

1.3.2. Identifying Functional Residues

The conservation scores described in the previous section may identify positions that are important for functional and or structural roles. However, merely identifying the conservation does not indicate the role that such residues may play. When interested in identifying the functional residues, other steps are often necessary. Many methods have been developed to predict functional residues and they have recently been reviewed by Pazos and Bang (2006). Some widely used methods are described below.

A standard approach for predicting functional residues is to split a protein family into subfamilies. In many cases, residues conserved only within subfamilies are thought to have a role determining functional specificity. Phylogenetics (Felsenstein, 2004) is often used by functional residue prediction methods to split a protein family into subfamilies. The combination of the phylogenetic tree, indicating the evolutionary history of the protein family, and its sequence alignment can distinguish residues that are conserved for functional rather than structural reasons. The Evolutionary Trace (ET; Lichtarge, et al., 1996) approach uses a phylogenetic tree to split a multiple sequence alignment into functional groups, at specific levels of sequence identity. A consensus sequence is generated for each functional group. Invariant residues in a functional group are retained in the consensus, while those positions with variable amino acids are considered to be neutral and are represented by an underscore (Figure 1.5). The consensus sequences are combined to generate the evolutionary trace, which identifies if a residue is neutral, conserved throughout the protein family, or if it is class specific (Figure 1.5). The trace is then mapped onto protein structure, with the aim that functional residues will cluster and form a site of conserved residues. The sequence identity cut off used to split the sequences into functional classes can have a significant effect upon the ET results. Therefore, at high cut offs, the sequences in the functional classes have high levels of conservation and the resulting traces are noisy, with many false predictions. At the other extreme, using low sequence identity cut offs, very few positions are conserved and the resulting trace may miss many of the functional residues. Initially this signal to noise problem had to be managed manually, but more recent variations of ET have automated the method by using clustering algorithms to assess the clustering of the trace residues (Aloy, et al., 2001; Yao, et al., 2003).

Modifications to ET include the use of substitution matrices to allow for conservative amino acid changes (Landgraf, et al., 1999) and ranking of trace residues (Lichtarge, et al., 2003). Ranking indicates the number of functional classes where an alignment position is not invariant. High ranking positions vary in many of the functional classes and are therefore unlikely to have important functional roles. Those with low rankings are invariant in most classes and therefore likely to have more important functional or structural roles. Evolutionary trace has also been used to infer protein function (Kristensen, et al., 2006), by using it to identify functionally important residues, which are then used to generate a 3-D motif to search query structures (see section 1.5.2.2).

ConSeq (Berezin, et al., 2004) offers a purely sequence based ET approach. It uses the Rate4Site (Pupko, et al., 2002) algorithm, which calculates the evolutionary rate of positions in a multiple sequence alignment, in a similar way to ET. In the absence of a representative structure to map the data onto, ConSeq uses neural networks to predict solvent accessibility and only classes residues that have a low evolutionary rate and are accessible to solvent as functional.

Rate4Site has recently been used by Chakrabarti, et al. (2007) to identify residues that are responsible for different functional specificities between protein subfamilies. Their SPEER (specificity prediction using amino acid properties, entropy and evolution rate) scoring system combines evolutionary rate with two other scores. The first of these calculates a distance based on amino acid physico-chemical properties to express the amino acid properties within and between subfamilies. Combined relative entropy is also calculated for each pair of subfamilies, to further assess the distribution of amino acid types between them. The resulting SPEER score was shown to obtain greater levels of sensitivity compared to each of the scores used individually.

Alfonso Valencia's group have developed numerous methods for the identification of functional residues. Some of their approaches relate to evolutionary trace, as they use phylogenetic trees to split a family of proteins into subfamilies and use the subfamilies to identify functional residues. The details of their approach differ from ET. For example, they use relative entropy and also consider the mutation rate of individual

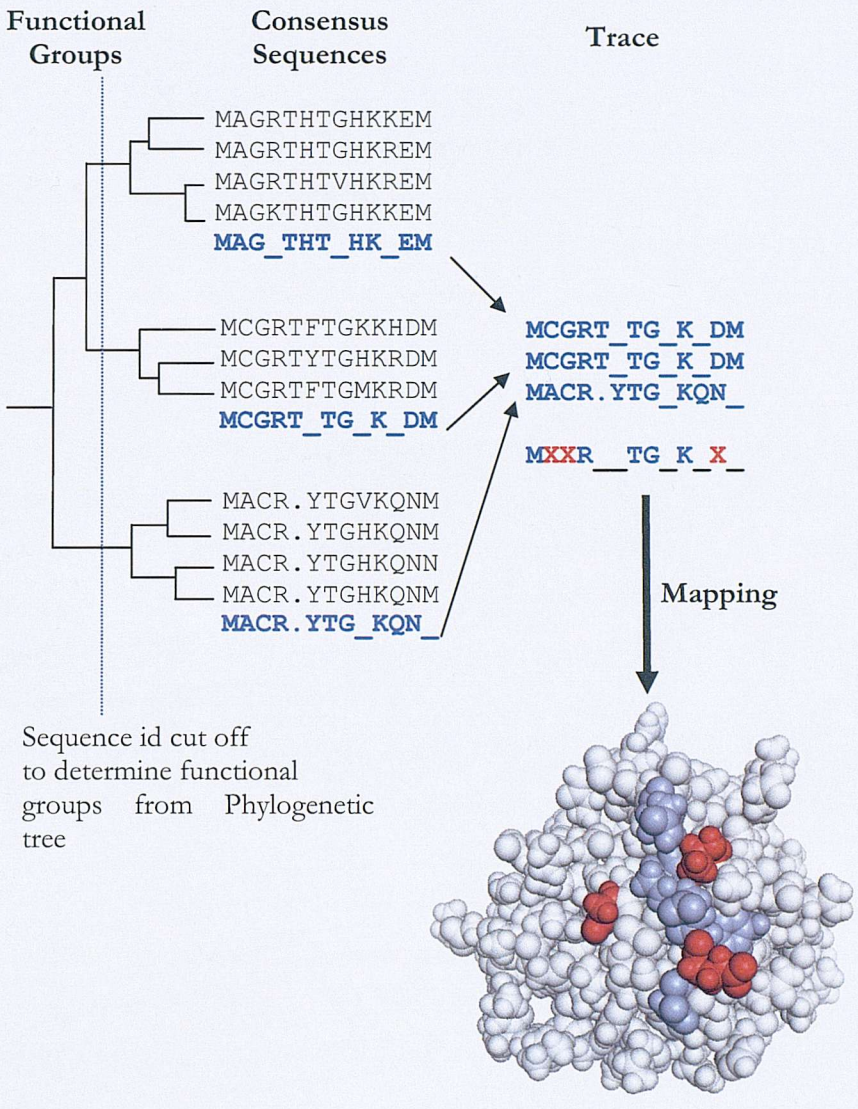


Figure 1.5 Evolutionary Trace. A phylogenetic tree is calculated for a set of homologous sequences. A sequence identity cut off is used to split the sequences in the tree into functional groups. Consensus sequences are obtained for each of the functional groups. Positions that are fully conserved in a functional group are represented in the consensus sequence by the amino acid at that position. Other positions are represented as an underscore. The trace is formed by comparing the consensus sequences. Positions that are conserved throughout the whole set of proteins are represented by the amino acid at that position. Other positions that are constant within each functional group but vary between functional groups are represented by an X. An underscore is placed at all other positions. The trace is mapped onto a representative protein structure. Adapted from Lichtarge et al. (1996) Figure 2.

positions in a multiple sequence alignment compared to the rate for the whole sequence (del Sol Mesa, et al., 2003).

The phylogeny methods described above rely on an assumption that functional differences are represented within the phylogenetic trees that are used to create subfamilies. The purpose of using subfamilies is generally that they should each be associated with different functions (most often different functional specificity). If the functional difference of interest is not split effectively by a phylogenetic tree then it may not be possible to use such an approach to infer functional residues. The Xdet method (Pazos, et al., 2006) is a phylogeny independent approach that can use functional classifications that are not represented by a phylogenetic tree. Xdet constructs two matrices, one describing the amino acid differences between each pair of proteins in a family at each position in a multiple sequence alignment. The second matrix is constructed using a hierarchical functional classification to express the similarity in function between each pair of proteins. Functional residues are inferred by considering the correlation between the two matrices for each position in the protein family. Highly correlated positions are inferred to be associated with determining functional specificity.

Other approaches that do not rely upon phylogenetics include firestar (Lopez, et al., 2007a) and conserved functional group analysis (CFG; Innis, et al., 2004). Firestar uses FireDB (Lopez, et al., 2007b), a database of functionally important residues from proteins of known structure, to predict functional residues. PSI-BLAST is run for query sequences against the sequences in FireDB and pairwise or multiple sequence alignments are generated for the identified hits. SQUARE (Tress, et al., 2004) is used to identify reliably aligned regions of the pairwise alignments. Functional residues are transferred where they occur in such regions.

CFG analysis identifies homologues of a protein of known structure and identifies patterns of conservation of amino acid residues that have similar functional groups in their side chains. Conserved areas are mapped onto the protein structure and functional residues inferred where the conserved positions cluster on the protein structure.

Wangikar, et al. (2003) use graph theory to identify functional sites. Families of protein structures are represented as graphs and functional sites are identified by searching families of structures for the occurrence of common side chain patterns.

Some approaches consider the prediction of catalytic residues and enzyme active sites. Gutteridge, et al. (2003) used neural networks that consider parameters including conservation, solvent accessibility, secondary structure and the location and depth of residues in a cleft, to identify potential enzyme catalytic residues. Enzyme active sites are predicted for sites on the protein structure where the potential catalytic residues cluster.

1.4. Relationships between Sequence, Structure and Function

1.4.1. The Relationship between Protein Sequence and Structure

Anfinsen first demonstrated that a protein amino acid sequence contains sufficient information for protein folding, by the denaturation and refolding of ribonuclease (Anfinsen and Haber, 1961; Sela, et al., 1957). Chothia and Lesk (1986) assessed the relationship between the similarity of proteins sequences and their structures. They

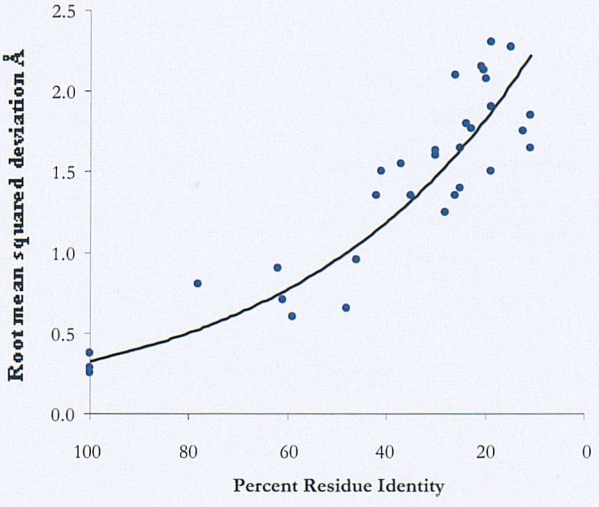


Figure 1.6 The relationship between protein sequence and structure. The RMSD of pairs of protein structures is plotted against their sequence identity. Adapted from Chothia and Lesk (1986) Figure 2.

compared the structures of homologous sequences and identified a correlation between increasing sequence identity and conservation of protein structure. Figure 1.6 provides a schematic representation of their results. The success of homology modelling methods for protein structure prediction demonstrates the ability to use this relationship between similar structures to identify protein structure.

1.4.2. The Relationship between Protein Sequence and Function

1.4.2.1. Single Domain Analyses

Numerous studies have considered the relationship between sequence and function. Hegyi & Gerstein (1999) initially assessed this relationship for a set of single domain Swiss-Prot sequences, each with high sequence similarity to a SCOP domain (enabling them to also assess the relationship between structure and function, see section 1.4.3). They considered the number of domains that have sequence similarity to proteins with different enzymatic functions (different three digit E.C. numbers). They used BLAST e-values (see section 1.2) as a measure of similarity between the sequences and plotted the percentage of domains that have matches to multiple biological functions as a function of the e-value. Their results are reproduced in Figure 1.7 and show that the more similar two sequences are the more likely they are to share the same function. At an e-value of $1x10^{-60}$ there is a less than two percent chance that two proteins will have different functions and even at e-values of $1x10^{-10}$ there is only a five percent chance that their functions will differ. At higher e-values, between $1x10^{-5}$-$1x10^{-1}$, the chance that two homologues have different functions increases from six percent to ten percent. Hegyi and Gerstein (2001) later performed a similar analysis, using a different dataset, they observed a greater percentage of domains with multiple functions (Figure 1.7). These differences are likely to be due to the different versions of Swiss-Prot and SCOP used.

Devos and Valencia (2000) surveyed four functional features: conservation of E.C. number, Swiss-Prot keywords, cellular function class and amino acid conservation in binding sites, for a set of structurally aligned sequences obtained from the FSSP database (Holm and Sander, 1994). The conservation of Swiss-Prot keywords and E.C. functions are of most interest to this study and will be discussed further. All four digits of the E.C.

number are conserved above 80% sequence identity, and between 30-80% identity there is a tendency for the first three E.C. digits to be conserved, with stronger conservation above 50% identity. Their observations are reproduced in Figure 1.8 along with the results from the other studies described in this section. Although these observations are similar in nature to those of Hegyi & Gerstein (1999; 2001), a direct comparison cannot be made because they used different ways of expressing sequence similarity, with Devos and Valencia using sequence identity, in contrast to Hegyi and Gerstein's BLAST e-values.

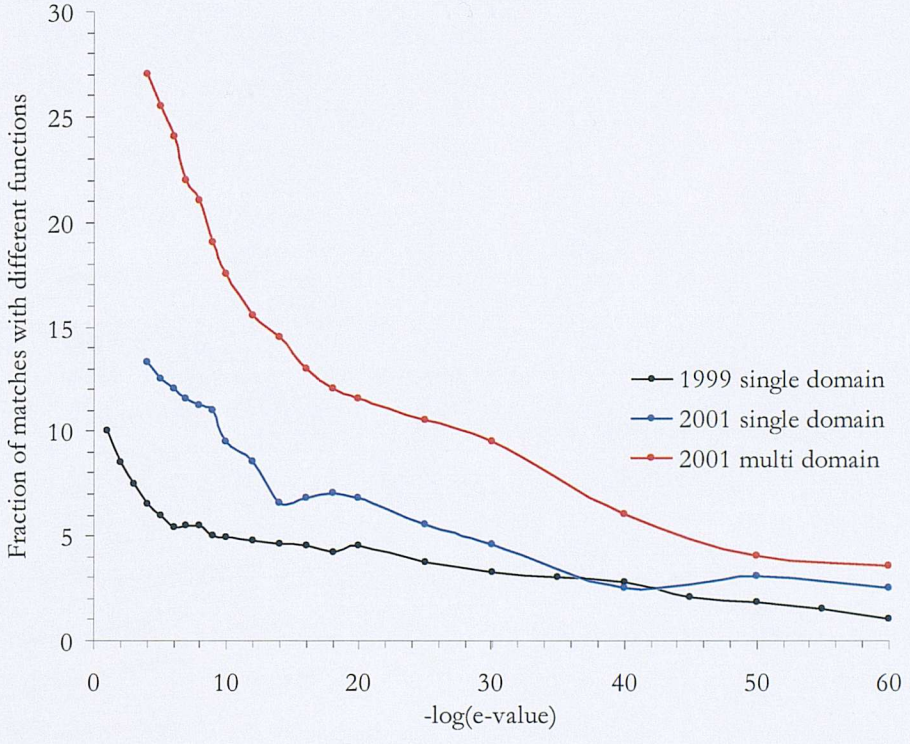


Figure 1.7 The relationship between protein function and sequence similarity e-values. Reproduced from Figure 7, Hegyi and Gerstein (1999) and Figure 4 Hegyi & Gerstein (2001). Values are extracted from these figures so they may have error associated with them.

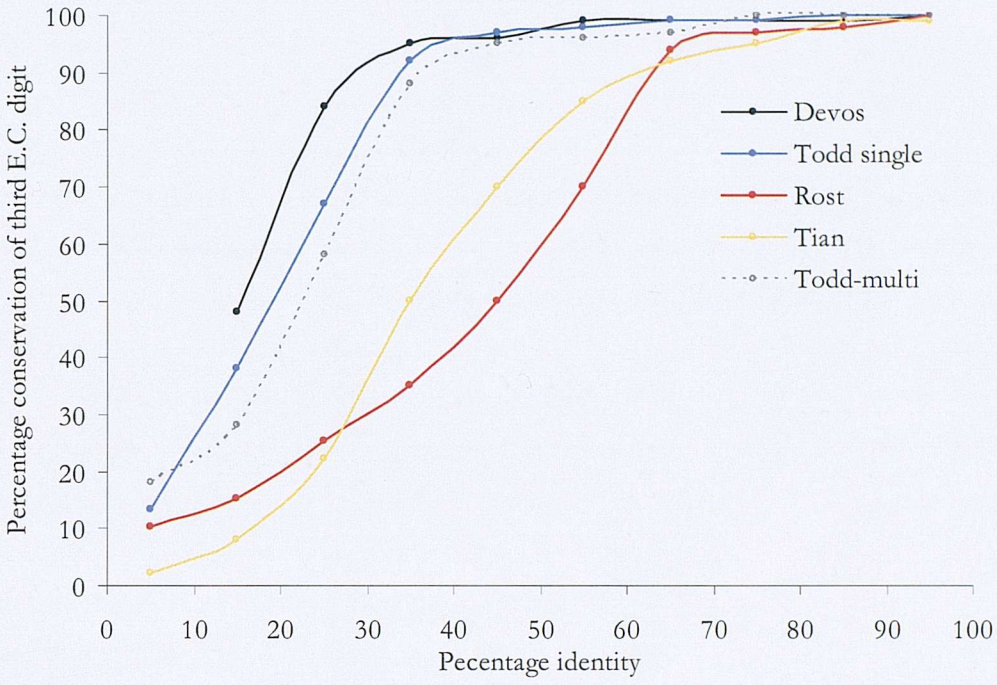
Devos and Valencia also observed poor conservation of Swiss-Prot keywords between sequences such that at 40% sequence identity a pair of proteins is only likely to share 70% of keywords. This highlights the problems associated with the use of free text to describe function.

Wilson, et al. (2000) performed a similar analysis comparing the function of pairs of protein domains from SCOP, with all pairs of proteins sharing at least the same fold. Their analysis combined the E.C. classification and the FLY database drosophila classification (Ashburner and Drysdale, 1994) to enable the comparison of both enzymes and non-enzymes. They identified that precise function (defined as first three E.C. digits), is highly conserved down to 30% sequence identity, and general function (first E.C. digit) is further conserved down to 20% identity. Below 20% identity, paired proteins often perform completely unrelated functions and in some cases enzymes are even matched with non-enzymes. They also observed less conservation of non-enzymatic function, with precise function only conserved down to 40% sequence identity.

1.4.2.2. Multi-domain Analyses

The previous analyses have only considered single domain proteins. Many proteins contain multiple domains, so understanding the conservation of function between multi-domain proteins is important. This has been studied by Hegyi and Gerstein (2001) and Todd et al. (2001). Hegyi & Gerstein used a similar approach to their previous study (1999, described above) studying the ability to transfer function between pairs of multi-domain proteins as a function of sequence similarity. They also compared this to the transfer of function between single domain proteins and identified that functional conservation is more than two fold less for multi-domain proteins than for single domain proteins (Figure 1.7). The only exception to this being multi-domain proteins that have exactly the same domain combinations and in such cases the probability of them sharing function is 80%, compared to 35% when the domains differ.

The presence of multiple domains in a protein sequence complicates our ability to transfer functional annotations as the function of a protein is determined by the combination of domains with the possibility of each domain performing a different function when present in a multi-domain protein than individually. This is demonstrated by the observation that of the 455 SCOP superfamilies present in the study only 70 are present in both single and multi-domain proteins. Interestingly only a small number (14) of these 70 superfamilies perform the same function in single and multi-domain proteins (Hegyi and Gerstein, 2001). Thus, although two proteins may share sequence similarity,



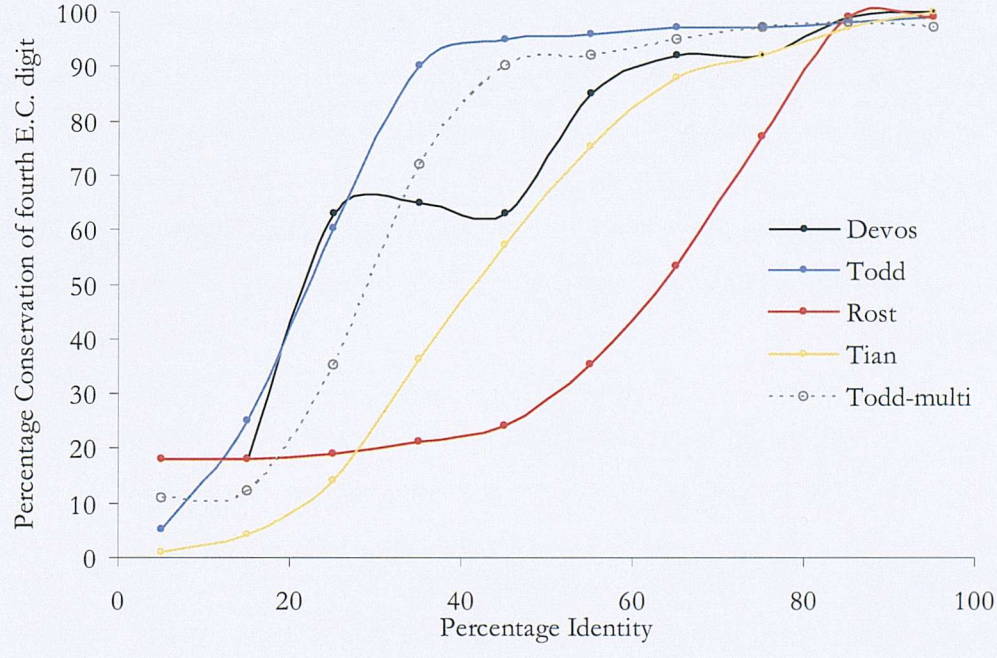


Figure 1.8 The relationship between protein function and sequence identity. The data shown are extracted from the results from Devos and Valencia (2000), Wilson et al. (2000), Rost (2002), Tian and Skolnick (2003), and Todd et al. (2001). Each graph legend shows the first author for the paper that each set of results is from. All results are for single domain proteins, with the exception of Todd-multi which considers a set containing both single and multi-domain proteins.

we must remain aware that they may have additional domains, not present in both sequences. Functional annotations may be associated with such domains, leading to incorrect annotation transfer.

Todd et al. (2001) considered functional conservation by comparing the function of enzymes from homologous superfamilies in CATH. For single domain proteins their results are in agreement with Wilson et al., with the first three E.C. digits being highly conserved (same in > 90% of pairs) above 30% identity. Their results further suggest that variation in the fourth digit is low, with more than 90% of pairs sharing exactly the same E.C. number above 40% sequence identity. A rapid decrease in conservation is observed below 30% identity (Figure 1.8). When multi-domain proteins are also considered, the first three E.C. digits are only conserved above 70% sequence identity. This emphasises the importance of different domain combinations in determining function and is in agreement with Hegyi & Gerstein (2001).

1.4.2.3. Analyses Accounting for Sequence Database Biases

Based on the analyses of the previous studies, 40% sequence identity became considered as a general threshold for confidence in functional transfer. However there is not complete agreement about the relationship between sequence similarity and functional conservation. Rost (2002) proposed that current sequence databases such as Swiss-Prot contain biases due to the presence of a few, well studied protein families with many sequences. Further he claimed that some previous studies had not used representative samples of proteins because the sequences were known to have similar structures. To reduce any biases present he split proteins into families based upon sequence identity and used a representative set of sequences from each family for comparison. Rost's results do not agree with the previous results (Figure 1.8), particularly at lower levels of sequence identity. In this study fewer than 30% of sequence pairs with greater than 50% identity shared the same four digit E.C. number and below 70% identity, enzyme function rapidly diverges. This is in clear contrast to the previous studies, which saw greater conservation of function with sequence similarity.

The results also suggest that pairwise sequence identity is not a good measure of functional similarity and BLAST/PSI-BLAST e-values and an HSSP-distance (Dodge, et al., 1998) all discriminate better between functional similarity and non-similarity. Further, the results show that even close PSI-BLAST hits with e-values less than $1x10^{-50}$ may not have identical E.C. numbers with only 86% percent of such hits sharing identical function. This contrasts with the 4% of multi-domain pairs that exhibited different functions at e-values of $1x10^{-50}$ in the Hegyi & Gerstein analysis (2001). Rost does not offer any further details on this result. It seems most likely that such hits probably differ in only the fourth E.C. digit (i.e. different substrate specificity) or have domains that are not shared by the paired proteins, which results in them having different function, as was seen by Hegyi and Gerstein (2001).

Tian & Skolnick (2003) extended Rost's approach by splitting their protein test set into groups by both sequence identity and function (either same 3 digit E.C. or same 4 digit E.C. number). They calculated the functional conservation of each family and averaged the values. This method shows that 60% identity is required to have confidence in the transfer of the fourth E.C. digit, which is less than the 70% identified by Todd et al. and Wilson et al. but is considerably greater than the level of identity suggested by Rost. However, Tian and Skolnick also observe that conservation of the first three digits remains high with 93% of sequence pairs having the same annotation in the range of 40-50% sequence identity. This supports the observation that function can be effectively transferred between sequences as low as 40% sequence identity.

1.4.2.4. Sequence-Function Relationship Summary

The studies predominantly compared enzyme classification using E.C. numbers, although the analyses by Hegyi & Gerstein and Wilson et al. incorporated some non-enzymatic functions. Each analysis adopted a different approach for the study of functional conservation and they used sequences from different sources and had different schemes for selecting the sequences included in the assessment. This makes it difficult to compare the results obtained. However, the results have been plotted in Figure 1.7, where functional conservation is assessed as a function of sequence similarity using e-values and in Figure 1.8 for the relationship between functional conservation and sequence identity.

From Figure 1.7 a definite trend is observed between sequence similarity (assessed as e-values) and conservation of function. This trend is present for both single and multi-domain proteins, with much greater conservation of function for single domain proteins. It is important to note that even for small e-values ($1x10^{-60}$), it is possible that proteins will have different functions. Figure 1.8 suggests that the relationship between sequence identity and conservation of function is less well established with varied results from the different studies. There is very high conservation (90% or more) of three digit E.C. numbers above 65-70% sequence identity in all studies and also for four digit E.C. numbers above 85-90% sequence identity. Below these levels of sequence identity the relationship between sequence identity and function is unclear, with large variations in the results obtained. The earlier studies (Devos and Valencia, 2000; Todd, et al., 2001) suggest strong conservation of both three and four digit E.C. numbers to much lower levels of sequence identity. It is possible that this is a consequence of their use of sequences with known structures. This may be particularly true for the Todd study as only protein pairs belonging to the same CATH homologous superfamily were considered. There are likely to be fewer different functions for proteins grouped in this way. Rost and Tian and Skolnick have attempted to ensure a fairer representation of proteins in their analyses and their results suggest much lower conservation of both three and four digit E.C. functions below 65% and 85% identity respectively. However they differ in their approach for obtaining better representation of protein families and thus obtain considerably different results.

Combining all the analyses it is only possible to extract that at very high identity (> 85%) proteins will probably have the same function, while at low identity (< 30%) they are more likely to have different functions. However, this is just a general trend and there will be exceptions to these guidelines. In between these levels it is impossible to gauge if functions will be the same or different. These observations contrast with the well defined relationship between protein sequence and structure (as described in section 1.4.1), however this seems reasonable considering the differences between protein structure and function. Each of the residues in a protein sequence has an effect upon its structure, and while the protein structure is important for its function, protein function is often determined by a small fraction of the residues in a protein. An enzyme active site, for example, and therefore, changes to many of the residues in a protein may not affect

function, but differences in those associated with function can drastically modify it. This is demonstrated by two different scenarios: enzymes sharing nearly identical sequences but catalysing different reactions and non-homologous enzymes, with no sequence identity performing exactly the same reaction.

Melamine Deaminase and Atrazine Chlorohydrolase (E.C. 3.8.1.8), provide an excellent example of this first scenario. They have 98% sequence identity and yet they catalyse different reactions (Babbitt, 2003; Seffernick, et al., 2001) with no catalysis of the alternate reactions present in either enzyme. Their sequences are the same length and only nine residues differ between them, demonstrating the importance of individual residues in determining protein function as opposed to overall sequence and structure.

The second scenario of proteins performing the same reaction with little or no sequence similarity is exemplified by the serine proteases (Matthews, et al., 1967). Serine proteases exhibit convergent evolution with unrelated structures (often with different SCOP folds) containing the same Ser-His-Asp catalytic residues and performing the same reaction. A collaboration that I have been part of (reported in chapter 2), with a group from Rome, has recently identified convergent evolution of enzyme active sites in approximately 15% of the three digit E.C. numbers considered (Gherardini, et al., 2007). This demonstrates the importance of active site residues in determining protein function.

1.4.3. The Relationship between Protein Structure and Function

It is difficult to separate the relationship between protein sequence, structure and function, as protein sequence and structure are themselves related and each are related to protein function. As a result, the previous section introduced the concept that a small number of critical residues are often important for protein function, with overall structure having a smaller influence over function. However, the overall protein structure is essential to ensure that the functional residues are present in the required orientations to be functionally active. For example mutations that disrupt protein folding will generally also destroy its function even if the residues critical for function have not been altered.

Martin et al. (1998) performed one of the first large scale analyses of protein structure and function. They surveyed the distribution of the principal E.C. enzyme classes (i.e. first E.C. digit) with the CATH structural classification. They used a single representative structure for each homologous family in the Protein Data Bank (PDB) and found no evidence of bias for enzyme class to particular CATH classes. They did identify a strong correlation between protein fold and ligand (both enzyme and non-enzyme). For example, Haem binding proteins are more likely to belong to the mainly α class or have mixed $\alpha\beta$ structures.

Later work by Hegyi & Gerstein (1999) also investigated the relationship between protein fold and function. Their approach was different as they did not only use protein structures but also assigned SCOP folds to Swiss-Prot sequences by using SCOP domains as queries for BLAST run against Swiss-Prot. This contrasts with Martin et al. who only used protein structures present in CATH and further only considered a single structure for each set of homologous proteins. The sequence based relationships of the Hegyi & Gerstein study were described in the previous section. Using single domain proteins they compared SCOP fold with E.C. number, effectively comparing protein general structure with enzymatic activity. Only 1.6% of all the possible fold and function combinations occurred in the data set and their results identified a correlation between protein fold and general function. For instance small folds and all-α folds are more likely to be associated with non-enzymes, while α/β folds are often associated with enzymes (particularly transferases and hydrolases). $\alpha+\beta$ folds are equally associated with enzymes and non-enzymes. Later work by Hegyi & Gerstein (2001) identified the same associations between fold and function in multi-domain proteins but with less pronounced differences between the occurrence of enzymes and non-enzymes in α/β folds. While these results show some correlation between structure and function, there is not a definitive rule linking protein fold and function as more than half the functions in the protein set were found in at least two protein folds.

More recent work (George, et al., 2004) mapped SCOP domains and the E.C. Classification. The resulting database, SCOPEC, contains catalytic domains for 731 different E.C. numbers, which is approximately twenty percent of all the complete 4 digit E.C. numbers in the ENZYME (Bairoch, 2000) database. Their analysis assessed the

relationship between function and structure at different levels of both the E.C. and SCOP classifications. Considering the E.C. to SCOP ratio gives an indication of functional divergence. At the SCOP family level, there are 1.7 four digit E.C. numbers per family. Similar numbers of E.C. terms per SCOP family are observed at the three, two and single digit E.C. levels. Structures present at the SCOP family level share a clear evolutionary relationship and so this result is not unexpected as it suggests that evolutionary related sequences are likely to share a common function. The coverage of E.C. numbers is low (731 4 digit E.C. numbers compared to a total of 3758 in the ENZYME database), which might also reduce the number of E.C. numbers identified per SCOP family.

For SCOP superfamilies there is an average of 3.0 four digit E.C. numbers per superfamily and 1.58 at the three digit E.C. level. The authors propose that many superfamilies may then perform a single enzymatic function but have members with different substrate specificity. Structures at the superfamily level are likely to have a common evolutionary origin and so this might suggest that more distant homologous enzymes (i.e. belonging to the different families but the same superfamily) are likely to share function but differ in substrate specificity. The comparison of SCOP class and general E.C. functional class (i.e. first E.C. digit) can be compared with Hegyi & Gerstein's (1999 & 2001) findings. At these levels there is an average of 5.6 E.C. numbers per SCOP class, and as there are only 6 E.C. general classes this suggests that each type of enzyme is present in most SCOP classes. There is no information for the frequency of each enzymatic class in each SCOP class, so it is not possible to know if there is a similar bias for functional classes in different SCOP classes.

The ratio of SCOP superfamilies to E.C. number gives an indication of the frequency of convergent evolution of enzyme function. Sixty nine percent of three digit E.C. numbers were observed to belong to multiple superfamilies, suggesting that convergent evolution is widespread among enzymes, if proteins belonging to different superfamilies are considered to be evolutionarily unrelated.

Enzymatic function and structure was thoroughly investigated by Todd et al. (2001). Their analysis of 31 homologous superfamilies in CATH shows the following relationships:

- Wide variation of substrate specificity within superfamilies. Some of these specificity differences are due to the presence of regulatory domains which allow different substrates to access the active site.
- Greater conservation of reaction chemistry, with some superfamilies completely conserving the overall reaction strategy while many others 'semi-conserve' i.e. one or more of the reaction steps are conserved.
- Active sites often have similar frameworks although in some cases active residues are located in different positions in the active site and even on different chains.
- The difference between enzyme and non-enzyme homologues is often the absence of a catalytic residue in the non-enzyme. Inaccessibility to the active site can also be a factor and will cause inactivity even if catalytic residues are present.

These findings demonstrate the complexity of the relationship between protein structure and function. While similar protein structures often conserve function this relationship is complicated by the roles of a small number of crucial residues. However the presence or absence of critical residues does not always infer conservation of function, which can be further affected by other factors including the presence of regulatory domains.

1.5. Protein Function Prediction Approaches

The complex relationship between protein sequence, structure and function often makes the inference of function using sequence and structure a difficult task. The simplest sequence methods utilise sequence similarity but to obtain accurate predictions they need to adopt more sophisticated approaches than transferring the annotation of the closest homologue. Such approaches include the use of sequence motifs, phylogenomics and clustering. State of the art sequence methods are described in the following sections. Greater emphasis is given to them as this thesis focuses on the development of a sequence based function prediction method. Approaches that use structural data are also important and are described in section 1.5.2. Other methods use different sources of data such as gene expression levels, genomic context, post-translational modifications and

interaction data. Alternative techniques for inferring function including machine learning and data mining.

1.5.1. Sequence Based Methods

Sequence based function prediction methods have an important role due to the overwhelming number of protein sequences present in databases compared to the number of protein structures in the PDB. This ratio will continue to increase as more genomes are sequenced and data from metagenomics projects such as the Sorcerer II project (Yooseph, et al., 2007) is added. Sequence based function prediction methods range from simple annotation transfer, to more complex algorithms that consider the annotations of many sequences. Sequence methods are often homology based. Such methods tend to share two main steps:

- Homologue identification
- Use of annotations present in homologues to predict function

Homologues can be identified by running BLAST (Altschul, et al., 1990), PSI-BLAST (Altschul, et al., 1997), FASTA (Pearson, 1996) or other sequence searching programs. The homologues of a sequence provide a set of possible annotations that could be assigned to it. Not all of these terms will be correct and it is essential for algorithms to discriminate between correct and incorrect functions. Current sequence based methods are described below. Some details of the performance of the methods are provided but a more thorough discussion and comparison of performance is provided in section 1.7.

1.5.1.1. Annotation Transfer – Top Blast

The simplest function prediction method is the direct transfer of annotation to the query sequence from its top BLAST or PSI-BLAST hit. The complexities of the sequence-function relationship (see section 0) make this method unreliable especially at lower levels of sequence identity, where homologues are likely to have a range of different functions.

1.5.1.2. GeneQuiz

GeneQuiz (Andrade, et al., 1999) uses a more sophisticated approach to annotation transfer from a homologue. It does not automatically transfer the annotation from the best hit, but uses rules to generate a reliability score for each homologue. The reliability score is based upon sequence identity, the annotation source, the sequence alignment method used and a lexical analysis of keywords associated with the homologues. These rules are applied to identify the most suitable homologue to transfer annotation to the query sequence.

1.5.1.3. GO Engine & FunCut

GO Engine (Xie, et al., 2002) and FunCut (Abascal and Valencia, 2003) are examples of methods that have built upon the simple process of annotation transfer. GO Engine uses homology, protein domain analysis and text mining to predict GO annotations. The method is based upon homology transfer similar to that of transferring the annotation of a top BLAST hit, but performs single linkage clustering of homologous sequences. The predicted annotations from the clustering are used in conjunction with GO annotations predicted by text mining. Genbank and Swiss-Prot entries along with reference articles are analysed and predictive words identified. A probabilistic method is applied to map these terms to predicted GO terms. Final predictions are made by applying an error weighting to the terms identified by the clustering and text analysis. In benchmarking, GO Engine obtains an accuracy of 78% for molecular function annotations.

The FunCut method also utilises clustering of proteins (Abascal and Valencia, 2003). It uses BLAST to identify sequences similar to a query sequence and then recursively runs BLAST for the sequences identified from the previous BLAST search. All the sequences identified are clustered using a normalised cut algorithm, which results in groups of related sequences (considered to be subfamilies). The functional annotations present in the query sequence cluster are investigated and those that are most representative of the cluster are transferred to the query. Unlike Go Engine, FunCut transfers E.C. numbers, Swiss-Prot descriptions and keywords and in benchmarking obtained 94% correct functional annotations. As has been explored in section 1.4.2.2, the domain nature of proteins can affect functional annotations, to avoid such problems

FunCut groups sequences in the cluster into four categories according to how they align with the query sequence. There is greatest confidence where the full length of both sequences align and least where neither is fully aligned.

1.5.1.4. Protein Family and Motif Methods

Hits to Pfam (see section 1.1.5) are often used as a first step for inferring function as they can identify the domain or family that a protein belongs to. In some cases specific functions may be associated with the family or domain, indicating potential functions for the query sequence. Protein families are often large and while family members often share similar general functions, they can have varied specific functions (Abhiman and Sonnhammer, 2005a; Abhiman and Sonnhammer, 2005b). In these cases it may only be possible to infer general function. Mappings exist to convert hits to Pfam and other domain databases to GO functional annotations. SMART, ProDom and motif methods such as ProSite, PRINTS and BLOCKS can all be used in a similar way to identify the potential family that a protein belongs to and in turn infer function. Interpro incorporates most of these and other methods and can also infer function from the protein families identified. Interpro is widely used for considering the functional annotation of genomes with many genome annotation papers expressing functional genome coverage as the percentage of the genome that has hits to Interpro (Waterston, et al., 2002).

ProDom and CDD (a combined database mainly comprising Pfam and Smart; Marchler-Bauer, et al., 2007) have been used to assign GO molecular function to sequences by using rules to associate GO terms with the domains in their databases (Schug, et al., 2002). GO annotated proteins were associated with domains by using BLAST to identify hits between the query sequences and the ProDom or CDD domains. Rules were then used to determine which of the GO terms present in the sequences for each domain should be associated with that domain. Using non-electronic annotations this approach associated GO functions with 11% and 38% of ProDom and CDD domains respectively (increased to 18% and 52% when including electronically inferred annotations). Many of the rules were based upon a single protein being associated with a domain. Benchmarking on over 4000 GO annotated proteins from the human proteome

obtained coverage of 81% with 74% of predictions agreeing with annotations (exact matches or more/less specific accepted).

SMART enables users to query for domains that are associated with GO terms by using the associations of domains between GO from Interpro. It will return lists of domains associated with the GO term of interest and the user can select particular domains and view the proteins belonging to them.

1.5.1.5. Catalytic Site Atlas – sequence based approach

The Catalytic Site Atlas (CSA; Porter, et al., 2004) is a manually maintained database of enzyme catalytic residues from proteins of known structure. The catalytic sites are often used as motifs for searching structures to predict function (see section 1.5.2.2) but they have also been used for sequence based function prediction (George, et al., 2005). This approach uses CSA entries as PSI-BLAST query sequences. The conservation of the CSA catalytic residues in the homologues identified is then used to infer function, by transferring the function of the query only if all of the CSA residues are conserved. As the number of CSA residues is very small, the accuracy of the alignment is important, so the sequences are realigned using CLUSTAL W (Thompson, et al., 1994). If full conservation is not obtained the sequences are further aligned using the Smith-Waterman algorithm. Where there is not full conservation of catalytic residues it is possible that the homologue has either a different function (or is possibly a non-enzyme), or a similar function but with a different catalytic mechanism. Benchmarking of this method obtained 87.1% accuracy and coverage of 77.4% using 482 CSA templates, making predictions for 23,776 unique homologues.

