DAROGAN: Enzyme Function Prediction from Multiple Sequence Alignments

DAROGAN
Enzyme Function Prediction

A thesis presented for the degree of

Doctor of Philosophy

at the University of Edinburgh

Russell S. Hamilton

Biocomputing Research Unit

Institute of Cell and Molecular Biology

University of Edinburgh

February 2006

Abstract

Genome sequencing projects are leading to large numbers of protein sequences being deposited in the sequence databases. Unfortunately the amount of accessible functional and structural information for these sequences is advancing more slowly, leaving an ever widening gap in our knowledge. To experimentally characterise each of the sequences would be a massive undertaking, being both time consuming and expensive. Hence there is an ever increasing need for computational tools to bridge this gap. A tool for predicting the function of a putative enzyme from its sequence alone, for example, would be extremely valuable, yet a universally successful method remains elusive.

The function of an enzyme is often dependent on a few key functional residues and the principal objective of this project was to develop a novel function prediction system which takes advantage of this by, comparing the conserved amino acids in known enzyme families to those in a putative enzyme. Multiple sequence alignments of well characterised enzyme families (with an E.C. number assigned) are used to create unordered sets of conserved functional residues, termed *Treads*. Comparison of a query proteins *Tread* to the reference *Treads* is undertaken by projecting them in multidimensional space and measuring distance between them. A major advantage of the prediction strategy implemented in DAROGAN is that it should be able to recognise similarities in the functions of enzymes that are not similar in structure or sequence.

To investigate the feasibility of the DAROGAN function prediction strategy the method has been tested with regard to its ability to predict cofactor-dependencies toward pyridoxal-5'-phosphate, thiamin, glutathione and folic acid utilising enzymes. An area of application for DAROGAN is the prediction of previously described enzyme functions in organisms with completed genomes to which no gene and protein sequences could be assigned through the standard annotation processes. Investigations were made into the potential of utilising the DAROGAN method to propose candidates for the missing pyridoxal-5'-phosphate utilising enzymes in the *E. coli* genome according to EcoCyc (Karp *et al.*, 1999). These missing enzymes have either no gene associated with them or have no sequence associated with their gene. Candidates are proposed by assessing the 511 sequences from the GeneQuiz project (Hoersch *et al.*, 2000), to which there are homologues in other species, but with uncertain functions. The assessment takes the form of using the DAROGAN method to determine the similarities of each of the sequences to the reference *Treads*.

The DAROGAN is implemented as a web service, *www.darogan.co.uk*

Declaration

I declare that this thesis was written by myself and that the work detailed in this thesis is my own, except where reference is made to the work of another.

Russell S. Hamilton
February 2006

Acknowledgements

I would like to thank my supervisor Dietlind Gerloff for her enthusiasm and guidance over the last three (and a bit) years. Special thanks go to Ayona Chatterjee for her expert guidance with GEV statistics and Alastair Kerr for proof reading and suggestions with this thesis. I would like to thank all the people in the Biocomputing Research Unit and the Structural Biochemistry Group who have made working in Edinburgh enjoyable. Special thanks go to Paul Taylor for his advice, friendship and sarcasm. I would also like to thank Ilan Davis for providing me with a source of income, thus helping fund me during the writing of this thesis.

And finally I would like to thank my parents, Dave and Sue Hamilton, for their support and encouragement during my PhD.

This work was funded by a BBSRC Special Studentship.

bbsrc
biotechnology and biological sciences research council

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

With the vast number of protein sequences deposited in the sequence databases, it is becoming impossible to determine functional information for each of the proteins experimentally. Proteins can be divided up into two major categories: enzymes and non-enzymes, this thesis is mainly concerned with enzymes so non-enzymes will be largely ignored. Determining the function of a putative enzyme is a very time consuming process, especially if there are little or no clues to the reaction the enzyme catalyses. If there are clear clues, for example the putative enzyme may show significant sequence similarity to an enzyme whose function has already been assigned, then the determination of function is more straightforward. However it is still a time consuming and expensive process to confirm the reaction the enzyme catalyses. To address this problem there has been a requirement to develop computational methods to successfully predict the function of these protein sequences, specifically those encoding enzymes. There

are many tools using a variety of different methods to predict the functions of these putative enzymes, however a universally accepted method remains elusive.

This chapter introduces the field of enzyme function prediction and the aims of this work. The novel method for predicting enzyme function, developed during this project, is introduced by discussing the general approach and specific aims of the method. Full details of the method are discussed in Chapter 3. To put the method into context existing function prediction techniques are outlined, as well as a specific example of a successful enzyme function prediction. Finally the enzyme families used in the development of this function prediction method and the reasons for their selection are discussed.

1.2 What is Enzyme Function?

The term enzyme is derived from the Greek for *in* and *yeast* (en + zyme) and is defined as a biological catalyst. It is therefore most logical to define the function of an enzyme by the reaction it catalyses, or biochemical reaction. Unfortunately the term *function* is used very broadly in the context of enzyme function.

The function of an enzyme can be taken at many different levels. At a high level an enzyme can be described as being globular, structural or being membrane bound. At lower levels the description can be in terms of biochemical reaction, defined by the chemical reaction or substrate specificity of the enzyme. Further details include cofactors and regulatory molecules. The function of an enzyme

can also be described at the cellular level, including interactions with other proteins and the location of the enzyme within the cell. The physiological function is determined by which metabolic pathway the enzyme is part of or the physiological role it plays in the organism. The phenotypic function is the role the enzyme plays in the organism as a whole, observed by deleting or mutating the gene encoding the enzyme (Skolnick & Fetrow, 2000). The definition of enzyme function will be taken to be the *biochemical reaction* the enzyme catalyses from this point forth.

The Nomenclature Committee of the IUBMB (International Union of Biochemistry and Molecular Biology)(Webb, 1992) has set out a classification scheme for enzymes. The scheme assigns an Enzyme Commission (EC) number to each enzyme based on the biochemical reaction it catalyses. The example below shows the classification of alcohol dehydrogenase which has an EC classification of 1.1.1.1.

```
1.-.-.-  Oxidoreductases.
1.1.-.-    Acting on the CH-OH group of donors.
1.1.1.-      With NAD(+) or NADP(+) as acceptor.
1.1.1.1        Alcohol dehydrogenase
```

All EC classification numbers consist of four numerical terms. The first term is the highest level of classification with the fourth term corresponding to the individual enzyme. The classification of alcohol dehydrogenase becomes clearer from the chemical reaction it catalyses:

$$\text{An alcohol} + \text{NAD(+)} \rightleftharpoons \text{an aldehyde or ketone} + \text{NADH}$$

There are six classes at the top of the EC numbering hierarchy. The first is for the oxidoreductases, as mentioned above, the full set of six classes is outlined

below:

```
1. Oxidoreductases
2. Transferases
3. Hydrolases
4. Lyases
5. Ismerases
6. Ligases
```

A typical enzyme is comprised of approximately 250-300 amino acids residues, however only a small subset of these are directly involved in the actual function of the enzymes and are discussed in the section below.

1.2.1 Enzyme Active Site Residues

Enzymes are made up of polypeptide chains consisting of amino acid building blocks. There are twenty (widely accepted) different amino acids, however not all of them contribute to the function of the enzyme. There are four generalisations that can be applied to the residues contained in an enzyme (Benner *et al.*, 1994)

```
1. Hydrophobic residues tend to lie within the folded structure
2. Hydrophilic residues tend to lie on the outside of structure
3. Conserved residues tend to lie inside or near to the active site
4. Variable residues tend to lie on the outside of the structure
```

Most residues contribute to maintaining the fold of the enzyme. By contrast, the active site, usually contains functionally important residues. There is considerable evolutionary pressure to conserve residues involved in the function of the enzyme.

Which types of amino acids are commonly involved in the function of any particular enzyme is not documented very generally in the literature. Bartlett *et al* defined four criteria for describing in which way a residue can be involved in the function of an enzyme (Bartlett *et al.*, 2002). The first criterion is fulfilled if a

residue has a direct involvement in the catalytic mechanism of the enzyme (e.g. histidine in the catalytic triad of serine proteases). The second criterion is fulfilled if the residue exerts its effect through another residue (satisfying criterion one) or water molecule. The third criterion describes residues stabilising a transition state intermediate of the reaction being catalysed. The fourth and final criterion is fulfilled if the residue has some involvement with the substrate or cofactor of the enzyme; this includes steric and electrostatic interactions. In the same study the occurrence of each of the twenty amino acids satisfying the four criteria in the active sites of the 178 enzymes in their data set were explored. The results showed definite preferences of certain amino acids over others:

H>C>D>R>E>Y>K>N>W>S>Q>T>G>F>M>L>P>I>A>V

In the development of this function prediction method a smaller subset of eleven residues (KRENDYCHQST) was chosen to represent the residues most likely to be involved in the function of the enzyme (Figure 1.1). The subset was chosen before the publication of the work of Bartlett *et al* and agrees well with their observations. Each of the eleven selected residues contains a potentially chemically reactive group (with oxygen and/or nitrogen atoms) capable of forming hydrogen bonds in the active site of the enzyme.

1.3 Project Aims

The main aim of the project was to develop an enzyme function prediction method and to investigate the feasibility of utilising the method as a foundation for a successful prediction method for the future.

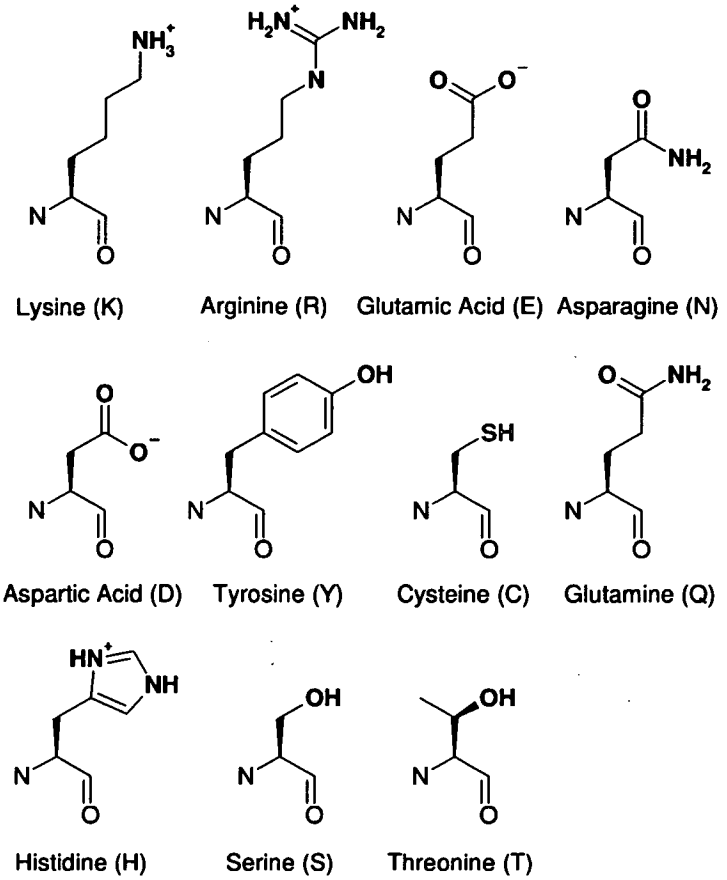


Figure 1.1: Functional Amino Acids. The structures of the eleven amino acids (KRENDYCHQST) utilised in the function prediction method. All eleven amino acids have oxygen and/or nitrogen atoms capable of forming hydrogen bonds with cofactors and/or substrates.

The method had been developed by exploring the potential of conserved positions in multiple sequence alignments for the prediction of cofactor binding and/or enzyme function. A key aspect of the design for the code to implement the method has been to allow straightforward updating and maintenance of the code and databases. While also ensuring the method is as fully automated as possible and easily expandable in the future.

Once developed to the aim was to test the method with regard to its ability to predict cofactor dependencies to-wards pyridoxal-5'-phosphate, thiamin (TPP), glutathione (GLU) and folic acid (FOL). A further aim of the project was to explore the application of DAROGAN for the prediction of previously described enzyme functions in organisms with completed genomes to which no gene and protein sequences could be assigned through the standard annotation processes. Investigations were therefore also made into the potential of utilising the DAROGAN method to propose candidates for the missing pyridoxal-5'-phosphate utilising enzymes in the *E. coli* genome according to EcoCyc (Karp *et al.*, 1999) (See Chapter 6)

1.4 Existing Function Prediction Techniques

There has been great interest in the enzyme function prediction field, most of which take the sequence of the protein into consideration, rely on the structure entirely for the prediction, use clustering of sequences or employ phylogenetic analyses to predict function (Figure 1.2). A total of 640 different enzymes have been structurally classified (number derived from the number of unique EC numbers in the PDB as of September 2000), out of a total of 3705 EC numbers in the ENZYME database (Erlandsen *et al.*, 2000).

1.4.1 Sequence to Function

Shah and Hunter conducted a systematic appraisal of using a protein sequence to predict enzyme function designated by EC class (Shah & Hunter, 1997).

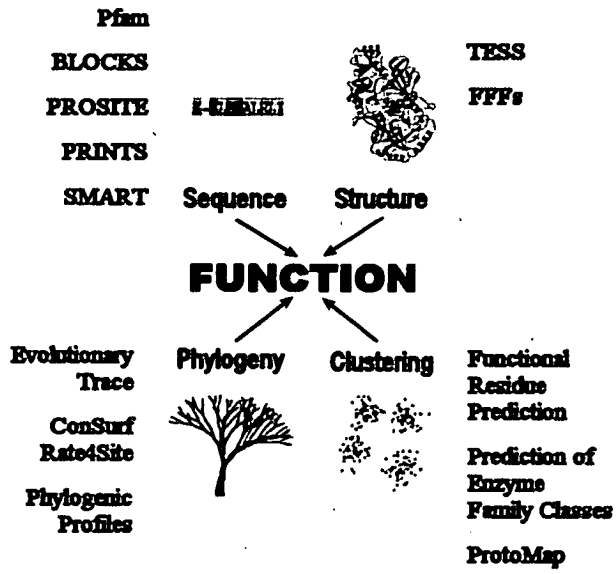


Figure 1.2: An illustration of the four main approaches to enzyme function prediction: from sequence, structure, phylogeny and clustering.

The appraisal used statistical analyses to determine how well EC class can be predicted from a protein sequence alone. The conclusion of the appraisal was that using sequence similarity is a *moderately* good method for predicting the functional class, but there were a high number of EC classes for which sequence similarity methods fail. The reason for this failure among certain EC classes is due to the under population of the classes with sequences, the lack of sequences means any sequence similarity is not likely to be statistically significant. This situation can only improve as more enzymes are assigned EC numbers through experimental classification. However not all members of an EC class share a high degree of sequence similarity. Analysis of the correlation between EC number and sequence identity (Wilson *et al.*, 2000) gave convincing evidence that the higher the sequence identity the more likely the sequences are to have the same biochemical function as denoted by the EC number. Single domain

proteins were shown to have three EC numbers conserved at a sequence identity of above 40% and that the variation of the fourth EC number is uncommon. Even at 30% sequence identity three EC numbers can be predicted with 95% accuracy (Wilson *et al.*, 2000). When multi-domain proteins were considered the conservation of the EC numbers is still evident, as a sequence identity of 30-40% has a 90% chance of three EC numbers being conserved (Todd *et al.*, 2001). An analysis of the relationship between sequence similarity using highly sensitive sequence comparison algorithms (Hidden Markov Models, BASIC and PSI-BLAST) and functional similarity was recently discussed (Pawlowski *et al.*, 2000). Well characterised proteins from the *E. coli* genome were used in the comparison and the functional categorisation used was by EC number. The main conclusion of the study was that even at low pairwise sequence identity (10-15%) the function of two proteins are more likely to be similar than by chance, with a pairwise alignment with a random sequence. In other words even very weak sequence identity between two proteins increases the chance of the proteins having the same function. However exceptions to this rule were found.

In the sections below, some of the more commonly available tools for enzyme function prediction from sequence are discussed. Some were conceived as enzyme function prediction tools, others have been found useful in prediction function. The aim is not to present an exhaustive list of tools, but rather provide an overview of some of the more popular and illustrative tools.

Pfam

Pfam (Bateman *et al.*, 1999; Pfam, 2005) is a database of multiple sequence alignments for protein domains created using profile Hidden Markov Models (HMM). Pfam-A is the main part of Pfam and contains a seed alignment, profile HMM, full alignment and an annotation for each of the included protein domain families. Pfam-B contains sequence segments not included in Pfam-A. The seed alignment is a hand curated alignment containing a set of sequences considered to be representative of the protein family. The seed alignment is used to create a profile HMM for the family, which is used to search a database for matching sequences. The full alignment contains all the sequences matching the profile HMM for the SwissProt and TrEMBL databases. To predict the function of a putative enzyme sequence it is possible to align the sequence the profile HMMs for each Pfam protein family. A high scoring match would indicate that the sequence is a member of the matching family and is likely to be functionally related to that family.

BLOCKS

BLOCKS (Henikoff & Henikoff, 1991; BLOCKS, 2005) is a system for identifying and assembling conserved regions for protein families to facilitate database searching. The BLOCKS themselves are short regions of multiple sequence alignments, created by extending ungapped aligned regions found using a local pairwise alignment algorithm. A set of related proteins may contain several BLOCKS, which are permitted to overlap, and a collection of BLOCKS is then used to define the family. A database of a collection of protein family defining BLOCKS can then be used to search for further sequences matching a protein family of interest.

As with Pfam, a putative enzyme sequence can be matched to the BLOCKS to indicate membership of a particular enzyme family giving an indication to the likely function of the sequence.

PROSITE

AA_TRANSFER_CLASS_1, PS00105, Aminotransferases class-I pyridoxal-phosphate attachment site (PATTERN)

Consensus pattern:	[GS] - [LIVMFYTAC] - [GSTA] - K - x(2) - [GSALVN] - [LIVMFA] - x - [GNAR] - x - R - [LIVMA] - [GA] *K is the pyridoxal-P attachment site*

Figure 1.3: An example of the output from a Prosite search on the sequence for aspartate aminotransferase (EC 2.6.1.1) (Swissprot ID:P00509; PDB ID:1ARS)

PROSITE (Hofmann *et al.*, 1999; PROSITE, 2005) is a database of biologically significant patterns and profiles within protein sequences arising from specific binding regions, catalytic residues and residues conserved for structural reasons. The patterns and profiles usually consist of a handful of ordered and contiguous residues, so represent a localised subsection of the sequence (See Figure 1.3 for an example). PROSITE patterns are regular expressions used to match the protein of interest. A profile is a table of position specific amino acid weights and gap costs, used to score an alignment with the protein of interest. Sequences can match any number of non-overlapping patterns and profiles. PROSITE can be used to predict the function of a putative enzyme sequence by matching PROSITE profiles and patterns to the sequence. Matches are then used to give insight into the likely function of the putative enzyme.