1.5.1.6. Prediction of Functional Sub-types

Hannenhalli and Russell (2000) developed a method to predict protein sub-types, where sub-type refers to proteins from the same family but with different function. Sequences from a sequence alignment are split according to their sub-types. Hidden Markov Model profiles are generated for conserved residues present in each sub-type. These are used to predict the sub-type of proteins known to belong to a specific family but of unknown

sub-type. The method uses Pfam sequence alignments and Swiss-Prot annotations to split the alignment into sub-types. This approach obtained an average accuracy of 96% in analysis of four well characterised protein superfamilies. This method performs best for large families of proteins where there is good knowledge of functional sub-types.

1.5.1.7. Goblet, OntoBlast and GoFigure

GOblet (Groth, et al., 2004), OntoBlast (Zehetner, 2003) and GoFigure (Khan, et al., 2003) were among the first sequence based function prediction methods to use Gene Ontology annotations. GOblet uses BLAST to identify high confidence homologues (e-value $\leq 10^{-20}$) and simply lists them and their annotation in order of significance, effectively performing simple annotation transfer. Similarly OntoBlast lists GO terms by e-value of the BLAST hits they are from. GoFigure also uses BLAST to identify homologues and then assigns each GO term present a score based on the number of homologues annotated with the term and weighted according to the homologues' e-values. These methods are suited to web servers as they give users the ability to view possible annotations of their query sequence rather than assigning annotations on a genomics scale.

1.5.1.8. GOtcha

GOtcha (Martin, et al., 2004) is a more advanced approach to function prediction using GO annotations. Like GoFigure and GOblet it performs BLAST searches to identify homologues of a query. GOtcha then obtains the set of GO terms present in these sequences and assigns each of them a score based upon the sum of the logarithm of the e-values of the sequences annotated with that term. This score is normalised to enable comparison between GO terms. A confidence score is calculated using the score obtained for the root node (e.g. the molecular function GO term). The authors further calculate an empirical accuracy to indicate confidence in their predictions.

In their analysis, the results from all GO ontologies are combined, making it difficult to identify the success of predicting function from the separate ontology categories of molecular function, cellular component and biological process. In testing

GOtcha obtained a 60% improvement over assignment based on the top BLAST hit for the combination of all three ontology categories. GOtcha performance was also tested in the benchmarking of SIFTER (Engelhardt, et al., 2005, see 1.5.1.11). In this analysis it achieved accuracy close to 80% for molecular function predictions. This provides a better indication of GOtcha performance as most of the methods described here consider only molecular function.

1.5.1.9. PFP – Protein Function Prediction

The PFP (Hawkins, et al., 2006) approach is similar to GOtcha, but initially uses PSI-BLAST to identify homologues of a query sequence. PFP uses the frequency of GO terms within these homologues and their level of sequence similarity with the query to determine the predicted functions. Each predicted GO term is associated with a probability score to give an indication of its confidence. PFP supplements these predictions by considering other GO terms that are strongly associated with the GO terms predicted in the initial step. They have generated a Function Association Matrix (FAM), which scores the co-occurrence of pairs of GO terms within UniProt (The UniProt Consortium, 2007). This enables PFP to predict functions that are not identified by PSI-BLAST and has been shown to increase sensitivity without reducing specificity.

1.5.1.10. Phylogenomics Approaches

The concept of Phylogenomics was introduced by Eisen (Eisen, 1998; Eisen, et al., 1995) as a method for inferring protein molecular function. The basic Phylogenomics approach requires the generation of a phylogenetic tree for a family of proteins and the subsequent overlaying of the known functions of these proteins onto the tree. Using this information, functional changes are traced back along the tree and used to infer the function of uncharacterised proteins in the tree. Phylogenomics resolves a common problem present in sequence similarity based functional annotation by distinguishing between gene duplication and speciation events. Gene duplication results in paralogous genes within a species. One of the genes must maintain its current function to ensure that the function is retained by the species, but the second gene is free to evolve a new function. As a result these similar sequences may perform different functions (Gerlt and Babbitt, 2000).

Phylogenomics approaches have been used to perform detailed studies of individual protein families (Citerne, et al., 2003; Eisen and Hanawalt, 1999) and whole genome analyses (Eisen, et al., 2002). A recent phylogenomics approach, SIFTER (Engelhardt, et al., 2005) is described in the next section.

1.5.1.11. SIFTER

SIFTER is a Bayesian phylogenomics approach for protein function prediction. Unlike many of the sequence based GO function prediction methods, SIFTER (Engelhardt, et al., 2005) does not use BLAST to identify homologues to a query sequence. Instead it obtains multiple sequence alignments by identifying a Pfam (Finn, et al., 2006) family for a query sequence and uses the Pfam alignment. The Pfam multiple sequence alignment is used to generate a phylogenetic tree. A model of molecular function evolution is applied to the tree to estimate evolutionary duplication events. The GO functions present in the GOA database are overlaid onto the tree and the combination of these data sources is used to infer function for the uncharacterised proteins in the phylogenetic tree. The benchmarking of SIFTER is detailed in section 1.7.

1.5.1.12. CORRIE – a probabilistic approach

CORRIE (Audit, et al., 2007; Levy, et al., 2005) is a probabilistic approach for the inference of enzyme function using the E.C. classification. CORRIE initially groups a database of annotated protein sequences into functional classes by their E.C. classification. They introduce correspondence indicators, which represent the relationship between protein sequences and functional classes. They are based upon the combination of bit score between a query sequence, and the sequences belonging to a functional class. They then use the correspondence indicators in univariate and multivariate Bayesian classifiers to obtain prediction probabilities for each functional class. In brief, for each functional class, the Bayesian methods consider the ten sequences with correspondence indicators closest to the correspondence indicator for the query sequence with this functional class. For the univariate approach, the probability of this function is then calculated as the percentage of these sequences which belong to the functional class being considered. So if one of the proteins belongs to this functional class, then it is predicted

with a probability of 0.1 and if all ten belong to the functional class, the probability is 1. The multivariate approach considers the functional classes together, so the resulting probabilities of all the functional classes sum to one.

For a test set of 28088 enzymes, coverage of 96% is obtained by the multivariate classifier with a small error rate of 0.20% (Levy, et al., 2005). This is a significant improvement over the best BLAST hit approach (see 1.5.1.1), demonstrating the potential of this approach.

1.5.2. Structure Based Methods

The sequence based homology methods described above are only effective for sequences with sufficient numbers of homologous sequences to perform analysis. In recent years the number of orphan (no known homologues) sequences has increased and the number of hypothetical (those with unknown function) proteins in the PDB has also increased. While homologues may perform the same function, it is well established that non-homologous proteins may perform the same function and may even utilise exactly the same residues to do so. As 15% of three digit E.C. reactions have been shown to exhibit convergent evolution of their active sites (Gherardini, et al., 2007), it is essential that function prediction methods are not limited to sequence based approaches.

1.5.2.1. Fold Matching

Methods such as DALI (Holm and Sander, 1995) and SSM (Krissinel and Henrick, 2004) identify overall structural similarities between proteins. Using such tools it may be possible to match a query protein to a protein fold in SCOP or CATH. Function predictions can then be made on the basis of protein fold. However, studies have shown that there is limited correlation between overall protein fold and function (see section 1.4.3), so predictions made are likely to be limited to general functions. ProKnow (Pal and Eisenberg, 2005 – see section 1.5.4.2), uses DALI as a one of its source of information to make function predictions.

1.5.2.2. 3-D Motifs

3-D motifs of functional residues from protein structures are regularly used by structural function prediction methods as they represent important functional motifs present in local structural elements that are not present in sequence. Such motifs are also ideal for applying to non-homologous proteins. 3-D templates are either extracted from existing structures of known function or are automatically generated. Enzyme active sites are commonly used as templates or the CSA provides a good example of manually extracted functional residues that can be used (Laskowski, et al., 2005a). Templates are not restricted to enzyme active sites and templates of ligand and DNA binding sites have also been used (Laskowski, et al., 2005b).

Numerous approaches have been used to predict or automatically identify 3-D templates. GASPS (Polacco and Babbitt, 2006) uses a genetic algorithm to generate motifs for enzymes. PDBSiteScan combines the residues present in PDB hetatom records with interaction sites from protein complexes. Other approaches (Jambon, et al., 2003; Wangikar, et al., 2003) use graph theory to identify common structural elements. For example, SuMo (Jambon, et al., 2003) represents protein structures as graphs of chemical groups rather than different amino acids and compares triangles of chemical residues between structures. An alternative approach proposed by Laskowski et al., (2005a) uses reverse templates, which are generated by splitting the target structure into templates (3 residues in length). These are then scanned against a representative set of PDB structures to identify matches. PINTS (Russell, 1998) does not need to rely upon pre-generated motifs and can identify common substructures in two complete protein structures. It is also used with libraries of predetermined motifs.

Once 3D templates have been generated, it is necessary to search structures for occurrences of them. Templates often consist of only a few residues and so it is possible that similar arrangements will be found in other proteins simply by chance and not reflect a similarity in function. It is therefore essential to be able to assess the significance of matches. Root Mean Squared Deviation (RMSD) is used as an indication of how similar matches are but it does not discriminate effectively between true and false positive hits. PINTS applies a statistical approach by calculating e-values for local structural similarities (Stark, et al., 2003). SiteSeer (Laskowski, et al., 2005b) considers the local environments

around the template residues in the template and target structures. It calculates the number of residues that are equivalent (or overlap) between the two structures in a sphere of 10 angstroms around the centre of the template. The SiteSeer score is then calculated by considering the relative position of the paired residues in the two amino acid sequences and is converted to an e-value to make the score comparable between different templates and structures.

1.5.2.3. Phunctioner

Phunctioner (Pazos and Sternberg, 2004) was one of the first structural prediction methods to utilise GO annotations, it uses structural alignments to identify sequence motifs. Phunctioner generates a structural alignment of proteins and then splits the alignment into subalignments according to the GO annotations of the structures. Conserved residues within the subalignments are identified and those that are not conserved in the full alignment are considered to be functionally important. These residues are used to generate profiles for each GO term against which a query sequence is scored. A Z score is calculated to discriminate between the GO terms and assign them to the query. Phunctioner demonstrated a significant improvement over the top BLAST method at low sequence identity levels where sequence methods are least effective. Phunctioner demonstrates the use of structural data to improve function prediction at identity levels where homology methods fail.

1.5.2.4. Surface Clefts and Binding Pockets

Identification of ligand or substrate binding pockets can give an indication of potential function. It has been shown that most small ligands bind to the largest or second largest cleft present in a protein (Laskowski, et al., 1996). It seems likely that the size of the cleft is important for function as the largest cleft is often much larger than other clefts in the structure. Therefore analysing protein clefts can identify likely binding and catalytic sites. Approaches for identifying surface clefts often use spheres to fill the voids enabling the cleft volume to be calculated (Glaser, et al., 2006; Laskowski, 1995; Levitt and Banaszak, 1992). Clefts can then be searched for against databases of known clefts and binding

pockets such as pvSOAR (Binkowski, et al., 2003). Hits can provide links to function if the dataset cleft is associated with a known function.

1.5.3. Other Methods

1.5.3.1. Machine Learning Methods

Machine learning approaches, such as Support Vector Machines (SVM; Vapnik, 1995) and Inductive Logic programming (ILP; Muggleton, 1991) have been used to predict function. Machine learning methods generally use training sets of proteins with known functional annotations to link their attributes to their functions by identifying patterns or generating rules. Predictions are then made for query proteins by comparing their attributes to the previously generated rules.

GOPET (Vinayagam, et al., 2006) first uses BLAST to identify homologues of the query sequence and then uses sequence similarity (e.g. e-value, bit score, alignment length) and other attributes (GO term frequency, GO relationships – i.e. GO structure) as the input to SVMs to predict function. In benchmarking, at 80% precision, GOPET obtained recall of 65% and 75% for molecular function and biological process predictions respectively.

Data Mining Prediction (King, et al., 2001) partly uses sequence homology but can also predict protein function in the absence of homology. ILP is used to identify patterns in a training set. The patterns consist of multiple types of data including basic features of the protein sequence including the number of each type of amino acid present, sequence molecular weight, length and theoretical isoelectric point. These are combined with sequence similarity features obtained from PSI-BLAST, secondary structure predictions and protein expression levels. The method was initially used on the *E. coli* genome function hierarchy (King, et al., 2000) and has more recently been implemented using GO for the *Arabidopsis thaliana* genome (Clare, et al., 2006). This method is most successful at generating general functional rules. In the analysis of the *A. thaliana* genome, rules for GO level one obtain 81% precision reducing to 74% at level 4 (no predictions are made at more specific levels of the ontology).

1.5.3.2. Protein Interaction Networks

Methods that use sequence and structural similarities rely on the observation that related sequences and structures may share the same function. Function prediction methods that use protein-protein interaction networks are generally based on an assumption that proteins that interact or are closely linked in an interaction network, may share the same function. Early methods used neighbour counting approaches to infer function (Schwikowski, et al., 2000). For a protein of unknown cellular function they assigned the most common function of its neighbours in the interaction network. A recent extension of this approach considers the function of indirect neighbours (i.e. nodes separated by 2 or three edges) when inferring functions (Chua, et al., 2006). Other methods have considered different approaches (reviewed in Zhang, et al., 2007) including: chi-squared statistics (Hishigaki, et al., 2001), clustering into functional classes (Samanta and Liang, 2003), simulated annealing (Vazquez, et al., 2003) and machine learning.

1.5.3.3. Genome Function Prediction

While it has been possible to use sequence and structural data to infer function for a considerable time, the sequencing of genomes over the past decade has only recently made it possible to infer function using genomic data. Domain fusion has been used to infer function (Marcotte, et al., 1999a). This approach infers functional relationships between proteins found individually in one genome, which have homologues of proteins in other organisms where the individual proteins are fused together.

Chromosome location can also be used to infer functional relationships. The most common approach uses gene context. This is most effective in prokaryotes because their genes are often arranged into operons, with many of the genes encoding proteins in the same biochemical pathway grouped together in the genome. Therefore it is possible to infer gene/protein function by considering the function of genes encoded close to genes with uncharacterised function. Strong et al. (2003) provides an example of this approach to infer protein function in *Mycobacterium tuberculosis*.

Other approaches consider correlated evolution, mRNA expression levels and domain fusion patterns to predict protein function as demonstrated in the yeast genome (Marcotte, et al., 1999b).

1.5.4. Combined Methods

The previous sections have demonstrated the range of information that is used to predict protein function, from sequence and structure to text mining and protein interactions. These individual methods perform well individually, but they all still make errors (most methods have a maximum accuracy of approximately 80%). Therefore combining different individual methods together to make predictions may result in increased accuracy. This approach has proved successful for protein structure prediction, with the development of meta-servers (Moult, et al., 2003), which obtain predictions from multiple individual servers and then make a consensus prediction. However, it is important to realise that the combination of different methods may also reinforce errors, if the individual methods make the same incorrect predictions.

Function prediction meta-servers have been developed. Approaches for meta-servers include ProFunc (Laskowski, et al., 2005a), which performs multiple analyses but requires the user to combine them and infer function, to ProKnow, which combines the results of the different methods to give overall predictions.

1.5.4.1. ProFunc

ProFunc (Laskowski, et al., 2005a) performs a range of structural and sequence based analyses to predict protein function. Sequence methods include: PSI-BLAST, InterPro searches and residue conservation calculations. Structural approaches include protein fold matching using SSM (Krissinel and Henrick, 2004), cleft analysis, nest analysis and 3-D template searches. Most of these methods have been described in the sequence based and structure based methods sections (1.5.1 and 1.5.2). The combination of these methods provides the user with a wealth of information that can be used to infer function. It would be useful, however, if these sources of information were combined to give overall function predictions. Such a step would be necessary to make ProFunc useful for

automated genomics functional annotation pipelines. ProFunc has recently been used to infer functions for all of the structures solved by Midwest Centre for Structural Genomics over the past five years (Watson, et al., 2007). This analysis shows that the fold matching using SSM and the reverse template searching (see section 1.5.2.2) outperform the other structural methods present in ProFunc. Their results also show that no single sequence or structural method is correct in all cases, emphasising the need to combine methods.

1.5.4.2. ProKnow

ProKnow (Pal and Eisenberg, 2005) extracts protein sequence and structure features and uses them to predict function. These include 3-D fold (DALI), sequence homologues (PSI-BLAST) and motifs (PROSITE and RIGOR (Kleywegt, 1999)). ProKnow combines these methods to make predictions, by using a knowledgebase to map the features of annotated proteins to GO terms, enabling the interpretation of the results for sequences of unknown annotation. Bayes theorem is used to weight each predicted GO term and give an indication of the confidence of the prediction.

ProKnow was benchmarked using a set of 1507 proteins of known structure. Predictions were made for both molecular function and biological process GO terms and they were analysed at each GO level. Eighty five percent of predictions are correct at level one, reducing to 60% at level two, and to 40% at level nine. Ninety-three percent coverage was observed over this range and overall seventy percent of predictions were correct. These results demonstrate that ProKnow accurately predicts general function but that as functional specificity increases (i.e. GO level) the ability to predict terms is greatly reduced. It shows that multiple function prediction methods can be combined to make functional predictions and the application of Bayesian statistics enables predictions to be weighted.

1.5.4.3. Jafa

Jafa (Friedberg, et al., 2006a) was the first function prediction meta-server to submit jobs to other automated function prediction servers and collate their results. Jafa output consists of three tables, one for each of the different GO categories, which show the level

of agreement between the servers for each of the predicted terms. A simple consensus approach is taken by multiplying the level of each predicted GO term by the fraction of servers that agree for the prediction.

1.6.Assessing Function Predictions

For benchmarking it is essential to predict the function for proteins of known function and to then compare the predictions with the correct functional annotation. The use of hierarchical functional classifications such as GO or E.C. complicates this assessment, as demonstrated by the example in Figure 1.9. The query protein is annotated with a function C and method 1 predicts function **E**, a child of **C** and therefore overpredicts the function. In the opposite case method 2 predicts function **B** a parent function of **C**, underpredicting the protein function. It is important to decide how each of these predictions should be dealt with; are they both correct, both incorrect or both more or less correct?

Ouzounis and Karp (2002) initially sought to describe the numerous types of errors that are associated with function predictions. They proposed the Transitive annotation-based scale (TABS) to give an indication of the quality of predicted functions. TABS gives each prediction a score based upon a set of criteria (described in Table 1.4). The score is scaled, so that the higher the score the more serious the error is. Numerous approaches have been proposed for the assessment of Gene Ontology based function predictions, which generally use either semantic similarity or measures of precision and recall.

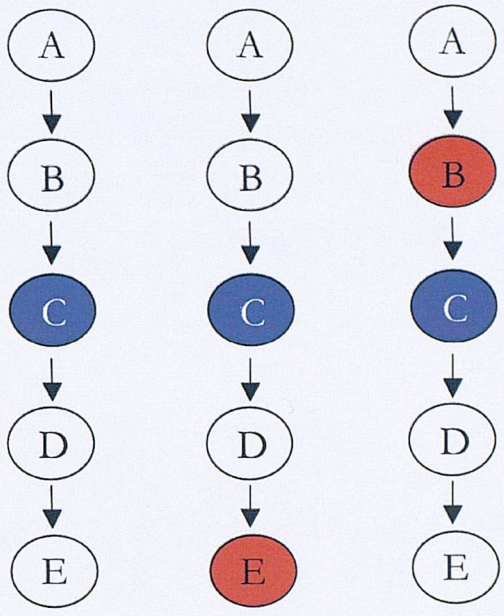


Figure 1.9 Overprediction and underprediction of protein function. The annotation of a protein is shown on the left. In the middle an example of overprediction is shown and on the right underprediction.

Score	*Description*	*Comment*
7	False positive	No supporting evidence for predicted function (i.e. does not match existing annotation).
6	Overprediction	Prediction of a specific biochemical prediction without sufficient supporting evidence (is evidence for a more general function).
5	Domain error	prediction overlooks different domain structure of query and reference proteins.
4	False negative	A function is not predicted although there is sufficient evidence to characterise the query protein.
3	Underprediction	Prediction of a nonspecific biochemical function although more detailed prediction could have been made.
2	Undefined source	Prediction contains undefined terms, non-homology based prediction etc.
1	Typographical Error	Prediction contains typographical errors that may be propagated in the database
0	Total agreement	Prediction is correct.

Table 1.4 The transitive annotation-based scale. Functional prediction categories are displayed with a description and score. Adapted from Ouzounis and Karp (2002).

1.6.1. Semantic Similarity Measures

Lord et al. (2003) first proposed the use of semantic similarity for the comparison of GO terms. Their measure is based upon the information content of GO terms, which first calculates the probability *p(c)* of each node as the fraction of times it or any of its children occur in the set of annotations (see Figure 1.10). *p(c)* is calculated using equation 1.13,

$$p(c) = \frac{n}{N} \qquad 1.13$$

where n is the number of times that the term occurs in the database of annotations being used and N is the total number of annotations in the database. Examples of the *p(c)* calculation are shown in Figure 1.10. *p(c)* increases as terms approach the root of the graph and the root term has a *p(c)* of 1.

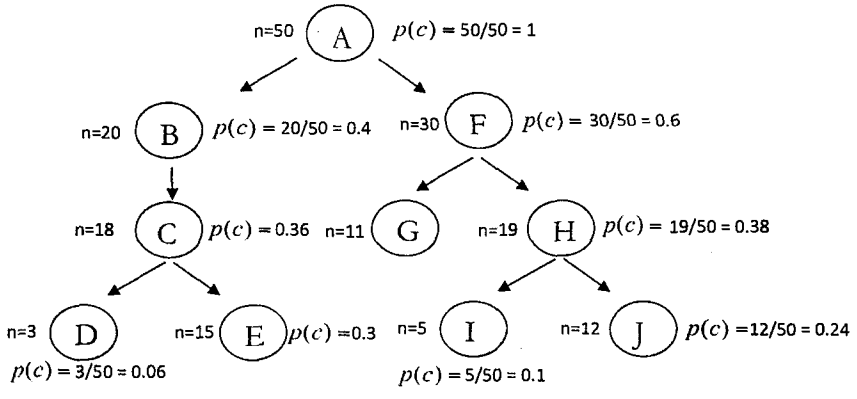
The semantic similarity *sim(c1,c2)*, between two terms c1 and c2 is calculated from the probability of their common parent. As GO is a DAG it is possible for two terms to have more than one common parent, so the parent with the smallest probability from *S(c1,c2)*, the set of common parents of *c1* and *c2* is used (this is known as the minimum subsumer). This is shown in equation 1.14, where $p_{ms}(c1,c2)$ is the parent with the smallest probability. The semantic similarity is then calculated using equation 1.15, by taking the natural logarithm of p_{ms}. Examples of the semantic similarity calculation are shown in Figure 1.10.

$$p_{ms}(c1,c2) = \min_{c \in S(c1,c2)} \{p(c)\} \qquad 1.14$$

$$sim(c1,c2) = -\ln p_{ms}(c1,c2) \qquad 1.15$$

Lord et al. demonstrated that the semantic similarity of two sequences correlates with their BLAST bit score. In general, specific terms occur less frequently than more general terms and they therefore have a smaller probability of occurring and result in higher similarity scores when they are a common parent. However, there are some limitations to this scoring scheme. It is based upon information content, so there is no

Term	No annotations
B	2
D	3
E	15
G	11
H	2
I	5
J	12



Term 1	Term 2	Common Parent	Common parent p(c)	Semantic similarity
D	E	C	0.4	1.02
G	J	F	0.6	0.511
I	E	A	1	0

Figure 1.10 Calculating semantic similarity. i)The table shows the number of annotations for a set of GO terms. The GO Graph for these terms is shown below. The number of times n, that each terms occurs is displayed to the left of the term. The *p(c)* of each term is shown to its right or below. ii) Example semantic similarity scores for these terms.

biological reasoning linking the similarity score of two terms; it is merely based upon the frequency of their common parent in the set of annotations. Secondly, annotations are not split evenly between the different types of function in the GO molecular function category, making it possible for two annotations to appear more similar to one another simply on the basis that there are fewer existing annotations of that type in the database.

1.6.2. Precision & Recall

Others have used the measures of precision and recall (Jones, et al., 2005), which are defined as

$$\text{Precision} = TP/(TP + FP) \quad \textbf{1.16}$$

$$\text{Recall} = \frac{TP}{N_A} \quad \textbf{1.17}$$

where TP is the number of true positive predictions, FP, the number of false positive predictions and N_A the number of annotations for the query sequence. Where a large test set is used these values are averaged over the full set, so N_A is equal to the total number of annotations in the test set.

Others have modified the calculation of recall and precision to account for the hierarchical structure of GO. Hierarchical precision (HP) and recall (HR) (Verspoor, et al., 2006) are defined as:

$$HP = \frac{1}{|G(x)|}\sum_{q \in G(x)} \max_{p \in F(x)} \frac{|\uparrow p \cap \uparrow q|}{|\uparrow q|} \quad \textbf{1.18}$$

$$HR = \frac{1}{|F(x)|}\sum_{p \in F(x)} \max_{p \in G(x)} \frac{|\uparrow p \cap \uparrow q|}{|\uparrow p|} \quad \textbf{1.19}$$

where $G(x)$ is the set of predicted annotations, $F(x)$ is the set of annotations and $\uparrow p$ is the set of parent terms for GO term p and likewise $\uparrow q$ is the equivalent for GO term q. As shown in Figure 1.11 for a prediction of $F(x)$ = GO:E with parent terms A,B,C, so that

$\uparrow p$ is {A,B,C,E} for a sequence with annotation $G(x)$ = GO:H, with parents A,B,D,F,G,K so $\uparrow q$ is {A,B,D,F,G,H,K} HR = 2/7 and HP = 2/4.

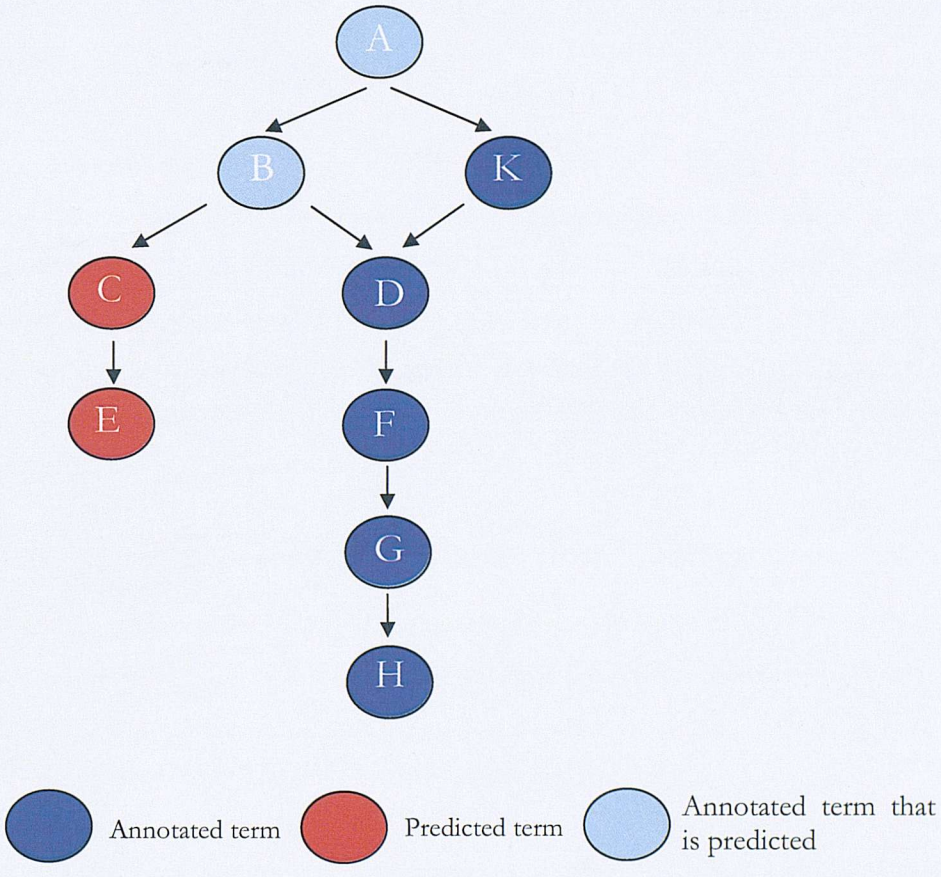


Figure 1.11 Hierarchical precision and recall. The example protein is annotated with terms A, B, D, F, G, H and K. GO terms A, B, C, and E are predicted as its function.

1.6.3. Compatible Terms

The beginning of this section introduced the different type of errors that are made when predicting protein function. The reliability of comparison of predictions with annotations is dependent upon the accuracy and completeness of the existing annotations. Unfortunately it is possible that functional annotations do not fully describe protein function. Annotations can be too general, with a general function assigned to a protein when a more specific related functional term better describes the function. Proteins (especially for multi-domain sequences) may also have more functions than those they are annotated with. So it is possible that predictive methods identify correct functions that are not currently annotated for the query protein.

Some approaches (e.g. Engelhardt, et al., 2005) account for such problems by considering a single annotation for a protein and then classing all terms that are more specific descendents of this term to be 'compatible' with it. More general terms (ancestors) are also considered to be compatible, as long as they are not too general. If the prediction is present in this set of compatible terms, then it is classed as a correct prediction and if not, it is an incorrect prediction.

1.7. Comparison of Function Prediction Methods

The previous sections have described numerous function prediction methods and some of the approaches used for assessing their performance. Many of the methods have been benchmarked using a set of proteins of known function. In most cases the performance of a new method is compared to that of annotation transfer using BLAST. Some benchmarks have used predictions from other methods (Engelhardt, et al., 2005) for comparison but have found that the complex setup of such methods makes it difficult to ensure that the assessment is performed fairly. Despite most methods being compared to BLAST annotation transfer, it is not easy to compare their performance because each benchmark generally uses a different set of proteins, annotations and other settings (e.g. some restrict to low levels of sequence identity). In addition most methods use different approaches to assess the performance of function prediction. Therefore it is not possible to directly compare results for benchmarking from one method to another. Further some

methods only make predictions for molecular function GO terms while others combine their predictions from all three GO categories (biological process and cellular component being the other two). In addition to all of these issues it is often impossible to know if an incorrect prediction is incorrect or if it is a function of the protein that has yet to be characterised. In general it is necessary to simply accept the existing annotations and consider any disagreement as incorrect. Despite these problems this section summarises the performance of current Gene ontology based function prediction methods and attempts to compare them.

The benchmarking of SIFTER (section 1.5.1.11; Engelhardt, et al., 2005) compared its performance to a number of methods including GOtcha (section 1.5.1.8) and BLAST annotation transfer for proteins from a set of 100 Pfam families. Only the top prediction made by each method was considered and both SIFTER and GOtcha made correct predictions for 80% of proteins. BLAST performance was worse only making correct predictions for 67% of proteins. A BLAST e-value cutoff of 0.01 was used, and as shown in section 0 BLAST annotation transfer is unreliable at such levels. Using this e-value threshold BLAST makes predictions for all of the proteins that SIFTER also makes predictions for. It would seem likely that reducing the e-value cut off would reduce the coverage but simultaneously increase the percentage of predictions that are correct.

There was only 40% agreement between SIFTER and GOtcha where predictions were identical and a total of 63% agreement if compatible predictions were also considered. SIFTER and BLAST show 24% and 25% identical agreement with BLAST, rising to 45% and 47% respectively, where compatible predictions are also considered. The authors suggest that the low level of agreement between the different methods indicates that each method is making a considerable number of incorrect predictions.

Assessment of the compatible predictions made by each of the methods shows that in many cases, SIFTER and GOtcha make predictions that are descendents (i.e. more specific) than the actual annotation (95% for SIFTER and 71% for GOtcha). In contrast only 34% of BLAST compatible predictions are descendents of the actual annotation. Therefore SIFTER and GOtcha have a tendency to overpredict, whereas BLAST

underpredicts. It is also possible that the apparent overprediction is a result of the existing annotations not being specific enough and the methods are correctly inferring a function that should be present in the annotations.

The ProDom and CDD domain based prediction approach proposed by Schug et al. (2002) used a similar approach for benchmarking. They made predictions for over four thousand human proteins and obtained coverage of 81% (percent of proteins that predictions were made for). For their predictions 74% had some agreement (identical or compatible) with the annotation.

GOEngine used a large set of proteins from nr and various genome databases (e.g. Flybase) and applied a similar approach of considering the number of proteins for which the predictions made were compatible with the existing annotations (Xie, et al., 2002). Under these conditions 78% of the molecular function predictions were correct, 76% for biological process annotations and only 65% of the cellular component annotations.

The benchmarking of the methods detailed above employed similar assessment schemes; they considered the top term and they classed a prediction as correct if it was compatible with the correct annotation. They were tested on different datasets and for these testsets between 70-80% of their molecular function predictions are generally correct. In contrast BLAST performance is slightly lower at 67% (from SIFTER benchmark).

The benchmarking of other methods use different assessment approaches. For example PFP (Hawkins, et al., 2006) primarily investigated the prediction of Biological process terms and were assessed at different depths within the GO graph (they defined the depth as the edge distance between the predicted term and the root term (i.e. biological process)). Benchmarking of PFP shows increasing specificity and annotation coverage with increasing GO depth. The PFP method has also participated in community wide blind assessments of function prediction methods (see section 1.8) and was one of the better performing methods in CASP7 (7^{th} Critical Assessment of Techniques for Structure Prediction) experiment.

ProKnow performance was also assessed at individual levels within GO (using the same classification of level/depth as PFP)(Pal and Eisenberg, 2005). ProKnow was assessed using a Receiver Operator Curve which plots the true positive rate against the false positive rate. ProKnow performed best at the first GO level (i.e. highest true positive rate and lowest false positive rate). Performance decreased at level 2 (lower true positive rate and higher false positive rate). A further reduction in performance was observed at level 3 and the performance at all subsequent levels was similar to that at level 3. ProKnow was also assessed using precision. It obtains precision of 90% at GO level 1 reducing to 60% at level 8.

Phunctioner assessed performance at low levels of sequence identity (less than 30%) and compared the top ranked prediction with BLAST predictions (Pazos and Sternberg, 2004). At a 30% identity threshold, 85% of Phunctioner predictions were correct, rising to 90% at a 15% sequence identity threshold. At the 30% identity cutoff BLAST performance was comparable to Phunctioner but at the 15% identity threshold only 70% of BLAST predictions were correct. ROC analysis also showed better performance of Phunctioner compared to BLAST at 15% identity at all settings.

The performance of the Data mining prediction (King, et al., 2001) approach was assessed for proteins from *A. thaliana* for the top four levels of the GO molecular function category. Using different settings (generating rules using different protein features) precision between 42%-85% is obtained. A composite setting, which combines all the protein features obtains precision of 79%, which probably provides the best indication of the overall performance of DMP. Recall isn't stated but the method focuses on greater precision and lower recall as opposed to greater recall and lower precision.

GOPET was benchmarked using a small set of only one hundred protein sequences (Vinayagam, et al., 2006). Its performance was assessed using precision and recall and compared to GOtcha. GOPET and GOtcha obtained precision of 89% and 80% respectively, however GOtcha had greater recall. A value of recall is not given but

296 of the 332 GOPET predictions were correct, while 430 of the 535 GOtcha predictions were correct.

While it was possible to make a limited comparison between the first few methods in this section, the various assessment approaches used for the subsequent methods demonstrate that it is not possible to compare the performance of different methods based upon their published benchmark results. At best, most methods appear to obtain close to 80% precision (in some cases only for specific levels of GO). While this level of precision is encouraging, there is still clearly room for improvement. The inability to effectively compare the results obtained by different methods provides an argument for community agreed benchmark sets of proteins and assessment methods, which would make it much easier to interpret the results of newly published function prediction algorithms. The structure prediction and protein docking fields have both benchmark sets (e.g. Hwang, et al., 2008) and methods of assessment, which have predominantly been driven by the community wide assessments that are performed in these fields. The establishment of a successful community wide experiment for protein function prediction would drive the development of standard datasets and assessment methods. Attempts to perform such community wide experiments for function prediction are described in the next section.

1.8. International Assessment of Function Prediction

The previous sections have described the many different approaches that are currently used to infer protein function. Section 1.6 introduced some of the most commonly used methods for assessing function prediction methods and Section 1.7 detailed the performance of function prediction and provided a limited comparison between them. As most methods use different data sets and methods of assessment and in addition generally compare their performance to BLAST, it is difficult to compare different methods and identify those that perform best. This is a common problem within Bioinformatics, most notably the structure prediction community, which introduced the concept of a large scale assessment of different methods. The resulting Critical Assessment of Techniques for Structure Prediction (CASP) is now well established, with CASP8 taking place later this

year. The docking community has also established an ongoing assessment programme CAPRI (Critical Assessment of PRediction of Interactions). Similar approaches have been piloted for the function prediction community, with the first assessment in this style undertaken at the first Automated Function Prediction Special Interest Group meeting (AFP SIG; Friedberg, et al., 2006b) at ISMB 2005. This assessment only considered automated function prediction and included seven targets for function prediction and around 7-8 participating servers. The outcome was that function prediction is difficult to assess. Unlike structure prediction, which has a ready source of structures from NMR and crystallography, there is not a plentiful supply of proteins, whose function has recently been elucidated but has yet to be published. Therefore, more than half the targets were hypothetical proteins and their function was still unknown at the end of the assessment. For the few cases, where the protein function was known, the servers made similar predictions, but often at different levels of functional specificity within GO.

The CASP meetings six and seven incorporated a degree of function prediction. The function assessment at CASP6 (Soro and Tramontano, 2005) suffered from the same problems as the AFP SIG, with the function of most of the targets unknown at the time of assessment. Predictors were allowed to predict function in multiple forms, including GO function, ligand binding sites, the role of individual residues and also post-translational modifications. Where the predictions could be analysed, the functional predictions tended to vary but there was often agreement at more general functional levels of GO. The results were re-analysed over a year later (Pellegrini-Calace, et al., 2006), with proteins considered that had not had an annotation at the time of the experiment but were now annotated. For this re-analysis, the assessors had knowledge of the individual methods used by each of the groups, which was not available as the original assessment was blind, with assessors not knowing who had made individual submissions. This allowed an analysis of only 11 targets, for which GO molecular function predictions were considered. Of 85 predictions, approximately 20% (18) matched the annotation in some way (i.e. more or less specific). While these results demonstrate the difficulty of inferring function, most of the predictions that were correct, saw some agreement between groups using different approaches and sources of information to make predictions. This suggests that combining predictions from separate sources can complement each other and lead to correct predictions. Unfortunately the assessors do not comment on the level of

agreement between the different methods where predictions were incorrect, so it is not clear if there was also agreement between different groups in such cases.

The CASP7 experiment encountered similar problems with a lack of experimental results to verify GO and E.C. based predictions (Lopez, et al., 2007c). This problem occurs as CASP targets are chosen because their structure has been solved and not on the basis of identifying their function. So in most cases there is no change in the known function of the targets between the beginning and end of the experiment. It was possible to assess twenty targets that did not have GO annotations present in UniProt (The UniProt Consortium, 2007) before predictions were submitted. The assessors compared each annotated term with the most similar predicted term and calculated the common ancestor depth, calculated as the depth of the first common parent. A prediction score was calculated for each target using the sum of common ancestor depths, divided by a target's maximum possible score. The best performing group obtained an average GO score of 0.5, suggesting that on average their predictions were half way between the correct annotation and the GO graph root. Like for CASP6, the combination of predictions using different approaches was considered, but showed only some improvement over the performance of most individual groups and was worse than the best performing group.

The prediction of ligands and ligand binding sites is more accessible for assessment in CASP because in some cases ligands are bound to the protein structures (Lopez, et al., 2007c). The assessors identified biologically relevant ligands in 21 of the target structures. They were assessed by calculating the overlap of predicted binding sites with the structural binding site. Only three groups submitted binding site predictions making it impossible to perform a thorough evaluation of the results, although some predictions were very accurate. Despite the lack of binding site submissions, it is likely that CASP8 will only consider binding site predictions in the function category. This is due to the problems associated with GO function remaining unknown at the end of the assessment.

1.9. Scope and Outline of this Thesis

The previous sections described state of the art methods for protein function prediction and highlighted the challenges associated with the function prediction field. This thesis details the development of ConFunc, an automated function prediction algorithm, focusing on improving function prediction at low levels of sequence similarity. The main aspects of this work are briefly described below. This chapter also described the classification systems of protein structure such as SCOP and catalogues of enzyme catalytic residues (the CSA). In addition to describing ConFunc, this thesis details a study of the convergent evolution of enzyme active sites.

- Chapter 2 reports on the study of the convergent evolution of enzyme active sites.

- Chapter 3 describes, in detail, the aims and objectives of this thesis. The ConFunc method is described, as well as the different approaches that were considered during development. The generation of the test set used to benchmark ConFunc and the methods to which ConFunc is compared are also described. Finally this chapter details how the function predictions made by the various methods are assessed against existing functional annotations.

- Chapter 4 covers the development of ConFunc, investigating the different options and parameters that were considered.

- The benchmarking of ConFunc is reported in Chapter 5. It includes the comparison of ConFunc performance to function predictions made by BLAST, PSI-BLAST and Pfam. Benchmarking is performed at low levels of sequence similarity and in one case sequence similarity is limited to the 'twilight zone'. Detailed analyses are performed for the prediction of different functional types and at different levels of the Gene Ontology graph.

- Chapter 6 describes the ConFunc server, the submission options available to users and the results that it outputs.

- Chapter 7 discusses the results obtained from benchmarking and compares the ConFunc approach to other function prediction methods. An analysis of the functional role of the conserved residues identified by ConFunc is also performed. Practical applications of ConFunc are also described.

- Chapter 8 concludes the thesis by detailing future work and planned additions to ConFunc.

Chapter 2

Convergent Evolution of Enzyme Active Sites

2.1. Overview

This chapter details the study of the convergent evolution of enzyme active sites, which was reported by Gherardini et al. (2007) in the *Journal of Molecular Biology*. This work was performed primarily by Pier Federico Gherardini while visiting the Structural Bioinformatics group at Imperial College London. I provided day to day supervision for Pier Federico and assisted him with the experimental design and analysis of results. I also made significant contributions to the writing of the manuscript.

2.2. Introduction

Convergent evolution refers to non-homologous proteins which evolve to perform the same biochemical function. Convergent evolution is best known amongst the serine proteases where it was first identified in the 1970s (Drenth, et al., 1972; Kraut, 1977). The observation that the Ser-His-Asp catalytic triad of the trypsin family has very similar geometry to that of subtilisin, provided the first example of convergent evolution as subtilisin and trypsin are unrelated proteins with different folds. Other examples of convergent evolution have since been identified, but to date no systematic study of the convergent evolution of enzyme active sites has been performed. This chapter describes the first such study to do this.

2.2.1. Mechanistic and Transformational Analogues

We initially defined two types of convergence. An overview of their definitions is shown in Table 2.1. Transformational analogues are pairs of non-homologous enzymes that perform the same transformation as defined by having the same four-digit E.C. number. The second type of convergence relates to non-homologous enzymes that perform the same or related transformation (i.e. they share at least the first three digits of the E.C. number) with similar arrangements of residues in their active sites and a similarity in enzyme mechanism. The enzymes involved in this type of convergence are referred to as mechanistic analogues. Trypsin and subtilisin are mechanistic analogues as they share the same three-digit E.C. number and both have the Ser-His-Asp catalytic triad. Pairs of enzymes may be both mechanistic and transformational analogues, by sharing the same four-digit E.C. number and a similar arrangement of catalytic residues. Both types of convergence make an assumption that enzymes belonging to different SCOP superfamilies are evolutionarily unrelated.

Transformational analogues have been widely investigated (see below) and as such mechanistic analogues form the focus of this research. The main aim is thus to identify how frequently similar active sites and enzyme mechanisms have been invented by convergent evolution.

	Mechanistic analgoues	**Transformational analogues**
Reaction catalysed	Same 3-digit E.C. number	Same 4-digit E.C. number
Evolutionary relationship	Different SCOP superfamily	Different SCOP superfamily
Mechanisms	Similar	Similar or different

Table 2.1 Definitions of Convergence – Mechanistic and Transformational analogues. The E.C., SCOP and mechanistic requirements for proteins to be considered mechanistic and transformational analogues.

2.2.2. Previous studies of Transformational analogues

Galperin et al., (1998) performed a systematic analysis of transformational analogues. They considered the sequence conservation for each four-digit E.C. number and identified 105 E.C. numbers which have sequences without detectable sequence similarity. They demonstrated that 34 of these pairs had distinct structural folds and that in ten other cases the enzymes shared the same fold. No further structural information was available for the other pairs of enzymes at that time.

To study protein evolution Hegyi & Gerstein (1999) combined the SCOP, Swiss-Prot and ENZYME databases. Part of their work has been described in Chapter 1, Section 1.4. They considered convergence in a similar way to the definition of transformational analogues, however they did so at the third E.C. digit as opposed to the fourth. In summary they found that more than half (51 out of 91) of the functions considered were present in more than one fold, providing an indication of the occurrence of transformational analogues. Further, for folds only associated with enzymes they found an average of 2.5 folds per function, suggesting fairly high levels of transformational convergence within enzymes.

The work of George et al., (2004) was also introduced in section 1.4. They combined the SCOP, Swiss-Prot, ENZYME and CSA databases. At the fold level they observed an average of 1.2 folds per four-digit E.C. number, increasing to 3.4 at the three-digit E.C. level. These results suggest a low level of convergence to exactly the same biochemical transformation as described by E.C. but much greater levels of convergence to similar reactions as described by common three digit E.C. numbers. This observation is slightly greater than the 2.5 functions per fold observed by Hegyi and Gerstein. George et al. observe similar results at the SCOP superfamily level with 1.2 superfamilies per four-digit E.C. number and 3.5 per three-digit E.C. number.

2.3. Method

2.3.1. Identification of Mechanistic Analogues

The process of identifying mechanistic analogues involved the use of four different sources of data. Function data was obtained from E.C., structural data from SCOP, catalytic sites from the CSA and details of enzyme mechanism were manually retrieved from the literature. Figure 2.1 outlines the steps taken to identify mechanistic analogues. Each of the steps is described below:

1. CSA entries with associated E.C. numbers were identified. Version 2.2.2 of the CSA contained catalytic site data for 169 different three digit E.C. numbers.
2. Structural similarities within the structures for each E.C. number were identified using Query3D (Ausiello, et al., 2005).
3. SCOP (version 1.71) was used to filter out similar active sites between proteins in the same superfamily. This resulted in 67 different three-digit E.C. numbers matching all of these criteria.
4. The pairs of structures for each E.C. number were grouped to remove redundancy and ensure that each of the matches contained a different pair of active sites.
5. A literature search was performed to ensure that the structural matches identified were definitely the result of similarities in enzyme mechanism.

2.3.2. Identifying Transformational Analogues

The identification of transformational analogues requires the identification of analogous proteins sharing the same four-digit E.C. number. To do this the PDBSprotEC (Martin, 2004) was used to obtain matches between PDB structures and E.C. numbers. SCOP was then used to obtain details of the domains in each PDB structure. False positives were removed by identifying the catalytic domain and filtering out all matches where the catalytic domains are homologous but the other domains present in the PDB structure differ. This was initially done using the CSA, and then literature where CSA data was unavailable. This approach enabled an analysis of transformational analogues throughout the entire PDB rather than just those that have CSA data. The subset with CSA data was also considered separately to enable a direct to comparison to mechanistic analogues.

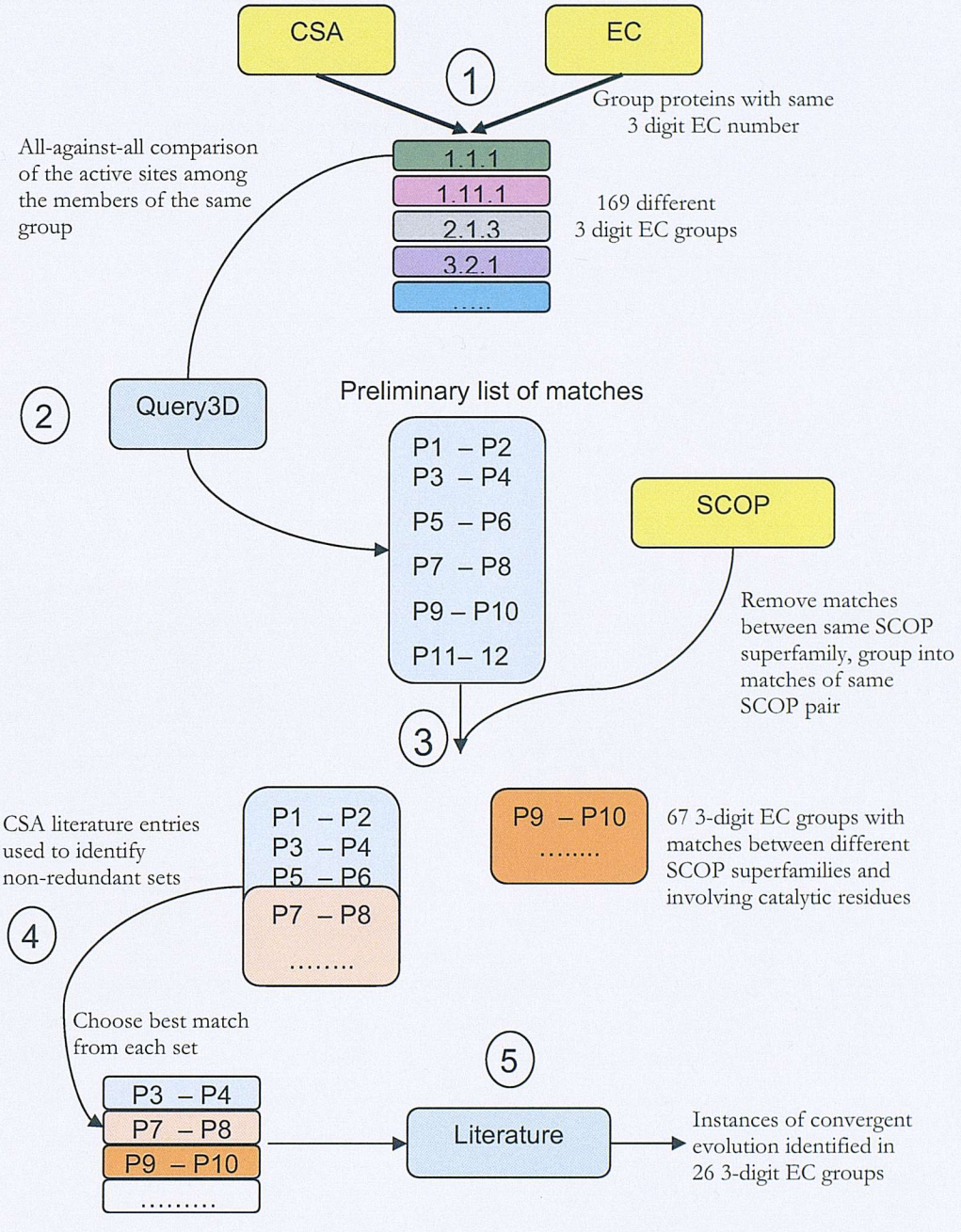


Figure 2.1 Identifying Mechanistic Analogues. Adapted from Figure 1 Gherardini et al 2007. The circled numbers refer to the steps in the method as described in the text.

2.4. Results

2.4.1. Mechanistic Analogues

The dataset considered contained 169 different three-digit E.C. numbers. Non-homologous proteins with similar catalytic sites were identified in 26 of the 169 E.C. numbers, representing approximately 15% of the three-digit E.C. numbers considered (see Figure 2.2). Of these eleven were also transformational analogues as they share the same four-digit E.C. number. This result demonstrates that convergent evolution of enzyme active sites is not a rare phenomenon and it is present in a significant number of different enzyme functions.

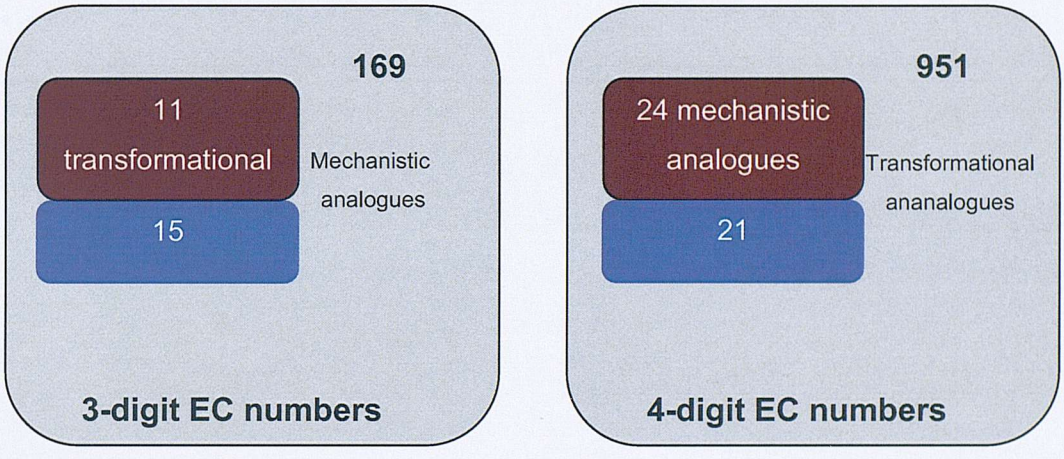


Figure 2.2 Overall results of convergent evolution study. The left diagram shows the total number of three-digit E.C. numbers considered (169) of which 26 are mechanistic analogues. Eleven of these are also transformational analogues. The results for the transformational analogues are shown on the right. Of 951 four-digit E.C. numbers considered, 45 have transformational analogues and 24 of these were also mechanistic analogues. Adapted from Figure 3, Gherardini et al. (2007).

It is interesting to consider how many times each catalytic site has evolved. This can be assessed by observing for each E.C. number with mechanistic analogues, how many superfamilies have the same active site. In many cases the analogues belong to only two superfamilies but in some cases many more superfamilies have converged to the same active site. For example glycosyl hydrolases (E.C. 3.2.1) exhibit two different mechanisms,

one of which is present in five different SCOP superfamilies, while the second is present in seven. The Ser-His-Asp catalytic triad is the most widespread example of convergent active sites. For E.C. 3.1.1 (carboxylic ester hydrolases) it is present in eight different superfamilies and six for E.C. 3.4.21 (serine endopeptidases). The catalytic triad has been found not only in serine proteases but also in active sites for many different types of enzymes, including acyltransferases (2.3.1), peptidases (e.g. 3.3.14 and 3.4.21) and amide hydrolases (3.5.1). In all, the catalytic triad was observed in 23 different SCOP superfamilies, making it the most widespread catalytic site observed.

2.4.2. Transformational Analogues

The results of analysis of transformational analogues in the CSA are shown in Figure 2.2. Transformational analogues were identified in approximately 5% (45 of the 951) of the four-digit E.C. numbers considered. More than half (24) of these were also mechanistic analogues sharing similar active sites and mechanisms. This supports the observation that convergent evolution of enzyme active sites is not a rare phenomenon. The transformational analogues identified are predominantly present in two superfamilies but in a few cases three superfamilies have converged to the same function, while the glycosyl hydrolases exhibit transformational analogues in six different superfamilies.

The extension of the analysis of transformational analogues to the whole PDB provides the first comprehensive and manually verified list of transformational analogues in the PDB. It considers a total of 1166 four-digit E.C. numbers. Twenty transformational analogues were identified in addition to those from the CSA set, resulting in transformational analogues being present in approximately 6% (65/1166) of reactions considered. In their analysis George et al. (2004) identified an average of 1.2 SCOP superfamilies per four-digit E.C. number. It is not possible to directly compare their results with ours because our results indicate the number of four-digit E.C. numbers with two or more SCOP superfamilies whereas the George et al. results simply provide the ratio of SCOP superfamilies to four-digit E.C. numbers. As some functions have been found in many different superfamilies their result of 1.2 Superfamilies per four-digit E.C.

number does not suggest that transformational analogues are present in 20% of four-digit E.C. numbers.

2.5. Conclusion

2.5.1. Limitations

This work is limited by the data sources that it used to perform the analysis. It is important to consider the coverage of these datasets. Version 2.2.2 of the CSA was used and it contains 98% (12,987 of 13230) of the PDB codes present in SCOP (Version 1.71) that also have an E.C. number assigned in PDBSprotEC. 1166 different four-digit E.C. numbers are linked to a PDB code in PDBSprotEC, of these 951 (82%) are present in the dataset used here.

Issues of accuracy concern SCOP, the CSA and E.C. The SCOP classification is used to provide an indication of evolutionary relationships, with the assumption that proteins belonging to different superfamilies are not evolutionarily related. This assumption could be affected by errors in the SCOP classification of assigning homology relationships. Recent work has shown that in some cases homology can be identified between superfamilies, folds and even classes (Cheng, et al., 2008). It seems likely that such observations are more likely to be the exceptions to the general observation that proteins belonging to different superfamilies are evolutionarily unrelated. It is also more likely that such observations will be observed at the superfamily level than at the fold or class levels. It is therefore important that all of the analogous active sites identified (for the three-digit E.C. analysis) belong to different folds and in 68% of cases the domains belong to different classes, thus reducing the potential for errors to be introduced by SCOP.

Active sites are compared using structural superposition of catalytic residues present in the CSA. The CSA has well defined criteria for catalytic residues, requiring them to play a direct role in the catalytic mechanism. It is possible that cases of

convergence could be missed if this definition does not fully capture all of the catalytic residues.

Finally the E.C. classification provides a source of potential error. Limitations of the E.C. system for describing enzyme function are well known (Babbitt, 2003; Riley, 1998). The first limitation is the choice of reaction direction, which while chosen arbitrarily, can change the class that an enzyme belongs to. A second limitation is the varying levels of specificity within E.C. Even though E.C. is a hierarchical classification, the E.C. numbers between and even within a class may describe different levels of functional specificity even though they are at the same E.C. level.

2.5.2. Concluding Remarks

Despite the limitations described above, the databases used are the most suitable resources to perform a large scale analysis of convergent evolution. It would be impractical to perform such an analysis without an automated first step to identify potential cases of convergent evolution which are then followed up with manual investigation.

This study provides a detailed insight into the occurrence of convergent evolution in enzyme active sites. The observation of mechanistic analogues in 15% of the three-digit E.C. numbers considered and also in more than half of the transformational analogues identified demonstrates that convergent evolution of enzyme active sites is not rare. In some cases the same catalytic machinery has evolved numerous times to perform the same or similar function. It seems likely that these same machineries have evolved independently because of specific mechanism constraints that are required to perform reactions. It is also possible that these mechanisms present efficient ways of performing the required reaction and as such are favoured by nature during evolution.

Chapter 3

ConFunc Methods

3.1. Overview

This chapter describes the methods that have been investigated during the development of ConFunc. Section 2.2 sets out the aims and objectives of this research and the resulting program ConFunc. Section 3.3 outlines the stages in the ConFunc predictive process and the following sections describe, in detail, the methods investigated for each step of the process. Section 3.4 details the method used for assessing ConFunc predictions. This includes the scoring scheme and sequence test set used and the alternative methods that ConFunc is compared to.

3.2. Aims and Objectives

The protein function methods described in the previous chapter demonstrate the numerous approaches that can be used to infer function. The use of homology is important for functional annotation as the number of known protein sequences far outweighs the number of known protein structures and as more genomes continue to be sequenced, this difference is escalating.

The previous chapter also described the complex nature of protein function and the challenges associated with its prediction. The main aim of this research was to develop a new method to complement existing methods, with the goal of improving the ability to infer protein function computationally. Consideration of existing methods suggested that the following points were important for the development of ConFunc:

- Studies assessing the relationship between sequence similarity and function (see section 0), demonstrate that at high levels of identity, sequences generally share the same function. Therefore inferring function for such sequences should be

relatively easy as annotation transfer from BLAST/PSI-BLAST hits will be very effective. However, at low levels of sequence similarity, sequences are less likely to share the same function and it is in this range that alternative more complex methods are essential. Therefore a main objective of this research was to develop a method that would accurately predict protein function at low levels of sequence identity. Despite the importance of this area of function prediction, it has been rarely the focus of existing methods.

- Numerous methods utilise BLAST or PSI-BLAST e-values to infer function, either by direct annotation transfer or by grouping hits, such as Gotcha (Martin, et al., 2004) or PFP (Hawkins, et al., 2006). These methods only utilise the statistically similarities between sequences and ignore the features of the sequences themselves. Therefore an aim of this research was to develop a method that uses the biological features of sequences, rather than merely their overall similarity with other sequences. As discussed in the previous chapter (see section 1.4), a small number of critical residues are often essential for protein function, so identifying residue conservation and using it in turn to infer function is a more biologically relevant approach, which has been used by a few methods (Hannenhalli and Russell, 2000; Pazos and Sternberg, 2004).

- The Gene Ontology (see 1.1.3) provides a universal classification for all types of function, including non-enzymes and as a result many recent function prediction methods have utilised GO to infer function.

- A requirement of the method is that it should meet the needs of the scientific community. The hundreds of sequenced genomes that require annotation demonstrate the need for an automated approach that can be applied on a large scale. Secondly, there is the need of the individual scientist, who would like to identify potential functions for their proteins of interest. So the method should also be scalable for use via a web server, making it available to the scientific community.

- Incorrect annotations can propagate when present in databases. It is important that new function prediction methods can accurately infer function. In this research precision and recall (see section 3.4) are primarily used to assess predictive performance, so greater emphasis should be placed on precision (proportion of predictions that are correct) as opposed to recall (proportion of annotations that are identified).

Phunctioner (see 1.5.2.3), a structural function prediction method developed in my laboratory, has shown the ability to identify functional residues and in turn use them to infer function. An equivalent method applicable to sequence space would complement existing sequence based methods and aid functional annotation. This thesis reports the research and development of such an algorithm and benchmarks the resulting program, ConFunc.

3.3. An Overview of the ConFunc approach

The ConFunc method developed in this research is loosely based upon the concept used by Phunctioner, a structural based function prediction method. Phunctioner (described in section 1.5.2.3) is a structural based method that uses the identification of functional residues to infer protein function. ConFunc follows a similar approach but in sequence space. For a query sequence, ConFunc runs PSI-BLAST to identify homologous sequences. GO annotated homologues are extracted from the output and a multiple sequence alignment is calculated. The GO annotations are then used to split the sequences into subalignments, with each subalignment representing a different GO function. Conserved positions are identified in each subalignment and a PSSM is calculated for these positions. The query sequence is scored against each GO term PSSM and the scores are converted to e-values and used to infer function for the query. These general steps are shown in Figure 3.1 and are visualised in Figure 3.2. The different approaches that were investigated for each of these during development are described in the following sections.

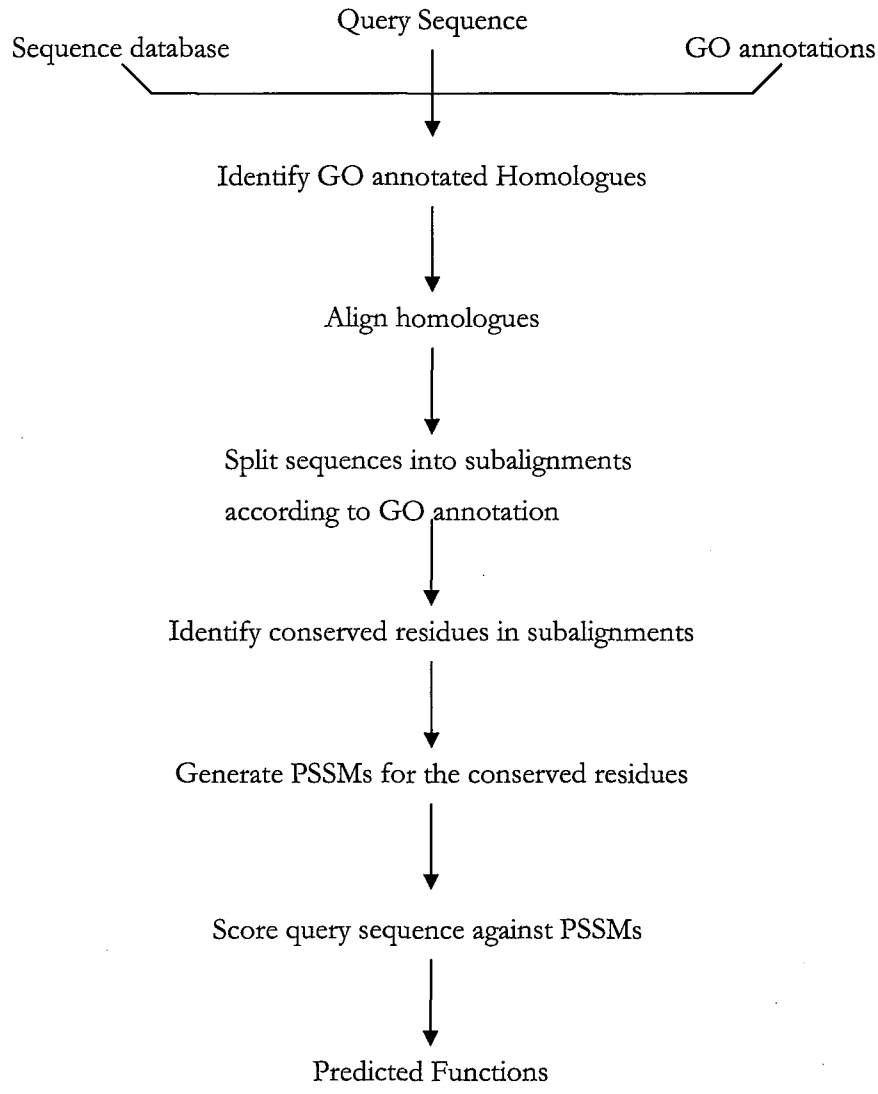


Figure 3.1 An outline of the steps in the ConFunc algorithm.

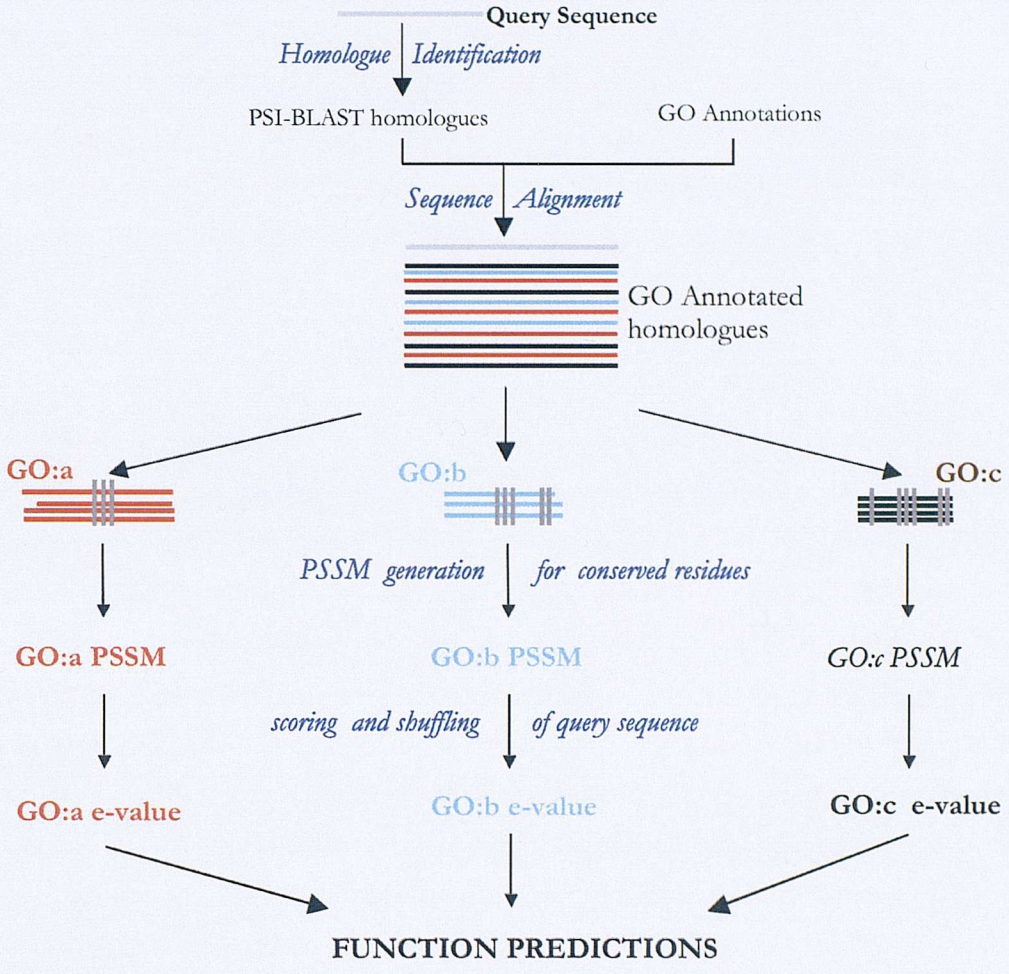


Figure 3.2 The ConFunc algorithm.

3.3.1. Identifying Homologues

ConFunc uses PSI-BLAST to identify homologues of a query sequence. PSI-BLAST is used because it is more sensitive than BLAST and able to identify more remote homologues. This is important because ConFunc is focussed on inferring function where annotated close homologues do not exist and therefore the identification of a greater number of remote homologues is necessary. The number of PSI-BLAST iterations is set at three because running many iterations increases running time and further iterations are only likely to identify very remote homologues of the query sequence, which are more likely to perform different functions than the query sequence and therefore offer little

benefit. Secondly it has been shown that PSI-BLAST alignment accuracy drops with increasing iterations (Friedberg, et al., 2000) and this could influence the resulting function predictions (Friedberg, 2006c). For the analyses described in Chapter 3 and 4, PSI-BLAST was run with a $1x10^{-8}$ e-value threshold. All sequences identified by PSI-BLAST with more than 30% sequence identity with the query sequence are removed. This simulates the scenario where query sequences do not have GO annotated close homologues.

It is important that the GO annotations used in the predictive process are accurate, as the use of incorrect annotations present in databases can result in the propagation of errors (Brenner, 1999; Devos and Valencia, 2001). The presence of GO evidence codes associated with each annotation provides a way of assessing their confidence. The GO evidence codes are shown in Table 1.2 and annotations with IEA, ND and NR evidence codes are ignored by ConFunc. Therefore electronically generated annotations and those for which there is no record of why they are present are removed from the predictive process, with the aim of reducing the propagation of errors. Further as the quality of the annotations is important, the Swiss-Prot database is used as the source of protein sequences because the EBI (European Bioinformatics Institute) provides a set of hand-curated annotations for Swiss-Prot sequences.

3.3.2. Aligning Homologues

Once GO annotated homologues have been identified, they need to be aligned. Sequence alignment methods were introduced in section 1.2. PSI-BLAST aligns the sequences the it identifies, but it is optimised for speed rather than precision and only stacks the individual pairwise alignments. As a result the PSI-BLAST alignment can generally be improved. Numerous tools exist to align multiple sequences (described in section 1.2.6) and the use of the following programs was investigated: T-Coffee (Notredame, et al., 2000), MUSCLE (Edgar, 2004) and MAFFT (Katoh, et al., 2002). The default running parameters were used for each alignment method. ConFunc uses multiple sequence alignments to identify conserved residues associated with particular GO terms, the quality of the alignments used is therefore important and may effect the residues that are identified and in turn the functions that are predicted.

3.3.3. Generating GO Term Subalignments

The Gene Ontology is central to ConFunc by supervising the function prediction process. Once the multiple sequence alignment is obtained, GO is used to direct the splitting of the multiple sequence alignment into subalignments. Each subalignment represents a different GO term and contains only the sequences annotated with that term. The parent terms of each annotation are identified and each sequence is added to the subalignments of its parent terms as well. An example is shown for the subalignments created from three sequences in Figure 2.3. In this example, sequence 1 is annotated with GO term ***E,*** whose parent terms are ***D,B,A*** and ***GO:Root*** (where ***GO:Root*** represents the molecular function root). Sequence 1 is therefore present in the subalignments for each of these terms.

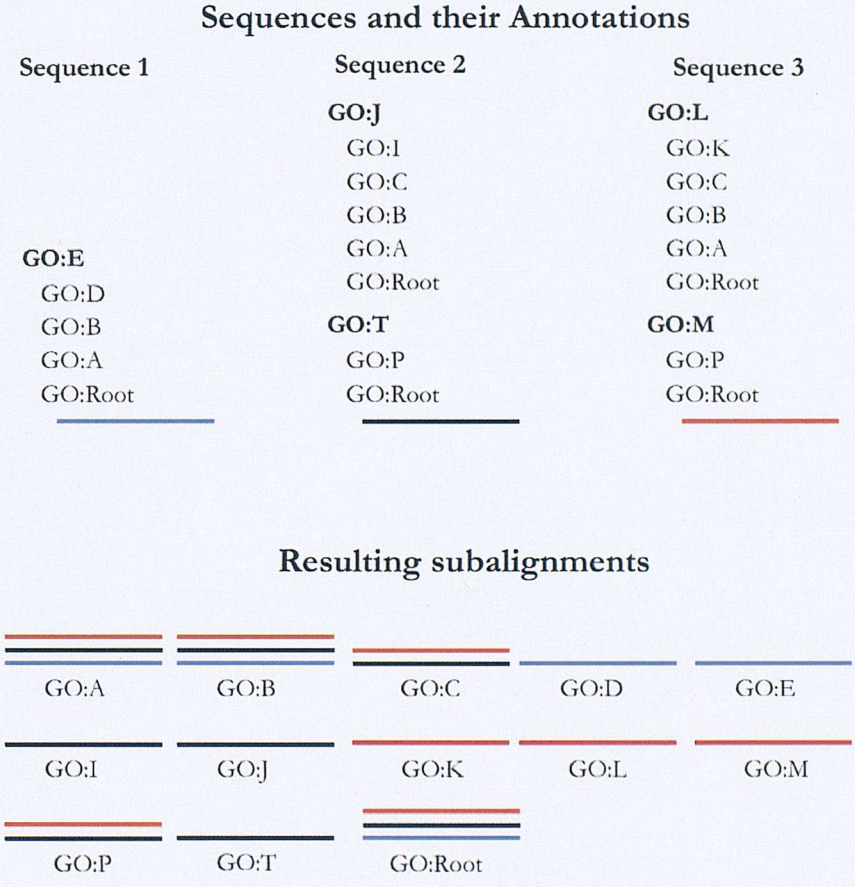


Figure 3.3 Generating GO subalignments. Three example sequences from a multiple sequence alignment are shown with their GO annotations. Each sequence is added to the subalignment for each of its annotated terms (including parent terms) as shown at the bottom of the figure.

Subalignments that represent specific functional terms tend to contain fewer sequences (e.g. terms J and L in Figure 3.3) and more general functions are present in more sequences (e.g. terms A and B in Figure 3.3). This process generates a subalignment representing each of the GO functions present in the set of annotated homologues. These are used to identify conserved residues associated with each function, which in turn are used to infer function. Only subalignments with three or more sequences are considered for prediction, any terms with fewer sequences are disregarded.

3.3.4. Identifying Conserved Residues

The subalignments generated provide a representation of how the sequences with the same GO function are related. While all the sequences identified are homologues, their sequences have diverged and it is important to identify the features that they share. It is generally considered that residues important for the structure and or function of a protein are under greater evolutionary constraints and therefore less likely to mutate. The conservation scores described in section 1.3 can therefore be used to identify positions that are conserved within each subalignment and may play a functional role in the protein. ConFunc uses a Vingron type sequence weighting method (Valdar, 2002; Vingron and Argos, 1989). Each sequence is weighted according to the average distance between it and the other sequences in the alignment. The weighting w_i of sequence i in a group of n sequences is

$$w_i = \frac{1}{n-1}\sum_{j\neq i}^{n} 1 - d(i,j) \qquad \textbf{3.1}$$

where $d(i,j)$ is the sequence identity of sequences i and j. The conservation of each position is calculated by summing the substitution scores of each pair of residues within the alignment, such that the conservation of position x in a subalignment of n sequences is given by

$$C_x = \frac{1}{\sum_{i}^{n}\sum_{j>i}^{n} w_i w_j} \sum_{i}^{n}\sum_{j>i}^{n} w_i w_j sub(i_x, j_x) \qquad \textbf{3.2}$$

where $sub(i_x,j_x)$ is the substitution score for amino acids in sequences i and j at position x.

3.3.5. Calculating Conservation Z Scores

The conservation of each position with respect to the other positions in the alignment is calculated. This is done because ConFunc aims to identify functional residues in the GO term subalignments. Not all residues will have a functional role and if functional and structural residues are more likely to be conserved, then identifying the most conserved residues in the alignment may identify functionally or structurally important positions. A Z score is calculated for each position in the subalignment, for position x as

$$Z_x = \frac{C_x - \overline{C}}{\sigma} \qquad 3.3$$

where C_x is the conservation score at position x, $\overline{C}$ is the average conservation value of all the positions in the subalignment and σ is the standard deviation. All residue positions with a Z score greater than a given threshold are considered to be potentially functionally important residues and used for the scoring of the GO term PSSMs against the query sequence. During development the use of different Z score thresholds was investigated (see section 4.3), as well as the ability to identify functional residues using this approach (section 7.3).

3.3.6. Generating Position Specific Scoring Matrices (PSSMs)

PSSMs (see 1.2.5) are generated for each of the GO term subalignments to capture the amino acid composition at each position of a GO term subalignment for comparison with the query sequence. The same method, as described in section 1.2.5, for PSI-BLAST PSSM calculation is used.

3.3.7. Scoring Query Sequences against PSSMs

The query sequence is scored against each GO term specific PSSM for only the positions that are above the Z score conservation threshold (see 3.3.5). The score S, of a sequence against a PSSM is

$$S = \sum_{i=1}^{n} P_{ik} \qquad 3.4$$

where P_{ik} is the value in the PSSM at position i for residue k, where k is the residue at position i in the query sequence.

3.3.8. Calculating GO Term E-values

Having calculated the score for the query sequence against each GO term PSSM, it is necessary to assess the significance of the scores obtained. E-values are calculated to do this. The approach used is similar to the statistics of sequence alignments that was introduced in section 1.2.4. In this case e-values are calculated by shuffling the query sequence. The query sequence is shuffled to generate two thousand randomly rearranged sequences. Expectation values are calculated by fitting the scores from the shuffled sequences to an extreme-value distribution using maximum likelihood fitting as described by Eddy(1997). The probability ($P(S \geq x)$) of obtaining a score, S, greater than x (the PSSM score for the query sequence) is calculated using the estimates of λ and μ (where λ and μ are the scale and location parameters of the extreme value distribution respectively) from the maximum likelihood fitting as shown in equation 3.5. E-values are then calculated using equation 3.6, where n is the number of sequences considered, which in this case is 2000. The e-values calculated indicate the significance of the query sequence score obtained compared to the scores obtained for scoring random sequences against the same positions in the PSSM.

$$P(S \geq x) = 1 - \exp(-e^{-\lambda(x-\mu)}) \qquad 3.5$$

$$E(x,n) = n(P(S \geq x)) \qquad 3.6$$

The fitting of scores to extreme value distributions was tested for a subset of 10,000 GO terms from the test set. Initially the fit was assessed using the Kolmogorov-Smirnov test (KS test; Mendenhali, et al., 1999). As the extreme value distribution parameters μ and λ are estimated from the data it necessary to perform the KS test using bootstrapping (Romano, 1988). The KS statistic for the experimental data is compared to the KS statistic obtained from 1000 randomly generated samples from an extreme value

distribution with parameters μ and λ. p-values are calculated as the fraction of random sample KS statistics that the experimental KS statistic is less than. So a p-value of 0.2 means that the experimental KS statistic is less than the KS statistic for 200 of the 1000 random samples. The fit to the extreme value distribution was tested at a number of different significance levels as shown in table 3.1. The test shows that only 14% of the fitted data are significantly different from the extreme value distribution they are fitted to at the p=0.05 significance level. Further nearly 62% percent have p-values greater than 0.4.

There are still however, 38% of extreme value fitted distributions for which the p-value is less than 0.4. Visualisation of some of the fitted distributions to the observed data suggest that the observed distribution of scores deviates most from the fitted extreme value distribution in the lower tail and centre of the distribution. There is greatest agreement in the upper tail of the distribution. An example of this is shown in Figure 3.4. In this example the KS test p-value for the fitting of the extreme value distribution was 0.002, however there appears to be good agreement between the data and the fitted distribution at higher values. The KS test statistic is calculated as the largest deviation between the empirical cumulative distribution function (ecdf) of the observed data and the cumulative distribution function (cdf) of the distribution that the data are being compared to. The ecdf and cdf of the data and fitted distribution for this example are

p-value (significance level)	*0.05*	*0.2*	*0.4*
% datasets that fit EVD	86.2%	74.7%	61.9%

Table 3.1 Kolmogorov-Smirnov test of fitting of random scores to the extreme value distribution. The percentage of the 10,000 sets of random scores that have KS test p-values greater than three different threshold is shown.