Ten top scoring fingerprints for your query. Detailed by motif									
FingerPrint Name	Motif Number	IdScore	PfScore	Pval	Sequence	Length	low	Pos	high
TRANSAMINASE	1 of 6	70.18	902	6.44e-12	VLFHGCCHNPTGIDPTLEQW	20	131	175	238
	2 of 6	65.70	511	1.57e-08	WLPLFDFAYQGFARG	15	162	206	269
	3 of 6	62.68	690	1.23e-12	ASSYSKNFGLYNERVGACTLV	21	198	241	305
	4 of 6	58.11	771	1.22e-15	SQMKAAIRANYSNPPAHGASVVATIL	26	230	273	337
	5 of 6	59.90	526	8.81e-10	IIKQNGMFSFSGLTKEQVL	19	298	341	406
	6 of 6	38.56	444	1.32e-08	VYAVASGRVNVAGMTPDNM	19	324	367	432

Figure 1.4: An example of the output from a PRINTS search on the sequence for aspartate aminotransferase (EC 2.6.1.1) (Swissprot ID:P00509; PDB ID:1ARS)

PRINTS

PRINTS (Attwood *et al.*, 1999; PRINTS, 2005) is a database of protein *fingerprints* characterising protein family signatures by using groups of aligned sequence motifs. Groups of motifs bypass the *match or no match* characteristics of the single motif matching methods. The matches are at the level of residues within single motifs and the number of residues between single motifs (See Figure 1.4 for an example PRINTS prediction).

SMART

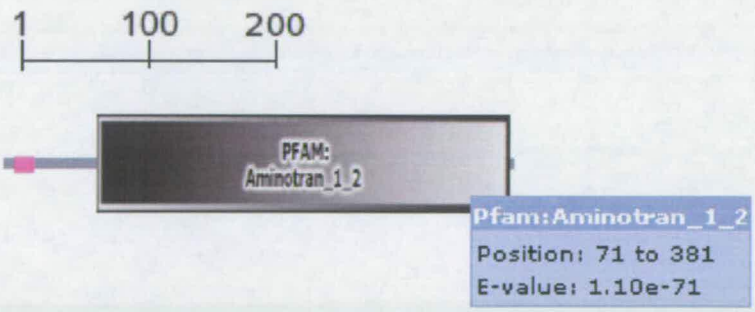


Figure 1.5: An example of the output from a SMART search on the sequence for aspartate aminotransferase (EC 2.6.1.1) (Swissprot ID:P00509; PDB ID:1ARS)

SMART (Simple Modular Architecture Research Tool) (Schultz *et al.*, 1998; SMART, 2005) is a collection of accurate alignments to aid in the annotation of protein domain sequences. The alignments are considered to be accurate as there is a minimisation of insertions/deletions and secondary structure is used to guide the alignments (where available). Daily updates are performed to ensure all new similar sequences are added to the alignments. Similar sequences are searched for using a combination of sequence searching techniques (pairwise alignments and profile HMM) to ensure no new similar sequences are missed. Before being considered for addition to the alignment, the sequence must pass a set of statistical significance measures designed to determine whether the sequence is a putative homologue or not. SMART is capable of annotating single protein sequences as well as large datasets and addresses to problem of annotating multi-domain proteins (See Figure 1.5 for an example SMART prediction).

1.4.2 Structure to Function

Studies of the correlation of the EC number and the common structural classes for proteins (α, β, α+β, α/β) revealed that there is little conservation of EC number. This is consistent with the view that enzyme function is determined by key residues in the active site (Martin *et al.*, 1998). Hegyi and Gerstein recently discussed the conservation of enzyme function and different protein folds. Few folds, α/β especially, were found to have a diverse range of functions. Glycosyl hydrolysis was found to be the most ubiquitous reaction as it was found to exist in seven different folds in all three of the fold classes (α, β, α+β, α/β) (Hegyi & Gerstein, 1999). The relationship between sequence identity to a PDB SITE

record and functional conservation of the active site was studied by Zhang et al. The SITE records in a PDB file contain information on the residues in the proteins active site(s). Ten percent of *E. coli* proteins with significant sequence identity to PDB entries with SITE records were studied, concluding that there is no conservation of SITE annotations and hence functional residues. However the inclusion of SITE records in PDB entries was found to be inconsistent (Zhang *et al.*, 1999).

There are several methods for predicting enzyme function from structure and will not be discussed at length here. Two of the more established methods are presented below; TESS and FFF.

TESS

Catalytic Site

Found by:
PsiBLAST alignment on 1ay4
(Structural analysis and templates exist for the 1ay4 family)
PsiBLAST alignment on 1b8g
PsiBLAST alignment on 1qgn

Residue Chain Number	Uniprot	Func
140	139	S
222	221	S
258	257	S

Figure 1.6: An example of the output from a PROCAT search on the sequence for aspartate aminotransferase (EC 2.6.1.1) (Swissprot ID:P00509; PDB ID:1ARS)

The TESS (TEmplate Search and Superposition)(Wallace *et al.*, 1997; PROCAT & TESS, 2005) algorithm is based on three dimensional templates, equivalent of PROSITE one dimensional templates. The three dimensional templates describe enzyme active sites and are used to search the PDB for structures containing a

match to the template. The TESS algorithm is based on geometric hashing, as utilised by many protein docking algorithms, each PDB structure is preprocessed into hash tables storing geometric information for the atoms in the structure. The preprocessing greatly speed up the algorithm, and it is the hash tables that are used to match against the query template. A database of three dimensional templates, defining enzyme active sites, accompanies the TESS algorithm. The TESS method provides a very effective method for finding structures that match the templates for enzyme active sites, giving insight into the functions of the matching proteins. However the requirement for a protein to have its three dimensional structure determined at high resolution limits the application of TESS in function prediction. As high throughput structure determination efforts come to fruition, TESS will have wider applications in function prediction. An example of the output from a PROCAT search is shown in Figure 1.6.

FFFs

Fuzzy Functional Forms (FFF)(Skolnick & Fetrow, 2000) are three dimensional motifs for enzyme active sites, similar to those used in TESS. The FFFs have the advantage of being *fuzzy*, meaning the motifs do not have to rigidly match the exact geometry in the structures being searched. The method therefore has a level of tolerance in the matching of the motifs. The disadvantage, as with TESS, is the requirement for a protein to have has its structure determined. However the fuzzy nature of the FFFs is more amenable to matching modelled structures as well as to experimentally determined structures. The FFFs are created by searching the literature to determine which of a particular enzymes residues are

directly involved in the function of the enzyme. The second stage is to find a set of functionally related enzyme structures, to create a consensus FFF of the residues determined to have functional roles. The method was found to be very successful in finding further functionally related proteins in the PDB (Fetrow *et al.*, 1998; Fetrow & Skolnick, 1998) and more recently FFFs have been applied to genome wide active site searches using predicted structures (Cammer *et al.*, 2003).

1.4.3 Sequence Clustering to Function

Functional Residue Prediction

Casari *et al* describe a method to predict functional residues in proteins through clustering the sequences in multidimensional space (Casari *et al.*, 1995). A multiple sequence alignment of similar sequences is taken as the sole input, where ideally sequences should contain fewer than 50% identical residues to any other sequence in the alignment. Each sequence in the alignment is represented as a vector occupying a single point in multidimensional space, the number of dimensions is defined by the number of positions in the alignment. Each dimension is therefore occupied by the type of residue present at that position in the alignment.

Principal component analysis techniques allow protein subfamilies to be defined as well as the ability to trace individual residues and position characteristics of the subfamilies. Both the direction and the magnitude of the vectors have biological meaning. The direction relates to specific patterns within the sequences, permitting discrimination between subfamilies. The length of the vectors represent the level of conservation of the residues in the sequence, the further away from the

geometric origin the more characteristic the sequence is for the sequence pattern (direction). By representing the sequences as vectors it is therefore possible to predict the functional residues of a sequence and whether they are completely conserved or subfamily specific. The disadvantage of this method lies with the quality of the input alignment, as a poor quality alignment will lead to poor prediction of the functional residues.

Prediction of Enzyme Family Classes

To predict the enzyme family a particular enzyme belongs to, first the enzyme family must be defined (Chou & Elrod, 2003). By taking all the sequences in a particular enzyme class and representing them in vector format, where dimensions are the amino acid compositions of the sequences, the vectors from each sequence in the class can be combined to produce a single vector. A query enzyme can also be represented as a vector with the same twenty dimensional space as the enzyme class vectors. A covariant discrimination function then measures the difference between the query vector and each of the enzyme class vectors. The smaller the difference the higher the probability the query enzyme belongs to a particular enzyme class. The method is somewhat dependent on the sequences used to define each of the enzyme classes. A poor selection of sequences would have a detrimental effect on the prediction of enzyme class. Despite this limitation the method is successful in producing rapid function predictions for query enzyme sequences. In the study 2640 oxidoreductases classified into 16 subcategories by substrate specificity, were used to assess the success of the method. A jack-knife test and a self-consistency test showed successful assignments of the sequences to

the 16 classes with 75% and 63% respectively.

ProtoMap

ProtoMap (Yona *et al.*, 1999; ProtoMap, 2005) provides a method for classifying large sets of protein sequences by clustering them in protein space. The first stage of the method is to perform all-against-all pairwise sequence alignment comparisons on the set of proteins being studied. The pairwise alignments are performed using the Smith-Waterman algorithm, BLAST and FASTA, the results of which are manipulated to a common scale. A directed graph represents the protein space, where the vertices are the protein sequences and the edges represent the dissimilarity between sequences. A recursive clustering algorithm is then applied to the graph, and repeated at varying levels of statistical significance to form the final clusters. The result is a hierarchical classification of the protein sequences, correlating well with the biological functions of the proteins. Comparisons to PROSITE and Pfam protein family classifications had 64.8% and 88.5% agrement respectively.

1.4.4 Phylogeny to Function

Enzyme functions are thought to evolve through several distinct mechanisms: Gene recruitment, Gene Duplication, Incremental Mutations, Gene Fusion, Oligomerisation, Post-translational Modification or combinations of the above (Todd *et al.*, 1999). The creation of the COG (Clusters of Orthologous Groups) database (Tatusov *et al.*, 2001; COG, 2005) has been used to predict the function of protein sequences using inferred evolutionary relationships. The COG

database contains homologous proteins from all of the sequenced genomes, each of the groups is assumed to have evolved from a common ancestral protein. Members of a COG are likely to perform the same functions in their respective organisms so if a protein is found to be similar to several members of a COG then its function is very likely to be the same as for the COG. The COG database allows the identification of probable orthologous proteins sometimes in spite of different evolutionary rates acting upon individual genes. This allows for increased sensitivity in the search for similar sequences. It is worth noting that the COG concept can only be applied to complete genome sequences.

Evolutionary Trace Method

The Evolutionary Trace method was developed by Olivier Lichtarge (Lichtarge *et al.*, 1996; Madabushi *et al.*, 2002; Lichtarge & Sowa, 2002) and was initially conceived with the purpose of identifying areas or patches on the surfaces of proteins involved in binding other proteins. The method is based around the two assumptions; the first is that residues in a protein involved in the function of the protein are less likely to undergo mutations during evolution than residues not participating in the proteins function. A second assumption is that functionally related proteins from a common ancestral protein will have maintained the positioning of the functional residues within the three dimensional structure of the protein.

To generate an evolutionary trace for a protein structure of interest, first a set of similar proteins must be found and aligned using a multiple sequence alignment

algorithm, which also calculates the required dendrogram. The dendrogram is then *partitioned* by taking the minimum percentage sequence identity for branches in the tree. A partition identity cut off (PIC) value defines which branches belong to the partitions or clusters. Generally the higher the PIC value the more clusters are formed. For each of the clusters of sequences, at a particular PIC, a consensus sequences is constructed. Only invariant residues in the cluster are put forward into the consensus sequence with variable positions left blank. Each of the consensus sequences for the clusters are then aligned together for the evolutionary trace. If residues are invariant this represents the residue being conserved across the entire set of sequences. If however there are different invariant residues for each of the consensus sequences then these represent cluster specific residues which are labelled as such in the trace. If any position in the cluster consensus has a blank then this is passed onto the final trace. Once the trace has been calculated the residues can then be mapped onto the structure of interest, coloured according to whether they are invariant, class specific or blank. By creating traces for a range of PIC values the traces can be mapped on to the structure and compared to give an indication of the degree of conservation of the residues across the evolution of the protein family.

The evolutionary trace method provides an effective way of mapping potentially important functional residues onto the surfaces of proteins, making the method particularly good at identifying surface patches involved in the binding of other proteins. Its use for the identification of functional residues involved in the catalytic reaction of the protein has not been explored. The main disadvantage of

the method is that there is a requirement for the three dimensional structure of the protein, although modelled structures could be used. The method also relies on the availability of a set of closely related proteins from a common ancestor, which is not guaranteed.

ConSurf and Rate4Site

The ConSurf (Armon *et al.*, 2001; ConSurf & Rate4Site, 2005) algorithm and it sister program Rate4Site take the evolutionary trace method a step further by taking into account the physiochemical properties of the functionally important residues. A phylogenetic tree is created, allowing amino acid changes in the branches of the tree to be tracked and weighted with an amino acid similarity matrix of physical and chemical properties.

The main difference from the evolutionary trace method is the usage of tree building algorithms that do not assume equal rates of evolution throughout the tree. Further to this, consensus sequences are not based on the three states of invariant, cluster specific and blank. Instead averaged conservation, using a similarity matrix, is used to calculate a consensus sequence. This averaging allows normalisation to be performed for the number of sequences in the branches, reducing the effect of bias toward highly similar sequences.

Despite taking the consensus sequence concept further, ConSurf and Rate4Site still suffer the same limitations of the requirement for a three dimensional structure of the protein of interest.

Phylogenic Profiles

By assuming that enzymes within the same biochemical pathway are put under similar evolutionary pressure, it is possible to assign phylogenic profiles to proteins based on their existence in other genomes. The enzymes in a particular pathway are likely to be either preserved together or eliminated all together as a species evolves. So by creating phylogenic profiles it is sometimes possible to predict the function of putative enzymes (Pellegrini *et al.*, 1999)

A profile for a protein is stored as a string with each character representing whether the protein is present in a particular genome or not. The number of characters is dictated by the number of genomes included in the study. Proteins are then clustered according to their phylogenic profiles, where closely clustered proteins implies some functional relatedness. At the time of the study only 16 genomes were utilised, so had limitations of coverage. The inclusion of larger numbers of genomes will increase the validity of the functional assignments utilising phylogenic profiles.

1.5 Specific Function Prediction Example

Individual examples where enzyme function has been successfully predicted provide an excellent starting point for genome-wide enzyme function prediction methods. The heat shock protein, Hsp90, structure was submitted to the CASP2 (Critical Assessment of Techniques for Protein Structure Prediction) competition for blind structure and function predictions. In fact the structure had not been

completely solved at the time of it submission, but was solved and released publicly soon after (Prodromou *et al.*, 1997).

Once the Hsp90 was submitted to the CASP2 competition it was now the turn of the modellers to predict the structure and function of the anonymous Hsp90 protein. The Hsp90 target was found to have similarity to DNA gyrase, and conserved positions in multiple sequence alignments pointed to the presence of a Mg^{2+} binding site and therefore possible ATP binding. However published experimental evidence contradicted the hypothesis that Hsp90 was able to bind ATP and exhibit ATPase activity. The predicted ATP binding site was also structurally unusual, however it had been seen before in DNA gyrase giving support to authenticity of the predicted ATP binding site (Gerloff *et al.*, 1997). At the time of the prediction of the ATP binding, the binding site in the solved structure was being misinterpreted as a substrate binding site. However personal communication from D. Gerloff pointed to the likelihood that the binding a site was in fact for ATP. With this information and further investigation the co-crystallised Hsp90 structure with bound ATP was solved, supporting the prediction (Obermann *et al.*, 1998).

The successful prediction of the structure and ATP binding site of Hsp90 illustrates the potential of using conserved functional residues in a function prediction context. In this case the functional residues were utilised to back up the hypothesis that there was indeed a Mg^{2+} and ATP binding site, despite experimental evidence to the contrary. The experimental evidence against ATPase activity

turned out to be incorrect as the ATPase activity was undetectable due to inhibition. For the successful structure and function prediction by Gerloff *et al* (1997) a comparison was also made between their manual and automated (yet transparent) prediction methods. The conclusion was that there is little difference between the methods as virtually the same model was predicted. The ability of an automated method to recreate the results of a manual method indicates that there is a role for computational methods in the prediction of the function of proteins where the function is unknown.

1.6 Background to Enzymes Studied

Enzymes can utilise a wide range of cofactors (Table 1.1), providing a convenient method of defining a set of enzymes. Conserved residues have been implicated in binding certain substrates and cofactors, and are of particular interest in the case of cofactors and their chemistry. It was therefore useful to select a training set of enzymes by the cofactor they utilise, as there are multiple examples of enzymes utilising a particular cofactor. The selection of a specific cofactor and the conservation of the residues binding the cofactor or participating in the catalytic mechanism will allow a function prediction method to exploit these residues. Potentially predicting the utilisation of a particular cofactor by a putative enzyme with unknown function.

To develop the function prediction method an enzyme family with examples of *similar folds - different functions* and *different folds - similar functions* was

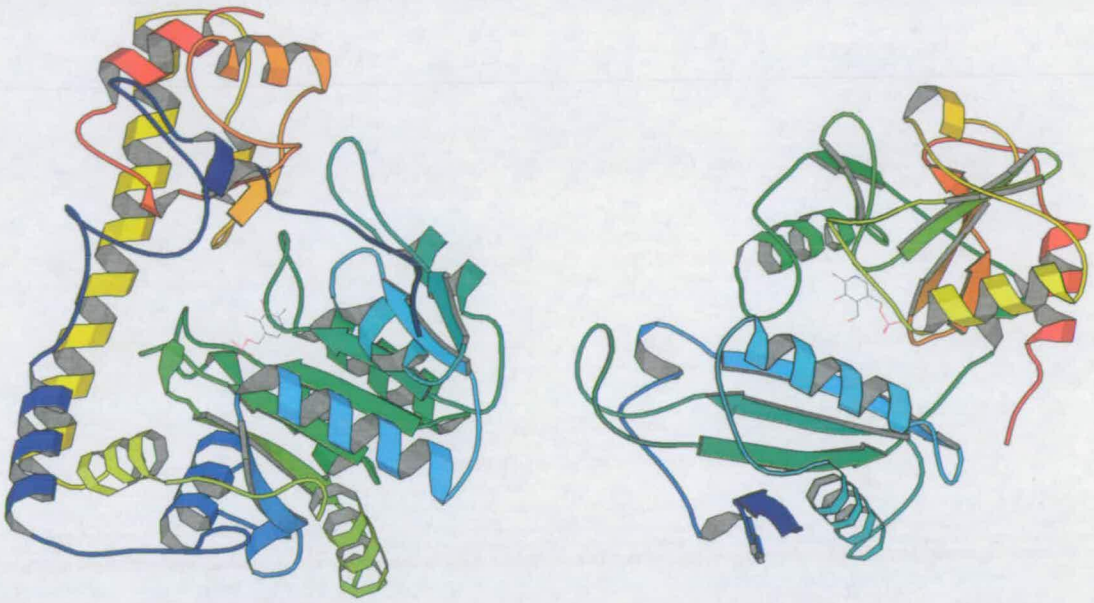


Figure 1.7: PLP: Same Cofactor, Different Structures (left:1ARS, right 1DAA)

chosen to decouple fold from function as much as possible. A training set of enzymes should also have many well characterised examples, having had their three dimensional structure determined to aid method development. The pyridoxal-5'-phosphate (PLP) utilising enzymes offer all of the above criteria and were chosen to develop the prediction method. See Figure 1.7 for an example of two different folds for PLP utilising enzymes using a similar mechanism of action. Thiamin utilising enzymes were also selected as they also bind the cofactor in the active site of the enzyme. The thiamin utilising enzymes would therefore offer a set of similar enzymes to discriminate from the PLP utilising enzymes (in addition to the glutathione and folic acid utilising enzymes).