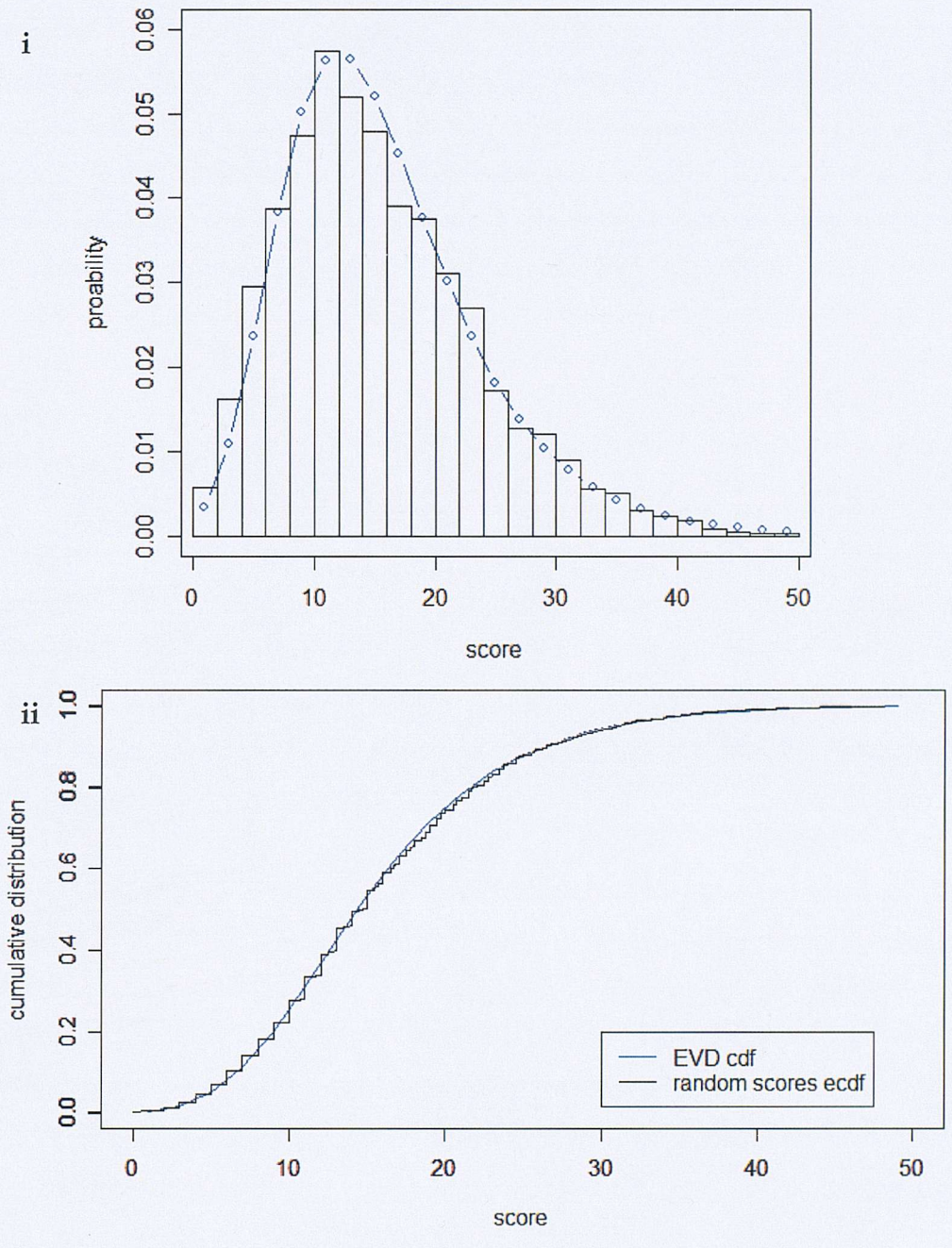


Figure 3.4 Fitting the random scores to an extreme value distribution. The example shown is for Swiss-Prot sequence Q64261 and GO term GO:0005021. i) Histogram of the observed probability density of the random scores. The probability density function of the fitted extreme value distribution with parameters μ=12.025 and λ = 0.155 is plotted in blue. ii) The empirical cumulative distribution function (ecdf) of the random scores is shown in black. The cumulative distribution function of the fitted EVD is shown in blue.

shown in Figure 3.4ii. At lower values the observed data has greater deviations from the fitted extreme value distribution cdf (which results in a low p-value) but there is much greater agreement between the two as the score increases. While it is preferable for there to be a good fit to the extreme value distribution over all values, it is the upper tail area where it is most important that there is agreement between the data and the fitted distribution. This is the area primarily used for calculating the GO term e-values. As will be described in section 3.3.9, only GO term e-values less than 0.001 are considered for prediction by ConFunc, therefore as the e-value is 2000 times the p-value (equation), then the minimum p-value considered is $5x10^{-7}$, which falls far in the upper tail of the distribution.

It is not practical to systematically visualise the data and fitted distributions for each case and it is desirable to use a test to quantify the agreement in the upper tail of distribution. To do this a version of the Anderson-Darling test, which up weights the contribution of the upper tail, was used to assess if the fit of the data is better in this area (Chernobai, et al., 2005). Like the KS-test, this test was performed using simulation, comparing the Anderson-Darling statistic to randomly generated samples from extreme value distributions with parameters μ and λ.

All EVD fitted distributions with KS test p-values less than 0.4 were tested. The goodness of fit was first assessed using the standard Anderson-Darling test; as for the KS test the p-values were predominantly less than 0.4. However, when the upper tail is up weighted a large increase in p-values is observed, with 97% (3695 of 3810 GO terms) having p-values greater than 0.5. This result demonstrates that there is good agreement between the data and fitted extreme value distribution in the upper tail region of the distribution.

3.3.9. Inferring GO Function from E-values

The steps described in the previous sections have resulted in the generation of e-values associated with a set of GO terms that are potential functions of the query sequence. The GO term e-values represent the significance of the similarity between the query sequence and the conserved residues associated with that term. These scores must be used to

discriminate between those functions that should be inferred for the query sequence and those that should be rejected as functions of the query. A number of approaches were considered to identify how best to discriminate between correct and incorrect GO terms. They range from simple thresholds to more complex methods that combine data other than the e-values and are described in the following subsections.

3.3.9.1. A Simple E-value Threshold

The annotations present in the homologues identified by PSI-BLAST are a pool of potential terms that can be inferred for the query sequence. In all examples, the term 'annotated terms' refers to those GO terms that the query sequence is annotated with.

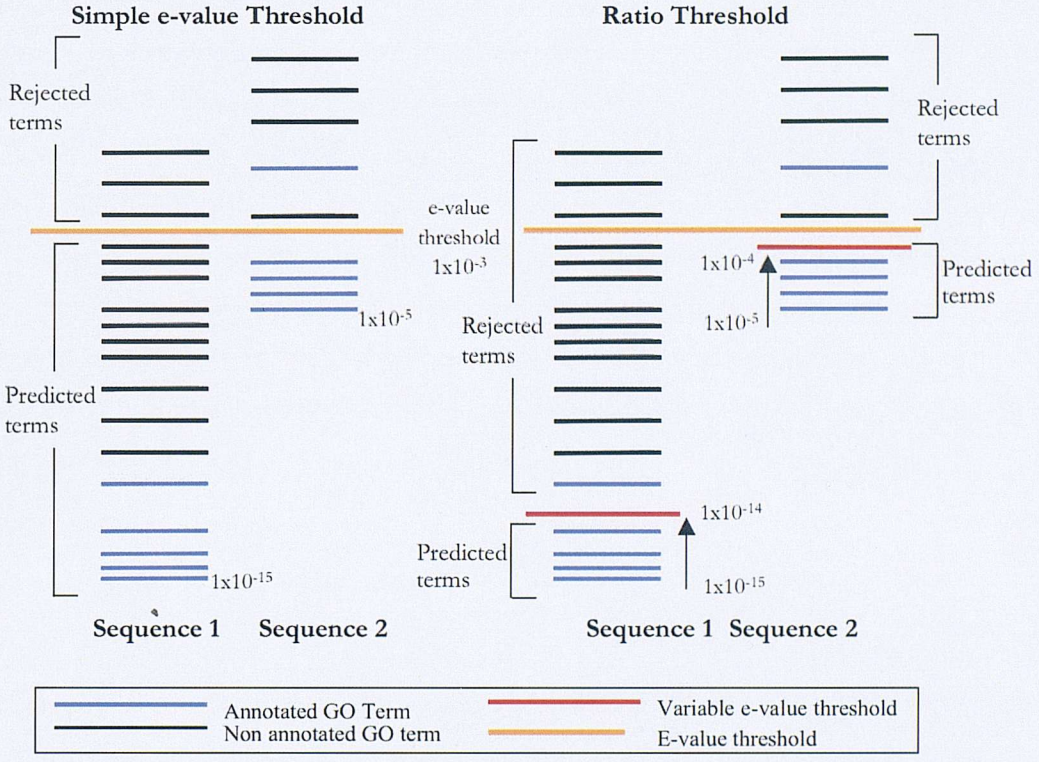


Figure 3.5 The e-value and ratio thresholds. The e-value and ratio threshold are shown for two example sequences. Blue lines represent the e-value of GO terms annotated for the sequence. GO terms present in the set of homologues but that are not annotated for the query sequence are shown in black.

Non-annotated terms are any other terms that the query sequence is not annotated with. Predicted terms are those that are predicted by ConFunc as functions of the query sequence, while rejected terms are those that are not inferred as functions of the query by ConFunc. The aim of any function prediction method is to maximise the prediction of annotated terms while minimising the prediction of non-annotated terms. The simplest way to determine the functions that are inferred for a query sequence is to reject all e-values above a given threshold and accept those below the threshold and annotate the query with them (Figure 3.5). Inferring a function also implies that all of its parent functions are inferred, this occurs regardless of their individual e-values.

3.3.9.2. The Ratio Threshold

An approach used by other function prediction methods, is not to use a threshold for the value that they calculate to indicate the likeliness of certain function but to simply predict the top result (Engelhardt, et al., 2005) or the top n terms (Jones, et al., 2005). As will be shown (see section 4.2.1) the range of e-values obtained between different sequences varies considerably. So an approach that considers only a set of the lowest e-values was developed. Instead of using an arbitrary approach of predicting the top n terms, the terms predicted were determined by their e-values. The ratio threshold takes into account the lowest e-value present for each query sequence and sets the threshold relative to this value. The ratio of the smallest e-value, e_{min}, and each individual e-value e_{xg}, is used such that the ratio R_{xg}, for GO term x is

$$R_{xg} = \frac{e_{xg}}{e_{\min}} \qquad 3.5$$

Using this threshold the GO term with the lowest e-value is predicted as a function of the query sequence and all other terms with an e-value ratio (R_{xg}) below a given threshold are also inferred as functions of the query sequence. Those with ratios greater than the threshold are rejected as functions of the query sequence (Figure 3.5). A range of R_{xg} thresholds were used from 5 to $1x10^7$. The ratio threshold also incorporates the approach of the simple e-value threshold, such that an e-values greater than $1x10^{-3}$ are automatically rejected, even if their GO term ratio is less than R_{xg}.

3.3.9.3. The Frequency threshold

The frequency threshold builds upon the ratio threshold. The frequency of each GO term, defined as the percentage of annotated sequences identified by PSI-BLAST that a GO is present in, is combined with the e-value ratio to assess which terms are inferred for the query sequence. In this way the frequency of a GO term determines the e-value ratio threshold for the term i.e. the maximum R_{xg} value that will be accepted. For example the ratio required by a GO term present in a small percentage of the homologues identified by PSI-BLAST requires a smaller ratio than one present in a high percentage of the homologues. This is demonstrated in Figure 3.6. The frequency threshold (R_{fg}) calculates an individual threshold for each GO term using equation 3.6 (rearranged in equation 3.7)

$$\log(R_{fg}) = \frac{n_g}{N} m \qquad \textbf{3.6}$$

$$R_{fg} = 10^{\left(\frac{n_g}{N} m\right)} \qquad \textbf{3.7}$$

where n_g is the number of sequences identified by PSI-BLAST annotated with GO term g, N is the total number of annotated sequences identified by PSI-BLAST and m is a factor for modifying the threshold. In equation 3.6 m is the gradient of a straight line (as shown in Figure 3.6). Like the ratio threshold all GO term e-values greater than $1x10^{-3}$ are automatically rejected. Different settings of the frequency threshold are obtained by varying m and it is varied from 2 to 40. A GO term g, is inferred as a function of the query sequence if its ratio R_{xg} is less than (or equal to) its R_{fg} threshold as shown below

$$g \text{ inferred if } R_{xg} \leq R_{fg} \qquad \textbf{3.8}$$

An example of the frequency threshold is shown in (Figure 3.6ii). The two graphs show the frequency threshold for m=9. The left graph shows how the frequency threshold is used to determine which functions are inferred for the query sequence. The straight line shows how the frequency threshold varies according to the frequency of a GO term in the set of homologues identified for the query sequence. Any terms below the line are

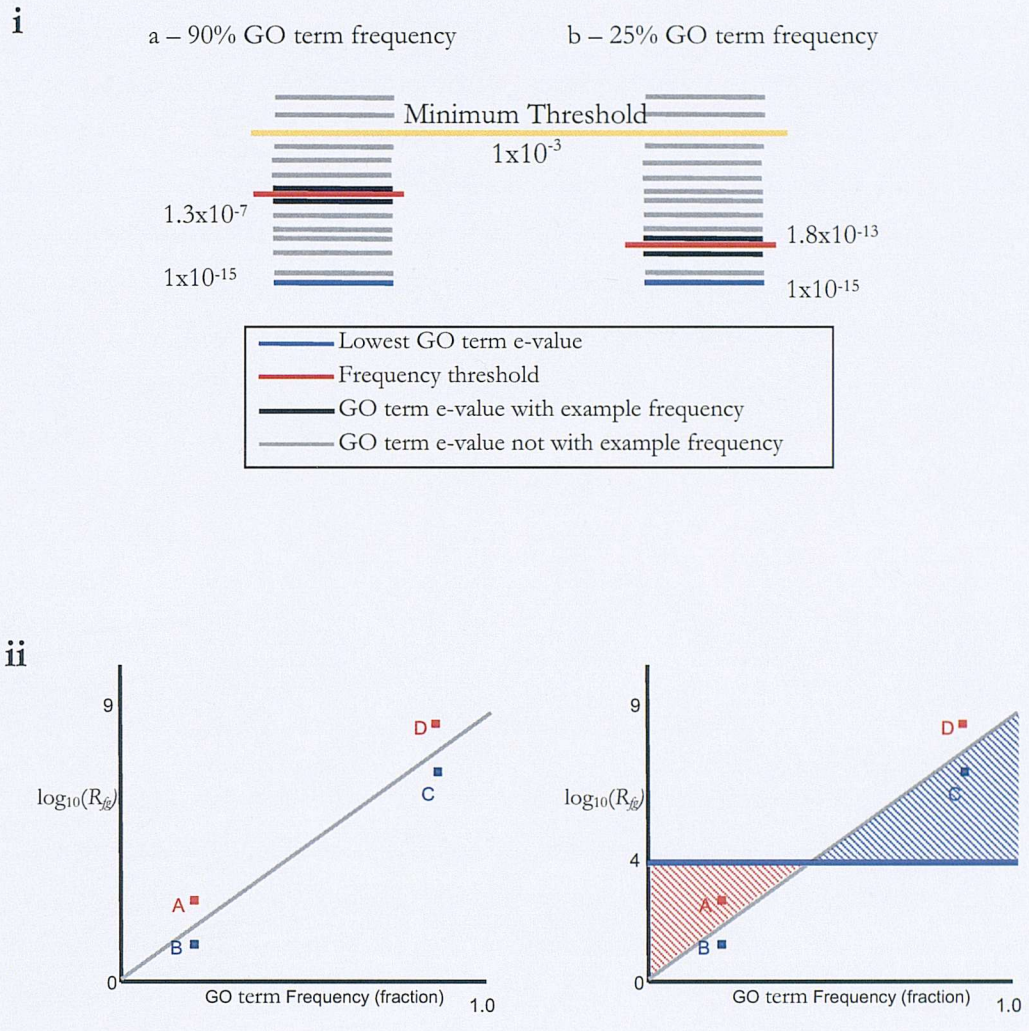


Figure 3.6 The frequency threshold. i). Each line represents the e-value of a different GO term, the properties of which are described in the key. The example on the left shows the frequency threshold for a term present in 90% of the homologues. On the right, the e-value ratio required when terms are present in only 25% of homologues. ii). A schematic representation of the frequency threshold. The GO term e-value ratio (y axis) is plotted against GO term frequency (x axis). The ratio threshold is greater for terms present in a higher percentage of sequences. Points A, B, C and D represent four different GO terms. Terms B and C are below the threshold line and are inferred as functions, whereas terms A and D are above the threshold line and are rejected as potential functions of the query sequence.

inferred as functions of the query sequence, while those above the line are rejected as functions of the query sequence. The graph on the right provides a comparison with the ratio threshold. The ratio threshold is the same for all GO terms independent of their frequency. The $1x10^4$ ratio threshold is shown in the figure, represented as a horizontal blue line. The frequency threshold is equivalent to a ratio threshold of $1x10^4$ when the GO term frequency is 0.44 (i.e. the term is present in 44% of the homologues) and this is the point at which the two line cross. In the example shown, terms A and B would be predicted and terms C and D rejected using the ratio threshold, whereas the frequency threshold predicts terms A and C and rejects terms B and D. The red striped area indicates the area in which the ratio threshold would infer terms but the frequency threshold would reject them and the blue area shows where terms rejected by the ratio threshold are predicted using the frequency threshold.

3.4. The Comparison Scheme Used Here

The concepts of compatible terms and hierarchical precision and recall have been introduced in section 1.6. The method for comparing predictions and annotations in this research builds upon these ideas. GO terms that are descendents of the most specific annotation of each protein are classed as compatible (see Figure 3.7). Accepting terms that are descendents of intermediate nodes in the annotation would allow too many different GO terms to be accepted as correct (see Figure 3.7). This definition of compatible terms is stricter than that used by other methods (Engelhardt, et al., 2005) and seeks to account for the potential that some existing annotations are not specific enough. The compatible terms are effectively ignored for the calculation of precision and recall, as shown by the examples in Figure 3.7. Precision and recall are defined as

$$\text{Precision} = \frac{|F(x) \cap G(x)|}{|G(x)|} \quad \textbf{3.9}$$

$$\text{Recall} = \frac{|F(x) \cap G(x)|}{|F(x)|} \quad \textbf{3.10}$$

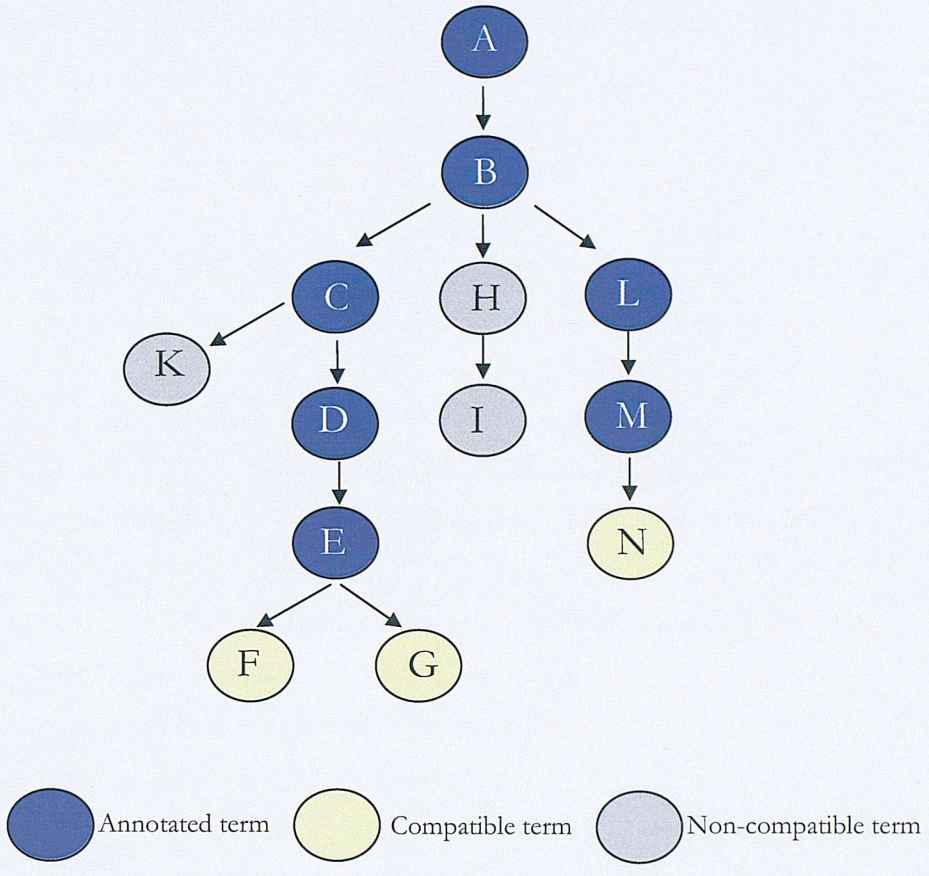
where *F(x)* is the set of predictions, which includes the parent terms of all the predictions made (e.g. if a prediction is GO:E with parent terms A,B,C,D, then *F(x)* is {A,B,C,D,E}. *G(x)* is the set of annotations (e.g. annotation GO:H whose parents are A,B,D,F,G then *G(x)* is {A,B,D,F,G,H}). While this calculation is similar to hierarchical precision and recall described in section 1.6, it differs from them for the case where a protein has more than one annotation as the hierarchical approach calculates a precision and recall value for each separate annotation and prediction and averages over them. The approach here simply takes the set of predictions and annotations and calculates the precision and recall values from them.

Results from the analyses in this research are generally displayed as precision recall graphs. Precision-Recall graphs provide a good assessment of the performance of methods where the class distribution is skewed (Davis and Goadrich, 2006). In this case, the number of annotations is much smaller than the number of potential functions that can be assigned. A perfect predictor would be represented by a point at 1,1 on a precision-recall graph, i.e. predicting all annotations without making any false predictions. Therefore the better a predictor the closer it will be to the top right corner of the graph. Receiver Operator Curves (ROC) are an alternative to the use of precision-recall graphs, however they use specificity and sensitivity defined as

$$specificity = \frac{TN}{(TN + FP)} \quad \textbf{3.11}$$

$$sensitivity = \frac{TP}{(TP + FN)} \quad \textbf{3.12}$$

where TN are true negatives and FN false negatives. Sensitivity is equivalent to recall. However, as the number of annotations is much smaller than the number of potential



Example Recall and Precision calculations

Query Annotation: **A,B,C,D,E,L,M**

Example 1	**Example 2**
Prediction: A,B,C,D,E,F,K	Prediction: A,B,C,D,E,L,M,H,I
TP: A,B,C,D,E (5)	TP: A,B,C,D,E,L,M (7)
FP: K	FP: H,I
Compatible: F (excluded from calculation)	Compatible: None
Recall = TP/N = 5/7 = 0.71	Recall = TP/N = 7/7 = 1.0
Precision = TP/(TP+FP) = 5/(5+1) = 0.83	Precision = TP/(TP+FP) = 7/(7+2) = 0.78

Figure 3.7 Example of precision and recall calculations. The GO annotation of a protein is shown in blue, with compatible and non-compatible terms displayed on the graph. Compatible terms are more specific functions than the most specific annotations (terms E and M in this case). Child terms of terms intermediate in the annotation are not compatible as shown by terms H, I and K. Below the graph the calculation of precision and recall is shown for two example predictions. The first example shows how compatible terms are considered.

functions, the true negatives will generally be much greater than the TP and FP values and as a result dominate the specificity calculation making it difficult to identify significant differences that are present between methods.

3.4.1. Generating a Test Set

It is important to have confidence in the annotations that are used by predictive methods and for the comparison of predictions with annotations. Swiss-Prot provides a hand-curated set of GO annotations, so release 47 was used to generate a protein sequence test set. GO annotated sequences in Swiss-Prot were identified using release 28 of EBI GOA. A number of filtering steps were performed to obtain a representative set of sequences with hand-curated annotations:

1. Initially all sequences with only IEA (inferred by electronic annotation) or NR (no record) GO annotations were removed.
2. Any sequences labelled as fragments in Swiss-Prot were also removed as were any containing 'X' in place of a residue.
3. A non-redundant test set of GO annotated sequences was generated from the remaining sequences, using CD-HIT (Li, et al., 2002) at 40% identity.
4. Finally any remaining sequences for which no GO annotated homologues were identified by three iterations of PSI-BLAST were removed.

The filtering steps resulted in a test set of 7150 sequences. This final step removes sequences for which neither ConFunc or PSI-BLAST would be able to make predictions. The test set has been generated in this way to reflect the different sequences that a user may wish to identify function for and the variation present within sequence space.

The test set contains proteins with annotations from all the main functional categories in the GO molecular function component. Catalytic activity and binding functions are the largest categories and account for 27% and 34% of the annotations respectively. Signal transduction, transcription regulation and transporter functions represent 12%, 8% and 7% of the annotations in the test set, with the final 12% of annotations split between the remaining molecular function categories.

3.4.1.1. Subsets of the Test Set

For some analyses subsets of the full test set have been used. One subset considers only those sequences for which all three methods (ConFunc, BLAST and PSI-BLAST) are able to make predictions for, thus giving an indication of how differently they perform in situations where they all make predictions. A second subset considers only sequences for which the top annotated BLAST hit has an e-value greater than $1x10^{-20}$. This, in conjunction with the removal of sequences with more than 30% sequence identity tested the performance of ConFunc in the twilight zone of sequence similarity where query sequences do not have close homologues.

3.4.2. Comparison with Other Methods

During benchmarking it is essential to compare ConFunc performance with other methods. It would be desirable to use other recently developed methods, such as SIFTER (Engelhardt, et al., 2005) or GOtcha (Martin, et al., 2004) for this comparison. However, the setup of these systems makes it difficult to ensure that query sequences are not used within the predictive process, as was encountered by Engelhardt et al (2005). Further ConFunc performance is assessed at low levels of sequence identity by removing homologues with greater than 30% identity and it would generally prove impossible to ensure this for third party methods. As it would not be possible to ensure a fair comparison with such methods, ConFunc was compared to annotation transfer by BLAST and PSI-BLAST. A limited comparison with Pfam based predictions was also performed, although Pfam had the advantage of not removing sequences with greater than 30% sequence identity. The settings for these comparisons are described below.

3.4.2.1. Comparison with BLAST and PSI-BLAST

The performance of ConFunc has been compared with the annotations predicted by the top BLAST and top PSI-BLAST hit for each query sequence against Swiss-Prot. The non-electronic set of GO annotations does not provide annotations for all sequences in Swiss-Prot, so where the top hit is not annotated, the first annotated hit is accepted. All sequences with greater than 30% sequence identity to the query were removed. To assess the range of performance obtainable by BLAST and PSI-BLAST the e-value cut off for inclusion of each top hit was varied between 0.1 to $1x10^{-100}$. For example, with an e-value

cut off of $1x10^{-10}$ predictions are only made for sequences that have a GO annotated homologue identified by BLAST (or PSI-BLAST) with an e-value of less than $1x10^{-10}$.

3.4.2.2. Comparison with Pfam

Pfam is a hand-curated set of sequence alignments and HMMs for protein families. It is therefore not possible to remove sequences with greater than 30% sequence identity from the Pfam alignments, so they may contain sequences with up to 100% identity with sequences in the test set. This gives Pfam an unfair advantage, however, the analysis was still performed to provide a comparison of ConFunc to a method other than BLAST.

Pfam release 17 was obtained. For each query sequence the Pfam mysql database was queried to identify all significant Pfam hits. Hits were converted to GO annotations using the PfamToGo mapping file downloaded from the Gene Ontology website (http://www.geneontology.org) in April 2005. Like the BLAST analysis, the Pfam e-value threshold for inclusion is varied to obtain a range of results.

Chapter 4

Development of ConFunc

4.1. Introduction

Chapter 3 described the individual steps within ConFunc and the different approaches that were investigated for each of these steps. This chapter presents the results obtained from these various approaches, including the effect of the following variables upon ConFunc performance:

- GO term e-value threshold
- Residue conservation Z score threshold
- Multiple sequence alignment method

In the analyses described ConFunc has been run for all sequences in the test set (see section 3.4.1). Performance is assessed using the recall and precision scoring scheme described in section 3.4, with ConFunc predictions compared to the same non-electronic GO annotations that are used for the predictive process. The results are plotted as precision-recall graphs to enable easy comparison between the results. In most cases the precision-recall graphs are plotted over the full possible range (0-1), to indicate clearly each methods range of performance. Occasionally the graphs will zoom into the area of most interest. This is done for clarity and to emphasise the performance differences between methods. Importantly, as the development of ConFunc is focussed on predicting function for sequences without high identity homologues, all homologues identified by PSI-BLAST with greater than 30% sequence identity with a query sequence, are removed from the analysis.

The statistical significance of different results is often tested using the McNemar test (McNemar, 1947). The McNemar test assesses the statistical significance between two methods by considering the correct predictions for each method that are incorrectly predicted by the other. It evaluates the probability of χ^2 with one degree of freedom, as

$$\chi^2 = \frac{(b-c)^2}{b+c} \quad 4.1$$

where b is the number of times that the first method makes an incorrect prediction while the same prediction is made correctly by the second method. Similarly, c is the number of times that the second method makes an incorrect prediction that is correctly predicted by the first method.

4.2. Inferring GO Functions from GO Term E-values

Using the e-values calculated by ConFunc to determine the functions inferred for a sequence is the last step in the predictive process. The threshold used for this has the greatest effect upon the results obtained and is therefore analysed first, in order for the best performing threshold to be used for the subsequent analyses. ConFunc was run with the following settings:

- $1x10^{-8}$ PSI-BLAST e-value threshold
- Residue conservation Z score threshold of 1.5
- MAFFT multiple sequence alignment

4.2.1. A Simple E-value Threshold

The results obtained using a simple e-value threshold (section 3.3.9.1), where a GO term is predicted if its e-value is smaller than a given threshold are plotted in Figure 4.1. The data points are obtained by varying the e-value threshold between $1x10^{-2}$ and $1x10^{-20}$. Using this simple e-value threshold it is only possible to obtain higher levels of precision with very low recall and vice versa. The greatest recall obtained is 0.57, at which precision is 0.14, meaning that for every correct prediction eight incorrect predictions are made. This performance is obtained at a very high e-value threshold (0.01), with ConFunc making predictions for 4711 of the 7150 sequences in the test set. Predictions are not made for the remaining sequences in the test set because either their GO term e-values are above this threshold or, as in the majority of cases, although PSI-BLAST identifies

GO annotated homologues for the sequences, insufficient sequences with similar annotations are identified. A minimum of three sequences is required to generate a profile for a GO term.

The highest level of precision (0.67) is obtained with recall of only 0.02 using a $1x10^{-13}$ threshold. Only 161 sequences in the test set have e-values below this threshold, explaining the low recall obtained. At lower e-value thresholds ($< 1x10^{-13}$) predictions are made for fewer sequences still and a reduction in precision is observed. Investigation of specific results suggests that they are heavily influenced by a few sequences. In these cases it appears that true functions of the proteins are predicted but are not present in the annotations. They are therefore classed as false positives and reduce the precision. This problem is observed in other analyses and considered in detail later (see section 5.3.3).

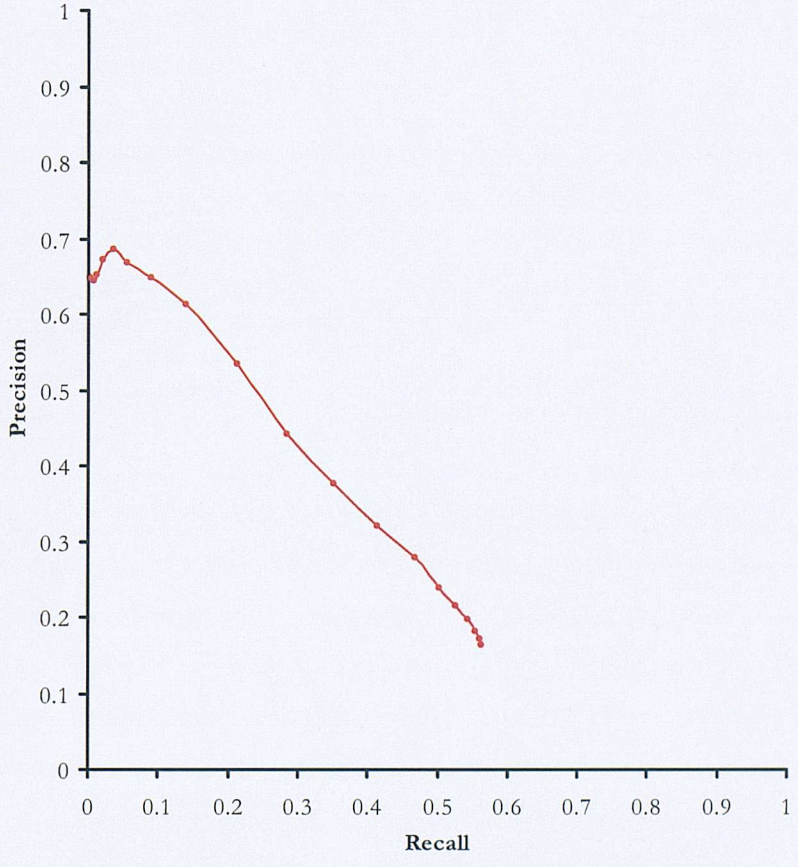


Figure 4.1 ConFunc performance using a simple e-value threshold. Precision-Recall graph for ConFunc results using a simple e-value threshold.

The e-value threshold cannot be used to differentiate successfully between correct and incorrect functions. The distributions of GO term e-values differ for annotated and non-annotated terms (Figure 4.2). However, the number of non-annotated terms is far greater than the number of annotated terms. Only at low e-values are there more annotated than non-annotated terms.

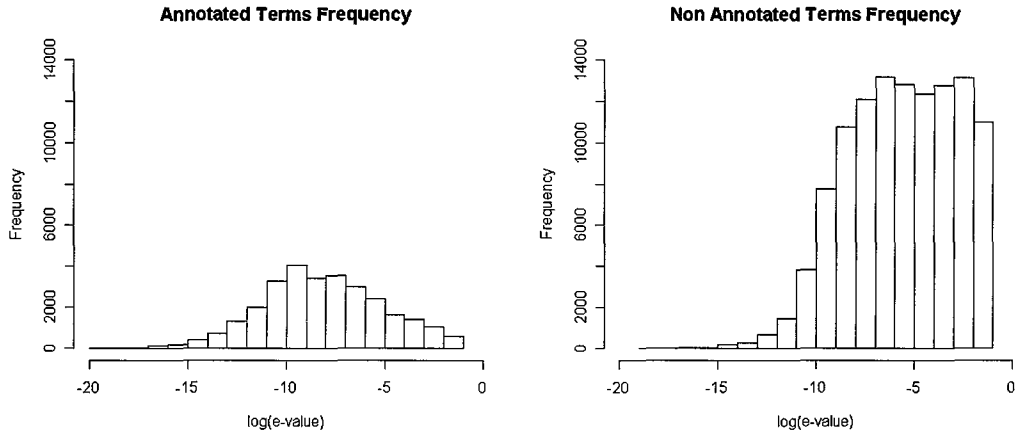


Figure 4.2 Distribution of GO term e-values. Histograms of $\log_{10}$(e-value) are plotted for both annotated and non-annotated terms for all sequences in the test set. Terms that are annotations of the query sequence (i.e. terms that are either true positives or false negatives) are shown in the annotated terms graph on the left. Terms that would be false positives if they are predicted (and True negatives if not predicted) for the query sequence are shown in the Non Annotated terms graph.

The range of e-values obtained for different sequences varies considerably as demonstrated by the distribution of lowest e-values (the lowest GO term e-value for each sequence) shown in Figure 4.3. This helps to explain why the e-value threshold does not discriminate between correct and incorrect GO terms. For sequences with very small e-values, high e-value thresholds will correctly predict the annotated terms but also potentially many other incorrect terms as shown in Figure 3.5. If the threshold is reduced then the number of false positive predictions for sequences with small e-values is also reduced (e.g. sequence 1 in Figure 3.5), but as a result predictions are not made for sequences with higher e-values (e.g. sequence 2 in Figure 3.5). This variation in e-values between sequences is not unusual, with similar variation observed for sequences

compared to Pfam HMMs using HMMer. It is necessary to use family specific e-value thresholds to discriminate better between e-values (Bateman, et al., 1999). Other function prediction methods also encounter the need for protein family specific thresholds (C Orengo personal communication).

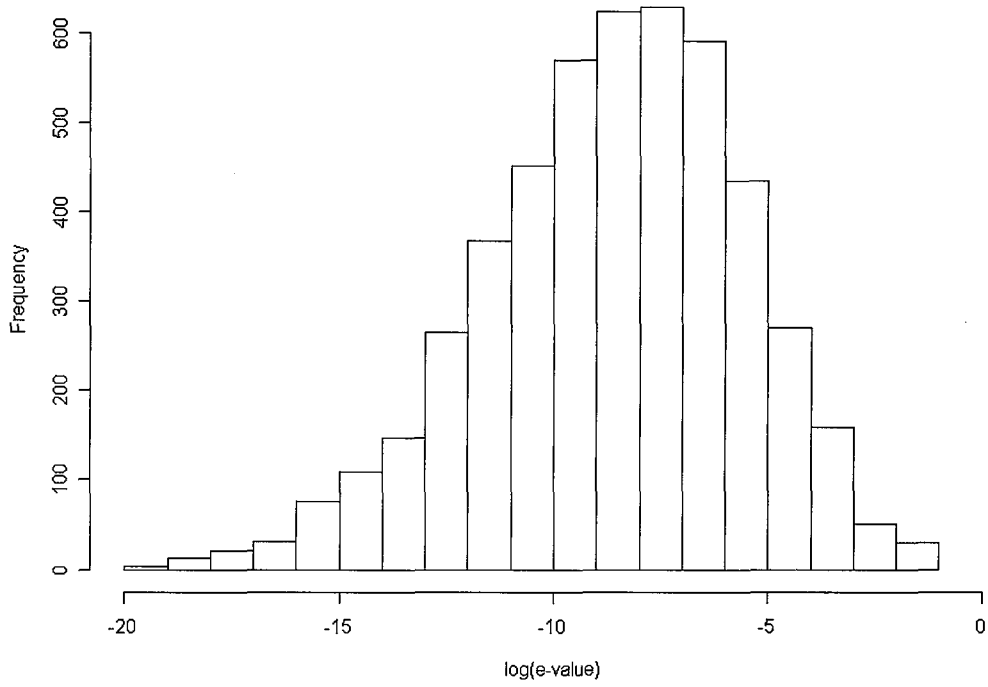


Figure 4.3 The distribution of lowest GO term e-values. The histogram shows the distribution of the smallest GO term e-value obtained for each of the sequences in the test set. The $\log_{10}$(e-value) is used on the x axis.

4.2.2. Ratio Thresholds

The ratio threshold (see section 0) attempts to account for the problems associated with the e-value threshold. A maximum e-value threshold is retained but it is kept constant at $1x10^{-3}$. The use of a constant maximum threshold ensures that ConFunc will make predictions for all sequences with an e-value smaller than this threshold. The ratio threshold then determines the functions that are inferred for these sequences. The ratio

threshold varies for different sequences because it uses the ratio between each GO term e-value and the lowest GO term e-value. Compared to using the simple e-value threshold, the ratio threshold obtains greater precision at equivalent levels of recall over its full range of performance (Figure 4.4), identifying the same proportion of the annotations, while making fewer errors. The greatest precision is obtained using the lowest ratio threshold setting of 5 and the precision decreases (recall increases) as the ratio threshold is increased. The maximum precision obtained using the e-value and ratio thresholds is similar, 0.67 and 0.71 respectively, however, the ratio threshold obtains 15 times greater recall (0.02 compared to 0.31) at this level of precision. This difference is statistically significant using the McNemar test at a $p=0.001$ significance level. The increase in recall is possible because by using the ratio threshold ConFunc is able to make predictions for 4711 sequences. The simple e-value threshold is only able to obtain high precision by accepting only low e-values, present in only very few sequences. The ratio threshold removes many of the GO term e-values of non-annotated terms that are smaller than the maximum threshold but are not close to the lowest GO term e-value.

The data points with low precision and high recall represent lax thresholds where GO terms with large e-value ratios are inferred for the query sequence. The simple e-value threshold results show the maximum recall that ConFunc is able to obtain at different e-value thresholds (i.e. the percentage of annotations in the test set that have GO term e-values below a certain threshold). The ratio threshold still incorporates a set e-value threshold, above which e-values are automatically rejected. This removes GO terms associated with very poor e-values and is set to $1x10^{-3}$ for all analysis using the ratio threshold. When this threshold was used for the simple e-value threshold recall of 0.56 was obtained (Figure 4.1). Only 56% of the annotations in the test set have e-values less than $1x10^{-3}$, this represents the maximum recall possible using the ratio threshold under these settings. As the ratio threshold settings are relaxed, large reductions in precision are observed with only small increases in recall. This is due to the recall approaching the maximum limit and therefore most of the additional terms are false positives, resulting in the large reduction in precision.

Investigation of the GO term e-values for individual sequences also showed that terms represented by fewer sequences have a tendency to have lower e-values than terms

represented by many sequences. Functionally specific GO terms (distant from the root) are generally present in fewer sequences in the set of homologues for a given query, while more general functional terms have a higher frequency. Specific functions often have much smaller e-values than those of related more general terms. This is likely to arise because sequences that share specific functions are more closely related and show greater conservation than a group of sequences that only share more general function.

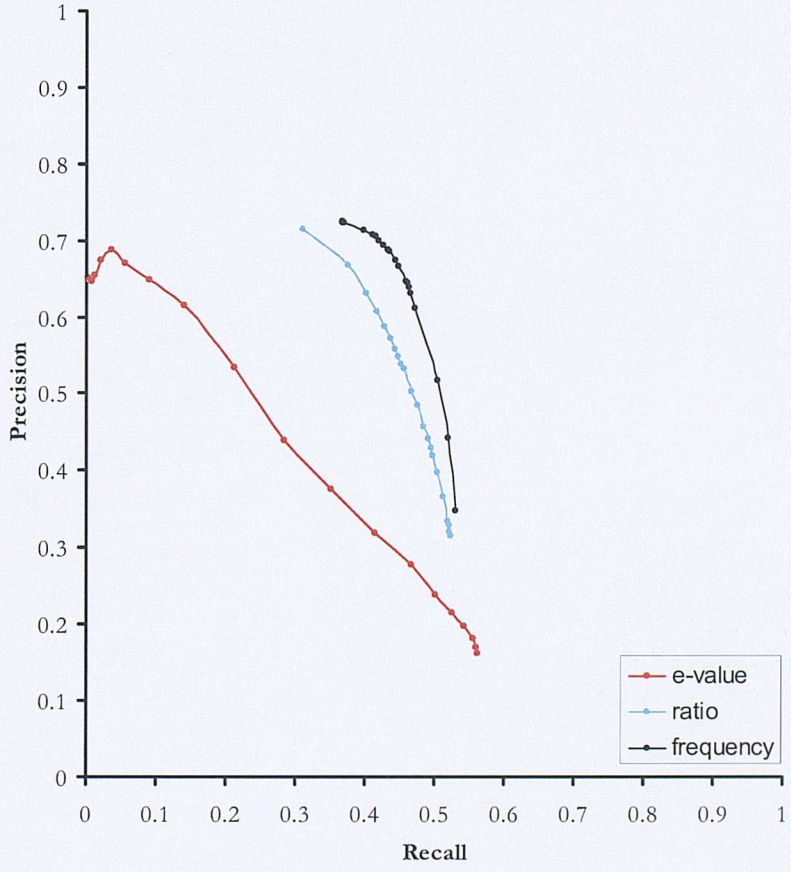


Figure 4.4 ConFunc performance using ratio and frequency thresholds. Precision-Recall graphs are for ConFunc results using the e-value, ratio and frequency thresholds to discriminate between the e-values associated with true and false GO term.

The frequency threshold (see section 3.3.9.3) builds upon the ratio threshold by incorporating the frequency of terms to account for the difference observed between the frequency of specific and general functions. It uses the frequency of each GO term to determine the ratio threshold for the e-value, such that the ratio threshold is lower for terms with low frequency and greater for high frequency terms. The frequency threshold shows an improvement over the ratio threshold (Figure 4.4), with all results obtaining greater recall at equivalent precision. The maximum precision is obtained with a frequency threshold setting of m=2 (see section 3.3.9.3). The maximum precision is increased to 0.72 with an associated recall of 0.36 compared to precision and recall of 0.71 and 0.32 for the ratio threshold. As for the ratio threshold, a wide range of different threshold settings were considered. As the frequency threshold is relaxed (i.e. the setting of m increased), the recall approaches it maximum value and there are large reductions in precision with only small increases in recall. The frequency threshold obtains the best performance and is used for all further ConFunc analyses.

4.3. Assessing the Residue Conservation Z Score

The residue conservation Z score determines the residues that are conserved and used for scoring the query sequence against the GO term PSSMs. The Z score threshold was analysed for values of between 0.0 and 2.5 to assess its effect upon performance. For all the thresholds considered, a similar range of precision of between 0.35 and 0.72 is achieved (Figure 4.5). This range of precision is obtained over different values of recall for the different thresholds. At low thresholds (Z=0.0 and 0.5) the recall is lowest. Further investigation of the results revealed that predictions were made for fewer sequences (Table 4.1) because they did not have GO term e-values below the maximum threshold (set at $1x10^{-3}$) and therefore no functions were predicted. This suggests that as less conserved positions are used, the GO term conserved residues, subsequent PSSM score and e-value are less representative of the function and the relationship between the GO term subalignments and the query sequence is lost.

The greatest recall is obtained using a Z score threshold of 1.5, although the performance at thresholds of 1.0, 1.8 and 2.0 is only slightly worse. Increasing the

threshold further to 2.2 and 2.5, results in a reduction in recall because fewer sequences have conservation Z scores above the threshold (Table 4.1). Some of the GO term e-values are also greater than the maximum e-value threshold and are not predicted which also acts to reduce the recall.

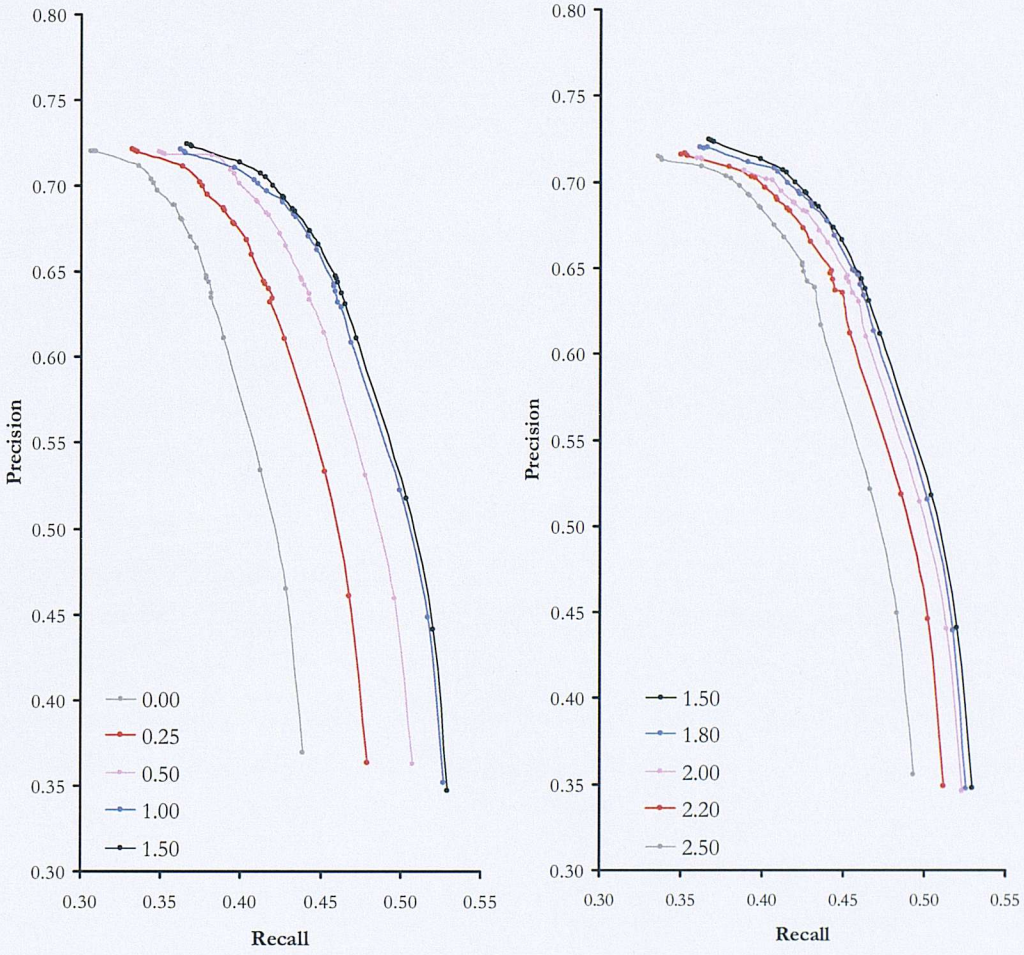


Figure 4.5 Assessing the residue Z score threshold. Precision-Recall graphs are plotted for ConFunc results using residue Z score thresholds ranging from 0.0 to 2.5. The results have been split into two graphs for clarity. The sequences were aligned using MAFFT. For clarity the graphs only show the area that results are present for. So the y-axis (precision) ranges from 0.30 to 0.80 and the x-axis (recall) from 0.30 to 0.55.

Z score threshold	*Number of sequences*
0.00	4252
0.25	4490
0.50	4625
1.00	4743
1.50	4711
1.80	4680
2.00	4661
2.20	4555
2.50	4412

Table 4.1 ConFunc coverage for different Z score thresholds. The number of sequences that ConFunc predicts function for is shown for each Z score threshold.

4.4. Assessing the Multiple Sequence Alignment Method

The previous section demonstrated the importance of ConFunc identifying conserved residues in order to make correct predictions. The quality of the conserved residues found is affected by the sequence alignment quality, so ConFunc performance was assessed using PSI-BLAST, MUSCLE and MAFFT alignments. Alignment using T-Coffee (Notredame, et al., 2000) was also considered but proved to be too processor and memory intensive, failing to generate alignments where the sequences were long or when there were a large number of sequences or both. It was therefore judged impractical to run ConFunc using T-Coffee and no further analysis was performed with it. Other studies have compared the performance of alignment methods (as described in section 1.2.6) by comparing the alignments generated against reference alignments. The task here is to assess the effect that the alignments from different programs have on ConFunc predictions.

PSI-BLAST does not perform a progressive alignment of sequences, it just combines the pairwise alignments obtained between the query sequence and each of the sequences identified from the database. Using the PSI-BLAST alignment in ConFunc has a considerable effect upon performance (Figure 4.6), reducing both the precision and recall obtained, compared to performance using MUSCLE or MAFFT. Similar performance is

obtained using either MUSCLE or MAFFT alignments, except at higher levels of precision, where MAFFT obtains higher precision at equivalent recall. These differences are statistically different using the McNemar test at the p=0.001 significance level.

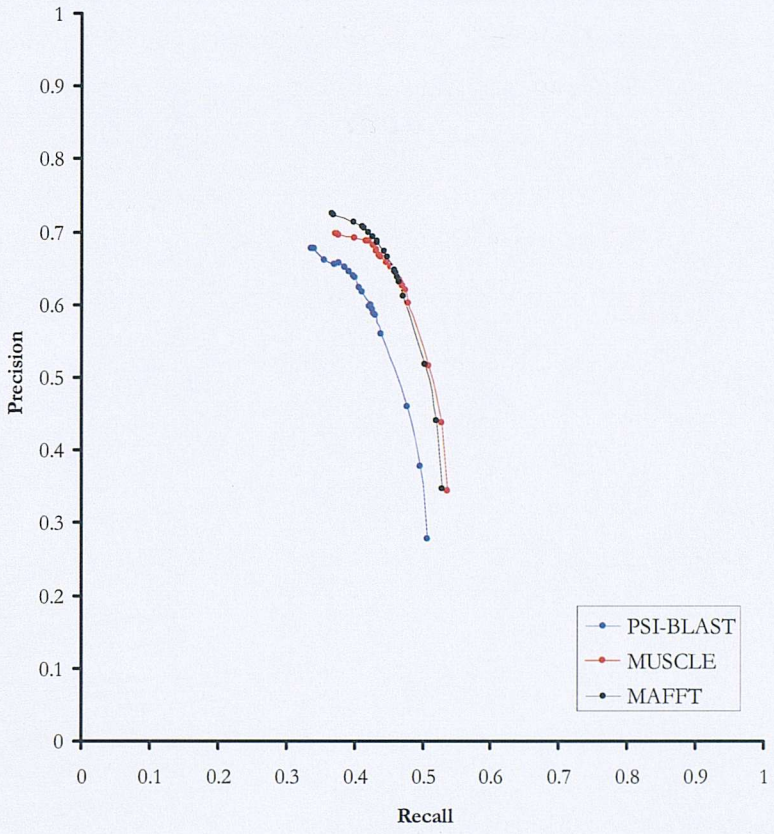


Figure 4.6 ConFunc performance using different sequence alignment methods. Precision-Recall graph showing the ConFunc performance using different multiple sequence alignments calculated using PSI-BLAST, MUSCLE and MAFFT.

4.5. Summary

This chapter has investigated the factors that affect ConFunc performance. Initially it was demonstrated that simple e-value thresholds cannot discriminate between correct and incorrect GO terms. The ratio and frequency thresholds were introduced and were found to provide much better performance. The ratio threshold accounts for the sequence

specific range of GO term e-values obtained. The frequency threshold also incorporates the percentage of homologues that each GO term is present in to account for differences in the PSSMs obtained due to differences in the number of sequences present in each subalignment.

The conservation Z score threshold also influenced the performance of ConFunc, with optimal results obtained at a value of 1.5. Similar ranges of precision are obtained at other threshold settings, but with lower recall. Finally, the sequence alignment method used by ConFunc was found to have a large effect upon performance. Better performance was obtained using progressive multiple sequence alignment methods, such as MUSLCE and MAFFT, compared to the PSI-BLAST alignment. Performance using MAFFT was better than with MUSCLE alignments, particularly at high levels of precision.

Chapter 5

Benchmarking ConFunc

5.1. Overview

When developing a function prediction method it is important to assess its performance against other methods. This chapter performs that analysis by comparing ConFunc to function predictions made by BLAST, PSI-BLAST and Pfam. Section 5.2 introduces the analyses that have been performed. Section 5.3 details the comparison of ConFunc with BLAST & PSI-BLAST. Section 5.4 describes an assessment of performance in the twilight zone of sequence similarity and section 5.5 details the comparison of ConFunc to Pfam. Many of the analyses reported in this chapter are also reported in Wass and Sternberg (2008), although in the publication, ConFunc uses MUSCLE used to align sequences. The analyses in this chapter use ConFunc results obtained using MAFFT generated sequence alignments.

5.2. Introduction

Measures of recall and precision are primarily used to assess the predictions made by ConFunc, BLAST and Pfam. In addition coverage is also quoted as the percentage of sequences in the test set for which predictions are made. In Chapter 3, predictions were compared with the same non-electronic annotations that were used in the predictive process. In this chapter some predictions are also compared with a larger set of annotations, which includes those inferred electronically. This larger set of annotations will be referred to as the electronic set and it also incorporates all the annotations present in the non-electronic annotations. The non-electronic annotations are limited by their requirement for clear evidence of function from literature or experiment and have limited coverage of Swiss-Prot. There is therefore the possibility that functions are not present in the annotations due to lack of evidence. The electronically inferred annotations provide a much larger set of annotations for which there is less confidence as they have not been

manually checked. However, the electronic annotations also contain correct functions that have been missed by the manual process. Many of the electronic annotations used by the EBI are generated from mappings between existing annotation systems (e.g. Swiss-Prot keywords and E.C. numbers) and GO, so there can still be a fair level of confidence in these annotations.

This approach of using a set of highly confident annotations for the predictive process (non-electronic in this case) and comparing the results with a large set of annotations (electronic in this case) from more varied sources with different reliability has previously been used in the assessment of SIFTER (Engelhardt, et al., 2005). In this case experimental annotations (with evidence codes IMP and IDA) from the Gene Ontology Annotation database (Camon, et al., 2004) are used to make predictions, which are then compared to the full set of annotations present in the GOA database.

All ConFunc analyses in this chapter are performed using the following settings:

- Residue conservation Z score threshold of 1.5
- Sequence alignment with MAFFT
- Frequency threshold

This chapter details a number of different analyses assessing ConFunc performance. First a detailed comparison of ConFunc with BLAST and PSI-BLAST annotation transfer is performed for a large test set of sequences. The differences between non-electronic and electronic annotations are also considered. An important analysis assesses ConFunc performance in the twilight zone of sequence similarity. This analysis was performed to identify if ConFunc could infer function in an area of sequence similarity where BLAST annotation transfer is likely to be limited.

Secondly, ConFunc performance over this test set is compared to Pfam based function predictions. Finally the ability of both ConFunc and BLAST to predict different types of function at different levels of functional specificity (GO level) is considered.

5.3. Comparison with BLAST & PSI-BLAST

ConFunc performance is first assessed against the annotation transfer for BLAST & PSI-BLAST top hits (method described in section 3.4.2.1). The annotation of the top BLAST or PSI-BLAST hit is transferred to the query sequence. A range of performance is obtained by varying the BLAST e-value threshold required for annotation transfer to be performed.

5.3.1. Assessment with Non-Electronic Annotations

The precision-recall graphs for the performance of ConFunc, BLAST and PSI-BLAST are shown in Figure 5.1. Both BLAST methods obtain greatest recall at high e-value thresholds (0.1). As the e-value threshold is reduced, the number of sequences with GO annotated homologues below the threshold reduces and so the coverage is also reduced (Table 5.1). A result of this is an increase in precision with reduction in recall as the query and top hit are more closely related and so more likely to perform similar functions. At low e-values, there are very few query sequences with annotated homologues below the e-value threshold and thus at this level the most precise predictions are made.

Figure 5.1 shows that BLAST generally outperforms PSI-BLAST. This occurs because a GO annotated homologue is not identified by BLAST for some query sequences, whereas one or more are retrieved by PSI-BLAST. PSI-BLAST identifies an annotated homologue for 993 more sequences than BLAST (Table 5.1). In such cases PSI-BLAST often identifies a remote homologue of the query sequence and transfers its annotation. Remote homologues are less likely to share the same function as the query sequence (see section 0), which may account for the difference in performance. For 4584 sequences in the test set the top annotated hit differs between BLAST and PSI-BLAST. This includes the 993 sequences for which BLAST does not identify a GO annotated homologue. For only 726 sequences, the different top hits have exactly the same annotation. 57% of the annotations present in the BLAST top hit are also present in the PSI-BLAST top hit and in reverse 41% of the PSI-BLAST annotations are present in the top BLAST hit. There is less agreement of the PSI-BLAST annotations with those from

BLAST because of the extra 993 query sequences that PSI-BLAST identifies annotated homologues.

ConFunc makes predictions for 4711 sequences in the test set and performs better at higher levels of precision than both BLAST and PSI-BLAST. The statistical significance of this difference in performance has been tested using the McNemar test. Individual results are compared separately with the most precise ConFunc result compared to the BLAST results that have the closest recall and precision to this ConFunc result. The result is significantly different from BLAST using a $1x10^{-10}$ e-value threshold (closest recall) at a $p=0.001$ level and also significantly different from BLAST performance using a $1x10^{-100}$ e-value threshold (closest precision) at the same $p=0.001$ significance level. At this level of precision, ConFunc recall is more than six times greater than BLAST, and BLAST coverage (percentage of test set that predictions are made for) is reduced to 6% compared to 66% for ConFunc.

At lower levels of precision, BLAST and PSI-BLAST obtain greater recall than ConFunc at equivalent precision. As descried previously (see section 4.2.2) these low precision points for ConFunc represent the use of very relaxed thresholds. As ConFunc approaches the maximum recall that it can obtain (see section 4.2.2), further relaxing the threshold mainly increases the number of false positives, resulting in the large drop in precision. These settings are not suitable when inferring function, but they are included to demonstrate a full range of performance. The trade off between precision and recall is demonstrated by the use of different thresholds for ConFunc and BLAST. Both recall and precision are important and a predictor aims to maximise both. However, in the case of function prediction there is a bias towards precision, as it is preferable to have a smaller set of mainly correct annotations, rather than a large set of annotations with a high proportion of errors. Indeed the acceptance of many false positive predictions can result in the accumulation of annotation errors in databases (Brenner, 1999; Devos and Valencia, 2001). However, in circumstances in which coverage is preferable then BLAST and PSI-BLAST may outperform ConFunc as they only require a sequence to have a single GO annotated homologue to be able to make a prediction.

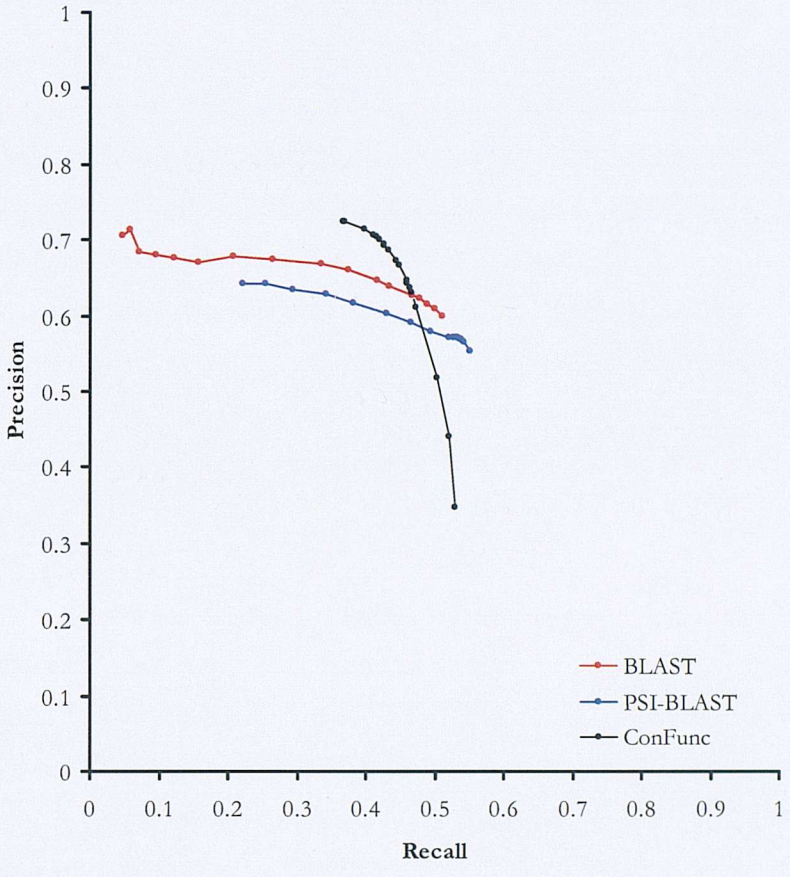


Figure 5.1 ConFunc benchmarking with BLAST and PSI-BLAST. Precision-Recall graph for ConFunc, BLAST and PSI-BLAST function predictions. Predictions are assessed against non-electronic annotations.

Method	*Coverage*	
ConFunc	4711 (66%)	
BLAST	min :428 (6%)	max : 6157 (86%)
PSI-BLAST	min :2314 (32%)	max : 7150 (100%)

Table 5.1 ConFunc, BLAST & PSI-BLAST Coverage. The coverage of each method over the test set is shown. For BLAST and PSI-BLAST the min value is the coverage using an e-value threshold of $1x10^{-100}$ and max with a 0.1 e-value threshold.

The recall values obtained for all three methods are low. However, all homologues with greater than 30% identity with each query sequence have been removed and as such overall performance is poorer compared to a standard case where no homologues have been removed. Further, recall is calculated over the complete test set of 7150 sequences. ConFunc and BLAST do not obtain one hundred percent coverage (Table 5.1), so the recall they can obtain is automatically reduced compared to PSI-BLAST, which can make predictions for all sequences in the test set. ConFunc does not obtain 100% coverage because it requires a minimum of three homologues annotated with the same term to generate a GO term subalignment and profile. So where only a few annotated homologues are identified by PSI-BLAST, ConFunc is often unable to make predictions.

To account for the differences in coverage between the methods, a similar analysis was performed using only the sequences in the test set for which all three methods make a prediction. The precision-recall graph for this analysis is shown in Figure 4.2. The increased recall of the methods is of most interest. This increase is greatest for ConFunc, which obtains recall between 0.54 and 0.77 compared to 0.37 and 0.53 in the previous analysis. This demonstrates that for the sequences that ConFunc is able to make predictions for, it performs better than both BLAST and PSI-BLAST.

In contrast to their performances against the full test set, BLAST and PSI-BLAST have similar performance for the sequences in the reduced test set. As previously stated, the poor PSI-BLAST performance across the full test set, is due to it using remote homologues to infer functions where BLAST does not identify an annotated homologue. This is shown in Figure 5.3 where PSI-BLAST performance is compared for the following three sets of sequences:

- The full test set.
- The reduced test set.
- Sequences that PSI-BLAST infers function for but BLAST does not.

The recall and precision of this last group of sequences is significantly worse than for the larger set of sequences. This shows that PSI-BLAST performance is hampered by its ability to identify remote homologues that do not share the same function as the query.

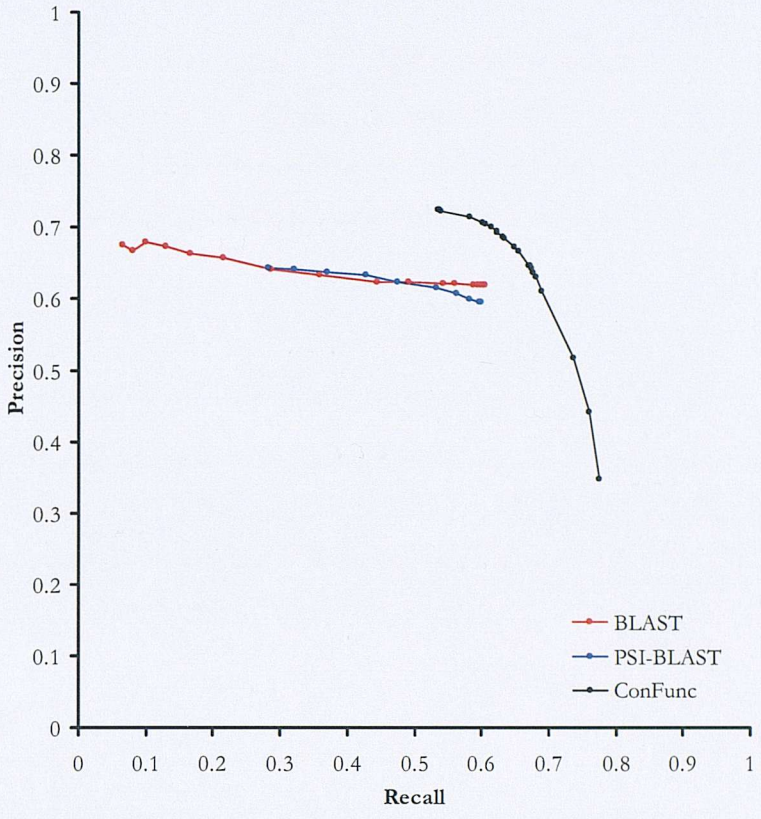


Figure 5.2 Assessment for reduced test set. Precision-Recall graph for sequences in the test set that all three methods (ConFunc, BLAST and PSI-BLAST) make predictions for. Predictions are assessed against non-electronic annotations.

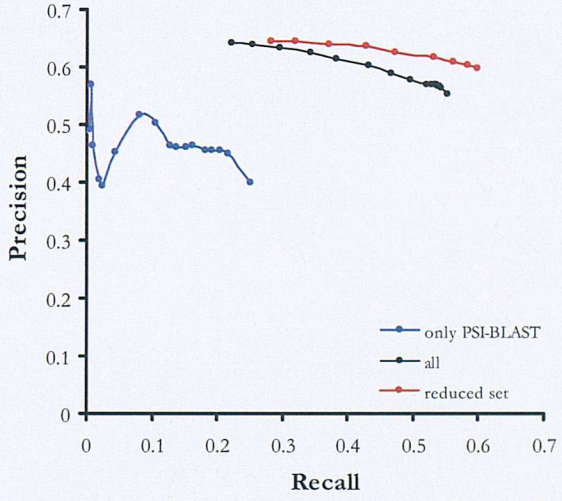


Figure 5.3 PSI-BLAST performance. The precision-recall graph for PSI-BLAST annotation transfer, considering subsets of proteins within the test set. The lines represent: *only PSI-BLAST* – sequences that only PSI-BLAST makes predictions. *reduced set* – sequences in the reduced test set, *all* –all sequences in the test set. Predictions are assessed against non-electronic annotations

The PSI-BLAST only set contains only 993 sequences, this reduces considerable as the BLAST e-value threshold is reduced. The fluctuations observed for this data set are a result of the small sample size. PSI-BLAST predictions are improved for the reduced test set because the predictions based on these remote homologues are removed.

5.3.2. Assessment with Electronic Annotations

The previous analysis compared function predictions to only non-electronic annotations. In this analysis, the predictions made by ConFunc, BLAST and PSI-BLAST are compared to a much larger set of annotations including those inferred electronically. Figure 5.4

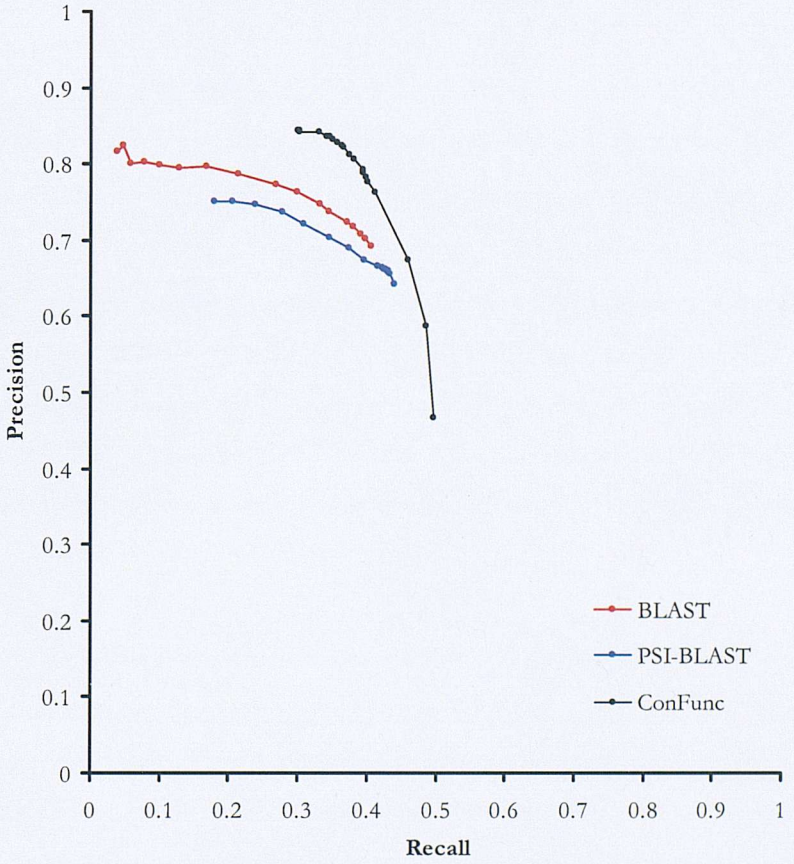


Figure 5.4 Precision-Recall analysis using electronic annotations. The precision-recall graph for ConFunc, BLAST and PSI-BLAST when predictions are compared to electronic annotations.

shows the range of recall and precision obtained by ConFunc, BLAST and PSI-BLAST when compared with this extended set of annotations. There is an increase in precision in all cases (compared with performance against non-electronic annotations - Figure 5.1). This indicates agreement between the electronic annotations generated from mappings of other functional annotation types (e.g. Swiss-Prot keywords and E.C.) and the predictions made by ConFunc and annotation transfer. A reduction in recall is also observed in comparison to the non-electronic annotation results because the increase in the total number of annotations present in the test set is greater than the increase of the number of true positive predictions. A clear difference between the predictive performance of ConFunc and BLAST is observed with ConFunc obtaining greater recall than BLAST at all levels of precision. The McNemar test was used to test the significance between the most precise ConFunc result and the BLAST results with equivalent recall and precision, ConFunc is significantly better in both cases at the $p=0.001$ level.

5.3.3. Differences between Non-Electronic and Electronic GO annotations

It is important to assess what causes the increased precision obtained when comparing predictions with electronic annotations. Is it is due to the electronic annotations containing correct functions, which the predictive methods have also identified or do the electronic annotations contain incorrect annotations, which have also been incorrectly predicted? The size of the test set makes it impractical to investigate the annotation of each individual sequence to identify the cause of the differences in performance when compared with different sets of annotations. Therefore a specific functional type is used to provide an example of one of the causes of these differences.

The GTPase enzymatic function provides an example of a difference between the non-electronic and electronic annotations. GTPases (GO term GO:0003924) hydrolyse GTP to GDP. They should therefore be annotated with this catalytic function and also with the GTP binding (GO:0005525) function. However, very few sequences annotated as GTPases are also annotated with the related binding annotation in the non-electronic set of annotations (Figure 5.5), demonstrating incompleteness in the non-electronic set of annotations. Most of these sequences are annotated with the binding function when electronic annotations are considered (Figure 5.5). While it might be clear to someone

using these annotations that a GTPase is likely to bind GTP, this difference in annotation has a greater effect upon the perceived performance of a function prediction algorithm. Predicting a GTP binding function for a GTPase that is not annotated with this binding function will be classed as a false positive prediction, therefore reducing its performance. For the majority of GTPase predictions, ConFunc also predicts GTP binding (40 out of

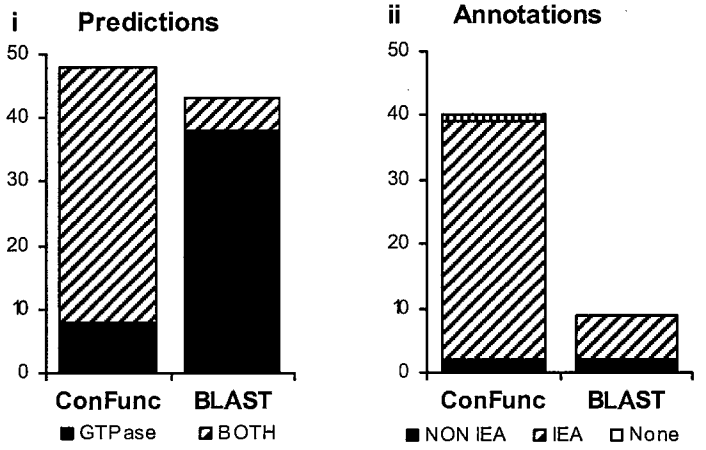


Figure 5.5 Prediction of GTPase functions. i) GTPase and GTP binding function predictions made by ConFunc and BLAST. GTPase predictions without GTP binding predictions are shown in black and sequences where both GTPase and GTP binding are predicted are shown with diagonal lines. ii) The annotations of the GTPase sequences predicted by ConFunc and BLAST. In black are the sequences that have both a GTPase and GTP binding annotation in the non IEA set, while those with both annotations in the electronic annotations are shown with diagonal lines.

48 see Figure 5.5). Only two of these sequences have non-electronic GTP binding annotations and as a result 38 of the 40 are classed as false positives. The electronic annotations include GTP binding for a further 37 of these sequences, so only one of the GTP binding predictions is classed as a false positive when compared to this annotation set (Figure 5.5ii). A similar pattern is observed for BLAST and PSI-BLAST predictions; of eight GTP binding predictions made for GTPase sequences, only one of these is present in the non-electronic set, while the remaining seven are all present in the electronic annotations (Figure 5.5). This shows that part of the increased precision obtained when electronic annotations are included in the assessment of predictions, is due

to the prediction of correct functional terms that should be present in the non-electronic set.

Similar annotation patterns have also been observed for other types of enzymes. For example there are 101 proteins with non-electronic ATPase annotations; only nine of these also have non-electronic ATP binding annotations, while a further 66 have electronic ATP binding annotations. NAD (nicotinamide adenine dinucleotide) is an important molecule in many biological functions and often acts as a coenzyme (Pollak, et al., 2007). The incompleteness of the GO annotations is highlighted by the absence of any proteins with non electronic NAD binding annotations. Further, only nineteen proteins have non electronic NADP annotations. Yet there are many non-electronic annotations that describe functions involving NAD, such as GO:0004022 (alcohol:NAD+ oxidoreductase), GO:0003954 (NADH dehydrogenase activity) and GO:0008137 (NADH dehydrogenase (ubiquinone) activity). In contrast there are 3868 NAD binding and 1259 NADP binding electronic annotations.

The analysis of predictions for GTPases also illustrates a difference between ConFunc and BLAST predictions. Both methods predict the GTPase function for a similar number of sequences (Figure 5.5i), but while ConFunc also predicts the GTP binding function for the majority of GTPase sequences, BLAST predicts GTP binding in very few cases. This is because BLAST transfers the annotation of the top hit, which often appears to be a sequence annotated as a GTPase but not as a GTP binding protein. ConFunc assesses all of the GO terms present in the set of PSI-BLAST homologues, giving it the potential to predict more and different terms than those present in the top BLAST hit. These differences result in more false positive predictions for ConFunc (when compared to non-electronic annotations), as BLAST only predicts eight GTP binding functions compared to 37 by ConFunc. The Gene Ontology now suggests that proteins annotated as GTPases should be considered for GTP binding annotations, so this problem will hopefully be reduced for newly annotated proteins.

5.4. Functional Annotation in the Twilight Zone

An important aim during the development of ConFunc was the ability to infer function for sequences which only have annotated homologues with low levels of sequence similarity. The previous analyses have assessed this scenario by removing homologues with greater than 30% identity to a query sequence. The following analysis assesses ConFunc performance for a more stringent set of proteins by only taking into account query sequences for which all three methods make predictions, and where the top annotated hit has a BLAST e-value greater than $1x10^{-20}$. This is in addition to the removal of all sequences with greater than 30% sequence identity. This results in a set of 1613 sequences from the original test set of 7150 sequences. These settings assess the ability of ConFunc to predict function in the twilight zone of sequence similarity.

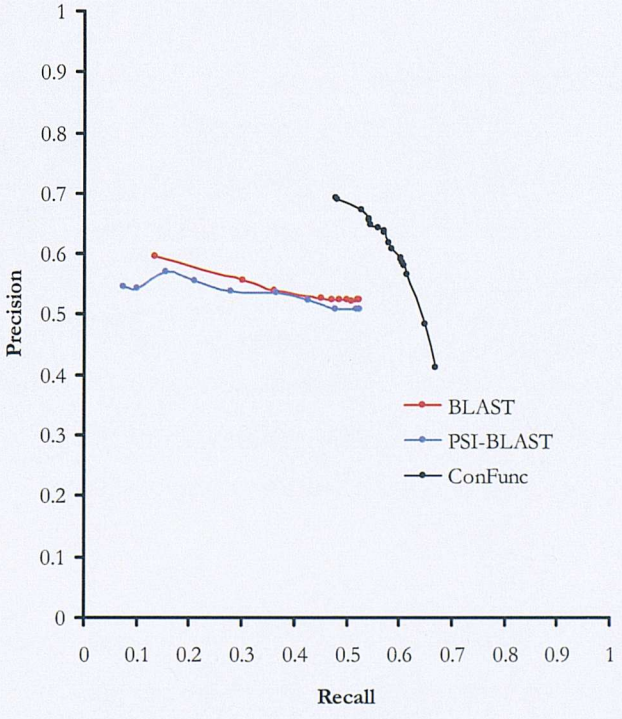


Figure 5.6 Functional annotation in the twilight zone. Precision-Recall graph for ConFunc, BLAST and PSI-BLAST predictions for sequences with only annotated homologues with BLAST e-values greater than $1x10^{-20}$ and also below 30% sequence identity. Predictions are assessed with non-electronic annotations.

Under these settings ConFunc performs much better than both BLAST and PSI-BLAST (Figure 4.5). The difference observed between ConFunc and BLAST is greater than in the previous analyses (see Figure 5.1 and Figure 5.4), with ConFunc obtaining levels of precision and recall that are not observed for BLAST. At high levels of precision, ConFunc reaches precision of 0.69 (values for comparison with non-electronic annotations), while BLAST is only able to reach a maximum precision of 0.59. Importantly, ConFunc recall is 0.48 compared to 0.14 for BLAST at these levels of precision, i.e. ConFunc is able to correctly infer a greater percentage of the annotations than BLAST and at the same time make fewer incorrect predictions. BLAST is able to obtain a comparable recall of 0.48 but only with precision equal to 0.52. Using the McNemar test, the performance of ConFunc at all settings is significantly better than BLAST performance at the $p=0.001$ level.

5.5. Comparison with Pfam

Identifying the Pfam hits for a sequence is a widely used approach for inferring function (see section 1.5.1.4) and is therefore a useful comparison to ConFunc. ConFunc performance has been assessed at low levels of sequence identity and unfortunately an equivalent assessment for Pfam is not possible. Pfam is a hand-curated library of sequence alignments and Hidden Markov Models that represent protein families and domains. The alignments and subsequent HMMs therefore contain sequences with more than thirty percent identity to the query sequences in the test set. Due to the hand-curated nature of the Pfam alignments it would not be possible or practical to recreate a version of Pfam that excluded such sequences. The Pfam analysis therefore has the unfair advantage of using sequences with higher levels of identity.

The method used to make Pfam based predictions is described in section 3.4.2.2. Predictions are only compared to non-electronic GO annotations as one of the mappings used for the EBI electronic annotations converts Pfam annotations to GO. Therefore, comparing the Pfam based predictions to the electronic set of annotations would, in some cases, result in comparing Pfam predictions to annotations originally made by mapping from Pfam. Despite Pfam's unfair advantage in this analysis, it does not outperform

ConFunc (Figure 5.7) with ConFunc obtaining greater recall at equivalent levels of precision. At the highest level of precision, Pfam obtains 0.71 precision with recall of 0.10 compared to 0.72 precision and 0.37 recall for ConFunc. Like BLAST, at this level of precision the coverage of Pfam predictions is low at 15% compared to 65% for ConFunc. Below a Pfam e-value threshold of $1x10^{-15}$, there is very little increase in precision but a gradual reduction in recall. The Pfam function predictions do outperform BLAST and PSI-BLAST at most settings (Figure 5.7). This difference is small and is likely to be due to the presence of close homologues in the Pfam alignments that have been used to make the predictions. BLAST/PSI-BLAST would outperform Pfam if they were not restricted to homologues with less than 30% identity with the query.