A brief introduction to PLP and thiamin utilising enzymes is provided in the section below, however there are more detailed reviews available in the literature (John, 1995; Kirsch *et al.*, 1984; Tai & Cook, 2001; Peisach, 1998; Kern *et al.*,

Cofactor	Search Term	No. in PDB (3Å cut-off)
Vitamin B1 (Thiamin, TPP)	thiamin	29
Vitamin B2 (Riboflavin, FAD, FMD)	flavin	349
Niacin (Nicotinic Acid, NAD, NADP)	nicotinamide	480
Lipoic Acid (lipoamide)	lipoic	2
Vitamin B5 (Pantothenic Acid, Coenzyme A)	coa	6
Vitamin B6 (Pyridoxine, PLP)	pyridoxal	255
Biotin	biotin	3
Vitamin B12 (Cobalamin)	cobalamin	16
Folic Acid (THF)	folic	59
Vitamin C (Ascorbic Acid)	ascorbic	3
Vitamin K	vitamin K	1
Adenosine (ATP)	adenosine	520
Guanosine (GTP)	guanosine	149
Uridine (UTP)	uridine	176
Glutathione	glutathione	60
S-Adenosylmethionine (SAM)	adenosylmethionine	20
Cytidine (CTP)	CTP	15
Coenzyme Q (Ubiquinone)	ubiquinone	2
Metal Cofactor	**Search Term**	**No. in PDB (3Å cut-off)**
Calcium	calcium	1194
Cobalt	cobalt	64
Copper	copper	145
Iron	iron	104
Magnesium	magnesium	738
Manganese	manganese	274
Potassium	potassium	179
Sodium	sodium	453
Zinc	zinc	985

Table 1.1: Commonly occuring cofactors, including metals cofactors. The approximate numbers of structures in the PDB determined for proteins utilising each of the cofactors are shown. The search term used to search the PDB to yield the numbers of structures in the database is also shown. Resolution was restricted to <3Å.

1997).

1.6.1 PLP Utilising Enzymes

Pyridoxal-5'-phosphate (PLP) is derived from vitamin B_6 and is utilised by a large number of enzymes to aid catalysis. In solution PLP will react with a substrate to produce several different products, however when bound to an enzyme PLP will react with a substrate to produce just one of these products. By restricting PLP reactions to just one product, the enzyme increases efficacy over the non enzymatic PLP reaction.

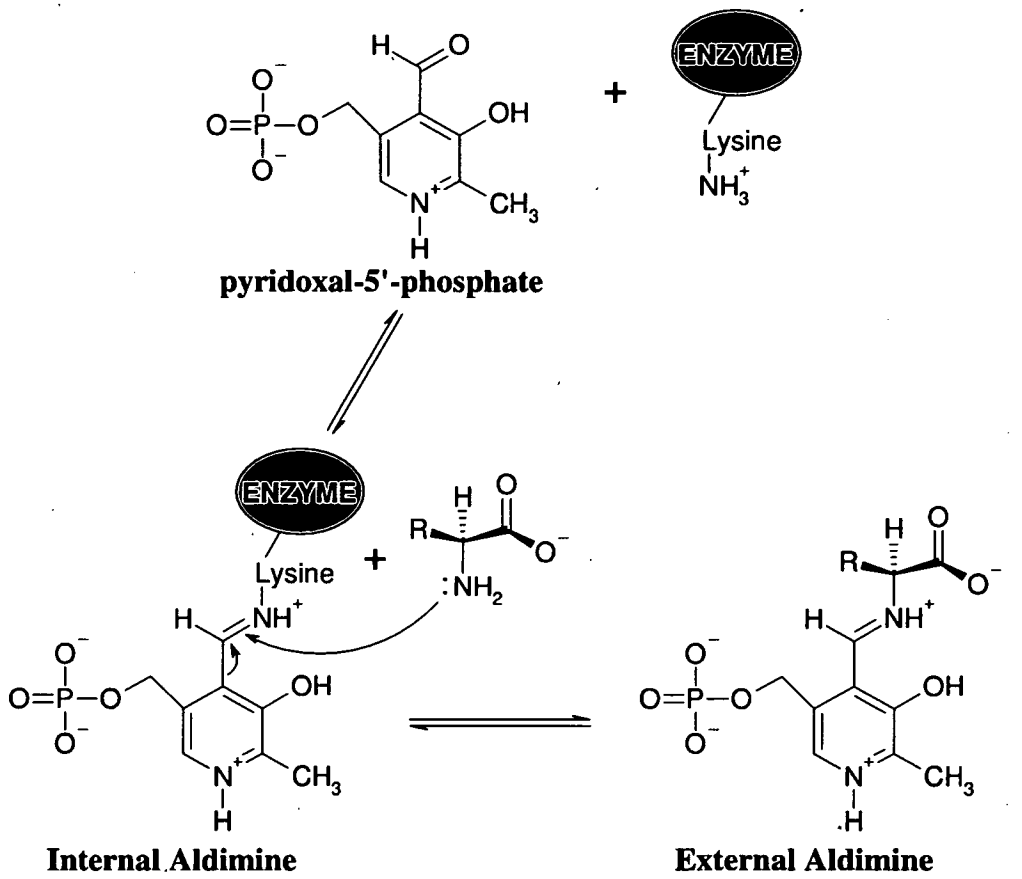


Figure 1.8: The pyridoxal-5'-phosphate (PLP) molecule is covalently bound to a lysine residue in the active site of the enzyme to form an internal aldimine. The displacement of the PLP from the lysine, by an amino acid substrate form the external aldimine from where the catalytic reaction can proceed.

In all PLP utilising enzymes the PLP molecule is covalently bound to a lysine (Lys) residue (Schiff base) in the active site of the enzyme, referred to as the internal aldimine. The reaction proceeds to form the external aldimine, where the PLP is displaced from its lysine tether by an amino acid substrate (Figure 1.8). The pyridine ring in the PLP molecule acts as an *electron sink*, stabilising a negative charge.

In the 1970's it was thought that PLP utilising enzymes had all evolved from a common ancestor (Dunathan & Voet, 1974), however with the sequencing of more PLP utilising enzymes in the intervening years it has become clear that this is not the case. The theory was not entirely incorrect as the majority of PLP utilising enzymes do appear to have evolved from a common ancestor, there are however examples of PLP utilising enzymes that have not evolved from this common ancestor. This minority of enzymes represent evidence of convergent evolution, where the usage of PLP has evolved independently. There is very low sequence identity between the enzymes evolved from a common ancestor and those from different ancestors. The folds of the enzymes are also completely different. For this reason PLP utilising enzymes represent an excellent development set for the function prediction method as there will be common residues involved in the function of the enzymes, but little or no similarity in sequence or fold, allowing a fold independent function prediction method to be developed.

Reaction	**Example**	**EC Number**
transamination	D-amino acid transferase	2.6.1.x
decarboxylation	dialkylglycine decarboxylase	4.1.1.x
racemazation	alanine racemase	5.1.1.x
aldol cleavage	serine hydroxymethyltransferase	2.1.2.x
β elimination	tyrosine phenol lyase	4.1.99.x
γ elimination	γ cystathionase	4.2.99.x

Table 1.2: The six classes of PLP utilising enzymes along with an example assigned EC number. Adapted from Peisach, 1998.

The different PLP utilising enzymes differ in the arrangement of residues in or around the active site, resulting in different substrate specificity and reaction

types (Table 1.2). The usage of PLP as a cofactor has permitted enzymes to catalyse a wide range of reactions.

1.6.2 Thiamin Utilising Enzymes

Thiamin pyrophosphate (TPP or ThDP) is the active form of thiamin (vitamin B_1), which is converted to TPP by binding a glutamic acid residue in the active site of the enzyme and with the addition of two phosphate groups. The TPP is then able to transform to the active form or *ylid* (Figure 1.9). The thiazolium ring of TPP is the most important part of the molecule for catalysis, acting as an *electron sink* stabilising negative charges. The pyrophosphate segment of the molecule functions to allow the TPP to bind in the active site of the enzyme keeping hold of the TPP for further reactions. Keeping the TPP in the enzyme is important as thiamin is not stored in significant amounts in vertebrates. In this respect there is a similarity to PLP utilising enzymes, distinguishing between PLP and TPP utilising enzymes was the first stage in developing the function prediction method. Thiamin utilising enzymes catalyse a wide range of reactions including decarboxylation (e.g. pyruvate decarboxylase, E.C. 4.1.1.1) and transketolation (e.g. transketolase, E.C. 2.2.1.1).

1.6.3 Glutathione Utilising Enzymes

Glutathione (GLU) is a tripeptide (γ-glutamylcysteinylglycine) containing the atypical γ amide bond (See Figure 1.10). Glutathione is utilised as a cofactor/co-enzyme in either its reduced (GSH) or oxidised (GSSH) form. As part of the reaction glutathione is converted to the alternate oxidation state (GSH to GSSH

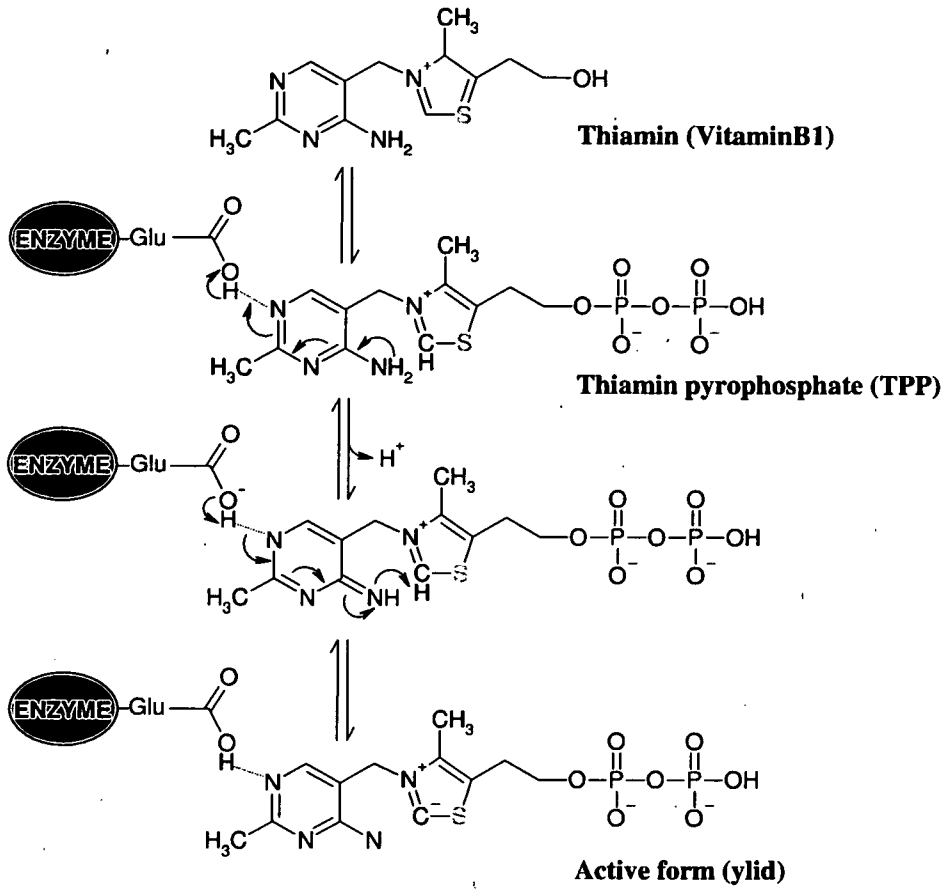


Figure 1.9: The steps involved in the processing of the water soluble thiamin molecule into the active form of thiamin, the *ylid*, present in the active site of the enzyme. The addition of the phosphate groups ensures the *ylid* can not escape the cell due to the negative charges, important as thiamin is not stored in significant levels in vertebrates.

or GSSH to GSH) relying on an additional enzyme (e.g. glutathione peroxidase) to catalyse the glutathione back to its original state. In humans glutathione deficiency has been linked to diabetes and reduced resistance to HIV.

Examples of glutathione utilising enzymes include prostaglandin-D synthase (E.C. 5.3.99.2), prostaglandin-E synthase (E.C. 5.3.99.3) and glyceryl-ether monooxygenase (E.C. 1.14.16.5). In prostaglandin-D synthase the glutathione

covalently binds a tyrosine residue during catalysis (Kanaoka *et al.*, 1997) and is postulated to do the same in prostaglandin-E (Yamada *et al.*, 2005).

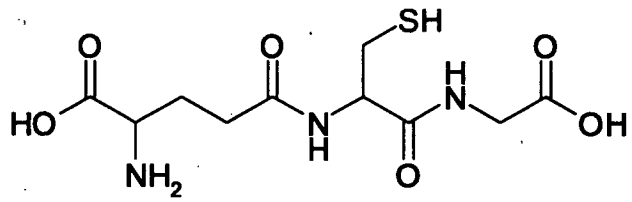


Figure 1.10: Glutathione

1.6.4 Folic Acid Utilising Enzymes

Folic acid (FOL) is the precursor to tetrahydrofolate (THF) and is utilised by many enzymes as a substrate (e.g. dihydrofolate reductase, E.C. 1.5.1.3), and less commonly as a cofactor (See Figure 1.11 for the structure). Folic acid is enzymatically reduced, first to dihydrofolate (DHF) before THF. Mammals are unable to synthesis folic acid so must be provided though their diet or intestinal microorganisms. In humans a deficiency can lead to megaloblastic anaemia.

THF as a cofactor is involved in the metabolism of C_1 units, most commonly in carboxylation reactions. Carboxylation is predominantly achieved through the use of biotin as a cofactor, however THF is able to act in several oxidation states (methanol, formate or formaldehyde). Enzymatic redox reactions allow the transition between the different oxidation states. Examples of THF being utilised as a cofactor include serine hydroxymethyl transferase (E.C. 2.1.2.1), glycine synthase (E.C. 2.1.2.10) and glutamate formino transferase (E.C. 2.1.2.5).

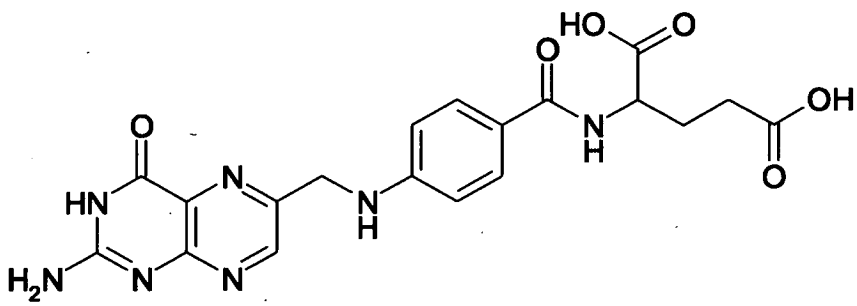


Figure 1.11: Folic Acid

Chapter 2

Fundamental Techniques and Resources

2.1 Overview

This chapter gives a brief guide to the diverse range of bioinformatics techniques and resources employed during this project. Each of the techniques is introduced and put into context by also discussing related techniques. For instance the BLAST algorithm is introduced and put into context by discussing pairwise sequence alignment algorithms and the related sequence search algorithm FASTA.

The chapter begins by introducing pairwise sequence alignment algorithms, followed by a discussion on the statistical significance of pairwise alignments. Next multiple sequence alignments algorithms are discussed, in particular introducing the different approaches employed by the Clustal, T-Coffee and AMAS alignment programs. The next topic discussed is the calculation of

conservation scores for multiple sequence alignments, several different methods are discussed providing background to the conservation score calculation in Chapter 3.2.4. Profile hidden Markov models (HMM) are explored to give background detail for the comparisons with the DAROGAN function prediction method in Chapter 5. The PISCES sequence culling server is discussed to give background to the method employed to cull sets of sequences to remove highly similar sequences. The enzymes included in the Tread database were derived from a set of sequences submitted to the PISCES server to obtain culled lists.

Finally the concept of minimising functions is outlined, specifically discussing the Downhill Simplex method in multi-dimensions. The purpose of this section is to provide some background to the minimisations performed during the calculation of the custom conservation score in Chapter 3.2.4.

2.2 Pairwise Sequence Alignment

Pairwise sequence alignment refers to the process of aligning two protein or nucleotide sequences. The purpose of aligning two sequences is to determine if the two sequences have diverged from a common ancestor through mutations (substitutions, insertions and deletions); insertion and deletion of residues to or from a sequence result in gaps in an alignment.

In a protein alignment the aim is to match up as many identical or similar amino acids together by the placement of non-identical amino acids and gaps. In the

example below a region from a pairwise alignment of two dihydrofolate reductase (DHFR) sequences (*Homo sapiens* and *Plasmodium falciparum*), clearly shows the effect of mutations, between the two sequences, on the alignment. The differences in the sequences from human and *Plasmodium* allow antimalarial drugs (e.g pyrimethamine) to selectively target the Plasmodium DHFR affecting folate metabolism while leaving human DHFR unaffected.

The central row shows invariant residues displayed by their single letter code, similar residues are denoted by a plus sign, and a space shows residues aligned with no similarity. Gaps in the sequences are represented by a dash.

```
P00374      DMVWIVGGSSVYKEAMNHPGHLKLFVTRIMQDFESDTFFPEIDL
               +I+GGS VY+E +      K++ TRI   +E D FFPEI+
Q9GN27      --CFIIGGSVVYQEFLEKKLIKKIYFTRINSTYECDVFFPEINE
```

To perform a pairwise alignment first a scoring scheme must be defined to give a measure of sequence similarity. It is trivial to assign a simple binary score for sequence similarity (0 for non-identical and 1 for identical residues). However this level of scoring is inadequate. A more complex, and more biologically meaningful, method of scoring sequence similarity makes use of substitution matrices. The most commonly used are the Dayhoff and BLOSUM matrices.

The Dayhoff matrices (Dayhoff *et al.*, 1978) make use of Point Accepted Mutation (PAM) distances as a measure of sequence similarity or more accurately sequence

dissimilarity. Two sequences with 99% identical residues are considered to be evolutionary 1 PAM apart. Further PAM matrices can be derived by multiplying the PAM1 matrix against itself N times. A PAM250 matrix corresponds to approx. 20% sequence identity. The scores in Dayhoff matrices are log-odds scores, representing the ratio of the probabilities of an amino acid substitution occuring during evolution to the amino acid substitution occuring by chance alone. This odds score is then converted to give a log-odds score. This last step is to make the calculation of sequence similarity computationally less expensive, it is more straightforward to add up log-odds scores rather than multiply odds scores.

The BLOSUM (Henikoff & Henikoff, 1994) matrices (BLOcks SUbstitution Matrix) are derived from the BLOCKS (Henikoff & Henikoff, 1991) protein sequence alignment database. The BLOSUM substitution matrices were developed by Henikoff and Henikoff to replace the Dayhoff matrices, to score more distantly related sequences and make use of the wealth of sequence alignment data produced since the Dayhoff matrices were created. The ratio of observed pairs of amino acids at any position to the expected number of pairs from overall amino acid frequencies from alignments not containing gaps in the BLOCKS database. Thresholds of sequence identity are used to filter sequences in the alignments to ensure closely related sequences do not influence the scores too heavily. A sequence identity threshold of 62% is used in the BLOSUM62 substitution matrix. As with the Dayhoff matrices the odds scores are converted to logarithms for convenience.