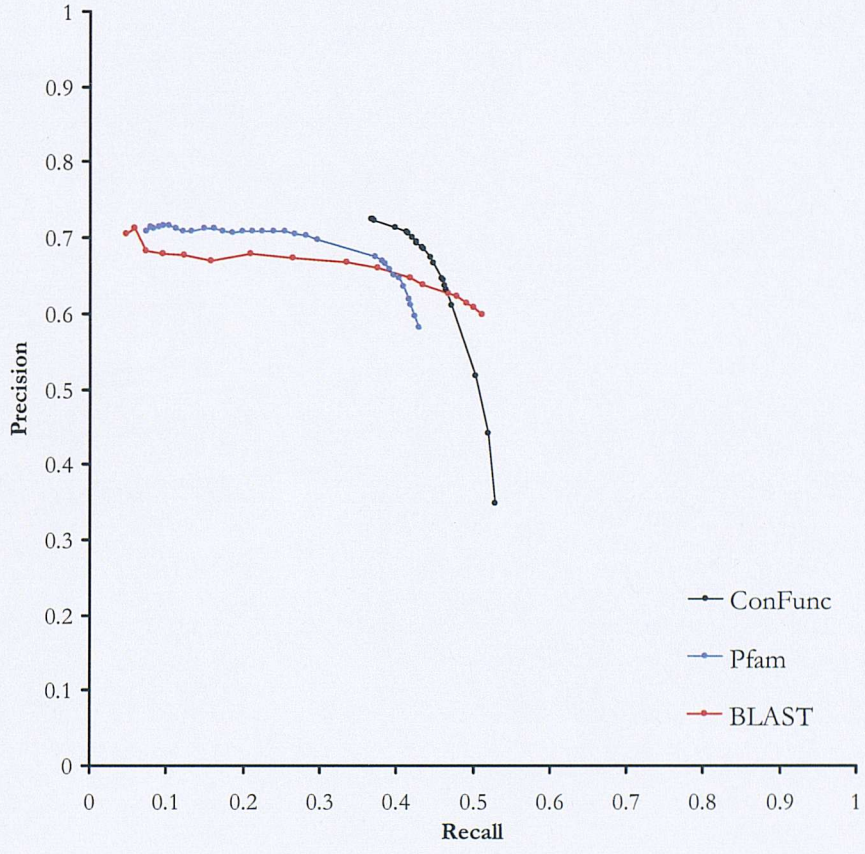


Figure 5.7 Comparison of ConFunc with Pfam. Precision-Recall graph showing the performance of Pfam, ConFunc and BLAST when compared to non-electronic GO annotations.

As the Pfam analysis does not simulate a low sequence identity scenario, it is surprising that the recall ranges obtained are similar to BLAST and ConFunc (Figure 5.7) This is likely to be caused by the variation in function within Pfam families, with family members often sharing similar general functions whilst a broader range of specific functions may be seen (Abhiman and Sonnhammer, 2005a). So, in many cases, only general functions are mapped from Pfam to GO. This hypothesis can be tested by assessing the performance of Pfam at different levels of GO. The results of this analysis showed reduced recall and precision between GO levels 2-5 compared to the predictions at level 1 (Figure 5.8). A different pattern is observed for GO levels 6 and 7. At these levels a large reduction in recall is associated with an increase in precision. The reduction in recall shows that Pfam

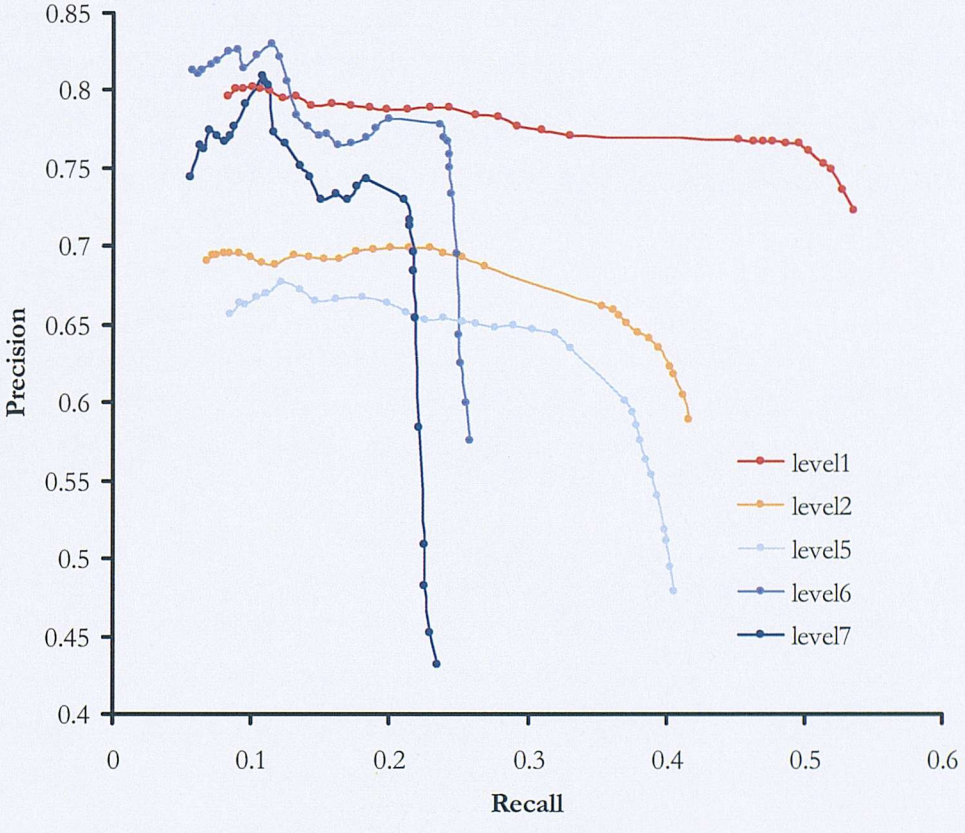


Figure 5.8 Pfam performance for individual GO levels. The precision-recall graph for Pfam performance at GO levels 1, 2, 5, 6, 7. Levels 3 and 4 are omitted for clarity, (they would appear between the lines for levels 2 and 5). Predictions are compared to non-electronic annotations.

can only infer functions at these GO levels for fewer sequences and the range of recall is considerably smaller than obtained by ConFunc or BLAST at the same GO levels (see section 5.7). For GO levels 6 and 7, there are initially large increases in precision with only small reductions in recall. This demonstrates that there are few Pfam hits with high e-values that infer function at these levels and where they do, they are generally incorrect, as there is very little reduction in the recall as the precision increases. After the sharp increase in precision, as the Pfam e-value threshold is reduced further, recall also reduces and is associated with fluctuations in precision, which reflects the composition of the sequences present below each threshold. This demonstrates that even Pfam hits with low e-values, may be incorrect. However, overall it is clear that if there is a low e-value Pfam prediction at a high GO level, then there is a good chance that it will be correct. A detailed analysis of the performance of ConFunc and BLAST at different GO levels is performed in Section 5.7. If the results at levels 6 and 7 are compared for all three methods, Pfam obtains lower recall than ConFunc and BLAST, but is able to obtain greater precision when only very confident Pfam hits are used. It is important to remember the Pfam predictions are not made with the same sequence identity restrictions as ConFunc and BLAST. So Pfam may be able to obtain greater precision at these levels, but it is only able to do this for a small proportion of sequences and also requires close homologues to do so.

5.6. Predicting Different Functional Types

The previous analyses have only considered the overall ability to predict GO functions and have not focused on the ability to predict different types of function within the molecular function category. The first level of GO terms, from the molecular function term, provides general functional groupings including: catalytic activity, binding and transcription regulator activity. It is important to consider the ability of ConFunc to predict different types of molecular function. The majority of annotations in the test set belong to the catalytic activity and binding categories (
Table 5.2), so the predictive performance for these types of function was considered in greater detail.

ConFunc performance differs considerably for catalytic and binding functions compared to the overall prediction of all functions as shown in Figure 5.9. For catalytic functions ConFunc obtains greater recall and precision than for all functions, while binding predictions obtain lower recall and precision. There are a number of potential explanations for these results. ConFunc associates conserved residues with GO terms to infer function. It is possible that the conserved residues identified for binding functions are unable to represent the function and therefore perform poorly when used by ConFunc. The converse of this would apply to catalytic functions, with ConFunc successfully identifying residues associated with catalytic functions and in turn using them to correctly infer function.

This scenario is plausible, however, similar differences in performance are also observed for BLAST based predictions of catalytic and binding functions (Figure 5.9). This suggests that the annotations themselves are also a source of the differences observed. Catalytic function is of major interest to the scientific community, and it is therefore likely that the catalytic annotations are well defined and complete. However, binding functions are often not the major function of a protein, so there is more chance that they are missed in an annotation, as has been shown for GTPase enzymes (see section 5.3.3). If this is the case, then the prediction of binding functions where they

Molecular Function Category	*Test Set Annotations*
Catalytic activity	34%
Binding	27%
Signal transduction	12%
Transcription regulation	8%
Transporter functions	7%
Other functions	12%

Table 5.2 Distribution of annotations in the test set. The percentage of annotations that are present for each of the main Molecular functional groups is shown.

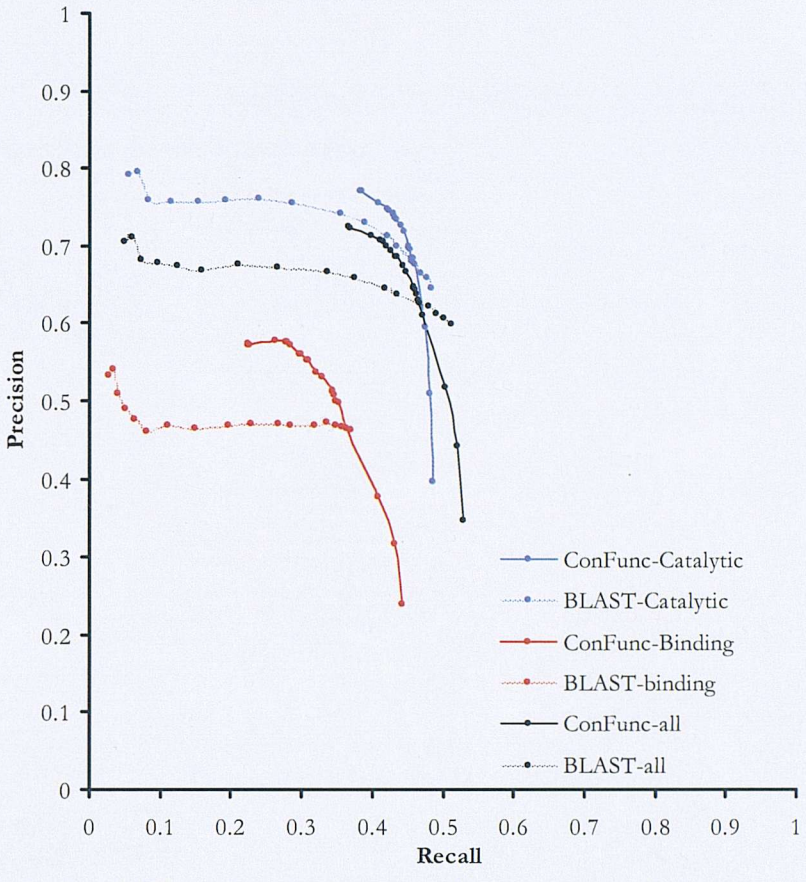


Figure 5.9 Prediction of different functional types. ConFunc performance is shown with solid lines and BLAST with dashed lines. Performance for all functional types is shown in black, catalytic functions in blue and binding functions in red.

are not annotated for a query sequence will result in lower precision. In the opposite case where a query sequence is annotated with a binding function, it is possible that few of the other homologues are also annotated with the binding function. This potentially reduces the chance of it being predicted for the query sequence and therefore results in lower recall.

Multi-domain proteins are a common problem for sequence based function prediction methods (Hegyi and Gerstein, 2001) and could be a potential explanation for the difference in performance between functional types. These problems arise because sequences may only be homologous over part of their lengths and have different domains that are not shared. Many approaches, particularly BLAST based annotation transfer, are

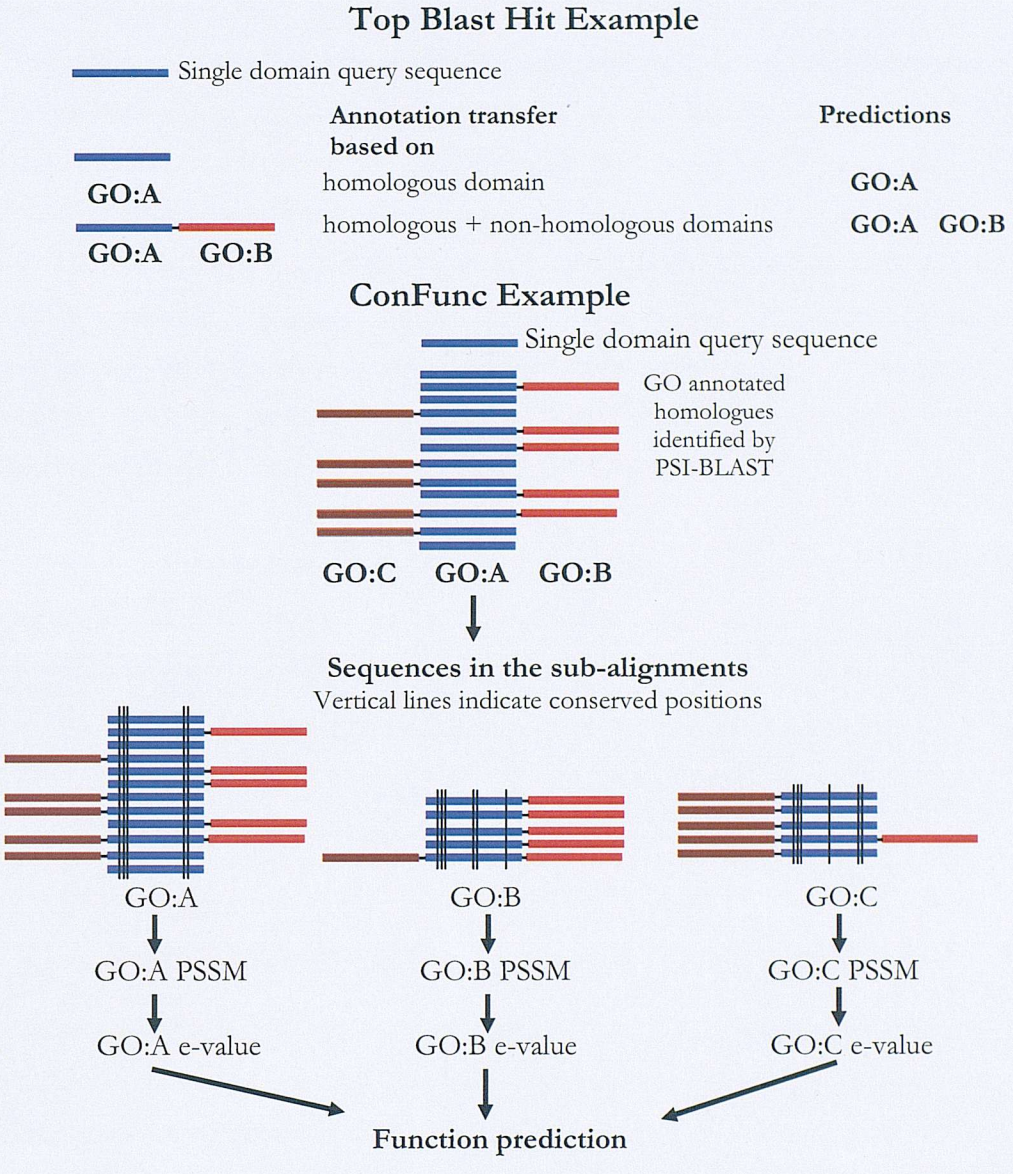


Figure 5.10 The function prediction domain problem. The potential effect of non-homologous domains is shown for both BLAST annotation transfer and ConFunc predictions. Both cases consider function predictions for a single domain query sequence. i) Annotation for two different top BLAST hits is shown, the first a single domain sequence and the second a multi-domain sequence with a domain not present in the query sequence. ii) ConFunc splits the PSI-BLAST homologues into subalignments. The annotations from non-homologous domains are inferred if the e-value based upon the conserved residues from the domain homologous to the query sequence is smaller than the e-value threshold being used.

susceptible to this problem by identifying the homologous regions and then transferring all the functions from one homologue to the other (Figure 5.10). ConFunc is also susceptible to this problem because GO annotations are assigned to sequences rather than domains or regions of a protein sequence. However, unlike BLAST annotation transfer, these unaligned functions are not automatically transferred to the query sequence. Subalignments are generated for the unaligned GO terms and whether they are inferred for the query depends upon the PSSM and subsequent GO term e-value obtained. The PSSM considers only aligned residue positions such that the unaligned functions are inferred on the basis of how well the aligned regions correspond to the query sequence. So for ConFunc to infer functions associated with other domains, requires the domain/function to be present in multiple homologues.

It is not currently practical to assess the extent to which functions associated with unaligned functions are inferred by ConFunc, although a mapping of GO functions to SCOP domains is currently being developed (S N Ali – personal communication). This would enable such an analysis and further, could be used within the ConFunc predictive process to reduce this problem.

5.7. Predictions at Different levels of Gene Ontology

Other studies (Clare, et al., 2006; Pal and Eisenberg, 2005; Pazos and Sternberg, 2004) consider GO based function predictions at different levels. They generally assign levels to GO terms based upon the number of nodes that they are from the root of the graph. Whilst it is possible to do this, there remains the problem that each of the branches within GO is different, with the same distance from the root not necessarily converting to the same level of functional specificity (see section 1.1.3). There is also ambiguity due to the possibility of a term having many parents and may therefore also have more than one potential level. Despite these problems such analyses do assess the performance of methods at different levels of functional specificity as long as it is accepted that each increasing level suggests greater functional specificity but that this increase cannot be quantified.

Figure 5.11 and Figure 5.12 show the performance of ConFunc and BLAST between GO levels one and seven, with the level representing the number of nodes from the molecular function term. Where terms have more than one potential level, the largest level is used. This is the greatest number of nodes it is away from the root term. When assessed using non-electronic annotations (Figure 5.11), a similar trend to overall ConFunc performance (Figure 5.1) is observed at most levels, with ConFunc outperforming BLAST at high levels of precision. At more specific functional levels (levels 6 and 7), ConFunc and BLAST performance is comparable. However, when compared to electronic annotations (Figure 5.12), level 1 and 2 performance is similar to the assessment with non-electronic terms. From level 3 onwards, ConFunc shows improvement over BLAST, with the difference greatest at levels 4 and 5.

When the ConFunc performance at different levels is compared (Figure 5.13) the recall and precision are greatest at level one, suggesting that few completely incorrect predictions are made in different functional categories. Instead, more false predictions are made at specific functional levels. A considerable drop in both recall and precision is observed for all other levels (compared to level 1). There is not a constant pattern of change in the performance observed at each increasing level, although the results at level 7 are most different from these obtained for levels 2-6, with lower levels recall and precision.

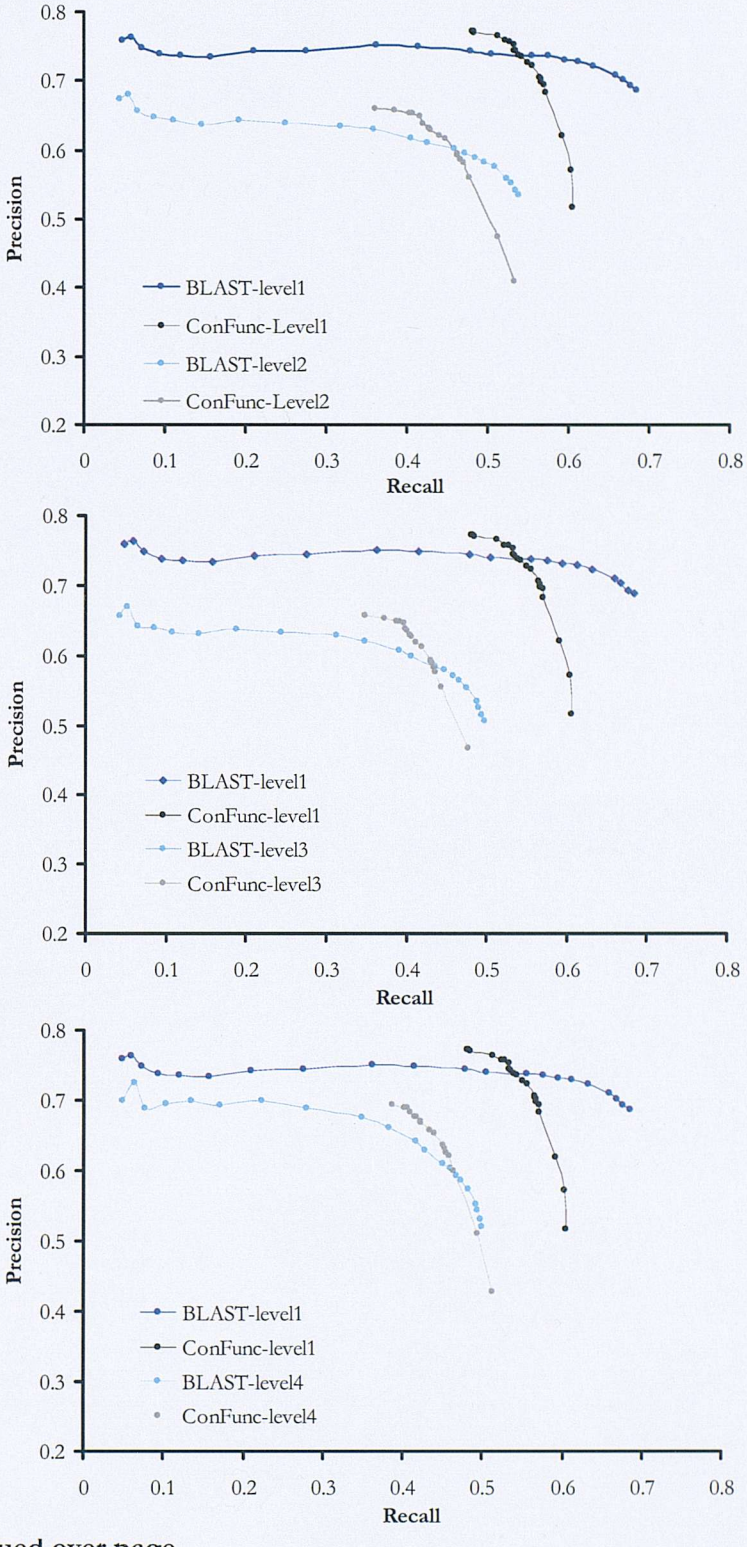


Figure continued over page.

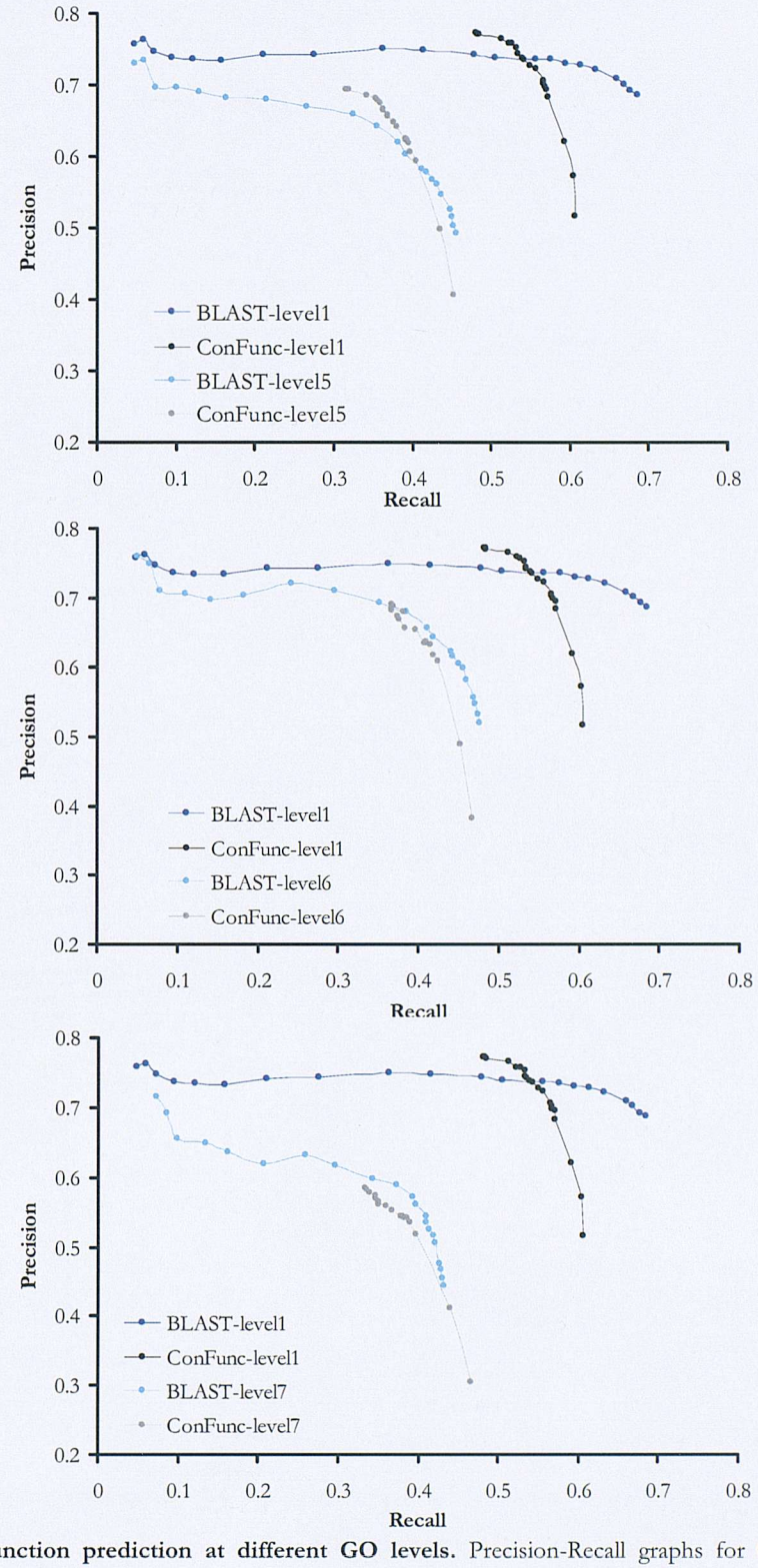
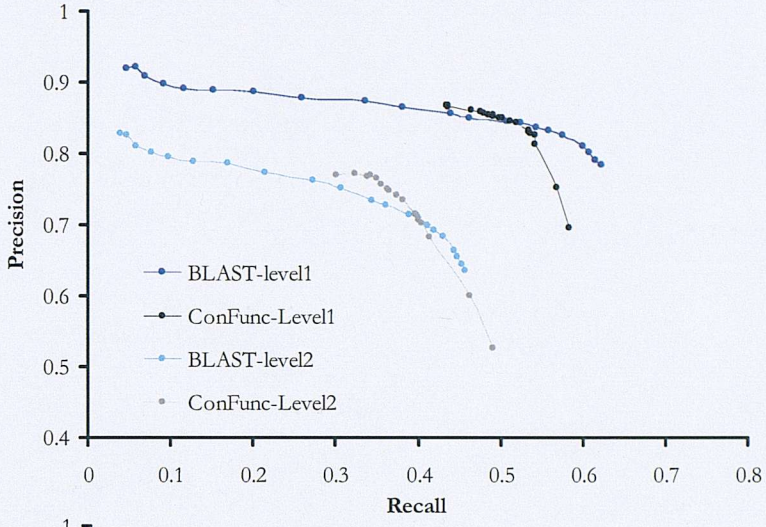
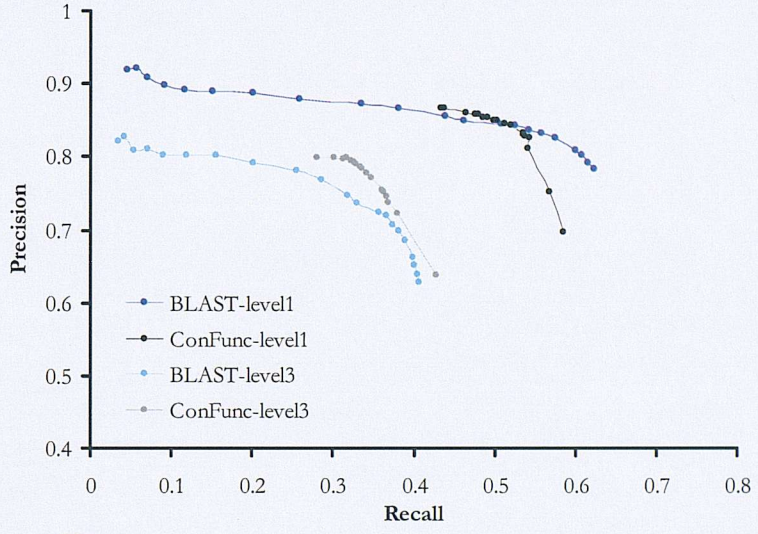


Figure 5.11 Function prediction at different GO levels. Precision-Recall graphs for ConFunc and BLAST performance for GO levels 1 to 7. Level 1 are plotted on all graphs for reference. Predictions were assessed against non-electronic annotations.





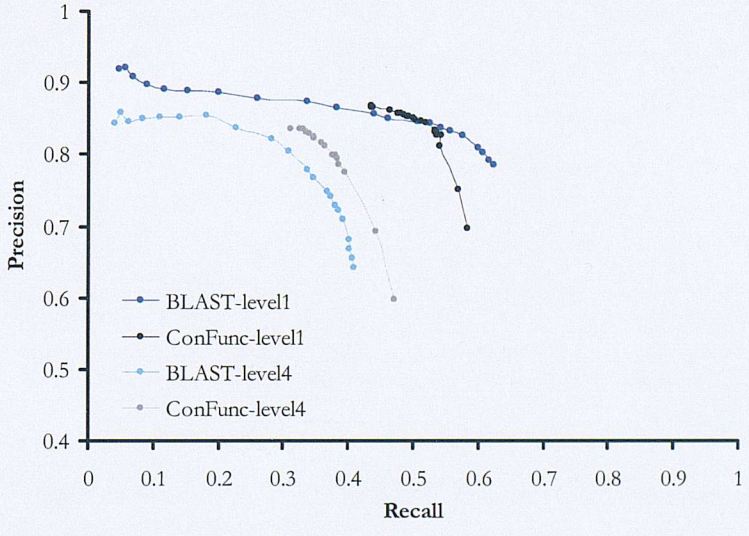


Figure continued over page.

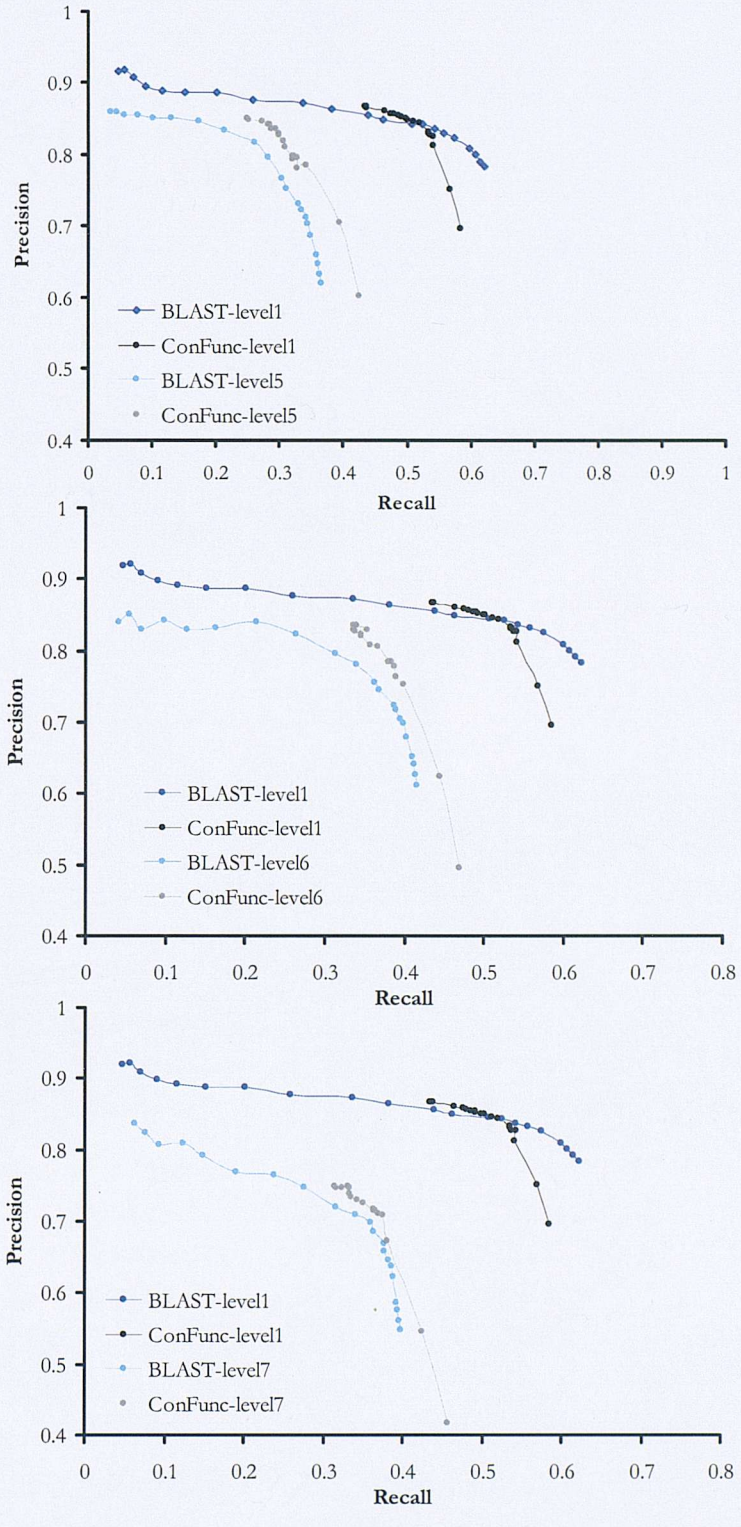


Figure 5.12 Function prediction at different GO levels. Precision-Recall graphs for ConFunc and BLAST performance for GO levels 1 (most general) to 7 (most specific). The level 1 results are plotted on each graph for reference. Predictions were assessed against electronic annotations.

i

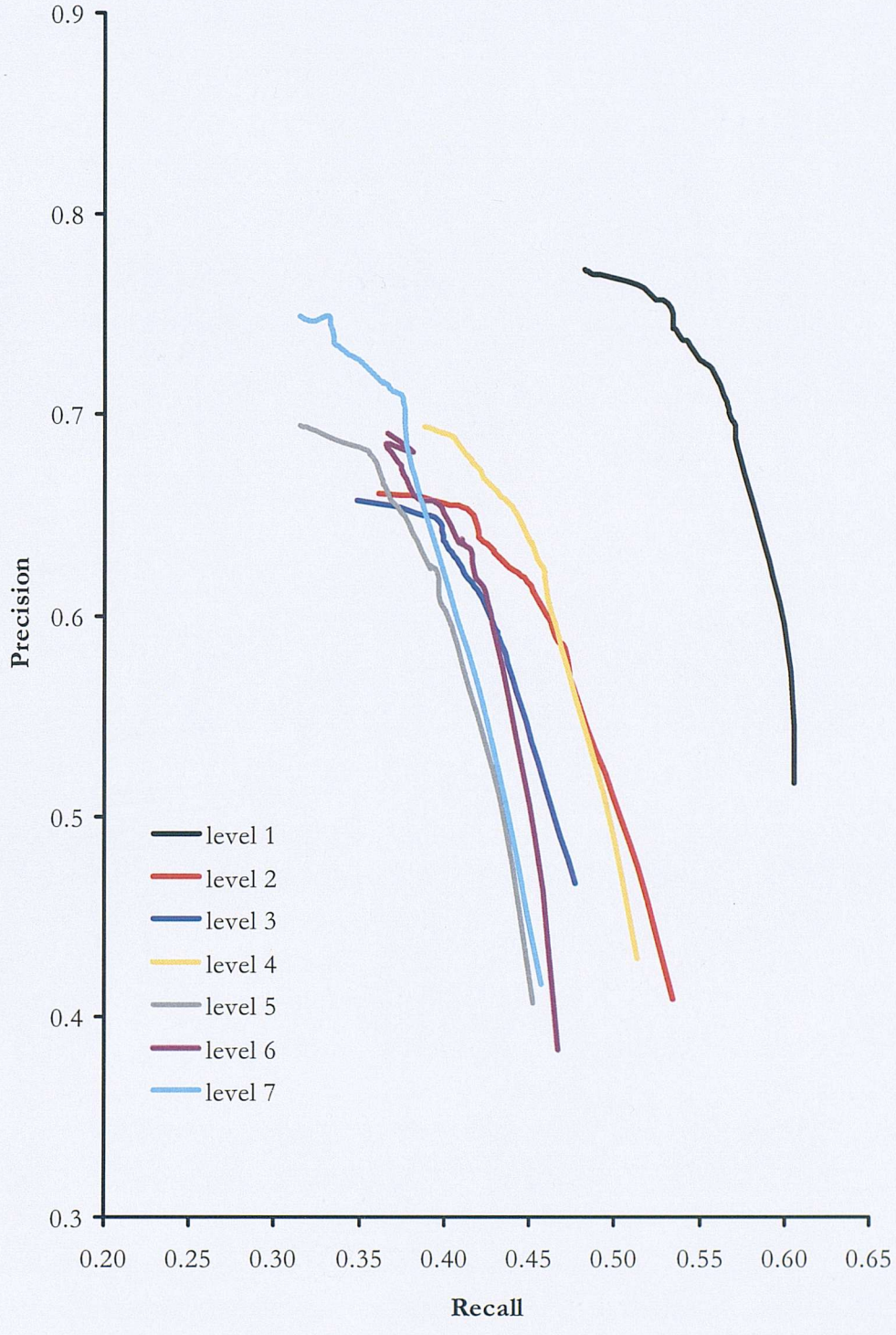


Figure continued over page.

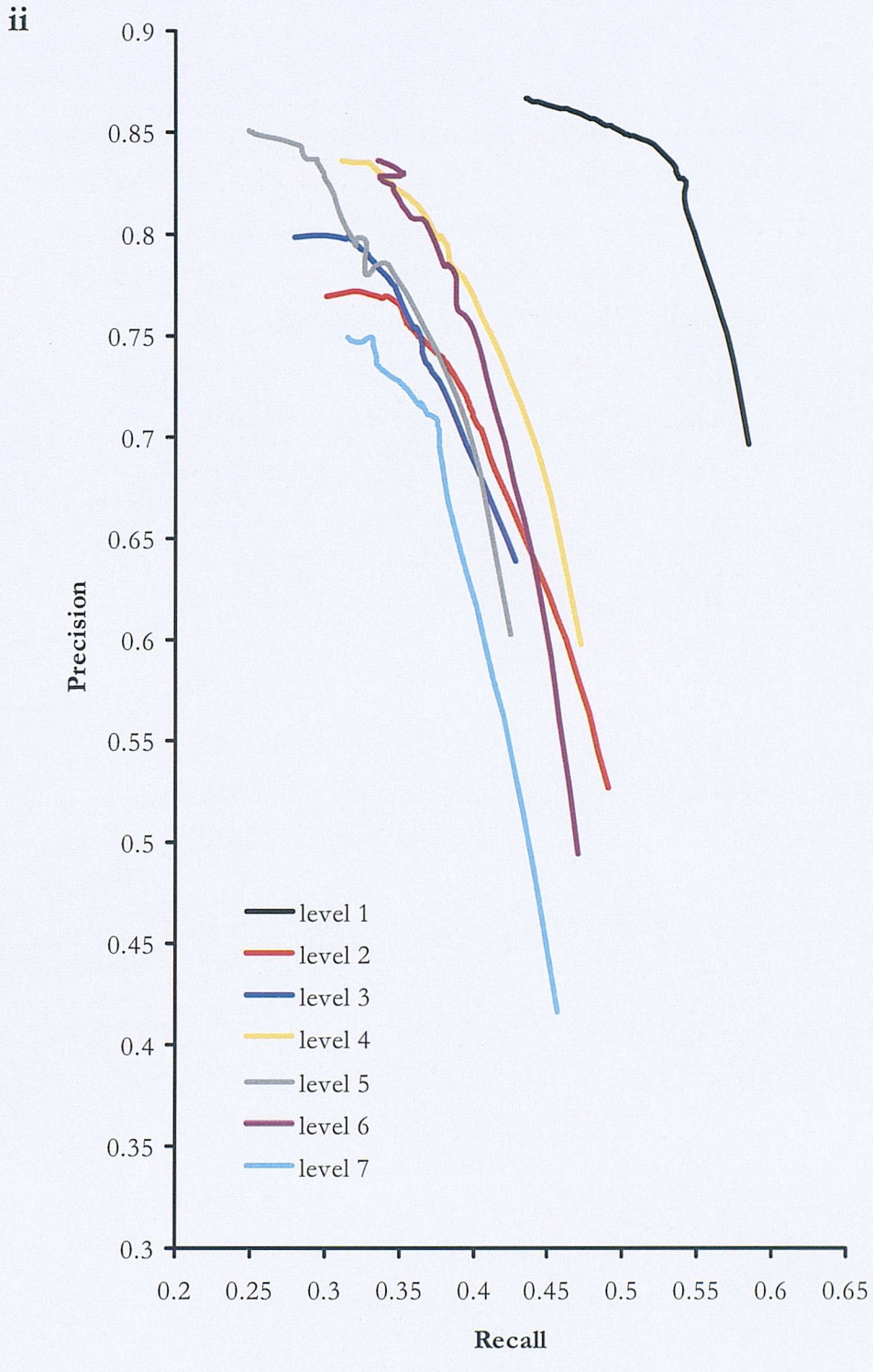


Figure 5.13 ConFunc performance at different GO levels. Precision-Recall graph for ConFunc predictions at individual GO levels 1 to 7. i) Predictions compared to non-electronic annotations. ii) Predictions compared to electronic annotations.

5.8. Summary

This chapter has assessed the performance of ConFunc against function prediction methods using BLAST, PSI-BLAST and Pfam. The following results have been observed:

- For the full test set at 30% sequence identity:
    - ConFunc outperforms BLAST and PSI-BLAST, obtaining greater recall at high levels of precision. BLAST and PSI-BLAST obtain greater coverage of sequences. However if only sequences that all three methods make predictions for are considered, then the performance difference between ConFunc and BLAST/PSI-BLAST is increased.

    - BLAST outperforms PSI-BLAST. This is likely to be due to PSI-BLAST identifying remote homologues with different functions in cases where BLAST does not find any annotated homologues.

    - ConFunc also outperforms Pfam despite the potential use of high sequence identity homologues by Pfam. In many cases Pfam is only able to infer general protein functions.

    - The comparison of ConFunc and BLAST with electronic GO annotations shows that the hand-curated non-electronic annotations are incomplete and may therefore underestimate the precision of the methods considered. The incompleteness of the non-electronic annotations is exemplified by GTPase proteins.

    - Predictive performance of ConFunc and BLAST varies between the different functional types in the molecular function GO category. The best performance is obtained for catalytic functions, while the prediction of binding functions is poor.

    - Predictive performance of both ConFunc and BLAST varies at different GO levels. Very few completely incorrect functions (i.e. incorrect GO

molecular function category) are predicted at low levels, while greater errors occur at more functionally specific levels of GO.

- The analysis of the methods in the twilight zone of sequence similarity highlights the area in which the performance differences between BLAST and ConFunc are maximised. ConFunc is able to combine the information from groups of remote homologues to correctly infer function, whereas the annotation transfer of the 'top' remote homologue by BLAST is often incorrect.

Chapter 6

The ConFunc Server

The ConFunc server (http://www.sbg.bio.ic.ac.uk/confunc) enables the scientific community to use the method for function prediction of proteins of interest. This chapter describes the features that are currently available on the server and those that are planned for future versions.

6.1. ConFunc Submission

Users can either paste a sequence into the server to infer function or enter a Swiss-Prot accession code (Figure 6.1). Three options are provided enabling the user to tailor their jobs. The PSI-BLAST e-value threshold and number of homologues options control the PSI-BLAST settings for identifying homologues. These user defined values effect ConFunc predictions especially in cases where there are only a few annotated homologues. Finally the user can select a confidence level of between 1 and 15. Each of these levels selects a frequency threshold setting to discriminate between those functions that are inferred for the query and those that are rejected. A confidence setting of 1 uses the frequency threshold that achieved greatest precision in benchmarking (section 4.2.2). Higher confidence levels select frequency threshold settings that obtained lower precision but greater recall in benchmarking. Users are required to enter their email address so that confirmation of their submission and results can be sent.

6.2. ConFunc Results

The server allows users to view the data files that have been generated during the ConFunc run. These include the PSI-BLAST output and MUSCLE sequence alignment (Figure 6.2). The GO molecular function terms present in the homologues identified by PSI-BLAST are presented in two lists. The GO terms that have been inferred as functions of the query are displayed first. These are followed by a list of the rejected terms. For each GO term its functional description, level in the ontology, GO e-value, count and frequency (percentage of homologues that term is present in) are displayed.

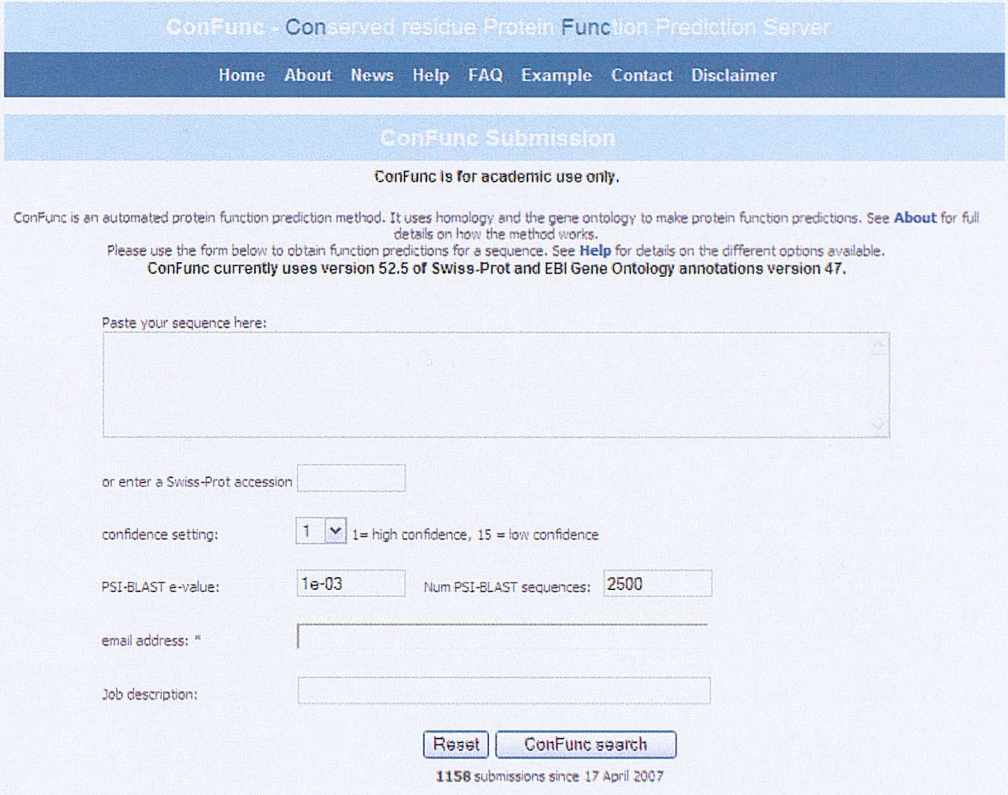


Figure 6.1 The ConFunc server submission page.

The user can also view each of the subalignments generated by ConFunc, by clicking on each individual GO term. These subalignments highlight the conserved residues used by ConFunc in the predictive process. They are colour coded to indicate different levels of conservation (Figure 6.3).

6.3. Future Developments

The ConFunc server currently provides users with limited options to control the parameters used by ConFunc and to explore their results. Updates to both are planned and these will provide users with a more powerful resource.

ConFunc Results
View intermediate data files

View PSI-BLAST results:	PSI-BLAST output
View Muslce alignment:	muscle alignment
View Muslce alignment (gaps removed):	muscle alignment no gaps
Number of annotated sequences:	154

ConFunc predictions
Click on each GO term to view its subalignment

Predicted terms

GO Term	Function	Level	GO c-value	Count	Freq
GO:0003924	GTPase activity	7	1.6592e-08	105	0.681818
GO:0017111	nucleoside-triphosphatase activity	6	1.6592e-08	105	0.681818
GO:0016462	pyrophosphatase activity	5	1.6592e-08	105	0.681818
GO:0016818	hydrolase activity, acting on acid anhydrides, in phosphorus-containing anhydrides	4	1.6592e-08	105	0.681818
GO:0016817	hydrolase activity, acting on acid anhydrides	3	1.6592e-08	105	0.681818
GO:0005515	protein binding	2	3.7978e-09	42	0.272727
GO:0016787	hydrolase activity	2	1.6592e-08	105	0.681818
GO:0005488	binding	1	3.7254e-09	74	0.480519
GO:0003824	catalytic activity	1	1.6592e-08	105	0.681818
GO:0003674	molecular_function	0	1.4684e-08	154	1
GO:0003673	Gene Ontology	0	1.4684e-08	154	1

Rejected terms

GO Term	Function	Level	GO c-value	Count	Freq
GO:0005525	GTP binding	5	1.0887e-08	38	0.246753
GO:0019001	guanyl nucleotide binding	4	9.5529e-09	41	0.266234
GO:0030742	GTP-dependent protein binding	3	4.2216e-09	2	0.012987
GO:0017076	purine nucleotide binding	3	9.5529e-09	41	0.266234
GO:0046872	metal ion binding	3	9.7828e-07	2	0.012987
GO:0005095	GTPase inhibitor activity	3	6.6945e-06	2	0.012987

Figure 6.2 ConFunc server results. The user can download the PSI-BLAST output and MUSCLE sequence alignment that are used by ConFunc (shown at the top). Results are expressed as a table of GO terms, which are displayed with their description and other properties including their level and GO term e-value. The list is split so that terms that ConFunc predicts to be functions of the query are shown first. These are followed by the terms that ConFunc has rejected as potential functions of the query. For space reasons the full list of rejected terms has not been shown.

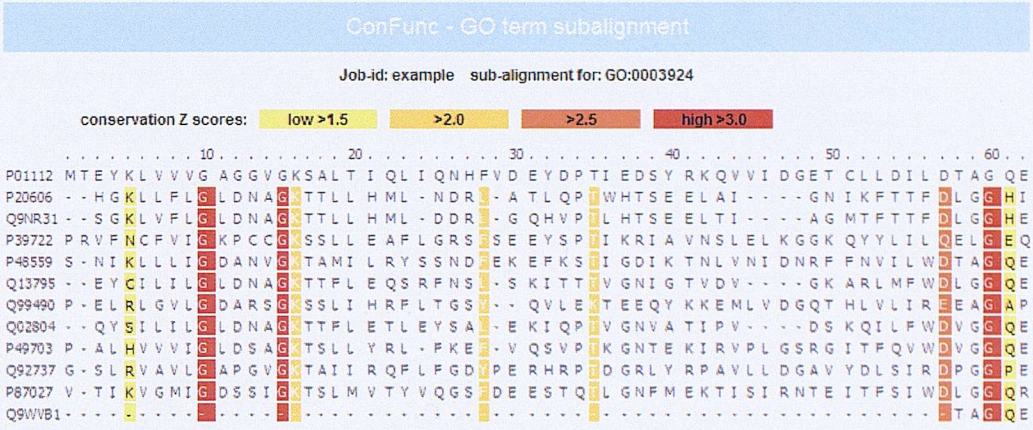


Figure 6.3 Subalignments on the ConFunc server. A section of a GO term subalignment is shown. Conserved residues are coloured as described by the key at the top.

6.3.1. Submission Options

The research has demonstrated the importance of sequence alignment quality to ConFunc predictions (Figure 4.6). The ConFunc server currently uses MUSCLE to align sequences. It is planned to add MAFFT and PSI-BLAST alignments as user defined options. The MAFFT option may result in better function predictions (see chapter 3), while the PSI-BLAST alignment would enable results to be obtained more rapidly as a progressive sequence alignment is not calculated. A further option would allow the user to upload an alignment. However this would also require the server to have access to GO annotations for the aligned proteins (it currently only considers sequences in Swiss-Prot). Other features to be added include batch submission and allowing the user to specify the conservation z-score threshold, which is currently set at a default of 1.5.

6.3.2. Updating the Display of Results

The ConFunc server output is currently predominantly text based. Future improvements to the server will include a graphical display that will allow the user to view the ConFunc results on a subset of the GO graph, indicating terms that have been predicted and those that have been rejected. Where a suitable protein structure is available, users will also be able to map the ConFunc conserved residues onto the structure, enabling the areas of conservation to be visualised (as shown later in section 7.3). The results of annotation

transfer will also be included on the results page, giving the user an indication of the function of the closest homologue.

Chapter 7

Discussion

This thesis has described the development and benchmarking of ConFunc, a new sequence based protein function prediction method. It has also emphasised some of the problems related to protein function prediction, particularly those relevant to ConFunc. The concept of ConFunc is simple, with homologous sequences grouped according to their functional annotations (from Gene Ontology). Sequence alignments of these groups are used to identify conserved residues associated with each function. These are then used to infer function. This chapter will discuss the results obtained during the benchmarking of ConFunc described in chapter 4. It will also report on the assessment of the conserved residues identified by ConFunc. Finally, the practical application of ConFunc is discussed.

7.1. Is ConFunc a success?

Section 3.2 set out the aims and objectives of this research. In brief they were to develop a sequence based function prediction method that:

- Uses Gene Ontology to enable the prediction of all types of molecular function
- Uses biological features of protein sequence and not just similarity scores.
- Performs well at low levels of sequence identity
- Is automated, enabling use in an genomics functional annotation pipeline

Does ConFunc fulfil these objectives? The first goal has clearly been achieved as the Gene Ontology is central to ConFunc, with GO being used to direct the predictive process and to express the predicted functions. The development of ConFunc also meets the second objective of using the biological features of homologous sequences, rather than just their overall similarity with the query. ConFunc does this by taking the approach of splitting homologues into subalignments according to their GO annotations and then using the conserved residues present in each subalignment for comparison with the query sequence, in order to determine which functions are inferred. This contrasts with methods such as GOtcha (Martin, et al., 2004) and PFP (Hawkins, et al., 2006), which consider the

annotations of a set of homologous sequences but primarily use their BLAST e-values or bit scores to infer function.

The analyses presented in chapter 4 assess the ability of ConFunc to infer function at low levels of sequence identity. The initial analysis using the full test set, demonstrates that ConFunc can outperform BLAST and PSI-BLAST at high levels of precision, obtaining up to six times greater recall than BLAST for a large set of proteins (Figure 5.1). This also demonstrates large scale application of ConFunc. More importantly, the analysis using the set of sequences in the twilight zone (section 5.4) highlights that ConFunc obtains the most significant improvement over BLAST annotation transfer at low levels of sequence similarity. This is of interest because many methods struggle to infer function at such levels of sequence similarity and therefore the improvement offered by ConFunc could lead to more accurate annotations for such sequences.

The comparison with Pfam function prediction is also important because Pfam is a widely used tool for genome annotation (International Chicken Genome Sequencing Consortium 2004; Waterston, et al., 2002). The ability of ConFunc to outperform functional annotation using Pfam occurs because Pfam can only associate general functional terms with many of its families. This is emphasised by the analysis of Pfam and ConFunc performance at different GO levels (Figure 5.8 and Figure 5.11). At maximum levels of precision, ConFunc obtains nearly four times greater recall than Pfam (Figure 5.7). This difference in performance is especially relevant given Pfam has the advantage of using high sequence identity homologues.

Some of the analyses have also highlighted the limitations of ConFunc, many of which are present throughout the function prediction field. The analysis at individual GO levels suggests that ConFunc is more likely to make erroneous predictions when inferring highly specific functions. This is seen in Figure 5.13 with the worst performance at level 7. A potential explanation for this observation is that at such levels, the differences in function between sibling terms are small. For example in the case of catalytic functions, this may relate to different substrate specificity. Such predictions may suggest a certain function, but with uncertain substrate specificity. This is therefore an obvious area for improvement. However, the non-standard levels that have been used within GO for this

analysis may be a cause of this observation. This is supported by the lack of a clear relationship between predictive performance and GO level, for both ConFunc and BLAST predictions (Figure 5.11).

A number of analyses suggest that current protein annotations may be incomplete. This is first seen in the performance differences when assessing predictions using different sets of annotations (electronic and non-electronic, see Figure 5.1 and Figure 5.4). The increase in precision when using electronic annotations shows agreement between ConFunc and BLAST and the automated methods used to generate the electronically inferred annotations for functions that are not present in the hand-curated annotations. The detailed analysis of GTPases further demonstrates a source of the incompleteness of the non-electronic annotations. The considerable differences between the predictions for different functional types (Figure 5.9) may also suggest incompleteness in the current annotations. The absence of many binding functions that should be present in the annotations could account for the poor prediction of these functions by both BLAST and ConFunc. These potential problems need to be addressed when developing and benchmarking function prediction algorithms.

7.2. ConFunc and Phunctioner

ConFunc and Phunctioner (Pazos and Sternberg, 2004) share the similar approaches of using GO annotations to group homologues and identifying conserved residues associated with individual functions. There are however, considerable differences between the two approaches. A major difference is the source of data that they use. ConFunc is completely sequence based, enabling the use of all GO annotated sequences present in Swiss-Prot. The data sources used by Phunctioner are more complex, with structural alignments used to obtain high quality sequence alignments. This is done by using structural families from the FSSP database (Holm, et al., 1992) to create the GO term suballgnments. This approach may have the advantage of potentially obtaining better alignments due to the inclusion of structural data. However, the FSSP database is not regularly updated (the FSSP files on EBI FTP site date to 2001, correct as of January 2008) and so the reliance on this outdated data, limits its ability to predict function for the

rapidly expanding protein sequence databases. It is conceivable that Phunctioner could replace the structural data obtained from FSSP by manually aligning structures. However, at the current time, with the impact of metagenomics projects, restricting the method to using proteins of both known structure and function would continue to limit the coverage of the approach. In contrast ConFunc is able to harness a much greater volume of information present in the vast number of protein sequences in databases, resulting in greater coverage than Phunctioner.

Another difference between ConFunc and Phunctioner is the values they use to infer function. Phunctioner calculates Z scores for each GO term and uses them to determine the inferred functions. ConFunc applies statistical approaches used for assessing sequence similarity and calculates an e-value between the query sequence and each GO term. Further, Phunctioner uses a simple Z score threshold to accept/reject functional terms, in the same way that the simple e-value threshold assessed here did (see section 4.2.1). So Phunctioner obtains good performance without the use of the ratio or frequency thresholds that were implemented in this work. This occurs because of the distributions of the GO terms considered between Phunctioner and ConFunc. In ConFunc benchmarking, there are many more false than true terms present in the homologues identified (Figure 4.2). The reverse is true for Phunctioner (see Figure 2D, Pazos and Sternberg, 2004), with many more true than false terms. Therefore it was possible to use a simple Z score threshold for Phunctioner as false terms did not overwhelm the true terms. There are two possible explanations for this observation. Firstly, as Phunctioner is a structural approach it is not subject to the domain problem associated with sequence based methods (see section 0), which is a source of false terms for ConFunc. Secondly, the GO annotations used for Phunctioner benchmarking are more than two years older than those used for ConFunc, so the difference observed could also be a consequence of the increasing number of annotated proteins.

The general approach taken by ConFunc and Phunctioner has similarities with methods such as Evolutionary Trace (ET; Lichtarge, et al., 1996) that use phylogenetics to infer functional residues (see section 1.3.2). ET uses a phylogenetic tree to split a protein family into subfamilies, which are then used to identify functional residues. This contrasts to ConFuncs use of GO to direct this splitting process. This enables ConFunc to infer

function, rather than just identify functional residues. In addition the conserved residues present in ConFunc subalignments appear to identify functional residues (see section 7.3).

It has also been shown that some functional characteristics may not be represented well by phylogenetic trees (Pazos, et al., 2006) and that splitting homologues using different features can prove more effective. For example Pazos, et al. (2006) show that grouping sequences according to the ligands that they bind, can be used to identify functionally important residues (Xdet method described in section 1.3.2). This approach of grouping sequences by functional features is similar to the approach of grouping sequences by GO annotations as used by ConFunc.

7.3. Assessment of ConFunc Conserved Residues

ConFunc identifies conserved residues associated with each individual GO term and uses them to determine the functions inferred for each query sequence. The prediction of functional residues is an important biological task and therefore ConFunc conserved residues have been compared to known functional residues.

Assessing if predicted functional residues (in this case conserved residues) have a functional role is not easily done. There are limited resources that catalogue functional residues and in general they are based upon protein structures rather than sequence, which complicates the comparison performed here. There are however two recent databases of functional residues, The Catalytic Site Atlas (CSA; Porter, et al., 2004) and FireDB (Lopez, et al., 2007b). The CSA is a curated set of catalytic residues for enzymes of known structure. The inclusion of a residue in the CSA strictly requires evidence of an active role in the enzyme mechanism. Residues that lack evidence of their role and those that have functional roles but are not directly involved in the enzyme mechanism, such as determining substrate specificity, are not included. As the CSA is hand-curated there is confidence in the residues present. However, its restrictive requirements for inclusion may mean that important functional residues are missed. FireDB is a second source of functional residue data. It is an automated approach that uses close atomic contacts between protein residues and ligands in solved protein structures to identify functionally important residues, based on the assumption that these residues bind the ligand and are

therefore functional. Unlike the CSA, FireDB is not restricted to enzymes and contains functional residues for a full range of different protein functions. To provide extra information about catalytic residues, FireDB also incorporates data from the CSA. Both the CSA and FireDB have been used for comparison with ConFunc conserved residues to exploit both the confidence of data in the CSA and the coverage of FireDB. It is important to note that these datasets are not perfect, they may include residues that do not have a significant functional role and they may also exclude residues that have an important functional role because the sources they use do not provide evidence of a functional role.

Both FireDB and the CSA contain functional residues present in proteins with solved structure, so only information for a small subset of the original sequence test set can be assessed. A number of steps were required to identify the sequences that could be used:

1) The Swiss-Prot to PDB mapping (Martin, 2005) was used to identify sequences in the test set with a structure in the PDB. This mapping also maps individual residues between the Swiss-Prot sequence and the PDB structure, which is important for comparing the residues.
2) For those sequences with mapping data, the relevant structures were searched for in FireDB and those with functional residue data retained. In cases where a sequence mapped to multiple FireDB entries, the one with the greatest coverage of the Swiss-Prot sequence was used.
3) The coverage between the Swiss-Prot sequence and PDB structure was considered (i.e. the percentage of the Swiss-Prot sequence residues that are present in the structure). Comparison between the conserved ConFunc residues is only possible where the equivalent residues are present in the structure. Sequences covered by less than 80% were filtered out.

Only nine sequences from the original test set met all these criteria. More sequences (74) had mapping and FireDB entries but their coverage was below 80% and in some cases very low, making them unsuitable for analysis. To increase the number of sequences that

Swiss-Prot Accession	*Z > 1.5*	*FireDB residues*	*Z > 1.5 & FireDB*	*%FireDB identified*	*% Z > 1.5 & FireDB*	*CSA res*	*Z > 1.5& CSA*	*%CSA identified*
P00390	41	44	9	0.20	0.22	7	0	0.00
P00491	30	16	5	0.31	0.17	4	2	0.50
P00746	24	14	3	0.21	0.13	6	2	0.33
P00797	35	38	9	0.24	0.26	7	4	0.57
P00915	22	20	2	0.10	0.09	3	1	0.33
P00950	16	14	2	0.14	0.13	4	1	0.25
P01009	38	5	0	0.00	0.00	0	-	N/A
P01112	19	25	9	0.36	0.47	1	1	1.00
P02692	11	42	5	0.12	0.45	0	-	N/A
P02693	10	17	3	0.18	0.30	0	-	N/A
P02768	32	85	0	0.00	0.00	0	-	N/A
P03958	27	20	4	0.20	0.15	4	2	0.50
P04746	42	18	0	0.00	0.00	6	2	0.33
P05132	34	31	12	0.39	0.35	5	2	0.40
P06766	32	6	1	0.17	0.03	1	1	1.00
P07170	22	47	12	0.26	0.55	6	2	0.33
P10114	14	25	8	0.32	0.57	1	1	1.00
P12931	44	30	14	0.47	0.32	4	3	0.75
P15121	34	32	10	0.31	0.29	4	3	0.75
P15289	52	19	7	0.37	0.13	8	4	0.50
P15454	20	24	7	0.29	0.35	0	-	N/A
P15700	22	34	10	0.29	0.45	6	2	0.33
P15848	50	24	5	0.21	0.10	7	3	0.43
P20160	25	8	0	0.00	0.00	3	2	0.67
P23946	26	14	4	0.29	0.15	6	4	0.67
P24280	24	21	3	0.14	0.13	0	-	N/A
P24941	27	23	9	0.39	0.33	6	4	0.67
P25114	55	40	8	0.20	0.15	6	2	0.33
P28161	17	12	3	0.25	0.18	1	1	1.00
P28482	33	28	8	0.29	0.24	5	2	0.40
P31151	10	9	3	0.33	0.30	0	-	N/A
P35813	29	8	5	0.63	0.17	0	-	N/A
P48052	48	5	0	0.00	0.00	3	0	0.00
P49773	12	18	5	0.28	0.42	3	2	0.67
P49789	16	15	5	0.33	0.31	3	3	1.00
P61586	16	24	10	0.42	0.63	2	1	0.50
P83686	21	24	6	0.25	0.29	1	1	1.00
Q01469	19	17	4	0.24	0.21	0	-	N/A
Q16539	38	18	7	0.39	0.18	5	4	0.80
Total	**1087**	**914**	**217**	**0.24**	**0.20**	**128**	**62**	**0.48**

Table 7.1. Comparison of conserved residues with functional residues. The individual data for the sequences mapped onto a structure are shown. The columns contain: *Z >1.5* - number of residues above the 1.5 Z score threshold, *FireDB residues* - number of FireDB residues, *Z > 1.5 & FireDB* - number of residues above the Z score threshold and in FireDB, *%FireDB identified* - percentage of FireDB residues that ConFunc identifies. %Z > 1.5 & FireDB - percentage of residues above the threshold and also in FireDB. Equivalent data is shown, in the remaining columns, for the CSA residues.

could be analysed, all sequences with non-electronic GO annotations were considered. 174 sequences had mapping and FireDB data, of which 70 were covered over more than 80% of their sequence. From these, only 39 were suitable for the comparison, the others had too few annotated homologues to run ConFunc. Thirty of the thirty nine structures were found to have CSA entries.

To assess if the conserved residues ConFunc identifies are functional residues of the query sequence, the conserved residues for the GO annotated term with the lowest e-value were compared to the CSA and FireDB entries for each of the 39 sequences. The individual results for each sequence using a residue Z score threshold of 1.5 are shown in Table 7.1. At this threshold, over the 39 proteins, ConFunc identifies 24% of the functional residues in FireDB and for one protein identifies 63% of them. There are also four sequences (P01009, P02768, P20160, and P48052) where none of the conserved residues match the FireDB entry. For those sequences that also have CSA entries 48% of catalytic residues are conserved and there are a few proteins (with more than 1 catalytic residue), for which all or nearly all of the catalytic residues are identified as conserved in the GO term subalignment (P00797, P15121, P15289, P23946, P24941, P49773, P49789 and Q16539). While identifying a good proportion of functional residues present in FireDB and the CSA, there are, in many cases large numbers of residues that are conserved but are not classified as functional by either database. In individual cases the percentage of conserved residues that are also in FireDB ranges from 0% to 63%. The results show that while there is some agreement between the ConFunc conserved residues there are many cases where ConFunc conserved residues are not classed as functional by either the CSA or FireDB. The mapping of conserved/functional residues onto the protein surface is reported below, which shows that the conserved residues are often close to the functional residues defined in the CSA and FireDB. Further the conserved residues identified by ConFunc may be functional or structural sites that have not previously been recognised. The high levels of conservation of such residues suggests that this is probable.

Mapping the conserved ConFunc residues and FireDB/CSA functional residues onto the protein structure can give further insight into the similarities between them. For some structures this shows that a fair proportion of the functional sites are identified by

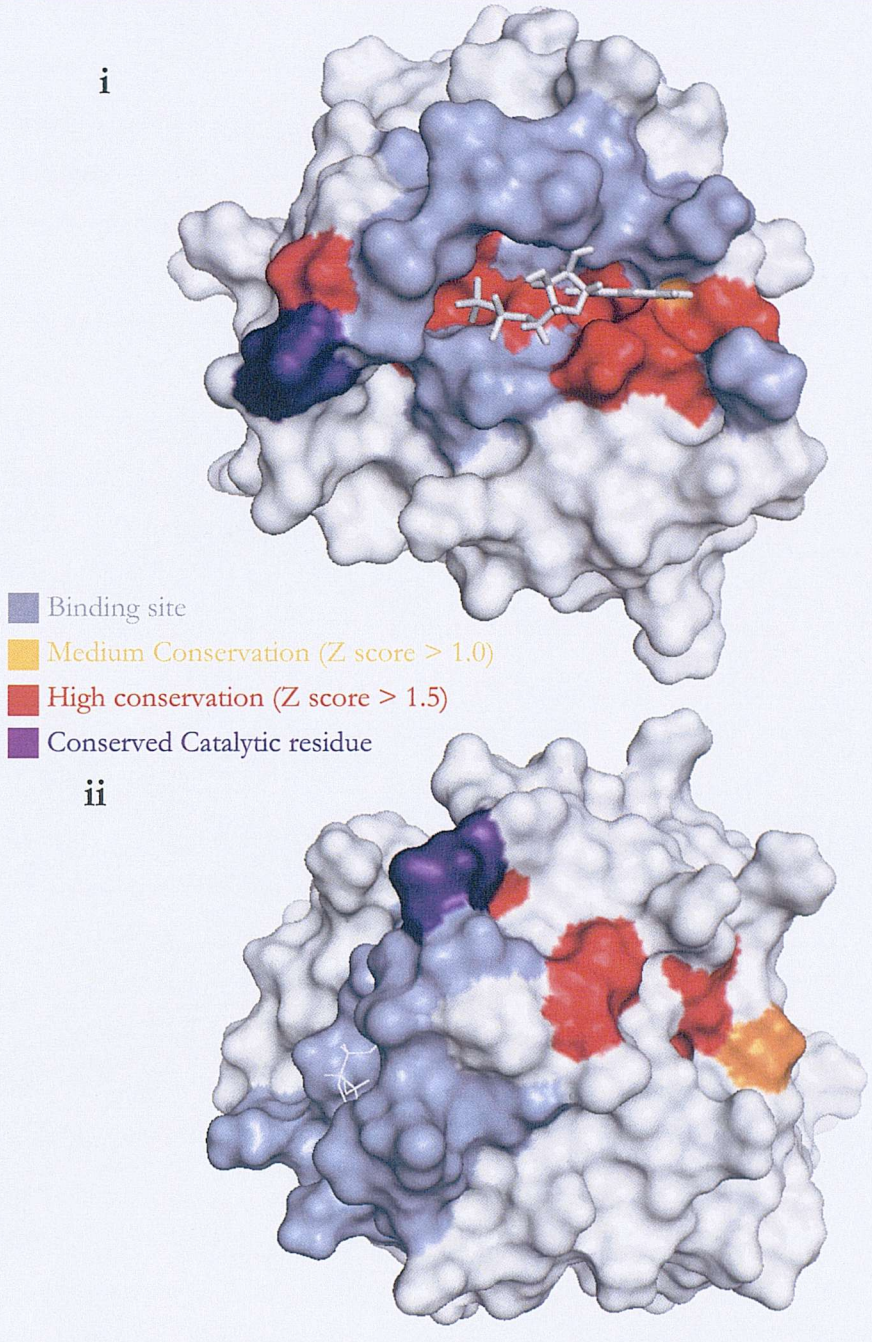
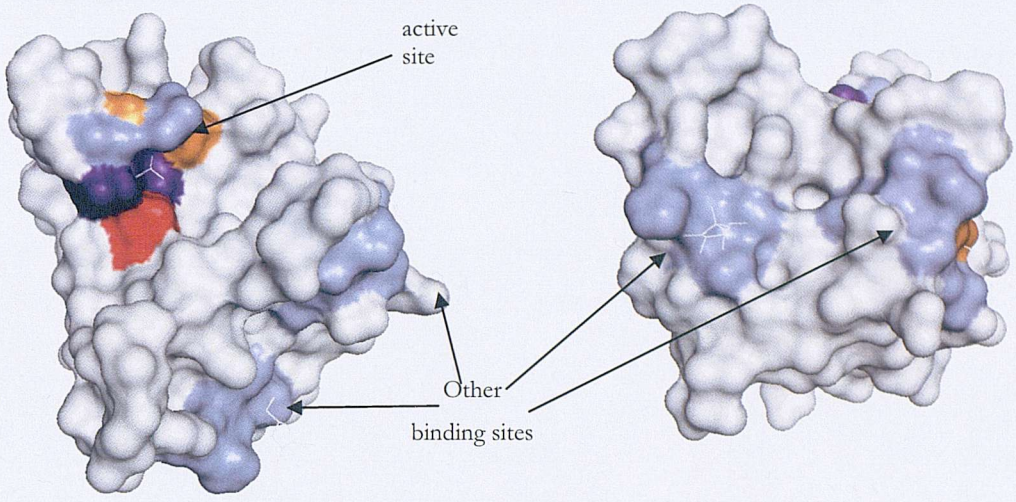


Figure 7.1 Conserved residues for the RAS protein sequence P01112 mapped onto structure 1aa9. Conserved residues with Z scores greater than 1.0 for the top annotated term for sequence P01112 are mapped onto protein structure 1aa9. Residues with a Z score greater than 1.0 are coloured orange and those greater than 1.5 are coloured in red. i) The overlap of conserved and functional residues is shown. The FireDB functional site is coloured teal and residues that are both conserved and functional are coloured. The single CSA residue is coloured purple. ii) Non overlapping residues. The FireDB binding site is shown in teal and only ConFunc conserved residues that do not overlap with the binding site are coloured.

ConFunc (e.g. structure 1aa9 for sequence P01112). In the case of P01112, ConFunc identified nine of the twenty-five FireDB residues and among these, its single CSA residue (Figure 7.1). Where the ConFunc and known functional residues do not overlap, they are often adjacent to one another as seen in Figure 7.1. In this case there are very few residues conserved away from the known functional site. This example also highlights another problem associated with using hetatom records from the PDB as indications of functional residues. Protein structures are often solved with different ligands, which in some cases may not be the ligand of biological interest and therefore the location that a ligand contacts the protein may not be the same as the correct ligand. 1aa9 is the structure of a GTPase crystallised with GDP, its product rather than its substrate. GTP is larger than GDP, as it has an extra phosphate group, and may therefore make more contact with the protein. It seems likely that this third phosphate group would be closer to the CSA catalytic residues and have greater interaction with the protein in this area. Other structures that map to sequence P01112 but with lower coverage of the sequence are available with substrate analogues, such as GNP bound (e.g. structure 221p). For the 221p structure, FireDB also classes two further residues near the CSA residue as functional with ConFunc identifying them as conserved.

The conservation of residues close or adjacent to defined functional residues is also seen for the mapping of sequence P49789 onto structure 2fit (Figure 7.2). In this case, FireDB identifies three different binding sites, one of which is the enzyme active site, which also has three CSA residues. The three CSA residues are identified as conserved by ConFunc and so is part of the enzyme active site, with residues close to this FireDB site also conserved (Figure 7.2). Most importantly this example highlights a problem associated with comparing predicted functional residues with FireDB. FireDB predominantly contains data relating to ligand binding sites. While these are often important to function, it is possible that they are not central to the function and may not be conserved within the protein family. For structure 2fit, the active site is clearly central to its function and it is likely that the other two binding sites, have less relevance to its function. When counting the number of conserved and functional residues that overlap, as has been done for Table 7.1, this can be misleading. It suggests that for P49789, only 31% of ConFunc conserved residues are also functional, despite the active site being correctly identified.

i) Overlap between ConFunc conserved residues with functional residues from CSA and FireDB



ii) ConFunc conserved residues that do not overlap with know functional residues.

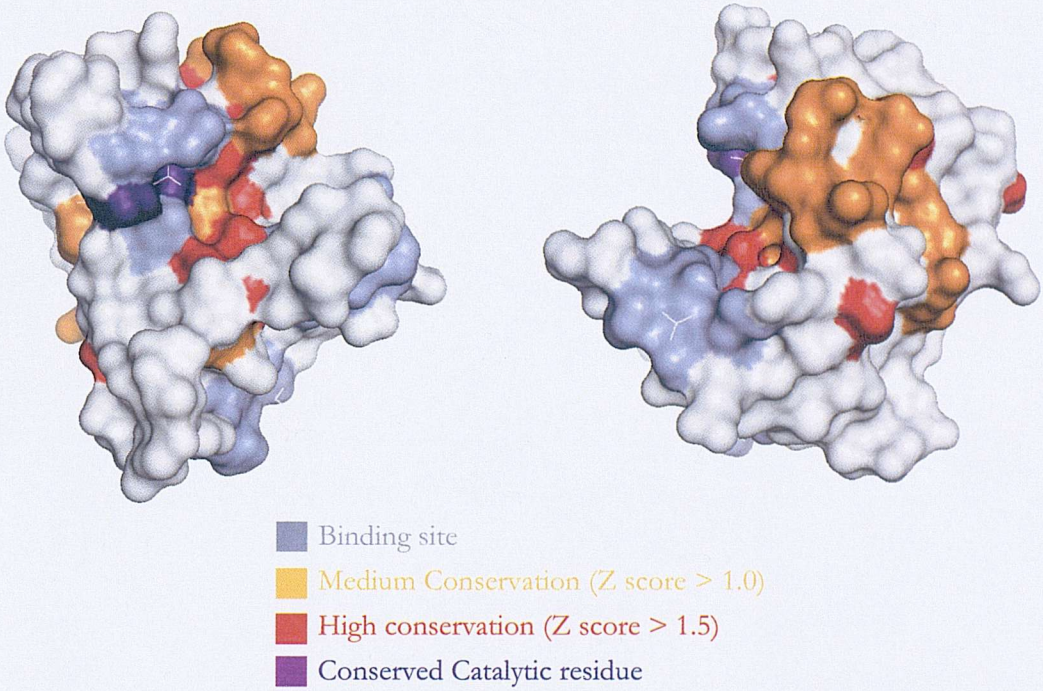


Figure 7.2 Conserved residues from P49789 mapped onto structure 2fit. The Three CSA residues are coloured purple, ConFunc conserved residues are orange (1.0 < Z score < 1.5) and red (Z score > 1.5). The FireDB binding sites are shown in teal. i) only ConFunc conserved residues that overlap with the binding site are coloured. ii) Only ConFunc conserved residues that do not overlap with the binding site are coloured. The binding site is coloured in teal.

The example of structure 2fit also shows that while ConFunc does identify known functional residues and those close to the functional site, other residues remote from the known functional site are also conserved (Figure 7.2). Residues are not only conserved for direct functional roles (such as catalytic residues) but they are also conserved to maintain protein structure. It is possible conserved residues remote from functional sites are conserved for this reason. It may be the case that residues in the protein core are likely to be conserved for structural reasons, so only using residues that are on the surface may improve performance. However, some use of solvent accessibility did not appear to improve Phunctioner predictions (M J E Sternberg, personal communication).

The Z score threshold determines how many residues are classed as conserved. Table 7.2 shows how the number of conserved residues and the proportion that are functional, varies with the Z score threshold. At low thresholds many residues are included and the greatest proportion of functional residues in the both FireDB and the CSA are identified, with up to 80% and 63% of the CSA and FireDB residues conserved respectively. Increasing the threshold reduces the number of conserved residues and also the percentage of functional residues that are identified. However the proportion of conserved residues that are also functional (present in FireDB) increases, such that at the highest thresholds very few residues are conserved but up to 31% of them are functional. While this proportion increases the total number of functional residues used represents a very small part of the total number of functional residues. The reduction in the number of

Z score threshold	*conserved res*	*FireDB res*	*con+ FireDB*	*%FireDB*	*%conserved*	*CSA residues*	*CSA found*	*%CSA found*
0.0	4812	914	578	0.63	0.12	128	98	0.77
0.5	3107	914	432	0.47	0.14	128	87	0.68
1.0	1850	914	323	0.35	0.17	128	81	0.63
1.5	1087	914	215	0.24	0.20	128	62	0.48
2.0	612	914	124	0.14	0.20	128	38	0.30
2.5	268	914	70	0.08	0.26	128	24	0.19

Table 7.2 Comparison of conserved residues with known functional residues at different residue Z score thresholds. Columns as for Table 6.1.

functional residues identified as the Z score threshold increases is observed for both the FireDB and CSA residues showing that both ligand binding and enzymatic functional residues are not necessarily very highly conserved. Over the full test set the Z=1.5 threshold obtains the best performance. In combination with the results from this subset, it suggests that this threshold best balances identifying functional residues with the noise of conserved non-functional residues.

The identification of protein functional sites is a separate and important problem in bioinformatics (see section 1.3.2) that can aid our understanding of function and the mechanisms used to perform these functions. Research is currently being performed in my laboratory to further investigate the ability of ConFunc to identify protein functional sites.

7.4. Practical Application of ConFunc

This work has described the testing of ConFunc against a set of sequences with known function. How can ConFunc be practically used by experimentalists? This section describes a current practical application of ConFunc.

7.4.1. Functional Annotation of *Campylobacter jejuni*

The Systems Biology Centre at Imperial College is currently investigating the functional annotation of four gene clusters in the *Campylobacter jejuni* genome. The gene clusters of interest encode carbohydrate polymerisation proteins and are categorised according to the type of carbohydrate they synthesise (the clusters are called: N-Linked Glycan locus, Lipo-oligosaccharide locus, O-Linked Glycan locus and the Capsule locus.).