2.2.1 Global: Needleman-Wunsch Algorithm

The Needleman-Wunsch Algorithm (Needleman & Wunsch, 1970) is a global pairwise sequence alignment algorithm that performs an alignment over their entire lengths. Therefore this algorithm is most suited to sequences which are similar over all or most of the lengths of the sequences.

The basic concept of the Needleman-Wunsch algorithm is to perform alignments of smaller sub-sequences, which are built up to form an optimum alignment. In more detail the first stage is to construct a matrix (F), where the values correspond to the score for the best alignment of segments from sequences i and j. $F(i,j)$ is then build up recursively from segments of x up to x_i against y Up to y_i. There are three ways to determine the best score $F(i,j)$ for a segment alignment, where the highest value of the options is taken as the best score (Equation 2.1). This process is repeated to completely fill up the matrix.

$$F(i,j) = max \begin{cases} F(i-1,j-1) + s(X_1, Y_i), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases} \tag{2.1}$$

An important step in the process is the recording of pointers to the cell from which $F(i,j)$ was calculated (Figure 2.1) and why the algorithm is an example of dynamic programming.

There are two special cases involved in the filling of F. These are known as

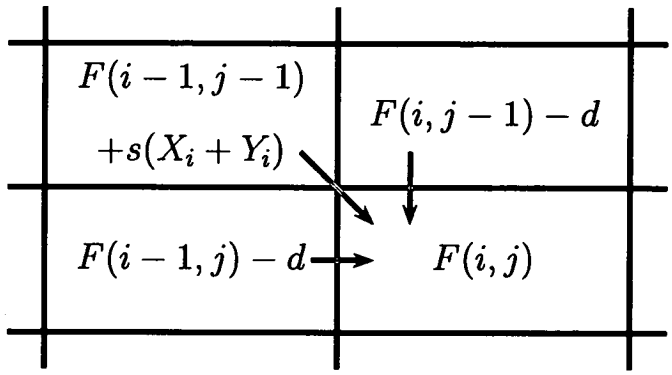


Figure 2.1: The bottom right cell of each set of four cells is filled with the highest value from three surrounding cells (left, above left and above). A pointer is kept recording the cell whose value was used to fill the bottom right cell.

bounding conditions, corresponding to the filling of the top row and the left column. In the top row where j always equals 1, this means $F(i-1,j-1)$ and $F(i,j-1)$ pose a problem. The alignments in these cases, $F(i,0)$, correspond to all gaps in y having a prefix of x. So this is treated as $F(i,0) = -id$. This method is also used in the left most column where $F(0,j) = -jd$.

Once the matrix has been completely filled, the $F(i,j)$ (bottom-right) cell will hold the score for the optimum alignment. Only one optimum alignment is determined by this method, so if two score derivations are identical then an arbitrary choice is made. Once this score has been determined the algorithm then works back through the matrix using the pointers recorded during the matrix filling stage. This working back through the matrix is known as a *trace back* and builds up the alignment in reverse. The trace back will give just one optimum alignment.

2.2.2 Local: Smith-Waterman Algorithm

The Needleman-Wunsch Algorithm (Needleman & Wunsch, 1970) was designed for sequences with similarity across their entire lengths, however it is often the case where there sequences only show similarity in regions of the sequences. This occurs in more distantly related sequences. To address this problem Smith and Waterman (Smith & Waterman, 1981) developed a local pairwise sequence alignments based on the Needleman-Wunsch algorithm. In fact there are only two significant differences between the two algorithms. The first difference is in the determination of the score in $F(i,j)$. The Smith-Waterman algorithm allows new alignments to be started, if the best alignment up to that point has a negative score. This starting of a new alignment allows the algorithm to ditch a previous alignment rather than continue extending an alignment unlikely to be the optimum alignment. The decision to start a new alignment is made by a modification to Equation 2.1, where an extra choice is added to give Equation 2.2.

$$F(i,j) = max \begin{cases} 0, \\ F(i-1,j-1) + s(X_1, Y_i), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases} \tag{2.2}$$

The choice of the 0 option is taken if all other values are less than 0, which has the effect of starting a new alignment. As a consequence the bounding conditions are different in the top row and left column. These are filled with a 0 rather than $-id$ and $-jd$ as in the global alignment algorithm.

The second difference to the Needleman-Wunsch algorithm is that the optimum alignment score is not necessarily in the bottom-right corner of the matrix. In fact the optimum alignment score can be anywhere in the matrix, so the algorithm searches for the highest score in the matrix to locate the optimum alignment score. The trace back through the matrix to find the optimum alignment is started from this point and works exactly as for the global alignment algorithm.

2.2.3 FASTA

The FASTA (Pearson & Lipman, 1988) algorithm uses a multi-step process of finding high scoring local alignments from a database. First short subsequence matches are found in the database, then extended via ungapped alignments to find maximal scoring alignments. This last step is then refined to find high scoring gapped alignments.

In more detail, a table is created for the query sequence, and each sequence from the database, to identify matching subsequences. The length of the subsequences is set with the parameter *ktup* which has default values of 1 or 2 for proteins and 4 or 6 for nucleic acid sequences. The next step is to find diagonals in the table that contain many subsequence matches. The best diagonals are then extended to find maximal scoring ungapped alignments. At this stage subsequences can be joined if they match. The maximal scoring alignments are then extended by gapped regions, with a gap penalty, to see if the matched maximal scoring alignments can be extended. The last step in the process is to align the highest

scoring matched with a full local dynamic programming algorithm (Needleman & Wunsch, 1970; Smith & Waterman, 1981). This final step is limited to the matching region found in the previous step, greatly reducing the search space for the optimum alignment.

The choice of the value for the *ktup* parameter determines the balance between speed and sensitivity of the algorithm. The higher the value of *ktup* the faster the algorithm, but the chance of missing significant matches increases. If *ktup* is set to 1 then FASTA gives the similar sensitivity to the full dynamic programming algorithms.

2.2.4 BLAST

BLAST (Basic Local Alignment Search Tools) (Altschul *et al.*, 1990) as the name suggests is actually a suite of programs rather than a single algorithm. BLAST performs local alignments of a query sequence against a database of sequences, so is analogous to the Smith-Waterman algorithm (Section 2.2). The first step in the BLAST method is to find short subsequences in the database that match with a high score to the query sequence. These subsequences are then used as *seeds*, investigated further by extending them to create longer alignments. The object of finding matching short subsequences is in the likelihood that true database matches will contain these short subsequences. To speed up the process of defining short subsequences and carrying out the comparisons to all other sequences in the database, the query sequence is preprocessed. A matrix is created of all possible subsequences from the query

sequence and their starting positions from within the original sequence. The default length of these short subsequence is 3 for proteins and 11 for nucleic acids.

Once the preprocessing of the query sequence has been carried out the next stage is to search the sequence database for these short subsequences. If the matches score above a threshold (2 bits per residue default) then a *hit extension* is started. The matching subsequence is extended in both directions as an ungapped alignment until the maximum scoring extension is found.

2.3 Statistical Significance of Pairwise Alignments

The two most widely used and most straightforward methods for expressing similarity between two sequences are Sequence Identity (Equation 2.3) and Sequence Similarity.

$$\% \text{ Sequence Identity} = \frac{\text{No. Identical Residues}}{\text{No. Residues in Lead Sequence}} X100\% \tag{2.3}$$

Sequence Similarity is calculated by summing up the scores from a substitution matrix for aligned pairs of residues. This sum is then divided by the lower of the two scores of when each of the sequences is alignment against itself. These two measures of sequence similarity are very useful, but what is really useful is the statistical significance of the alignments (i.e. is the alignment biologically

meaningful or are the sequences unrelated?).

2.3.1 Pairwise Alignment Score Significance

The statistical significance of a similarity score between a query sequence and a putative relative is usually estimated from a distribution of scores. A distribution of scores is obtained by aligning a large set of random unrelated sequences. The distributions are therefore individual to the particular alignment algorithm being used. The distribution of the scores is well approximated by a normal distribution, as it is basically the sum of a series of independent observations (i.e. matches to random sequences). It should be noted at this point that in contrast to local ungapped alignments, the score distribution for global alignments is not well understood. The distributions can be approximated by shuffling the sequences being compared to derive a score distribution.

The simplest method of calculating a significance score from a score distribution is to use a Z score (Equation 2.4). A Z score for an alignment being evaluated (Z_i) is basically a measure of how many standard deviations (σ) the alignment score (x_i) is from the mean score ($\bar{x}$).

$$Z_i = \frac{x_i - \bar{x}}{\sigma} \tag{2.4}$$

Because the Z scores are calculated from the same probability distribution as for other significance scores they are related to those significance scores (e.g. Z score

can be mapped to the P-value). The Z score is considered to be quite simplistic and has some significant drawbacks (not discussed here), so a more reliable method is usually implemented. The most common method is the Extreme Value Distribution or EVD.

The distribution of scores for a set of random sequences is well approximated by the tail of the distribution termed the extreme value distribution, as the tail decays more slowly than for a typical normal distribution. The tail of the distribution has a linear relationship between the score and the log of the frequency of the score being observed (Figure 2.2).

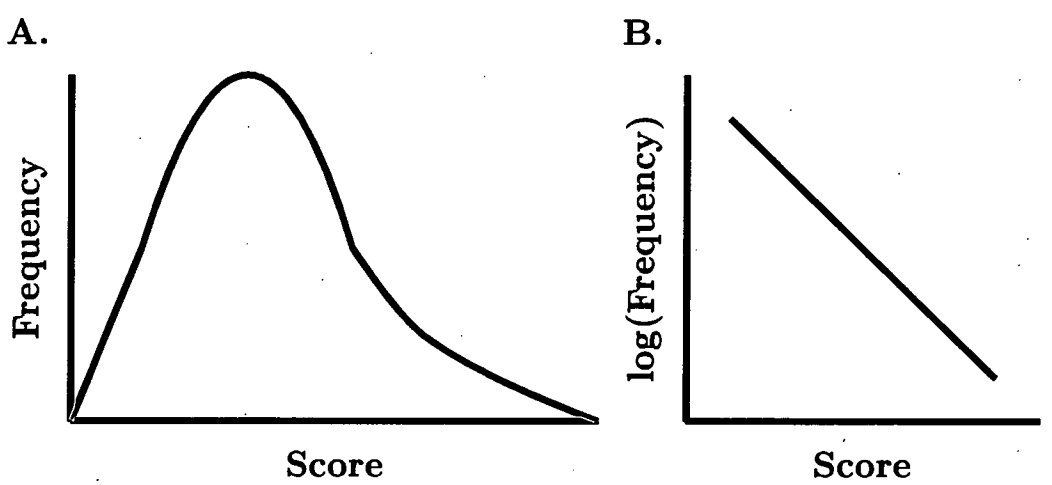


Figure 2.2: (A.) An example Extreme Value Distribution showing the slow decline in the tail differing from the usual bell shaped curve of a normal distribution. (B.) The linear relationship of the tail region of the extreme value distribution of the score with the log of the frequency for the scores.

The EVD can be used to derive several measures of statistical significance, three

of which are discussed here. The first of which is the P-value (Equation 2.5) describing the probability of a score occurring by chance rather than through biological relatedness. Where K is a constant which has the effect of correcting for the non-independence of possible starting points for matches in the alignment. The other constant, λ, is a scaling factor to transform the scores from the substitution matrices, used to calculate the score, to a natural scale.

$$P = Ke^{-\lambda S} \tag{2.5}$$

The second statistical significance measure is the E-value, Expectation Value or simply Expectation. There are several ways of calculating E-values depending on the algorithm being assessed. These boil down to two main methods; the BLAST method and the pairwise alignment algorithms method. The equation for calculating the E-value is the same for both methods, differing only in the term m (Equation 2.6). In the pairwise method m denotes the length of the sequences being aligned and in BLAST m denotes the total length of all the sequences added together in the database being searched. The total length of the database is taken in to account to allow for the large variation in the lengths of the sequences in the database. This assumes that a query is more likely to be related to a long sequence rather than a short sequence.

$$E = Kmne^{-\lambda S} \tag{2.6}$$

The E-value can also be calculated directly from the P-value (Equation 2.5) as shown in Equation 2.7.

$$E = mnP \tag{2.7}$$

The third statistical significance measure is the bit score. These scores are most commonly seen in BLAST searches and are given to each sequence returned in the database search. A bit score can be derived by normalising the raw scores using K and λ (Equation 2.8). Bit scores from different alignment methods can then be compared.

$$S\prime = \frac{\lambda S - lnK}{-ln2} \tag{2.8}$$

An E-value can also be derived from the bit score (Equation 2.9).

$$E = mn2^{-S\prime} \tag{2.9}$$

2.4 Multiple Sequence Alignments

Pairwise alignments are designed for looking at pairs of highly similar sequences either locally (Section 2.2.1) or globally (Section 2.2.2). The problem arises when studying more than just pairs of sequences, for instance all the members of a family of functionally related proteins. The members of a protein family do not necessarily have highly similar sequences, however the family is likely to have key residues conserved over evolution for functional or to a lesser extent structural roles. Multiple sequence alignment algorithms are designed to align sets of related proteins to allow the study of these evolutionarily conserved

residues.

It is theoretically possible to use pairwise sequence alignment algorithms to create a multiple sequence alignment. However this would require levels of computing power that is currently not practical, therefore heuristic multiple sequence alignments have been developed, three such methods are discussed below. Clustal based methods and T-Coffee are both based on progressive alignments, while AMAS uses an iterative refinement method.

2.4.1 Clustal Family

The Clustal family of programs, at present, are arguably the most commonly used set of multiple sequence alignment programs. The command line version, Clustal W (Thompson *et al.*, 1994), and the graphical interface version, Clustal X (Thompson *et al.*, 1997), are freely available for most computer platforms.

Clustal uses a fast approximation of the full dynamic programming pairwise alignment algorithms (Smith & Waterman, 1981; Needleman & Wunsch, 1970) to perform pairwise alignments for each sequence against every other sequence in the set of sequences being aligned. The results of the all-against-all pairwise alignments is a matrix of all the alignment scores. A guide tree is then derived from the pairwise score matrix. Clustal uses the *Neighbourhood Joining* method to create the tree, unrooted with branch lengths proportional to the estimated evolutionary distance. The *mid-point* method is used to place a root at a point where the means of branch lengths on either side of the root are equal. Once the

tree has been constructed weights can be assigned to each of the sequences, dependent on the distance the sequences are from the root of the tree. This means sequences from the same branch of the tree will be assigned the same weighting, high weightings are assigned to diverse sequences. The next step is to create a progressive multiple sequence alignment, where a series of pairwise alignments are used to align larger and larger groups of sequences. The sequences to be added are dictated by the branch order of the guide tree.

2.4.2 T-Coffee

Tree-based Consistency Objective Function for alignment Evaluation or T-Coffee (Notredame *et al.*, 2000), like Clustal, is also a progressive alignment program. However T-Coffee is a much more rigorous method, so is considered to produce more accurate alignments (Thompson *et al.*, 1999*b*; Thompson *et al.*, 1999*a*). However this comes at a considerable time cost compared to Clustal.

T-Coffee creates a library of pairwise alignments, both global and local alignments, for all the sequences to be aligned. The library is then used to find the optimum multiple sequence alignment for the information contained in the library. A progressive alignment method using all the library information is used to create a multiple sequence alignment.

In more detail, two sets of pairwise alignments are performed; one set performed using a global pairwise sequence alignment and the other using local pairwise sequence alignment. For the local sequence alignments the top ten non-intersecting

alignments for each pair of sequences are obtained using the Lalign program (Pearson & Lipman, 1988). Each pairwise alignment is represented in a library as a list of pairwise residue matches, which are then used as constraints. Each of these constrains is also assigned a weighting by the sequence identity of the two sequences aligned. Each set of pairwise alignments create two primary libraries, one for local alignments and one for global alignments. The two libraries are then combined by a simple process of addition where duplicates are merged with a weighting equal to the sum of the two original weights. Once all the pairwise alignment data has been pooled, the next step is to examine the consistency of each pair of residues for each pair of alignments. This consistency is used to give each pair of residues a weight, a process called *library extension.*

The Clustal progressive alignment method is used to create an all-against-all pairwise sequence alignment score matrix and from this create a guide tree. The guide tree is then used to align the sequences in branch order. In the T-Coffee method the weights in the extended library are used to align residues in the two closest sequences. Then the next two closest sequences are aligned or a sequence added to an existing alignment depending on the guide tree. Two groups of aligned sequences are added together using averaged library scores for each column in the alignments. This process continues until all sequences are aligned.

2.4.3 AMAS

One problem of the progressive alignment methods is the freezing of the alignments of groups of sequences once they have been aligned so they can not

be altered with the addition of further sequences to the alignment. AMAS (Livingstone & Barton, 1993; AMAS, 2005) is an iterative refinement multiple sequence alignment program. The method starts, as with the progressive alignment methods, by creating a matrix of all-against-all pairwise alignment scores. The pair of sequences with the highest pairwise similarity are then aligned. Then the sequence with the highest similarity to the profile of the previously aligned sequences as found and aligned to the first pair of sequences by profile sequence alignment. This step is repeated until all the sequences are included in an unoptimised multiple sequence alignment.

The key stage in the method is the optimisation of the multiple sequence alignment created. A single sequence is removed from the alignment and it is realigned to the profile of the alignment (missing the removed sequence). Each sequence is removed and realigned. This continues iteratively until a set number of iterations is complete or the score for the alignment is no longer improved.

2.5 Conservation Scores

Following on from the creation of multiple sequence alignments it is very useful to determine levels of conservation for amino acids in the alignment. Identifying invariant residues in a column of an alignment is a straightforward task, often accomplished visually, however it is less trivial to determine the level of conservation if there is any variance in the amino acid types. Having a measure of conservation for each column in an alignment is very useful as a

high level of conservation gives an indication of the importance of that position for functional or structural roles within the protein itself. Unfortunately there is no accepted standard way of calculating conservation scores, and they are sometimes referred to as conservation indices rather than scores.

Below is a short guide to the different techniques for calculating conservation scores, however Valdar has written extensive reviews on conservation scores (Valdar, 2002; Valdar, 2001; Orengo *et al.*, 2003*b*). The examples outlined below are adapted from Valdar's reviews and references to columns refer to columns from multiple sequence alignments.

2.5.1 Symbol Diversity Scores

Scores based on symbol diversity treat amino acid residues simply as independent alphabet characters, so no further information of the properties of the individual amino acids is considered. The most straightforward of the symbol diversity scores is that developed by Wu and Kabat (Wu & Kabat, 1970).

$$\text{Kabat Score} = \frac{k}{n_1} N \tag{2.10}$$

Equation 2.10 shows the calculation of symbol diversity where k is the number of different amino acids types in the column, n_1 is the number of times the most commonly occurring amino acid in the column appears and N is the total number of sequences in the column. The symbol diversity scores have many failings due to the fact that amino acid properties are not taken into

consideration. Also there is no provision for taking into account gaps in the score.