This project is a collaboration between experimentalists (Brendan Wren at the London School of Hygiene and Tropical Medicine) and theorists based at Imperial. They aim to obtain greater functional annotation of the gene clusters and ConFunc is being used in a number of ways to achieve this. Firstly, for proteins with limited functional annotations, ConFunc predictions are providing functional suggestions for the experimentalists to investigate. Where putative functions have been identified, ConFunc is

being used, where possible, to corroborate these annotations and suggest potential functions for investigation in cases where there is considerable disagreement between the putative function and the ConFunc prediction. For proteins with experimentally defined functions, ConFunc predictions are being made to assess its performance and to provide feedback for future development of the algorithm.

There are also plans to perform a similar analysis on the proteome of *Plasmodium falciparum*, the parasite responsible for malaria. This research will be conducted in association with the Imperial College Malaria Centre. Only a small proportion of the *P. falciparum* proteome is experimentally characterised. The aim of this collaboration would therefore be to infer function, using ConFunc and other methods, for those unannotated proteins, using the predictions to direct experiments. Further the experimental results will feedback into the prediction process enabling the improvement of future predictions.

Chapter 8

Conclusions

This thesis has described the development of ConFunc, an automated function prediction algorithm. It has shown the ability to group protein sequences by their function, identify conserved residues associated with each function and use these to infer the function of other sequences. This approach has been thoroughly benchmarked and the results and problems associated with it have been discussed. This chapter will describe the scope for future work to further the research described here.

8.1. Future Work

Chapter 3 detailed how the use of different methods and settings for some of the steps within ConFunc can affect its performance. From these results it is clear that the residues used for scoring the query sequence against each GO term PSSM is crucial. The quality of the sequence alignment and the conservation threshold both influence the conserved residues and had significant effects upon performance. It is therefore possible that further improvements to ConFunc could be obtained by modifying these two steps.

Sequence alignment techniques are under continual development (see section 1.2.6). However, sequence alignments are still generally not as good as those obtained from structural alignments. Some alignment methods (such as 3D-Coffee (such as 3D-Coffee O'Sullivan, et al., 2004) are now beginning to incorporate some structural information to obtain improved sequence alignments. A typical approach to do this maps the homologues onto a single structure and uses this mapping to improve the alignment. The availability of such methods could improve the alignments obtained and in turn increase ConFunc performance.

The ConFunc identification of conserved residues uses a simple weighted score that considers the substitution of different amino acids using a BLOSUM matrix. There are many different conservation scores but there is not currently a consensus on which

score, if any, performs the best (Valdar, 2002). The use of different conservation scores has not been investigated in this thesis, so future work could assess if improved results can be obtained using different scores.

The purpose of ConFunc identifying conserved residues is based on the assumption that functionally important residues are likely to be highly conserved. There are numerous approaches that use more than just residue conservation to predict functional residues. Some of these methods could be incorporated into ConFunc and used to determine the residues that are used to characterise each individual function. FRcons (Fischer, et al., 2008) a recent functional residue prediction method, obtains improved predictions by using predicted solvent accessibility and secondary structure predictions. Similar information could be used for ConFunc identification of conserved residues.

Future work could also investigate improving the e-value thresholds used by ConFunc. Chapter 3 demonstrates that it is crucial that such thresholds are able to discriminate between true and false functions. This is made difficult by the majority of GO terms being false. The analysis of ConFunc performance at different levels of GO and for different functional types shows that performance is not uniform across GO. Further investigation of these differences could lead to improved methods for discriminating between GO terms. For example, thresholds could vary depending upon the type of function; binding functions are poorly predicted by ConFunc, so a more stringent threshold could be applied for this functional type.

The domain problem associated with inferring function using sequences that have functional domains not present in the query sequence, is a source of false predictions for ConFunc (see section 0). The association of GO functions with complete sequences, rather than domains or regions of sequence causes this problem. A resource that maps GO functions to domains will soon be available (S. N. Ali, personal communication). This will enable ConFunc to remove from the predictive process, functions present in homologues that are associated with domains not shared with the query. This should lessen the problems caused by multi-domain proteins and could be applied to other sequence based methods.

The use of 3-D motifs is a widely used function prediction approach (see section 1.5.2.2). Predefined libraries of motifs, each representing particular functions, are used to search through target structures to identify similar arrangements of residues. Most of these methods rely upon the use of hand-curated motifs, such as the CSA (Porter, et al., 2004). This thesis has shown that when using sequences with known structure as ConFunc queries, the conserved residues identified by ConFunc are functional and can be mapped onto the structure (section 7.3). Therefore, ConFunc conserved residues could be used to automatically generate motifs associated with particular functions, resulting in a new structural function prediction method.

8.2. Outlook

At the beginning of this research (October 2004), there were relatively few sophisticated function prediction algorithms, especially those that use Gene Ontology to assign function. The past three years has seen the development of a number of advanced methods for inferring function from both sequence and structure. These include GOtcha (Martin, et al., 2004), PFP (Hawkins, et al., 2006), SIFTER (Engelhardt, et al., 2005) and of course ConFunc. All of these methods have demonstrated the ability to infer function better than standard methods of prediction, particularly annotation transfer using BLAST or Pfam hits. However, the predictions are also far from perfect and when tested over a large set of proteins have rarely obtained precision greater than 80%. There is clearly room for improvement within the field. It is now time for the field to consider combining the approaches that have been developed in recent years. Servers using combined methods (section 1.5.4) have begun to appear, first with ProFunc (Laskowski, et al., 2005a) and ProKnow (Pal and Eisenberg, 2005) and more recently Jafa (Friedberg, et al., 2006a). These servers perform numerous different analyses but with the exception of ProKnow, they make little attempt to generate consensus predictions from different sources, which is left for the user to address. Unfortunately, with the escalating levels of sequence data, it is essential that such methods are automated.

The simplest approach would be to rank predictions based on the number of methods that agree. More sophisticated approaches might include the use of machine

learning techniques, such as SVMs (Support Vector Machines; Vapnik, 1995) or ILP (Muggleton, 1991) to identify and use patterns in the predictions. The various prediction methods use different data sources to make their predictions, so it seems likely that the best approach would be to consider the sources and evidence used for the predictions. This is best explained with an example. Consider a protein of known structure that sequence methods predict to be an enzyme. Now other sources of prediction can be used to assess if there is further evidence for this. Are structural motifs associated with this enzymatic function present in the structure? The same could apply to ligand binding predictions, to check that the protein does in fact have a suitable site to bind the proposed ligand. Due to chance hits, 3-D motifs are often associated with high levels of false positives. Such predictions can be further supported by considering the location of the motif. Enzyme substrates and other ligands have a tendency to bind to the largest or second largest cleft on a protein surface (Laskowski, et al., 1996), so 3-D motif hits present in large clefts are more likely to be biologically relevant. Such an approach has already been used for a set of enzyme active sites (Chen, et al., 2006). An approach that logically combined the available data in this way could prove very effective for inferring function.

The function prediction assessments during AFP and CASP6 and 7 demonstrate the difficulty of using predictive methods to correctly identify protein function. The development of further novel methods, in conjunction with combining existing approaches will continue to result in further improvements for this biologically important problem. Indeed the level of interest at the first three AFP meeting suggests that the field is growing and continuing to innovate, proposing new solutions to this problem.

Acknowledgements

Looking back at this work, I realise the amount of effort that has been required to complete it, not just from myself but also those around me. So special thanks go to Sara Dobbins and Victor Lesk, who sit closest to me in the lab and have had to try to answer my many questions over the past three years. More annoyingly they have had to tolerate my moaning and complaints! I cannot forget the other members of the structural bioinformatics group, who have provided support along the way and most importantly have ensured an enjoyable environment to work in. Thanks to Suhail Islam for the excellent systems and graphics support has been provided.

Thanks also to Prof. Michael Sternberg for the supervision that he has provided. He has also provided many opportunities over the past three years, particularly to travel to conferences and get an insight into the rest of the function prediction world.

Finally, I would like to thank my friends and Family for the support they have given me during my PhD, although most of the time they have no idea what I am talking about!

References

Abascal, F. and Valencia, A. (2003) Automatic annotation of protein function based on family identification, *Proteins*, **53**, 683-692.

Abhiman, S. and Sonnhammer, E.L. (2005a) FunShift: a database of function shift analysis on protein subfamilies, *Nucleic Acids Res*, **33**, D197-200.

Abhiman, S. and Sonnhammer, E.L. (2005b) Large-scale prediction of function shift in protein families with a focus on enzymatic function, *Proteins*, **60**, 758-768.

Aloy, P., Querol, E., Aviles, F.X. and Sternberg, M.J. (2001) Automated structure-based prediction of functional sites in proteins: applications to assessing the validity of inheriting protein function from homology in genome annotation and to protein docking, *J Mol Biol*, **311**, 395-408.

Altschul, S.F. and Gish, W. (1996) Local alignment statistics, *Methods Enzymol*, **266**, 460-480.

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.

Altschul, S.F. and Koonin, E.V. (1998) Iterated profile searches with PSI-BLAST--a tool for discovery in protein databases, *Trends Biochem Sci*, **23**, 444-447.

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.

Andrade, M.A., Brown, N.P., Leroy, C., Hoersch, S., de Daruvar, A., et al. (1999) Automated genome sequence analysis and annotation, *Bioinformatics*, **15**, 391-412.

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**, D226-229.

Anfinsen, C.B. and Haber, E. (1961) Studies on the reduction and re-formation of protein disulfide bonds, *J Biol Chem*, **236**, 1361-1363.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nat Genet*, **25**, 25-29.

Ashburner, M. and Drysdale, R. (1994) FlyBase--the Drosophila genetic database, *Development*, **120**, 2077-2079.

Attwood, T.K., Blythe, M.J., Flower, D.R., Gaulton, A., Mabey, J.E., et al. (2002) PRINTS and PRINTS-S shed light on protein ancestry, *Nucl. Acids Res.*, **30**, 239-241.

Audit, B., Levy, E.D., Gilks, W.R., Goldovsky, L. and Ouzounis, C.A. (2007) CORRIE: enzyme sequence annotation with confidence estimates, *BMC Bioinformatics*, **8 Suppl 4**, S3.

Ausiello, G., Via, A. and Helmer-Citterich, M. (2005) Query3d: a new method for high-throughput analysis of functional residues in protein structures, *BMC Bioinformatics*, **6 Suppl 4**, S5.

Babbitt, P.C. (2003) Definitions of enzyme function for the structural genomics era, *Curr Opin Chem Biol*, **7**, 230-237.

Bairoch, A. (2000) The ENZYME database in 2000, *Nucleic Acids Res*, **28**, 304-305.

Balaji, S., Sujatha, S., Kumar, S.S. and Srinivasan, N. (2001) PALI-a database of Phylogeny and ALIgnment of homologous protein structures, *Nucleic Acids Res*, **29**, 61-65.

Bateman, A., Birney, E., Durbin, R., Eddy, S.R., Finn, R.D. and Sonnhammer, E.L. (1999) Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins, *Nucl. Acids Res.*, **27**, 260-262.

Bellman, R. (1957) *Dynamic Programming.* Princeton University Press, Princeton, New Jersey, USA.

Berezin, C., Glaser, F., Rosenberg, J., Paz, I., Pupko, T., Fariselli, P., Casadio, R. and Ben-Tal, N. (2004) ConSeq: the identification of functionally and structurally important residues in protein sequences, *Bioinformatics*, **20**, 1322-1324.

Binkowski, T.A., Adamian, L. and Liang, J. (2003) Inferring Functional Relationships of Proteins from Local Sequence and Spatial Surface Patterns, *Journal of Molecular Biology*, **332**, 505-526.

Bork, P., Dandekar, T., Diaz-Lazcoz, Y., Eisenhaber, F., Huynen, M. and Yuan, Y. (1998) Predicting function: from genes to genomes and back, *J Mol Biol*, **283**, 707-725.

Brenner, S.E. (1999) Errors in genome annotation, *Trends Genet*, **15**, 132-133.

Camon, E., Barrell, D., Lee, V., Dimmer, E. and Apweiler, R. (2004) The Gene Ontology Annotation (GOA) Database--an integrated resource of GO annotations to the UniProt Knowledgebase, *In Silico Biol*, **4**, 5-6.

Chakrabarti, S., Bryant, S.H. and Panchenko, A.R. (2007) Functional Specificity Lies within the Properties and Evolutionary Changes of Amino Acids, *Journal of Molecular Biology*, **373**, 801-810.

Chen, B.Y., Bryant, D.H., Fofanov, V.Y., Kristensen, D.M., Cruess, A.E., Kimmel, M., Lichtarge, O. and Kavraki, L.E. (2006) Cavity-aware motifs reduce false positives in protein function prediction, *Comput Syst Bioinformatics Conf*, 311-323.

Cheng, H., Kim, B.-H. and Grishin, N.V. (2008) Discrimination between Distant Homologs and Structural Analogs: Lessons from Manually Constructed, Reliable Data Sets, *Journal of Molecular Biology*, **377**, 1265-1278.

Chernobai, A., Rachev, S. and Fabozzi, F. (2005) Composite goodness-of-fit tests for left-truncated loss samples, *Technical report, University of California Santa Barbara*

Chothia, C. and Lesk, A.M. (1986) The relation between the divergence of sequence and structure in proteins, *Embo J*, **5**, 823-826.

Chua, H.N., Sung, W.K. and Wong, L. (2006) Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions, *Bioinformatics*, **22**, 1623-1630.

Citerne, H.L., Luo, D., Pennington, R.T., Coen, E. and Cronk, Q.C.B. (2003) A Phylogenomic Investigation of CYCLOIDEA-Like TCP Genes in the Leguminosae, *Plant Physiol.*, **131**, 1042-1053.

Clare, A., Karwath, A. and King, R.D. (2006) Functional bioinformatics for Arabidopsis thaliana, *Bioinformatics*.

Collins, J.F., Coulson, A.F.W. and Lyall, A. (1988) The significance of protein sequence similarities, *Comput. Appl. Biosci.*, **4**, 67-71.

Consortium, I.C.G.S. (2004) Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution, *Nature*, **432**, 695-716.

Davis, J. and Goadrich, M. (2006) The Relationship Between Precision-Recall and ROC Curves, *23rd International Conference on Machine Learning (ICML), Pittsburgh, PA, USA, 26th - 28th June, 2006*

Dayhoff, M.O., Schwartz, A.S. and Orcutt, B.C. (1978) *Atlas of Protein Sequence and Structure vol. 5 Suppl. 3*. Natl. Biomed. Res. Found., Washington, DC, USA: .

del Sol Mesa, A., Pazos, F. and Valencia, A. (2003) Automatic methods for predicting functionally important residues, *J Mol Biol*, **326**, 1289-1302.

Devos, D. and Valencia, A. (2000) Practical limits of function prediction, *Proteins*, **41**, 98-107.

Devos, D. and Valencia, A. (2001) Intrinsic errors in genome annotation, *Trends Genet*, **17**, 429-431.

Do, C.B., Mahabhashyam, M.S., Brudno, M. and Batzoglou, S. (2005) ProbCons: Probabilistic consistency-based multiple sequence alignment, *Genome Res*, **15**, 330-340.

Dodge, C., Schneider, R. and Sander, C. (1998) The HSSP database of protein structure-sequence alignments and family profiles, *Nucleic Acids Res*, **26**, 313-315.

Drenth, J., Hol, W.G., Jansonius, J.N. and Koekoek, R. (1972) A comparison of the three-dimensional structures of subtilisin BPN' and subtilisin novo, *Cold Spring Harb Symp Quant Biol*, **36**, 107-116.

Eddy, S.R. (1997) Maximum Likelihood Fitting of Extreme Value Distributions.

Eddy, S.R. (1998) Profile hidden Markov models, *Bioinformatics*, **14**, 755-763.

Edgar, R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Res*, **32**, 1792-1797.

Edgar, R.C. and Batzoglou, S. (2006) Multiple sequence alignment, *Curr Opin Struct Biol*, **16**, 368-373.

Eisen, J.A. (1998) Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis, *Genome Res*, **8**, 163-167.

Eisen, J.A. and Hanawalt, P.C. (1999) A phylogenomic study of DNA repair genes, proteins, and processes, *Mutation Research/DNA Repair*, **435**, 171-213.

Eisen, J.A., Nelson, K.E., Paulsen, I.T., Heidelberg, J.F., Wu, M., et al. (2002) The complete genome sequence of Chlorobium tepidum TLS, a photosynthetic, anaerobic, green-sulfur bacterium, *Proceedings of the National Academy of Sciences*, **99**, 9509-9514.

Eisen, J.A., Sweder, K.S. and Hanawalt, P.C. (1995) Evolution of the SNF2 family of proteins: subfamilies with distinct sequences and functions, *Nucleic Acids Res*, **23**, 2715-2723.

Engelhardt, B.E., Jordan, M.I., Muratore, K.E. and Brenner, S.E. (2005) Protein Molecular Function Prediction by Bayesian Phylogenomics, *PLoS Comput Biol*, **1**, e45.

Felsenstein, J. (2004) *Inferring Phylogenies*. Sinauer Associates, Sunderland, Massachusetts.

Feng, D.F. and Doolittle, R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees, *J Mol Evol*, **25**, 351-360.

Finn, R.D., Mistry, J., Schuster-Bockler, B., Griffiths-Jones, S., Hollich, V., et al. (2006) Pfam: clans, web tools and services, *Nucleic Acids Res*, **34**, D247-251.

Fischer, J.D., Mayer, C.E. and Soding, J. (2008) Prediction of Protein Functional Residues from Sequence by Probability Density Estimation, *Bioinformatics*, btm626.

Friedberg, I. (2006c) Automated protein function prediction--the genomic challenge, *Brief Bioinform*, **7**, 225-242.

Friedberg, I., Harder, T. and Godzik, A. (2006a) JAFA: a protein function annotation meta-server, *Nucl. Acids Res.*, **34**, W379-381.

Friedberg, I., Jambon, M. and Godzik, A. (2006b) New avenues in protein function prediction, *Protein Sci*, **15**, 1527-1529.

Friedberg, I., Kaplan, T. and Margalit, H. (2000) Evaluation of PSI-BLAST alignment accuracy in comparison to structural alignments, *Protein Sci*, **9**, 2278-2284.

Galperin, M.Y., Walker, D.R. and Koonin, E.V. (1998) Analogous enzymes: independent inventions in enzyme evolution, *Genome Res*, **8**, 779-790.

George, R.A., Spriggs, R.V., Bartlett, G.J., Gutteridge, A., MacArthur, M.W., Porter, C.T., Al-Lazikani, B., Thornton, J.M. and Swindells, M.B. (2005) Effective function annotation through catalytic residue conservation, *Proc Natl Acad Sci U S A*, **102**, 12299-12304.

George, R.A., Spriggs, R.V., Thornton, J.M., Al-Lazikani, B. and Swindells, M.B. (2004) SCOPEC: a database of protein catalytic domains, *Bioinformatics*, **20 Suppl 1**, I130-I136.

Gerlt, J.A. and Babbitt, P.C. (2000) Can sequence determine function?, *Genome Biol*, **1**, REVIEWS0005.

Gherardini, P.F., Wass, M.N., Helmer-Citterich, M. and Sternberg, M.J. (2007) Convergent Evolution of Enzyme Active Sites Is not a Rare Phenomenon, *J Mol Biol*, **372**, 817-845.

Glaser, F., Morris, R.J., Najmanovich, R.J., Laskowski, R.A. and Thornton, J.M. (2006) A method for localizing ligand binding pockets in protein structures, *Proteins*, **62**, 479-488.

Gough, J., Karplus, K., Hughey, R. and Chothia, C. (2001) Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure, *Journal of Molecular Biology*, **313**, 903-919.

Greene, L.H., Lewis, T.E., Addou, S., Cuff, A., Dallman, T., et al. (2007) The CATH domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution, *Nucleic Acids Res*, **35**, D291-297.

Groth, D., Lehrach, H. and Hennig, S. (2004) GOblet: a platform for Gene Ontology annotation of anonymous sequence data, *Nucleic Acids Res*, **32**, W313-317.

Gutteridge, A., Bartlett, G.J. and Thornton, J.M. (2003) Using a neural network and spatial clustering to predict the location of active sites in enzymes, *J Mol Biol*, **330**, 719-734.

Haft, D.H., Selengut, J.D. and White, O. (2003) The TIGRFAMs database of protein families, *Nucleic Acids Res*, **31**, 371-373.

Hannenhalli, S.S. and Russell, R.B. (2000) Analysis and prediction of functional sub-types from protein sequence alignments, *J Mol Biol*, **303**, 61-76.

Hawkins, T., Luban, S. and Kihara, D. (2006) Enhanced automated function prediction using distantly related sequences and contextual association by PFP, *Protein Sci*, **15**, 1550-1556.

Hegyi, H. and Gerstein, M. (1999) The relationship between protein structure and function: a comprehensive survey with application to the yeast genome, *J Mol Biol*, **288**, 147-164.

Hegyi, H. and Gerstein, M. (2001) Annotation transfer for genomics: measuring functional divergence in multi-domain proteins, *Genome Res*, **11**, 1632-1640.

Henikoff, J.G. and Henikoff, S. (1996) Blocks database and its applications, *Methods Enzymol*, **266**, 88-105.

Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks, *Proc Natl Acad Sci U S A*, **89**, 10915-10919.

Henikoff, S. and Henikoff, J.G. (1994) Position-based sequence weights, *J Mol Biol*, **243**, 574-578.

Hishigaki, H., Nakai, K., Ono, T., Tanigami, A. and Takagi, T. (2001) Assessment of prediction accuracy of protein function from protein--protein interaction data, *Yeast*, **18**, 523-531.

Holliday, G.L., Bartlett, G.J., Almonacid, D.E., O'Boyle, N.M., Murray-Rust, P., Thornton, J.M. and Mitchell, J.B. (2005) MACiE: a database of enzyme reaction mechanisms, *Bioinformatics*, **21**, 4315-4316.

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.

Holm, L. and Sander, C. (1994) The FSSP database of structurally aligned protein fold families, *Nucleic Acids Res*, **22**, 3600-3609.

Holm, L. and Sander, C. (1995) Dali: a network tool for protein structure comparison, *Trends Biochem Sci*, **20**, 478-480.

Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., De Castro, E., Langendijk-Genevaux, P.S., Pagni, M. and Sigrist, C.J. (2006) The PROSITE database, *Nucleic Acids Res*, **34**, D227-230.

Hwang, H., Pierce, B., Mintseris, J., Janin, J. and Weng, Z. (2008) Protein-protein docking benchmark version 3.0, *Proteins*.

Innis, C.A., Anand, A.P. and Sowdhamini, R. (2004) Prediction of functional sites in proteins using conserved functional group analysis, *J Mol Biol*, **337**, 1053-1068.

Jambon, M., Imberty, A., Deleage, G. and Geourjon, C. (2003) A new bioinformatic approach to detect common 3D sites in protein structures, *Proteins*, **52**, 137-145.

Jones, C.E., Baumann, U. and Brown, A.L. (2005) Automated methods of predicting the function of biological sequences using GO and BLAST, *BMC Bioinformatics*, **6**, 272.

Jones, C.E., Brown, A.L. and Baumann, U. (2007) Estimating the annotation error rate of curated GO database sequence annotations, *BMC Bioinformatics*, **8**, 170.

Karlin, S. and Altschul, S.F. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes, *Proc Natl Acad Sci U S A*, **87**, 2264-2268.

Karlin, S. and Altschul, S.F. (1993) Applications and Statistics for Multiple High-Scoring Segments in Molecular Sequences, *Proceedings of the National Academy of Sciences*, **90**, 5873-5877.

Katoh, K., Kuma, K., Toh, H. and Miyata, T. (2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment, *Nucleic Acids Res*, **33**, 511-518.

Katoh, K., Misawa, K., Kuma, K. and Miyata, T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform, *Nucleic Acids Res*, **30**, 3059-3066.

Khan, S., Situ, G., Decker, K. and Schmidt, C.J. (2003) GoFigure: automated Gene Ontology annotation, *Bioinformatics*, **19**, 2484-2485.

King, R.D., Karwath, A., Clare, A. and Dehaspe, L. (2000) Accurate prediction of protein functional class from sequence in the Mycobacterium tuberculosis and Escherichia coli genomes using data mining, *Yeast*, **17**, 283-293.

King, R.D., Karwath, A., Clare, A. and Dehaspe, L. (2001) The utility of different representations of protein sequence for predicting functional class, *Bioinformatics*, **17**, 445-454.

Kleywegt, G.J. (1999) Recognition of spatial motifs in protein structures, *J Mol Biol*, **285**, 1887-1897.

Kraut, J. (1977) Serine proteases: structure and mechanism of catalysis, *Annu Rev Biochem*, **46**, 331-358.

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.

Kristensen, D.M., Chen, B.Y., Fofanov, V.Y., Ward, R.M., Lisewski, A.M., Kimmel, M., Kavraki, L.E. and Lichtarge, O. (2006) Recurrent use of evolutionary importance for functional annotation of proteins based on local structural similarity, *Protein Sci*, **15**, 1530-1536.

Landgraf, R., Fischer, D. and Eisenberg, D. (1999) Analysis of heregulin symmetry by weighted evolutionary tracing, *Protein Eng.*, **12**, 943-951.

Laskowski, R.A. (1995) SURFNET: a program for visualizing molecular surfaces, cavities, and intermolecular interactions, *J Mol Graph*, **13**, 323-330, 307-328.

Laskowski, R.A., Luscombe, N.M., Swindells, M.B. and Thornton, J.M. (1996) Protein clefts in molecular recognition and function, *Protein Sci*, **5**, 2438-2452.

Laskowski, R.A., Watson, J.D. and Thornton, J.M. (2005a) ProFunc: a server for predicting protein function from 3D structure, *Nucleic Acids Res*, **33**, W89-93.

Laskowski, R.A., Watson, J.D. and Thornton, J.M. (2005b) Protein function prediction using local 3D templates, *J Mol Biol*, **351**, 614-626.

Levitt, D.G. and Banaszak, L.J. (1992) POCKET: a computer graphics method for identifying and displaying protein cavities and their surrounding amino acids, *J Mol Graph*, **10**, 229-234.

Levy, E.D., Ouzounis, C.A., Gilks, W.R. and Audit, B. (2005) Probabilistic annotation of protein sequences based on functional classifications, *BMC Bioinformatics*, **6**, 302.

Li, W., Jaroszewski, L. and Godzik, A. (2002) Tolerating some redundancy significantly speeds up clustering of large protein databases, *Bioinformatics*, **18**, 77-82.

Lichtarge, O., Bourne, H.R. and Cohen, F.E. (1996) An evolutionary trace method defines binding surfaces common to protein families, *J Mol Biol*, **257**, 342-358.

Lichtarge, O., Yao, H., Kristensen, D.M., Madabushi, S. and Mihalek, I. (2003) Accurate and scalable identification of functional sites by evolutionary tracing, *J Struct Funct Genomics*, **4**, 159-166.

Lopez, G., Rojas, A., Tress, M. and Valencia, A. (2007c) Assessment of predictions submitted for the CASP7 function prediction category, *Proteins*, **69 Suppl 8**, 165-174.

Lopez, G., Valencia, A. and Tress, M. (2007b) FireDB--a database of functionally important residues from proteins of known structure, *Nucl. Acids Res.*, **35**, D219-223.

Lopez, G., Valencia, A. and Tress, M.L. (2007a) firestar--prediction of functionally important residues using structural templates and alignment reliability, *Nucleic Acids Res.*

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.

Marchler-Bauer, A., Anderson, J.B., Derbyshire, M.K., DeWeese-Scott, C., Gonzales, N.R., et al. (2007) CDD: a conserved domain database for interactive domain family analysis, *Nucleic Acids Res*, **35**, D237-240.

Marcotte, E.M., Pellegrini, M., Ng, H.L., Rice, D.W., Yeates, T.O. and Eisenberg, D. (1999a) Detecting protein function and protein-protein interactions from genome sequences, *Science*, **285**, 751-753.

Marcotte, E.M., Pellegrini, M., Thompson, M.J., Yeates, T.O. and Eisenberg, D. (1999b) A combined algorithm for genome-wide prediction of protein function, *Nature*, **402**, 83-86.

Martin, A.C. (2004) PDBSprotEC: a Web-accessible database linking PDB chains to EC numbers via SwissProt, *Bioinformatics*, **20**, 986-988.

Martin, A.C. (2005) Mapping PDB chains to UniProtKB entries, *Bioinformatics*, **21**, 4297-4301.

Martin, A.C., Orengo, C.A., Hutchinson, E.G., Jones, S., Karmirantzou, M., Laskowski, R.A., Mitchell, J.B., Taroni, C. and Thornton, J.M. (1998) Protein folds and functions, *Structure*, **6**, 875-884.

Martin, D.M., Berriman, M. and Barton, G.J. (2004) GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes, *BMC Bioinformatics*, **5**, 178.

Matthews, B.W., Sigler, P.B., Henderson, R. and Blow, D.M. (1967) Three-dimensional structure of tosyl-alpha-chymotrypsin, *Nature*, **214**, 652-656.

McNemar, Q. (1947) Note on the sampling error of the difference between correlated proportions of percentages., *Psychometrica*, **12**, 153-157.

Mendenhall, W., Beaver, B.M. and Beaver, R.J. (1999) *Introduction to Probability and Statistics*. Duxbury Press.

Mi, H., Lazareva-Ulitsky, B., Loo, R., Kejariwal, A., Vandergriff, J., et al. (2005) The PANTHER database of protein families, subfamilies, functions and pathways, *Nucleic Acids Res*, **33**, D284-288.

Moult, J., Fidelis, K., Zemla, A. and Hubbard, T. (2003) Critical assessment of methods of protein structure prediction (CASP)-round V, *Proteins*, **53 Suppl 6**, 334-339.

Muggleton, S. (1991) Inductive logic programming, *New Gen. Comput.*, **8**, 295-318.

Mulder, N.J., Apweiler, R., Attwood, T.K., Bairoch, A., Bateman, A., et al. (2005) InterPro, progress and status in 2005, *Nucleic Acids Res*, **33**, D201-205.

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.

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.

Notredame, C., Higgins, D.G. and Heringa, J. (2000) T-Coffee: A novel method for fast and accurate multiple sequence alignment, *J Mol Biol*, **302**, 205-217.

O'Sullivan, O., Suhre, K., Abergel, C., Higgins, D.G. and Notredame, C. (2004) 3DCoffee: combining protein sequences and structures within multiple sequence alignments, *J Mol Biol*, **340**, 385-395.

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.

Ouzounis, C.A., Coulson, R.M., Enright, A.J., Kunin, V. and Pereira-Leal, J.B. (2003) Classification schemes for protein structure and function, *Nat Rev Genet*, **4**, 508-519.

Ouzounis, C.A. and Karp, P.D. (2002) The past, present and future of genome-wide re-annotation, *Genome Biol*, **3**, COMMENT2001.

Pal, D. and Eisenberg, D. (2005) Inference of protein function from protein structure, *Structure*, **13**, 121-130.

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, *Journal of Molecular Biology*, **284**, 1201-1210.

Pazos, F. and Bang, J.-W. (2006) Computational Prediction of Functionally Important Regions in Proteins, *Curr Bioinform*, **1**, 15-23.

Pazos, F., Rausell, A. and Valencia, A. (2006) Phylogeny-independent detection of functional residues, *Bioinformatics*, **22**, 1440-1448.

Pazos, F. and Sternberg, M.J. (2004) Automated prediction of protein function and detection of functional sites from structure, *Proc Natl Acad Sci U S A*, **101**, 14754-14759.

Pearson, W.R. (1996) Effective protein sequence comparison, *Methods Enzymol*, **266**, 227-258.

Pearson, W.R. and Lipman, D.J. (1988) Improved Tools for Biological Sequence Comparison, *Proceedings of the National Academy of Sciences*, **85**, 2444-2448.

Pellegrini-Calace, M., Soro, S. and Tramontano, A. (2006) Revisiting the prediction of protein function at CASP6, *Febs J*, **273**, 2977-2983.

Polacco, B.J. and Babbitt, P.C. (2006) Automated discovery of 3D motifs for protein function annotation, *Bioinformatics*, **22**, 723-730.

Pollak, N., Dolle, C. and Ziegler, M. (2007) The power to reduce: pyridine nucleotides--small molecules with a multitude of functions, *Biochem J*, **402**, 205-218.

Porter, C.T., Bartlett, G.J. and Thornton, J.M. (2004) The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data, *Nucleic Acids Res*, **32**, D129-133.

Pupko, T., Bell, R.E., Mayrose, I., Glaser, F. and Ben-Tal, N. (2002) Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues, *Bioinformatics*, **18**, S71-77.

Riley, M. (1998) Systems for categorizing functions of gene products, *Curr Opin Struct Biol*, **8**, 388-392.

Rison, S.C., Hodgman, T.C. and Thornton, J.M. (2000) Comparison of functional annotation schemes for genomes, *Funct Integr Genomics*, **1**, 56-69.

Romano, J.P. (1988) Bootstrap and Randomization Tests of some Nonparametric Hypotheses.

Rost, B. (2002) Enzyme function less conserved than anticipated, *J Mol Biol*, **318**, 595-608.

Rost, B., Liu, J., Nair, R., Wrzeszczynski, K.O. and Ofran, Y. (2003) Automatic prediction of protein function, *Cell Mol Life Sci*, **60**, 2637-2650.

Russell, R.B. (1998) Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution, *J Mol Biol*, **279**, 1211-1227.

Samanta, M.P. and Liang, S. (2003) Predicting protein functions from redundancies in large-scale protein interaction networks, *Proc Natl Acad Sci U S A*, **100**, 12579-12583.

Schug, J., Diskin, S., Mazzarelli, J., Brunk, B.P. and Stoeckert, C.J., Jr. (2002) Predicting gene ontology functions from ProDom and CDD protein domains, *Genome Res*, **12**, 648-655.

Schwikowski, B., Uetz, P. and Fields, S. (2000) A network of protein-protein interactions in yeast, *Nat Biotechnol*, **18**, 1257-1261.

Seffernick, J.L., de Souza, M.L., Sadowsky, M.J. and Wackett, L.P. (2001) Melamine deaminase and atrazine chlorohydrolase: 98 percent identical but functionally different, *J Bacteriol*, **183**, 2405-2410.

Sela, M., White, F.H., Jr. and Anfinsen, C.B. (1957) Reductive cleavage of disulfide bridges in ribonuclease, *Science*, **125**, 691-692.

Servant, F., Bru, C., Carrere, S., Courcelle, E., Gouzy, J., Peyruc, D. and Kahn, D. (2002) ProDom: automated clustering of homologous domains, *Brief Bioinform*, **3**, 246-251.

Sigrist, C.J., Cerutti, L., Hulo, N., Gattiker, A., Falquet, L., Pagni, M., Bairoch, A. and Bucher, P. (2002) PROSITE: a documented database using patterns and profiles as motif descriptors, *Brief Bioinform*, **3**, 265-274.

Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences, *Journal of Molecular Biology*, **147**, 195-197.

Smith, T.F., Waterman, M.S. and Burks, C. (1985) The statistical distribution of nucleic acid similarities, *Nucleic Acids Res*, **13**, 645-656.

Soro, S. and Tramontano, A. (2005) The prediction of protein function at CASP6, *Proteins*, **61 Suppl 7**, 201-213.

Stark, A., Sunyaev, S. and Russell, R.B. (2003) A Model for Statistical Significance of Local Similarities in Structure, *Journal of Molecular Biology*, **326**, 1307-1316.

Strong, M., Mallick, P., Pellegrini, M., Thompson, M.J. and Eisenberg, D. (2003) Inference of protein function and protein linkages in Mycobacterium tuberculosis based on prokaryotic genome organization: a combined computational approach, *Genome Biol*, **4**, R59.

The_Uniprot_Consortium (2007) The Universal Protein Resource (UniProt), *Nucleic Acids Res*, **35**, D193-197.

Thompson, J.D., Higgins, D.G. and Gibson, T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucl. Acids Res.*, **22**, 4673-4680.

Thompson, J.D., Koehl, P., Ripp, R. and Poch, O. (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark, *Proteins*, **61**, 127-136.

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.

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.

Tress, M.L., Grana, O. and Valencia, A. (2004) SQUARE--determining reliable regions in sequence alignments, *Bioinformatics*, **20**, 974-975.

Valdar, W.S. (2002) Scoring residue conservation, *Proteins*, **48**, 227-241.

Valdar, W.S. and Thornton, J.M. (2001) Protein-protein interfaces: analysis of amino acid conservation in homodimers, *Proteins*, **42**, 108-124.

Van Walle, I., Lasters, I. and Wyns, L. (2005) SABmark--a benchmark for sequence alignment that covers the entire known fold space, *Bioinformatics*, **21**, 1267-1268.

Vapnik, V. (1995) The Nature of Statistical Learning Theory, *Springer-Verlag,New York, USA.*

Vazquez, A., Flammini, A., Maritan, A. and Vespignani, A. (2003) Global protein function prediction from protein-protein interaction networks, *Nat Biotechnol*, **21**, 697-700.

Verspoor, K., Cohn, J., Mniszewski, S. and Joslyn, C. (2006) A categorization approach to automated ontological function annotation, *Protein Sci*, **15**, 1544-1549.

Vinayagam, A., del Val, C., Schubert, F., Eils, R., Glatting, K.H., Suhai, S. and Konig, R. (2006) GOPET: a tool for automated predictions of Gene Ontology terms, *BMC Bioinformatics*, **7**, 161.

Vingron, M. and Argos, P. (1989) A fast and sensitive multiple sequence alignment algorithm, *Comput Appl Biosci*, **5**, 115-121.

Wangikar, P.P., Tendulkar, A.V., Ramya, S., Mali, D.N. and Sarawagi, S. (2003) Functional Sites in Protein Families Uncovered via an Objective and Automated Graph Theoretic Approach, *Journal of Molecular Biology*, **326**, 955-978.

Wass, M.N. and Sternberg, M.J. (2008) ConFunc - Functional Annotation in The Twilight Zone, *Bioinformatics*, **24,** 798-806.

Waterman, M. and Vingron, M. (1994) Sequence Comparison Significance and Poisson Approximation, *Stat. Sci.*, **9**, 367-381.

Waterston, R.H., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J.F., et al. (2002) Initial sequencing and comparative analysis of the mouse genome, *Nature*, **420**, 520-562.

Watson, J.D., Sanderson, S., Ezersky, A., Savchenko, A., Edwards, A., Orengo, C., Joachimiak, A., Laskowski, R.A. and Thornton, J.M. (2007) Towards Fully Automated Structure-based Function Prediction in Structural Genomics: A Case Study, *J Mol Biol*, **367**, 1511-1522.

Webb (1992) Enzyme Nomenclature 1992. Recommendations of the Nomenclature Committee of The International Union of Biochemistry and Molecular Biology, Academic Press, New York.

Wheeler, T.J. and Kececioglu, J.D. (2007) Multiple alignment by aligning alignments, *Bioinformatics*, **23**, i559-i568.

Wilson, C.A., Kreychman, J. and Gerstein, M. (2000) Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores, *J Mol Biol*, **297**, 233-249.

Wu, C.H., Apweiler, R., Bairoch, A., Natale, D.A., Barker, W.C., et al. (2006) The Universal Protein Resource (UniProt): an expanding universe of protein information, *Nucleic Acids Res*, **34**, D187-191.

Wu, C.H., Nikolskaya, A., Huang, H., Yeh, L.S., Natale, D.A., et al. (2004) PIRSF: family classification system at the Protein Information Resource, *Nucleic Acids Res*, **32**, D112-114.

Wu, T.T. and Kabat, E.A. (1970) An analysis of the sequences of the variable regions of bence jones proteins and myeloma light chains and their implications for antibody complementarity, *J. Exp. Med.*, **132**, 211-250.

Xie, H., Wasserman, A., Levine, Z., Novik, A., Grebinskiy, V., Shoshan, A. and Mintz, L. (2002) Large-scale protein annotation through gene ontology, *Genome Res*, **12**, 785-794.

Yao, H., Kristensen, D.M., Mihalek, I., Sowa, M.E., Shaw, C., Kimmel, M., Kavraki, L. and Lichtarge, O. (2003) An Accurate, Sensitive, and Scalable Method to Identify Functional Sites in Protein Structures, *Journal of Molecular Biology*, **326**, 255-261.

Yeats, C., Lees, J., Reid, A., Kellam, P., Martin, N., Liu, X. and Orengo, C. (2007) Gene3D: comprehensive structural and functional annotation of genomes, *Nucleic Acids Res.*

Yooseph, S., Sutton, G., Rusch, D.B., Halpern, A.L., Williamson, S.J., et al. (2007) The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families, *PLoS Biol*, **5**, e16.

Zehetner, G. (2003) OntoBlast function: From sequence similarities directly to potential functional annotations by ontology terms, *Nucleic Acids Res*, **31**, 3799-3803.

Zhang, S., Jin, G., Zhang, X.S. and Chen, L. (2007) Discovering functions and revealing mechanisms at molecular level from biological networks, *Proteomics*, **7**, 2856-2869.