A more complex branch of the symbol diversity scores are the symbol entropy family of scores. Like the basic symbol diversity scores, the symbol entropy scores only consider amino acids as alphabet characters, however the scores take into account the relative frequencies (using Shannon's Entropy) of the amino acids from the column.

2.5.2 Physiochemical Property Scores

In contrast to the symbol diversity scores the physiochemical property scores take in to account the different physical and chemical properties of each of the twenty amino acids. The amino acids can be divided up into overlapping categories of the properties of the amino acids. These categories can include: hydrophobic, polar, non-polar, charged, small, tiny, aromatic and aliphatic. The overlapping categories are often set out in score tables, where amino acids are assessed for each property. The assessment is a binary score of, for example, whether an amino acid is hydrophobic or not.

$$\text{Zvelebil Score} = n_{const} X \frac{1}{10} \tag{2.11}$$

An elegant example of the physiochemical property scores was developed by Zvelebil (Zvelebil *et al.*, 1987) outlined in Equation 2.11. Where n_{const} is the number of properties shared by all the amino acids appearing the in the column.

2.5.3 Substitution Matrix Scores

The next set up in complexity is through the use of substitution matrices as empirical measures of amino acid substitutions. The scores in a substitution matrix quantify how likely one amino acid will be substituted by another. One of the best well known substitution matrix scores was developed by Karlin and Brocchieri (Karlin & Brocchieri, 1996). The first step is to normalise a substitution matrix (Equation 2.12) so that any amino acid being substituted by itself has a score of 1. The range of values in the matrix are therefore between the values -1 and 1.

$$S(i,j) = \frac{s(i,j)}{\sqrt{s(i,i)s(j,j)}} \tag{2.12}$$

One the substitution matrix has been normalised Equation 2.13 is used to calculate the conservation score for the column. Where $S(i,j)$ is the normalised substitution matrix and N is the total number of sequences.

$$\text{Karlin Score} = \sum S(i,j) X \frac{2}{N(N-1)} \tag{2.13}$$

The Karlin score is an example of a *sum of pairs* score as the score is a sum of all the possible substitutions of pairs of amino acids. The Karlin score fails to take gaps in to consideration, thought to be a major weakness of the score.

2.5.4 Weighted Scores

So far the conservation scores discussed have assumed that the sequences in the multiple sequence alignment contribute equally to the conservation score. This is

not the ideal situation as highly similar sequences should not contribute as much as diverse sequences as sets highly similar sequences offer little biological information to the conservation score beyond what a single member of the set contributes.

To address this the sequences can be normalised to act against sequence redundancy in the alignment. Vingron and Argos (P. Vingron, 1989; Vingron & Sibbald, 1993) have developed a simple weighting scheme (Equation 2.14) to assign weights to each of the sequences in the alignment. Where N is the total number of sequences in the alignment and $d(i,j)$ is some evolutionary distance between the sequences (e.g. sequence identity). The resulting weightings for each individual sequence in the alignment can then be incorporated in other conservation scores.

$$\text{Vingron Weight Score} = \frac{1}{N-1}\sum_{j\neq 1}^{N} d(i,j) \tag{2.14}$$

One such score that incorporates a weighting score is the Valdar score (Valdar & Thornton, 2001) which is considered to be one of the best scores currently available. Equation 2.15 outlines the calculation of the score. Where $S(i,j)$ is the score for substituting amino acids from a sequence.

$$\text{Valdar Score} = \left(\sum_{i}^{N}\sum_{j>i}^{N}\right)^{-1}\sum_{i}^{N}\sum_{j>i}^{N} w_i w_j S(i,j) \tag{2.15}$$

The score is a weighted substitution matrix score. Amino acids are compared in a pairwise fashion for each pair of sequences in the alignment taking into account the individual weightings of the sequences.

2.6 Profile Hidden Markov Models

Hidden Markov Models (HMM) are computational structures holding information on the patterns found in families of related protein sequences. The uses of HMM range from the detection of distant homologues in sequence databases to fold recognition as used in the Critical Assessment of Techniques for Structure Prediction competition (CASP)(Dunbrack *et al.*, 1997). Hidden Markov Models were originally developed in the 1970's in the field of speech recognition. A HMM is a first order Markov chain, defined by the choice of a state being determined by the previous state.

In the case of protein sequences the physical processes leading to the observed sequence are hidden. So a statistical model relates the physical processes to the observed sequence by probability distributions. This statistical model is known as a profile Hidden Markov Model as it is described by a Markov chain, but the physical processes can not be observed directly so it is termed hidden. With protein sequences the physical processes can be though of as evolution (i.e. mutation, insertion and deletion of residues in the sequence).

There are two methods of generating a profile Hidden Markov Model (pHMM), either from a multiple sequence alignment created by other unrelated software or they can be built from a set of unaligned sequences. In the latter case the unaligned sequenced are aligned as part of the pHMM creation process which defines the parameters included in the pHMM.

Profile HMM always have a beginning and an end state and there are three other types of state. The first state is a *match* state modelling the distribution of the different residues in a column of a multiple sequence alignment. The second and third states relate to insertions or deletions, equating to the creation of a gap or insertion of a residue between the current column and the next. Each state in a pHMM has associated with it a symbol *emission probability distribution* and states are connected by *transition state probabilities.*

Figure 2.3 demonstrates how a pHMM is generated from a multiple sequence alignment, showing all possible transition states. From the initial *start* state (B) the pHMM structure is transversed by analysing every column in the input alignment until the final *end* (E) is reached. The result is a hidden state sequence and an observable gapped amino acid sequence.

Once a pHMM has been created there are two standard dynamic programming algorithms used for aligning and scoring sequences to the pHMM. In this way databases can be searched to find best scoring matches. The two standard algorithms are the Viterbi and the Forward algorithms. The Viterbi algorithm is involved in the alignment of sequences to the pHMM and the Forward algorithm provides scoring of sequences matching the pHMM.

There are several pHMM software packages freely available. The most widely used packages are HMMer (Eddy, 1998) and SAM (Karplus & Hu, 2001).

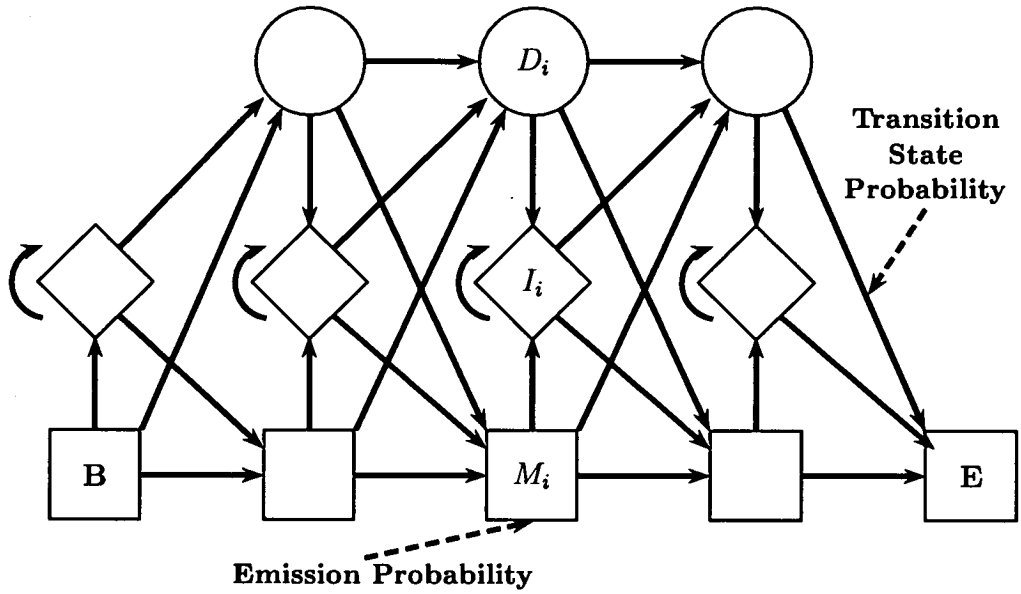


Figure 2.3: Structure of a profile Hidden Markov Model. Each of the red squares represents a column in a multiple sequence alignment. This match state (M_i) for each column *emits* a residue based on the distribution of amino acids in the column of the multiple sequence alignment. From this state, the next step is determined by the *transition state probability*, which decides whether is is best progress to a *delete* state (D_i)(Green circle) or an *insertion* state (I_i)(Blue diamond). A *delete* state will step over a column resulting in a gap insertion or gap extension. The *insertion* state allows for residue insertion between columns. Once the structure has been traversed, where every column has been analysed and the end state reached, the result is the creation of a hidden state sequence and a gapped amino acid sequence. Diagram adapted from Orengo *et al* (Orengo *et al.*, 2003*a*).

2.7 PISCES Sequence Culling

PISCES (Wang & Dunbrack, 2003; PISCES, 2005) currently provides three options for obtaining culled lists of protein sequences. The first option is to provide a culled list of the entire PDB (Berman *et al.*, 2000) by sequence similarity. The second option is to obtained a culled list of sequences from a user defined set of sequences; this subset is in the form of SwissProt, GenBank or PDB identifiers. The third option is to submit a list of sequences in FASTA

format or the output from a BLAST search. The culled lists for PLP, thiamin, glutathione and folic acid utilising enzymes were obtained using the third option described above so the PISCES method will be described specific to this option.

PSI-BLAST (Altschul *et al.*, 1997) is used to create a position specific similarity matrix (PSSM) using each entry sequence as a query. Three iterations of PSI-BLAST are performed using an E-value of 10^{-4} as a cut off. These PSSM's are pre-calculated and updated weekly to include all new database entries. The PSSM is used to score sequence identity between sequences in the final part of the PISCES method.

Along with the submission of the sequence lists, users also submit a set of criteria for the culling. The criteria are sequence identity cut off, minimum and maximum sequence lengths. Once a set of sequences and criteria has been submitted to the PISCES web site, the main culling process begins.

The first step in the method is to ensure all the sequences pass the criteria submitted by the user. The first sequence in the list is designated to be included in the final list, then moving down the list if the next sequence has a sequence identity higher than the cut off then this sequence is designated to be rejected from the final list. This process is then repeated for the second sequence and so on. The final list comprises of all the sequences that have been designated to be included.

2.8 Minimisation of Functions

The minimisation of a function is equivalent to maximising a function and should be correctly termed function optimising. However the usage of optimisation algorithms as discussed in this thesis is specifically minimisation and will therefore be referred from here on as minimisation.

The concept of a minimisation is to find the values of one or more independent variables of a function (F). Conceptually the most straightforward method of determining the values of the variables at the minimum of the function would be to try all possible values for the variables in an exhaustive search. This method however is computationally prohibitive. In fact some problems become so computationally demanding they are unsolvable with even the most powerful of supercomputers in a reasonable amount of time (less than several lifetimes!). For this reason minimisation algorithms have been developed to solve minimisation problems without using valuable resources; memory, processor power and time.

The simplest cases of minimisations are one-dimensional functions where just one variable is being minimised. This is complicated by the addition of further variables to create multidimensional minimisation problems. Minimisation functions must be able to find the global minimum for a function rather than the various possible local minima (Figure 2.4)

A common example in minimisation is the travelling salesman problem. The problem is to find a route between a set of cities that the salesman must

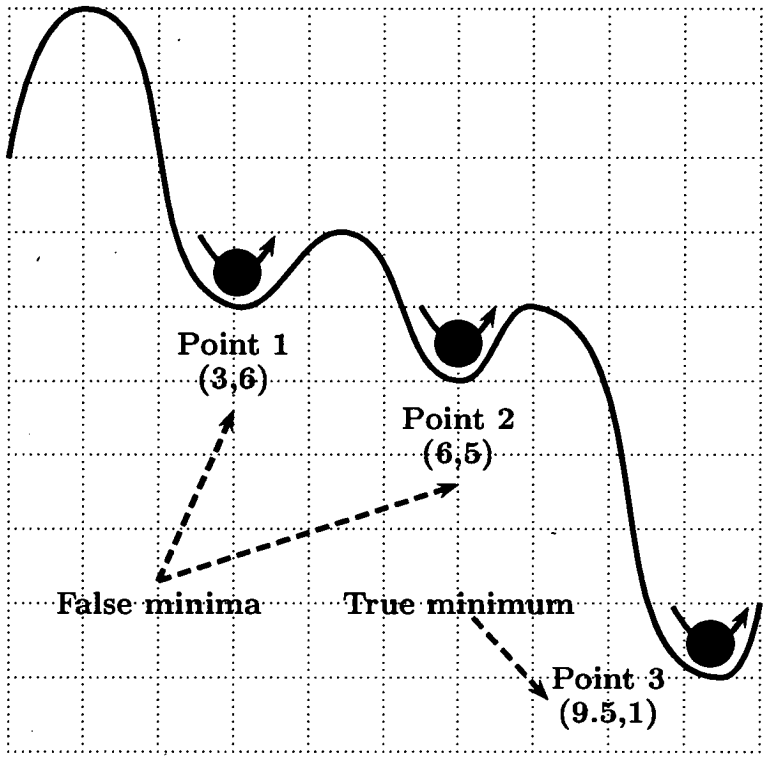


Figure 2.4: Finding Local Minima of a 1D function. Three troughs are observed at points 1,2 and 3. A minimisation algorithm will attempt to find the global minima (green) for the function, in this case point 3. The algorithms can sometimes *get stuck* in local minima, points 1 and 2, resulting in a false minima (red) being given for the function.

travel, but only visiting each city once. The only guaranteed solution is found by exhaustively searching every possible route. There are several different techniques for minimisation and there is no best minimising function that will enable all minimisation problems to be solved. The best method is defined by the function being minimised. The downhill simplex is considered to be one of the most straightforward methods and is often the first choice to find a quick solution.

2.8.1 Downhill Simplex Minimisation in Multi-dimensions

The simplex in the title of this minimisation technique refers to a geometrical shape (Nelder & Mead, 1965). The number of vertices in the shape is directly related to the number of dimensions or variables being minimised. The number of vertices is one more than the number of dimensions (N), in a two dimensional minimisation the simplex is a triangle ($N + 1$). The term downhill refers to the inability of the algorithm to backtrack along the function, so is thought of as always proceeding downhill.

In one-dimensional minimisations two values can be used simultaneously to bracket a minimum. With multiple dimensions this is not possible so a best guess has to be given as a starting point. This starting guess represents a simplex of N values. The use of a simplex means that derivatives need not be used as the minimum is bracketed by the simplex. The algorithm then proceeds downhill through the function until a minimum is reached, hopefully the global minimum for the function. The way the algorithm proceeds downhill is by movements of the start guess simplex. The simplex undergoes four possible movements (reflection, reflection and expansion, contraction, multiple contractions). The direction that the algorithm proceeds along the function is determined by the values of the function at points in the simplex, highest to lowest. The simplex is contracted until the minimum is bracket by the smallest simplex within the defined accuracy. The algorithm for the movements of the simplex is called

amoeba, as the movements of the simplex can be thought of as analogous to that of an amoeba changing its shape to move through variable terrain.

A tolerance factor is given to the algorithm to define the end point of the minimisation. If no progress in moving the simplex has been made within the value of the tolerance factor then the algorithm is stopped and values for the minimum taken. The value for the tolerance factor is determined usually by taking the square root of the precision of the measurements inputted to the function. The downhill simplex method works well with relatively few dimensions. If larger numbers of dimensions are used other minimisation algorithms should be used. The downhill simplex method has been coded up in several languages including Perl and is freely available to download as a Perl Module, Amoeba.pm (Perl Module: Math::Amoeba, 2005)

Chapter 3

Method Development

3.1 Overview

A function prediction for a putative enzyme is conducted by comparing conserved functional amino acid residues (KRENDYCHQST) from within enzyme families with known function, referred to as reference enzymes, to the query putative enzyme. To extract conserved functional residues for a reference enzyme, first a multiple sequence alignment must be created, residues are then extracted using a custom built conservation score. The extracted residues are considered an unordered set of conserved functional residues, termed a *Tread* and Treads for each of the reference enzymes are stored in the reference Tread database. An additional step during the creation of a reference Tread is to assign functional roles to each of the conserved residues in the Tread. In the current release of the function prediction software (DAROGAN) the putative enzyme is submitted as a multiple sequence alignment as provided by the user. The quality of this alignment greatly affects the results of the function prediction so an assessment

of the quality of the alignment is made, highlighting any deficiencies in the users alignment. The aim of this assessment is to attempt to ensure only high quality alignments are used to perform function predictions. In reality the user is presented with warnings of potential low quality alignments and given the option to continue with the prediction or to abort.

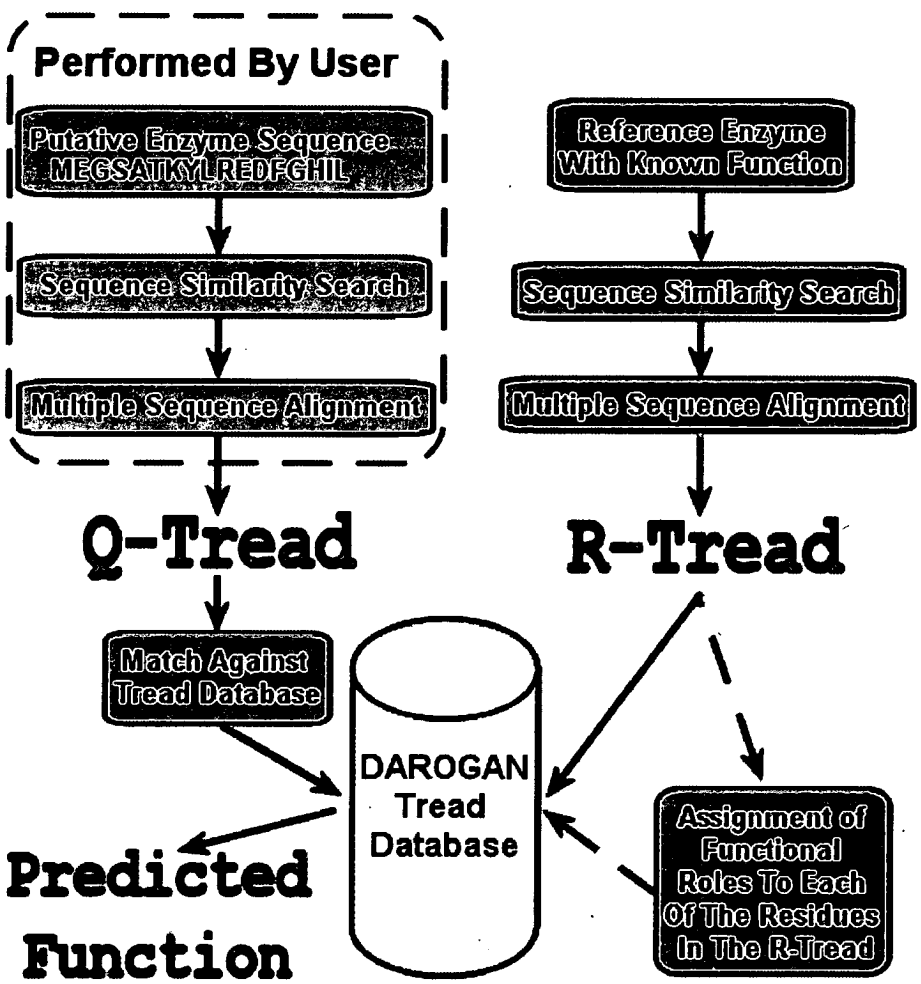


Figure 3.1: DAROGAN Flow Diagram describing the procedure for creating reference Treads, unordered set of conserved functional residues (KRENDYCHQST), and storing them in the DAROGAN Tread database (right). The procedure for performing a function prediction against the reference database is also shown (left). Key: Q-Tread - Query Tread, R-Tread - Reference Tread.

Treads are stored in vector format and an adaptation of the *Vector Space Model* was implemented to perform the actual comparisons. Each comparison is scored with a *Relevance* score, quantifying the similarity between Treads. A E-value is also calculated to assess the statistical significance of the matches, by calculating the probability that a match could have occurred by chance alone (See Section 3.7). Functional information, including the cofactor most likely utilised, for the putative enzyme can then be inferred from the statistically significant matches from the Tread database. See Figure 3.1 for a flow diagram summarising the DAROGAN function prediction method.

3.2 Reference Tread Creation

Similar sequences are acquired by performing BLAST searches (Section 2.2.4) against the SwissProt sequence database (SwissProt & TrEMBL, 2005) and are then aligned using a multiple sequence alignment program. The suitability of the alignment, for the DAROGAN method, is dictated by two main factors. The first is the number of sequences returned; a suitable alignment will contain at least twenty sequences. Any fewer than twenty sequences limits the amount of information that can be gained from the alignment as the conserved positions could occur by chance. The second measure is the number of conserved functional residues appearing in the alignment. To be suitable an alignment must contain approximately fifteen functionally conserved residues (± five residues). The figure of fifteen residues was derived from a preliminary, literature based, study into

how many conserved residues are typically involved in the function of the enzyme (data not shown). It was found that a fifteen residue cut off should be sufficient to include all the important functional residues in the case of enzymes catalysing bimolecular reactions. There are several ways a residue can contribute to the function of an enzyme and these are discussed further in Section 3.3. Creating a multiple sequence alignment with approximately fifteen conserved functional residues is accomplished by tailoring the E-value cut off in a BLAST search.

3.2.1 Tailoring the E-Value

The E-Value associated with a similarity score is a measure of the number of times one would expect to get a score of at least X by chance alone (See Section 2.3.1 for further details). By tailoring the E-Value cut off in a BLAST search it is possible to alter the evolutionary width of the sequences returned by the search; the lower the E-Value cut off the narrower the evolutionary width of the sequences returned. As a consequence of the sequences being more closely related the number of conserved positions in a multiple sequence alignment of the sequences will be larger.

Figure 3.2 shows a typical example (PDB 1ASP) of the effect of tailoring the E-Value cut off on the number of conserved residues appearing in a subsequent multiple sequence alignment. The number of conserved residues is seen to rise steadily, but not linearly, with the logarithm of the selected E-Value cut off. Unfortunately it is impossible to determine the exact number of conserved functional residues that will appear in an alignment of sequences from a BLAST search with-

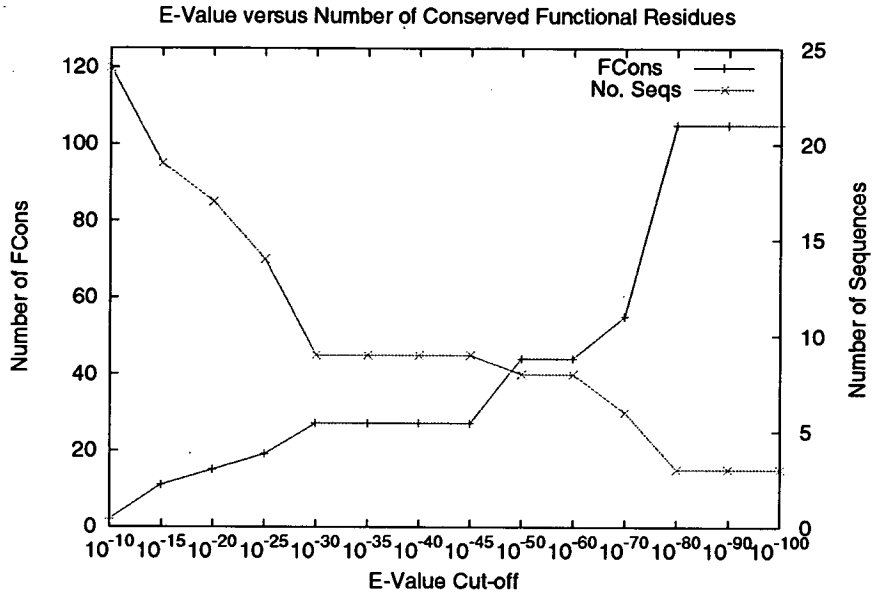


Figure 3.2: The number of conserved functional residues for a multiple sequence alignment is seen to increase with decreasing E-Value cut off for the BLAST search (left hand *y axis*). The number of sequences appearing in the resulting multiple sequence alignment are seen to decrease with decreasing E-Value cut off (right hand *y axis*). Example used was Ascorbate Oxidase (PDB:1ASP)

out first aligning the sequences. Creating an alignment requires a considerable amount of computational time, so for this reason an intelligent alignment selection heuristic was developed to reduce the computational time required to find the optimum alignment (See Section 3.2.2), but first the process of extracting conserved residues from a multiple sequence alignment is discussed.

3.2.2 Intelligent Alignment Generation and Selection

An intelligent alignment generation and selection heuristic was developed to reduce the computational time required to create an optimum multiple sequence alignment containing at least twenty sequences and approximately fifteen conserved functional residues. To create an alignment that meets the required

criteria, it is necessary to tailor the E-value cut off in the BLAST search to find similar sequences (Section 3.2.1).

Tailoring of the E-value cut off can be an extremely tedious and time consuming process; a BLAST search at a particular E-value cut off has to be performed followed by the creation of a multiple sequence alignment, before the number of functionally conserved residues in the alignment can be determined. It is often the case that several BLAST searches and alignments have to be performed before an optimal one is found. It was for this reason an intelligent alignment generation and selection heuristic was developed to greatly reduce the computational time required to produce an optimal alignment.

It was found that the most common E-value cut off, for which the optimal alignment BLAST search was performed, was 10^{-40} (See Section 3.2.1). It is often the case that a BLAST search at 10^{-40} will yield an optimum alignment, however this will not occur for all cases. The heuristic has a defined set of E-value cut offs (between 10^{-10} and 10^{-150}) for which BLAST searches can be conducted and uses the 10^{-40} cut off as a starting guess. The number of conserved functional residues extracted from the alignment at the first guess cut off is then used to determine at which E-value the next alignment should be conducted (See Figure 3.3 for a summary of the method). Only a single BLAST search needs to be performed by the heuristic which is then parsed to a set of files for each of the E-value cut offs. This is equivalent of performing BLAST searches for each of the required E-value cut offs.

In earlier versions of the heuristic two stages of multiple sequence alignment were performed. The first stage was to use ClustalW to produce initial alignments then assess how many conserved functional residues appear in the alignment. Once an optimum alignment was identified, it was then realigned using T-Coffee; a more accurate, but more CPU intensive multiple sequence alignment program. In the current version of the heuristic, the more recent MUSCLE (Edgar, 2004) alignment program is used as it has been shown to produce alignments matching or improving on the standard set by T-Coffee, but with out the computational expense.

The intelligent alignment heuristic will perform a minimum of two alignments, but will not perform more than nine alignments. In the worst case the heuristic is nearly twice as efficient as aligning each of the sets of sequences at varying E-Value cut-offs. The algorithm is not guaranteed to to find an optimum alignment within the defined E-value range, if the number of sequences or number of conserved functional residues is not optimal, then it will fail. To reduce the number of failures, BLAST searches are performed against two different sequence databases; SwissProt and TrEMBL. It is generally the case that if the algorithm fails to find an optimum alignment against one of the databases it will succeed with the other.

3.2.3 Extracting Conserved Functional Residues

Once the optimum alignment has been created the next stage is to extract the conserved functional residues. It would be simple to extract all 100% conserved functional residues, however there are several disadvantages with this approach.

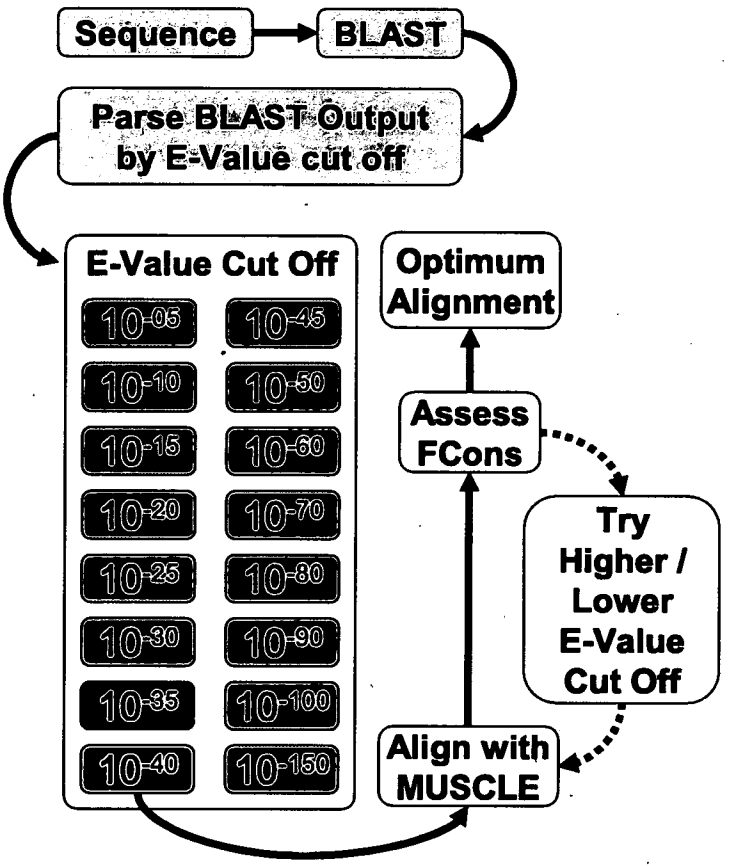


Figure 3.3: Similar sequences, to the sequence being used to produce a Tread, are found by performing a single BLAST search, parsed into separate files by E-Value cut off. The intelligent alignment heuristic then produces an optimal alignment containing approximately 15 conserved functional residues (FCons).

The first problem is that there might be a sequencing error in one of the sequences in the alignment, masking a functionally conserved residue (See Figure 3.4A for an example). The second problem is with potential alignment errors (See Figure 3.4B for an example). Although multiple sequence alignment algorithms are fairly accurate as a rule, they are by no means infallible. It is common for researchers to create a multiple sequence alignment using an automated method (e.g. ClustalW) and then edit the alignment manually. This two step process is very effective for producing very accurate alignments, however it is a very time

consuming and tedious process, so if more than a handful of alignments need to be generated then this two step method is no longer realistically viable.

A.

```
+ + + − +
A B C D E
A B C G E
A B C D E
```

Sequencing Error

B.

```
+ + + − − −
A B C F D E
A B C D E H
A B C F D E
```

Alignment Error

Figure 3.4: Errors in Multiple Sequence Alignments. **A**. Sequencing errors. In the 4rd column of the alignment the 2nd sequence has a glycine residue (G) in place of an aspartic acid residue (D) due to a sequencing error, masking a conserved position. **B**. Alignment error. In the 2nd sequence there should be a gap between columns 3 and 4 masking conserved positions in columns 5 and 6.

The problems leading to sequencing and alignment errors are uncontrollable for DAROGAN, so a method for extracting conserved functional residues must be tolerant toward these errors. One possibility is the use of a similarity score to quantify conservation at each alignment position (Section 2.5). Similarity scores are also known in the literature as conservation scores, quality scores and conservation indices.

3.2.4 Conservation Scores

A brief look into the literature reveals a wealth of methods for scoring conservation in a multiple sequence alignment, examples of conservation scores are introduced in Section 2.5. Unfortunately there is no widely accepted *best*

method, making selecting a method difficult, especially as new methods are still being published (Armon *et al.*, 2001; Pupko *et al.*, 2002; Valdar, 2002)

For extracting conserved functional residues a conservation score must be able to identify positions as being conserved with a degree of tolerance to take into account possible sequencing and/or alignment errors. As there is currently no widely accepted best method for scoring conservation it was decided that the best approach would be to develop a new method for scoring conservation optimised to one of the best currently available methods. The aim of this is to allow the conservation score to reproduce the current best method with the minimum of re-coding. If a new better method is developed then the conservation score can be optimised to reproduce the results of this new method.

The conservation score developed is based on a simplification of the *Trident* score (Valdar, 2002). The simplified *Trident* score has been termed the *Bident* score as it combines two components, rather than the three used in the *Trident* score. Although the *Bident* score resembles the *Trident* score in the way that they are both composites of other scores, the components of *Bident* are different representing a new method for scoring conservation in alignments.

Trident Score

The Trident method combines three conservation scores into a single score (Figure 3.5) and the three components are a symbol diversity score, a stereo-chemical score and a gap score (See Section 2.5). Each of the three components has a

weighting associated with it, allowing the score to be weighted in favour or against each of the components. This tailoring can be very useful in developing a custom conservation score, however the Trident score can also be optimised to another score by selecting appropriate values for each of the component weightings.

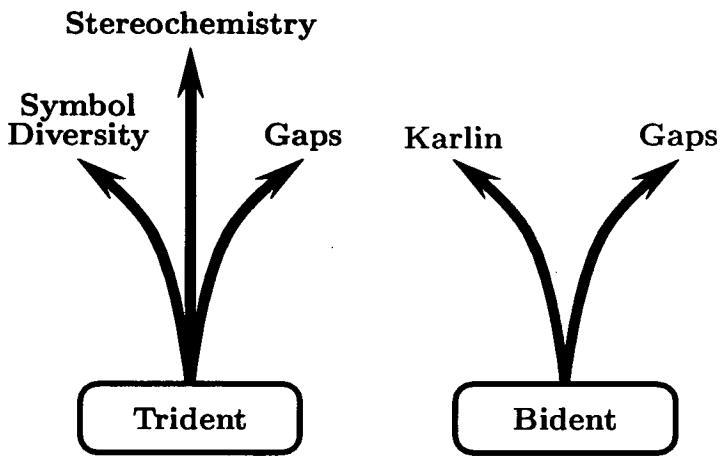


Figure 3.5: The Trident score combines a symbol diversity, stereochemical and gap score into a single score and the Bident score combines Karlin and gap scores into a single score.

Bident Score

The Bident score, like the Trident score, combines scores into a single score, however there are only two components in Bident score and only one of the components is the same in both. The main advantage of only having two scores, rather than three, is the simplification of tailoring the weighting values. The two scores in the Bident score are the Karlin score (Karlin & Brocchieri, 1996) (Section 2.5) and a gap score. The Karlin score was chosen as its calculation is both straightforward and intuitive, the score also performs well for identifying functionally conserved residues. The Karlin score is a robust score, but lacks

gap scoring so it has been combined with a gap score to address this.

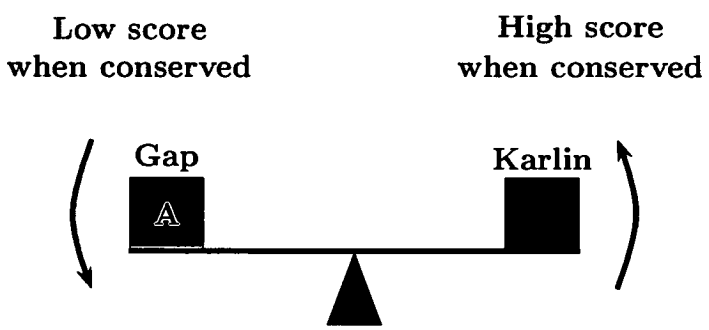


Figure 3.6: Bident Score Weighting.

The aim in developing the Bident score was to optimise it to recreate the best currently available conservation score. Currently one of the best conservation scores is one developed by Valdar (Valdar, 2002), satisfying all the properties required of a good score. These properties are discussed in detail in Valdar's paper, so will not be discussed here. To optimise the Bident score to the Valdar score it is necessary to find weightings for each of the score components (Karlin and gap, Figure 3.6). To accomplish this two functions were developed, termed the *Powers* function (Equation 3.1) and the *Linear* function (Equation 3.2). The weightings are X for the gap score (A), low when the position in the alignment is conserved and Y for the Karlin score (B), high when conserved. Two functions were developed with the aim of determining the best function for finding optimum values for X and Y.

$$\text{Powers Score} = (1 - A)^X B^Y \tag{3.1}$$

$$\text{Linear Score} = (1 - A)X + BY \tag{3.2}$$

To determine values for the weighting factors X and Y to best match the Valdar score a minimisation algorithm was used against a reference set of multiple sequence alignments.

Minimisation

The basic approach for the minimisation is to take a multiple sequence alignment and calculate the Valdar similarity score for each position in the alignment and then find optimum values for the weighting factors X and Y in the Bident score, that best match the Valdar score. The goal is to find values for X and Y that will be applicable to any multiple sequence alignment. To accomplish this minimisations were performed against a set of 26 multiple sequence alignments (Table 3.1). The alignments are all for PLP utilising enzymes, the reason for selecting this set of alignments is discussed later (A review of minimisation and different minimisation algorithms is discussed in Section 2.8).

In more detail the first step is to develop minimisation functions for each of the two Bident score functions; Powers (Equation 3.1) and Linear (Equation 3.2). The minimisation functions are designed to find values of X and Y, that best match the Valdar similarity scores over the entire alignment. The two minimisation functions for the Powers and the Linear score are found in Equations 3.3 and 3.4 respectively.

$$\text{Powers Optimisation Function} = \frac{\sum_{i=0}^{n} \mid Valdar_i - ((1 - A_i)^X B_i^Y) \mid}{n} \tag{3.3}$$

$$\text{Linear Optimisation Function} = \frac{\sum_{i=0}^{n} \mid Valdar_i - ((1 - A_i)X + B_i Y) \mid}{n} \tag{3.4}$$

Although the Valdar score is considered to be the best conservation score currently available (Valdar, 2002), the minimisations were also performed against the ClustalX (Jeanmougin *et al.*, 1998) quality score. The aim of using the ClustalX score was to provide a control for the minimisation procedure, to ensure the data being generated were as expected. The minimisation algorithm employed for the minimisations was the Downhill Simplex method in multiple dimensions, publicly available in the Math::Amoeba Perl module (Perl Module: Math::Amoeba, 2005).

The first step in choosing which scoring function, Powers or Linear, best matches the Valdar score was to determine if the functions are amenable to the minimisation procedure. The aim of minimisation is to find a global minimum for the minimisation function; this of course assumes there is a global minimum to be found. The global minimum must also be bracketed, meaning the limits of the global minimum can be found, a bracketed minimum ensures that the global minimum can be found; for the range of X and Y values given. The easiest way of determining whether the global minimum can be bracketed is to plot the surface of the minimisation for a series of crude values for X and Y. Plotting

the surfaces for the Powers and the Linear minimisation functions revealed that the minima can indeed be bracketed (Figures 3.7 and 3.8). Plotting the surfaces, although crude, will give an approximation of minimised values for X and Y, which can be used as a sign that the minimisation algorithm is probably finding the correct global minimum for the function.

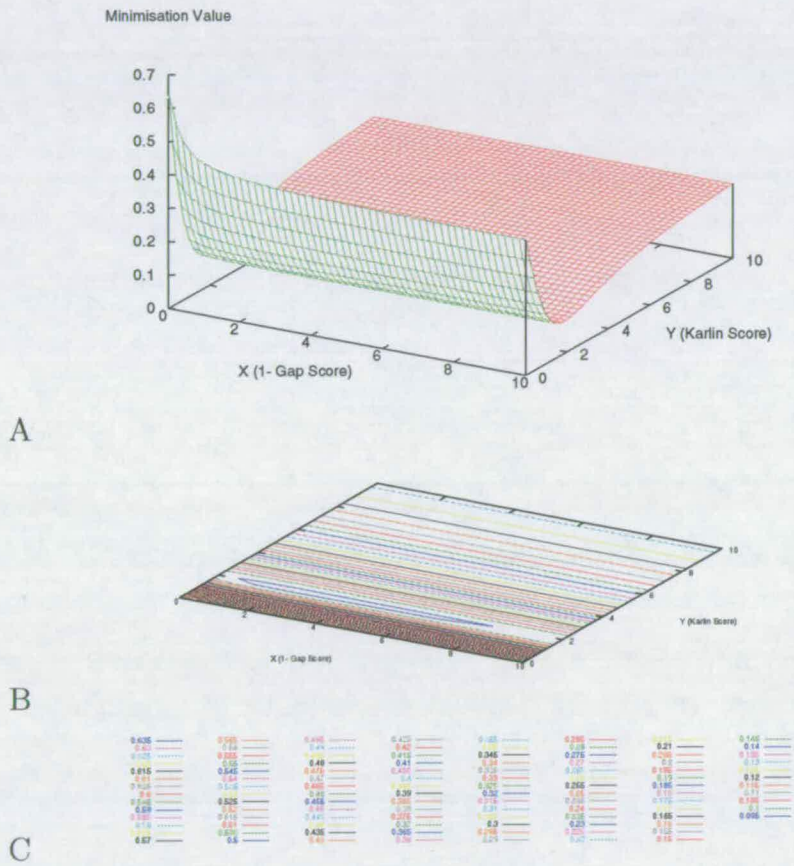


Figure 3.7: Powers Function Surface. **A**. Plot of the 3D surface of the Powers function. **B+C**. 2D plot of the same function surface with a colour key underneath.

Minimisations were performed against a reference set of 26 alignments for both the Valdar similarity score and the ClustalX quality score and for both the

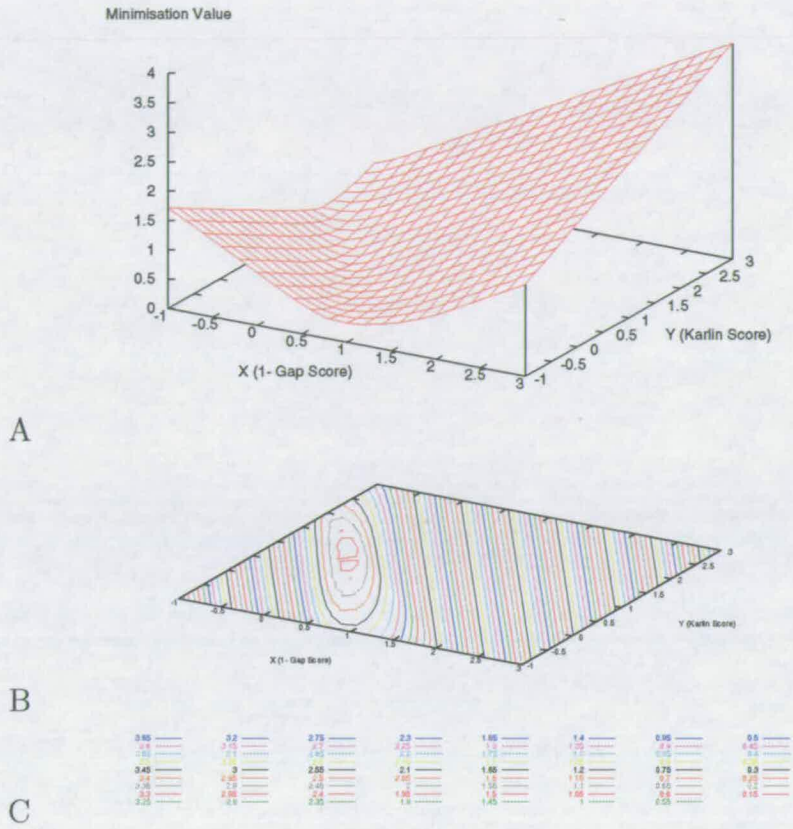


Figure 3.8: Linear Function Surfaces. **A**. Plot of the 3D surface of the Linear function. **B**+**C**. 2D plot of the same function surface with a colour key underneath.

Powers and the Linear scoring minimisation functions. In total 104 minimisations were run, which took several days CPU time.

The reference set of alignments were all from PLP and thiamin utilising enzymes, which means the optimised values for X and Y will be, to an extent, specific to these two families of enzymes. Ideally a reference set of enzymes containing a large number of unrelated alignments should be used. However this would be a massive undertaking computationally and would take a prohibitively long time to compute.

Minimisation Results

In total 104 minimisations were performed as there were two functions (Powers and Linear), two conservation scores (Valdar and ClustalX) and 26 reference alignments (Table 3.1). The results for each of the minimisations are minimised values for X and Y at the global minimum for the function. The values for X and Y represent the values which best simulate the conservation scores being minimised against.

			Powers Function				Linear Function			
Alignment Details			**Clustal**		**Valdar**		**Clustal**		**Valdar**	
PDB	*E-Value*	*DB*	X	Y	X	Y	X	Y	X	Y
1ars	E-50	SP	2.0312	1.9375	1.2515	0.8647	-0.2240	1.0761	0.0992	0.8819
1bfd	E-30	SP	1.7704	2.2262	0.9976	1.1455	-0.3264	1.1101	-0.0814	1.0586
1daa	E-20	SP	2.3594	2.2031	0.9214	1.0081	-0.0820	0.6380	0.0881	0.7736
1dje	E-25	SP	2.3594	2.2031	1.0854	0.9915	-0.2763	1.0154	-0.0144	0.9661
1f2d	E-15	SP	0.7634	2.5344	0.7217	1.1660	-0.5010	1.3627	-0.0441	0.9945
1f2d	E-40	NR	2.9688	2.0625	1.2634	0.9536	-0.2149	1.0308	0.1163	0.8313
1f3t	E-60	SP	2.0000	2.0000	1.2615	1.0062	-0.2770	1.0766	-0.0297	1.0093
1fc4	E-45	SP	2.3594	2.2031	1.0890	0.9709	-0.3166	1.1105	-0.0021	0.9671
1fg7	E-45	NR	3.7063	1.6948	1.9506	0.9123	-0.1419	0.9491	0.0807	0.8636
1g79	E-35	SP	1.3000	2.4919	1.0400	1.2119	-0.6340	1.6030	-0.1712	1.1711
1g79	E-40	NR	2.0000	2.0000	0.9219	0.9062	-0.1295	0.8518	0.0730	0.9019
1gde	E-50	SP	1.3672	2.0156	1.1884	0.9638	-0.2584	1.0809	0.0323	0.9467
1kta	E-20	SP	2.0000	2.0000	1.3155	0.9159	-0.2716	1.0784	0.0342	0.9364
1kta	E-40	NR	3.2227	1.8672	1.3467	0.9160	-0.1938	0.9506	0.0961	0.8473
1lk9	E-60	NR	2.6838	1.9158	0.7614	0.9480	-0.4444	1.2919	-0.0566	1.0373
1lw4	E-50	NR	3.5312	1.9375	1.1318	0.9082	-0.1287	0.8862	0.1287	0.8153
1n2t	E-20	NR	3.0820	2.4648	0.8992	1.1218	-0.1371	0.7063	0.0626	0.7863
1qj5	E-40	SP	1.8867	2.1797	1.2296	1.0594	-0.2378	0.9968	0.0147	0.9342
1qj5	E-45	NR	3.8516	1.7969	1.3340	0.9062	0.1425	0.2897	0.2200	0.6010
1sft	E-40	SP	2.5625	1.8750	1.3467	0.9160	-0.1225	0.8517	0.0971	0.8478
1trk	E-40	SP	3.0000	2.0000	1.2991	0.9214	-0.2590	1.0981	0.0574	0.9155
2cst	E-15	SP	3.0312	1.9375	1.2148	0.8698	-0.2102	1.0496	0.0964	0.8913
2tps	E-25	NR	2.7812	1.9375	1.3672	1.0156	0.0270	0.5938	0.1795	0.6408
2tps	E-30	SP	1.5312	1.9375	0.8728	1.1216	-0.5455	1.5456	-0.0997	1.0997
7aat	E-15	SP	2.6641	1.9219	1.2506	0.8752	-0.2021	1.0421	0.1085	0.8694
7odc	E-60	SP	2.0000	2.0000	1.2108	0.9866	-0.2758	1.1048	-0.0464	1.0376

Table 3.1: Minimisation Results. PDB sequence, E-Value and database searched are shown for each of the 26 alignments. Accompanied by the results of the minimisations for the Powers and Linear functions against the Clustal and Valdar conservation scores. In total 104 minimisations were performed.

A scatter plot shows how the X and Y values correlate for the 104 minimisations allowing the two functions to be compared (Figure 3.9). The Powers function produces a tight clustering for the Valdar score and a wider clustering for the ClustalX score. These clusterings are as would be expected for the 26 different alignments and the difference in the spread within these clusters can be explained by the fact that the Valdar score calculation is more similar to the Karlin score component of the Bident score than the ClustalX score. The Linear function produces slightly unexpected clusters for minimisations against the Valdar and ClustalX scores. It is difficult to account for the clusters aligning the way they do, one possible explanation is that the linear function is based on the classic $y = mx + c$ formula to describe a linear relationship translating into the clustering also being linear. The aim of comparing the powers and linear functions was to select the best one for rapidly and efficiently determining values for X and Y, applicable to all alignments, to best simulate the Valdar score using the Bident score. The linear function was therefore discarded in favour of the powers function.

	Powers Function				**Linear Function**			
	Clustal		**Valdar**		**Clustal**		**Valdar**	
Statistics	X	Y	X	Y	X	Y	X	Y
Average	2.4159	2.0516	1.1643	0.9839	-0.2400	1.0150	0.0310	0.9087
Median	2.3593	2	1.2128	0.9587	-0.2309	1.046	0.0600	0.9087
S.D.	0.7690	0.2084	0.2472	0.0979	0.1654	0.2754	0.0899	0.1288

Table 3.2: Summary of minimisation results for the Powers and Linear functions against the Clustal and Valdar scores (X; Gap weighting, Y; Karlin weighting).

A summary of the minimisation results showing the average, median and standard deviation values are given in Table 3.2. The values of 1.16 and 0.98 for the powers

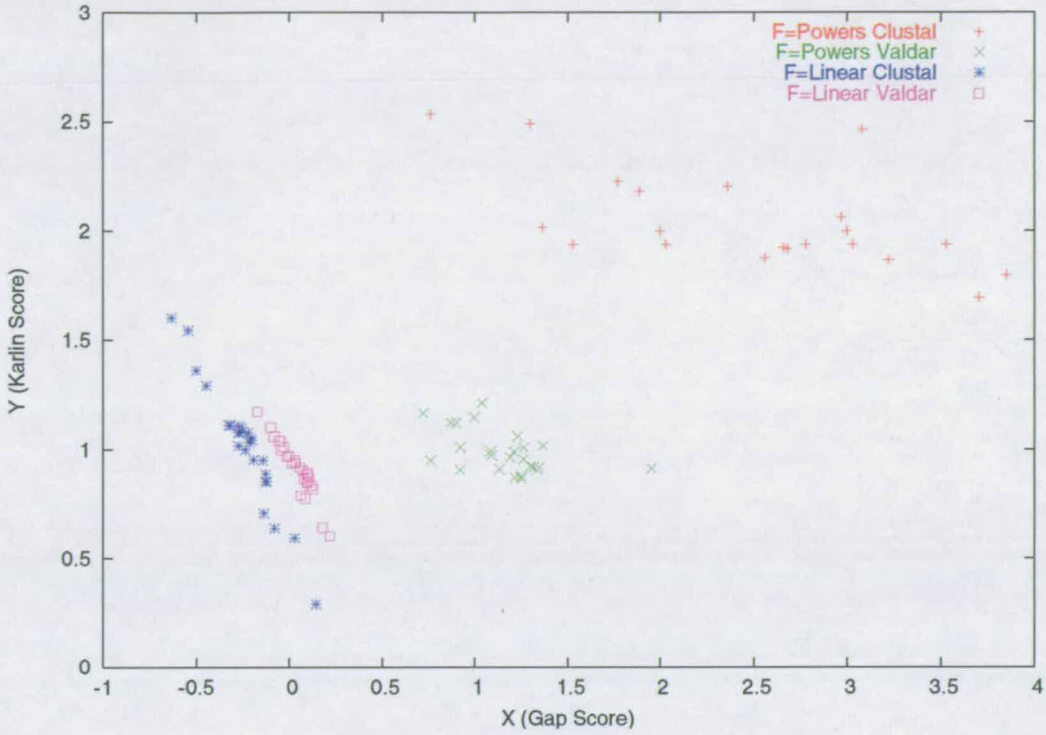


Figure 3.9: Comparing the minimisation results produced by the Powers (+,×) and Linear (∗,□) functions against the Clustal and Valdar conservation scores. (Plotted from Table 3.1)

function were selected for X and Y respectively to best simulate the Valdar score.

Tread Creation

After an optimal multiple sequence alignment has been generated and the conserved functional residues have been extracted using the Bident conservation score then a Tread can be created. A Tread is simply an unordered set of conserved functional residues extracted from a multiple sequence alignment. However is useful to store some additional information along with the actual residues that have been conserved for functional reasons. When the Treads are compared (Section 3.6) only the amino acid type and how many times it occurs are used in the comparison. However the additional information could be used

in more complex comparisons methods (Section 3.6.4). An example of a Tread as stored in the reference Tread database is shown in Table 3.4.

The main reason for storing the additional information is to aid with the process of assigning functional roles to each of the residues in the Tread. Each conserved residue in the Tread can be assigned a functional role, used in conjunction with different weighting schemes (Section 3.6.4) in the Tread comparisons. At the time of the Tread creation a *Tread Report* is automatically created dynamically in LaTeX, which can be viewed in the *Tread View* section of the DAROGAN interface (Section 4.2.4).

3.3 Functional Role Assignment

The simplest method of comparing Treads is to compare the types and the number of times each residue appears in the Treads being compared. This simple method makes the very unlikely assumption that each amino acid in the Tread contributed equally to the function of the enzyme in question. For this reason each of the residues appearing in the Tread can be assigned a functional role summarising the residues contribution to the function of the enzyme. Seven general categories have been defined to cover as best as possible the functional roles a residue may have in an active site. The seven categories were based on a literature study and the practical consideration of keeping the number of categories limited, but sufficiently descriptive. The seven categories are set out in Table 3.3.

ID	Category
1	General substrate binding
2	General cofactor binding
3	Covalent binding of substrate/cofactor
4	General acid/base
5	Positive charge stabilising leaving group
6	Metal binding residue
7	Proton shuttle partners (charge networks)
8	Unknown Function or Non functional role

Table 3.3: Functional Role Categories

The seven categories are not necessarily mutually exclusive as functionally conserved amino acids quite often have dual roles. For this reason functionally conserved residues can be assigned primary, and where appropriate, secondary functional roles. An example for a PLP utilising enzyme is outlined to demonstrate the functional role assignment (Figure 3.10). The enzyme in question is aspartate aminotransferase with EC number 2.6.1.1 (PDB code 1ARS). PLP utilising enzymes are described in more detail in Section 1.6.1.

To infer functional roles for each of the residues in the Tread it is necessary to study the crystal structure of the enzyme, preferably with the cofactor and substrate co-crystallised. This is the reason all reference Treads have been created for enzymes which have had their structure determined experimentally to at least 3Å. By investigating each of the residues in the structure it is possible to assign functional roles based on their positions and potential interactions they may form. In addition to utilising the crystal structures it is possible to infer functional roles from the literature and software such as LIGPLOT (Section (R.A. Wallace & Thornton, 1995).

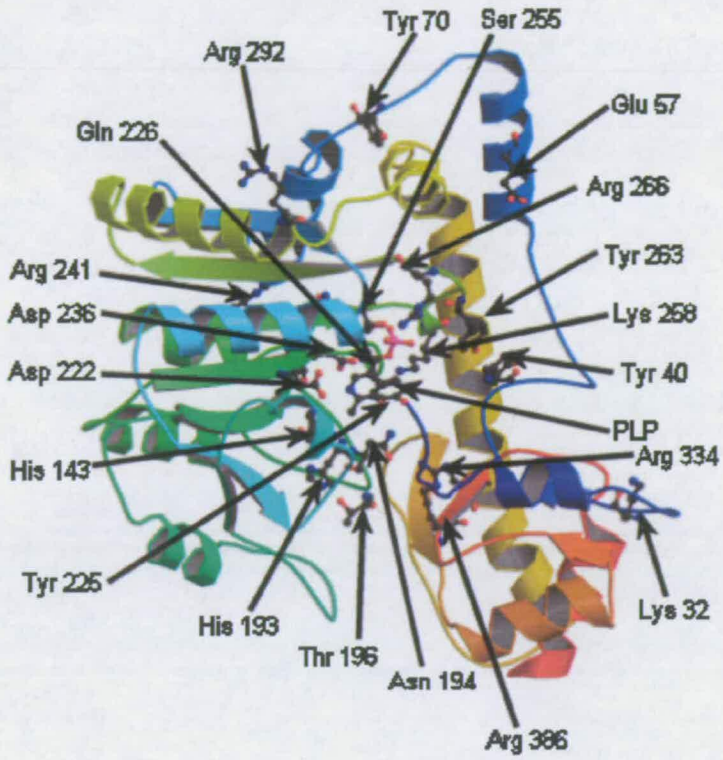


Figure 3.10: Functional Role Assignment of the PLP utilising enzyme Aspartate Aminotransferase (EC 2.6.1.1)(PDB:1ARS). K258 covalently binds the PLP cofactor. H143,H193,N194,D222,Y225,S255,Y263 and R266 all interact directly with the cofactor. Y40,T196 and D236 possibly stabilise other conserved functional residues interacting directly with the cofactor. K32,E57,Y70,Q226.R241,292,R334 and R386 have no obvious functioanl roles, but could have been conserved for structural reasons such as the formmation of salt bridges.

In the example (Table 3.4) each of the fifteen (+ optional five additional) residues in the Tread has been assigned a functional role based on their positions in the crystal structure (Figure 3.10). Assigning functional roles to all the residues in a Tread is a very time consuming process and can take several hours per Tread.

Alignment	Lead	Residue	Score	1	2	3	4	5	6	7	8
79	32	K	100	○	○	○	○	○	○	○	●
87	40	Y	100	○	○	○	○	●	○	○	○
104	57	E	100	○	○	○	○	○	○	○	●
119	70	Y	100	○	○	○	○	○	○	○	●
197	143	H	100	○	●	○	○	○	○	○	○
256	193	H	100	○	●	○	○	○	○	○	○
257	194	N	100	○	●	○	○	○	○	○	○
259	196	T	100	○	○	○	○	●	○	○	○
285	222	D	100	○	●	○	○	○	○	○	○
288	225	Y	100	○	●	○	○	○	○	○	○
289	226	Q	100	○	○	○	○	○	○	○	●
299	236	D	100	○	○	○	○	●	○	○	○
304	241	R	100	○	○	○	○	○	○	○	●
327	255	S	100	○	●	○	○	○	○	○	○
330	258	K	100	○	○	●	○	○	○	○	○
335	263	Y	100	○	●	○	○	○	○	○	○
338	266	R	100	○	●	○	○	○	○	○	○
381	292	R	100	○	○	○	○	○	○	○	●
423	334	R	100	○	○	○	○	○	○	○	●
476	386	R	100	○	○	○	○	○	○	○	●

Table 3.4: Tread for Aspartate Amino Transferase (1ars) EC 2.6.1.1.
Key: Alignment - Alignment Numbering, Lead - Lead Sequence Numbering, Residue - Amino acid appearing at position in Lead Sequence, Score - Conservation Score. Functional role assignments: 1 - General substrate binding, 2 - General Cofactor binding, 3 - Covalent binding of substrate/cofactor, 4 - General acid/base, 5 - Positive charge stabilising leaving group, 6 - Metal binding residue, 7 - Proton shuttle partners (charge network), 8 - Unknown Function or Non functional role.

3.4 User Alignment Quality Analysis

As the query enzyme input is in the form of a multiple sequence alignment pre-calculated by the user performing a function prediction it is prudent to analyse the quality of the alignment. A poor quality alignment will severely affect the quality of the function prediction. The aim of the alignment quality analysis is to provide warnings to the user so that the alignment quality can be improved so a more accurate function prediction can be made. There are two

methods employed to aid visual assessment of alignment quality by the user.

Figure 3.11: Schematic view of the sequences in the alignment for rapid identification of any unusual distribution of gaps or the inclusion of fragment sequences. Sequences are represented as blue lines perforated by gaps.

The first method is to create a schematic view of the sequences in the alignment for rapid identification of any unusual distribution of gaps or the inclusion of fragment sequences (Figure 3.11). Sequences are represented as blue lines perforated where gaps appear, giving a very quick visual indication of any gap anomalies in the alignment.

The second method takes a set of measurements from the alignment and are displayed in two forms, tabular and in the form of a *Quality Dial*. The table

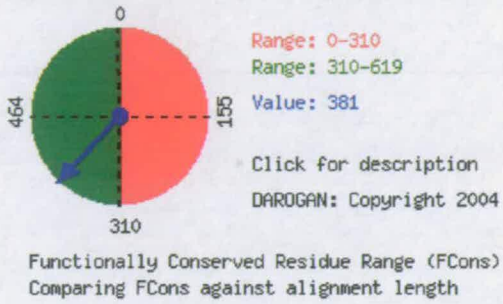


Figure 3.12: Dial: Alignment visual quality assessment. The Dial shows, via the blue needle pointing to the red on the dial face, whether there are any possible anomalies in the alignment. The quantity being measured is described underneath the Dial. If the needle is in the green the alignment is considered to be within acceptable range for the measurement.

provides raw data for the measurements. The tabular output provides users experienced in working with alignments with a means of detecting potential problems with the alignment. The dials were developed to give an at-a-glance method of determining potential problems with the alignment (Figure 3.12). The dials are analogous to a pressure gauge, where a needle points to a value on the dial. The value on the dial is coloured red if it is deemed to be outside acceptable range and quickly highlights a potential problem with the alignment. There are several dials created for each alignment and are summarised in Table 3.5.

Each of the alignment quality assessments are also preformed on each of the alignments used to create the reference Treads. The results of these analyses can be viewed in the *Tread View* section of the DAROGAN interface (Section 4.2.4).

Dial No.	Measurement Description	Threshold for Warning
1	Comparing functionally conserved residue range against alignment length.	Range of first and last Functionally conserved residues must be over half the entire alignment length
2	Comparing shortest against longest sequence length	The longest sequence length must not be more than two times the shortest sequence length
3	Comparing standard deviation against average sequence length	Standard deviation of sequence lengths must not be more than half the average sequence length
4	Comparing average sequence length against alignment length	Alignment length must not be more than two and a half times the average sequence length

Table 3.5: Dial: Alignment visual quality assessment. There are four Dials in the current release of DAROGAN, each measurement taken is described.

3.4.1 Cysteine Residue Warnings

The selection of the functionally conserved functional residues (KRENDY-CHQST) ensures that all the residues from an enzyme that are capable of participating in the function of the enzyme are included. However the inclusion of cysteine residues in the set of functionally conserved residues has drawbacks as cysteine can also be involved in disulphide bond formation. Disulphide bonds are covalent, but reversible, and make important contributions to the stability of the structure of the protein. $-SH$ groups from two cysteine residues are oxidised to form one $-S-S-$ (disulphide) bond. Disulphide bonds are notably found in the IgG class of immunoglobulins forming the hing region in the heavy chain and joining the light chain to the heavy chain.

To take into account this potential additional role of cysteine residues, warnings are given if there is more than one cysteine residue appearing in a Tread. As conserved pairs of cysteins are candidates for disulphide bond formation (provided

they are located close together in the three dimensional structure of the protein).

3.5 Query Tread Creation

A query Tread is created in a similar fashion to a reference Tread, however the process is somewhat simplified. The input of a query enzyme sequence, for which a function prediction is to be performed, is in the form of a multiple sequence alignment. The reason for the query enzyme being submitted as an alignment with similar sequences is due to the prohibitive computational power required to calculate the alignment. Reference Treads are preprocessed and stored in the Tread database so require little computational power to prepare them for a comparison to the query Tread. In the case of the query enzyme a BLAST search must be performed to find similar sequences, then aligned before a comparison can be made. It is therefore expected the query enzyme will be presented in the form of a pre-calculated multiple sequence alignment before a function prediction can be made. Future versions of DAROGAN may include the option to submit the query enzyme as a sequence (Section 7.4.1).

3.6 TREAD Comparisons

With a set of reference Treads in the database, it is now possible match a query Tread to the reference Treads. Functional information for the query Tread is inferred from the best matching reference Treads. Treads comprise of the number of times the fifteen conserved functional residues (KRENDYCHQST) occur in a

multiple sequence alignment. This information is best represented as a vector with the dimensions of the vector being the types of conserved functional residues. The frequency of each of the conserved functional residues is entered in their respective dimensions. Once the Treads have been represented as vectors they can then be compared using an adaptation of the *Vector Space Model.*

3.6.1 Vector Space Model

The *Vector Space Model* (VSM) was developed in the field of Information Retrieval by Gerard Salton at Cornell University during the 1960's (Salton, 1970). Today the most common application of the VSM is in document search engines, where a query term is matched against a set of documents to find the most similar documents. This is most commonly seen in Internet search engines, although the more widely used (and more recent) search engines such as Google (*http://www.google.co.uk*) do not use the VSM.

A vector is a mathematical term to describe a quantity having both magnitude and direction. For example the velocity of a moving ball is specified by the speed (magnitude) and the direction that the ball is travelling. The magnitude of a vector is the absolute value of the length of the vector and the direction of a vector is given by an angle from a reference line (e.g. the ground).

In the field of Information Retrieval search terms are searched for in a set of documents, both of which are represented as vectors. The magnitude of the document vector is determined by how many times each term occurs and direction

determined by the number of terms appearing in the document (See Figure 3.13).

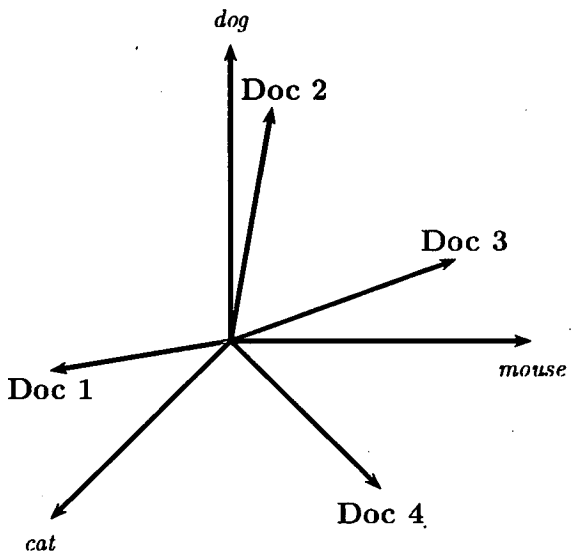


Figure 3.13: Four text documents (Doc1, Doc2, Doc3, Doc4) are compared by the occurrence of three different terms (dog, mouse, cat) to illustrate the Vector Space Model. Document 1 (Doc1) contains the terms *dog* and *cat*, but not *mouse* and is represented as a vector. Dimensions of the vector are for the presence or absence of the terms; positive scores for term occurrence. By representing the documents and the terms they contain as vectors, similarities between the vectors can be calculated, to find the most similar documents.

The VSM eliminates the order of terms in a document, in fact in natural language the order of the terms indicates very little about the content of the document. The document vectors contain many zero values so the document vector is sparse meaning the VSM can be scaled up to many terms without having the requirement for huge amounts of memory or disk access. The method of the VSM can be divided up into three distinct stages. Document indexing, Term Weighting and the determination of similarity between vectors.

The vector space model has the advantage over other information retrieval techniques such as reverse index look up tables in that no memory storage is necessary. Therefore all the searches take place in RAM with no disk or database access. Another advantage of the VSM is that queries can be of any length and there is no need for the use of regular expressions or boolean logic in the search terms.

Problems with the VSM are mainly concerned with high dimensionality of the search space. This means it is impossible to visualise the space, however the application of dimensionality reduction techniques such as *Principal Component Analysis* (Greenacre, 1984), allow the dimensionality of the data to be reduced to two or three dimensions. This dimensionality reduction is without loss of the overall trends in the data, allowing them to be easily visualised. However similarity is still based on the number of common terms between vectors beyond three dimensions (See Section 7.4.2).

3.6.2 Vector Form of Treads

Treads are converted into vectors with twenty dimensions, one for each of the amino acids. There are only eleven functionally conserved residues, so nine of the dimensions are redundant (Figure 3.14). The extra dimensions were included to allow for future expansion/refinement of the function prediction method to include more than the current set of eleven functional amino acids. The empty dimensions do not adversely affect the computational time required to perform

the vector comparisons.

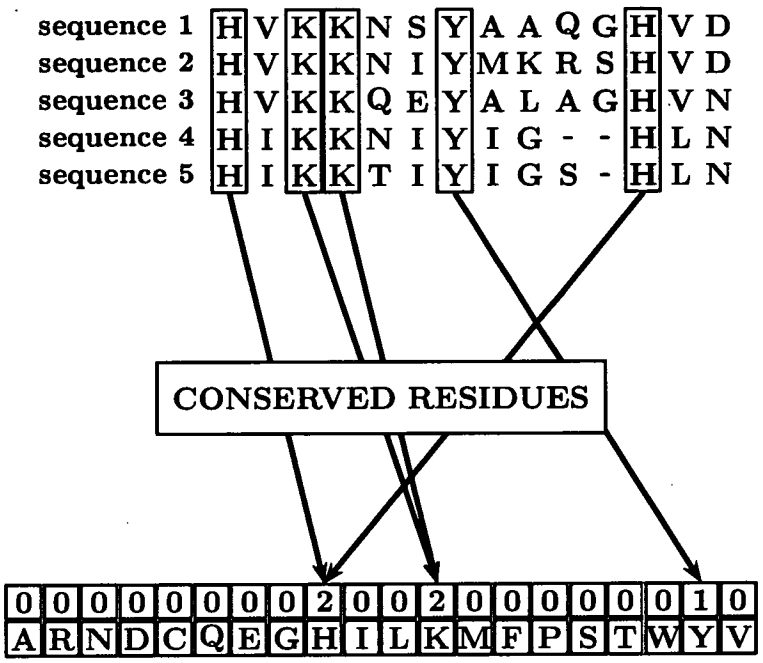


Figure 3.14: Vector Representation of Treads. Conserved functional residues (KRENDYCHQST) from a multiple sequence alignment are extracted using the custom conservation score (Section 3.2.4). Once extracted the residues are represented in vector form. Each of the different amino acids is represented as a dimension in the vector. For instance the two conserved histidines in the alignment are represented as 2 in the histidine dimension of the vector.

3.6.3 Tread Similarity

Now that the Treads are expressed in vector form the next stage is to quantify the similarity between Treads. There are a wide variety of similarity measures, or similarity metrics, available to calculate the similarity between vectors. The most commonly used metric in the VSM is the Cosine similarity measure and is used as the default method in DAROGAN. Five additional metrics have been included as alternatives to the Cosine measure; the definitions of each of the

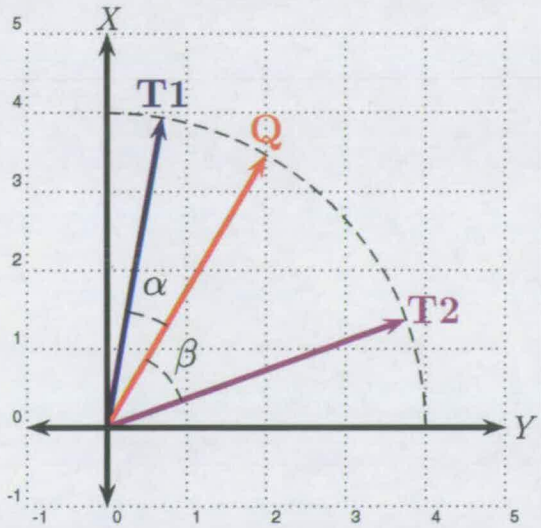
metrics can be found in Table 3.6.

Similarity Metric	Definition
Cosine	$cos(q,r) = \frac{\sum_{i=1}^{d} q_i r_i}{\sqrt{\sum (q_i)^2 \sum (r_i)^2}}$
Manhattan Distance (L_1)	$man(q,r) = \sum_{i=1}^{d} \mid q_i - r_i \mid$
Euclidean Distance (L_2)	$euc(q,r) = \sqrt{\sum_{i=1}^{d} (q_i - r_i)^2}$
Canberra	$can(q,r) = \sum_{i=1}^{d} \frac{\lvert q_i - r_i \rvert}{\lvert q \rvert \lvert r \rvert}$
Dice	$dic(q,r) = \frac{\text{2(common q,r)}}{\text{non zero q + non zero r}}$
Jaccard	$jac(q,r) = \frac{\text{common terms}}{\text{non zero q + non zero r - common q,r}}$
Minowski (L)	$min(q,r) = \sqrt[\lambda]{\sum_{i=1}^{d} \mid q_i - r_i \mid^{\lambda}}$

Table 3.6: Six commonly used similarity metrics for vectors. q referes to the query Tread, and r the refernence Tread; $\mid q \mid$ indicates the magnitude of vector q. The Manhattan, Euclidean and Chebychev metrics are part of the Minowski family of metrics. The value of λ determined which measure is calculated. $\lambda = 1$ gives the Manhattan Distance, $\lambda = 2$ gives the Euclidean Distance and $\lambda = \infty$ gives the Chebychev Distance (not discussed here).

The Cosine metric as the name suggests involves calculating the cosine angle between the two vectors. The larger the angle the more dissimilar the vectors. A simplified worked example is outlined in Figure 3.15. An additional step of first normalising the vectors is included in the DAROGAN method.

The five additional metrics were included to provide alternatives to the Cosine similarity measure and to investigate the most appropriate metric to use in the



T1 = (0.69,3.94)
T2 = (3.75,1.37)
Q = (2.00,3.46)

$$sim(Q, T1) = \cos\alpha = 0.94$$
$$= \frac{(2.00*0.69)+(3.46*3.94)}{\sqrt{[(2.00)^2+(3.46)^2]*[(0.69)^2+(3.94)^2]}}$$

$$sim(Q, T2) = \cos\beta = 0.77$$
$$= \frac{(2.00*3.75)+(3.46*1.37)}{\sqrt{[(2.00)^2+(3.46)^2]*[(3.75)^2+(1.37)^2]}}$$

Figure 3.15: Worked example of cosine similarity measure. The cosine similarity is derived from cartesian coordinates of the vectors (T1, T2, Q), used to calculate the angles between the reference vectors (T1 and T2) and the query vector (Q). The cosine of these angles gives the similarity between the vectors.

calculation of Tread similarity. A comparison of the different metrics is provided in Chapter 5. The six metrics are the Cosine, Manhattan, Euclidean, Canberra, Dice and Jaccard Similarity metrics.

The Manhattan distance, also known as the city block or L1 metric, is a member of the Minowski (L) metric family along with the Euclidean distance. The Manhattan distance is named after the typical city blocks found in Manhattan, so the distance in analogous to a taxi driving between opposite corners of a city block. The Euclidean distance is similar the the Cosine similarity measure, however the direct euclidean distance is measured rather than the cosine angle. The Canberra metric is also similar to the cosine and Minowski metrics, but is especially sensitive to small changes in the vector values near zero.

The Dice and Jaccard metrics are also considered to be similar to the metrics mentioned above, however are more orientated to measure highly similar vector values. The Jaccard metric (Jaccard, 1912) in particular penalises a small number of shared vector entries between the vectors being compared. The Jaccard metric is also referred to as the Tanimoto coefficient in the literature.

3.6.4 Weighting Schemes

The method of comparing a query Tread to the reference Treads has been discussed above, however modifications to the data stored in a Tread can be made (in future releases of DAROGAN). The modifications are designed to provide weightings to each of the residues appearing in a Tread. The weightings are implemented in several different combinations (Section 4.2.2).

The first scheme takes into account the functional role assignments made for each of the residues appearing in the Tread (Section 3.3). If a residue has been assigned a defined functional role this gains a weighting above that of residues assigned a functional role of *unknown*. The magnitude of the actual weighting can be tailored by the user. The second weighting scheme utilises the conservation score assigned to each of the conserved residues (Section 3.2.4). In a Tread the number of residues extracted from the multiple sequence alignment is always fifteen, however all of these are not necessarily 100% conserved. If fifteen conserved functional residues appear in the alignment then the next five highest scoring residues are also extracted, to take the total number of residues in the Treads to twenty. The

conservation score weighting scheme weights each of the residues in the Tread by its conservation score. The third scheme combines the functional role and conservation score schemes. The default available weighting scheme is that of applying no weightings to the individual residues in the Treads.

3.7 Statistical Significance of Results

When a query Tread is matched against the Treads in the reference Tread database each match is given a score quantifying how similar the Treads are. This measure does not give any indication whether the match is due to genuine functional relatedness or whether the match is down to chance alone. By calculating the statistical significance of each of the matches it is possible to estimate the probability of the matches being down to chance alone. The statistical significance of pairwise sequence alignments was discussed in Section 2.3, and the calculation of a statistical significance score for Tread comparisons is discussed below.

The text below details the steps involved in calculating the statistical significance of the Cosine similarity measure scores. Details for the other vector similarity metrics are not discussed in detail here, however a summary is provided in Appendix A.

3.7.1 Frequency Distribution

The basis of the significance score is to determine whether a match against the reference Tread database is likely to have occurred by chance. In order to model *chance* 10,000 randomly generated Treads were matched against the Tread database. Each random Tread was matched against each reference Tread and the highest relevance score was recorded. The number of random Treads generated was fairly arbitrary and was selected as a compromise between a large number, for good coverage of random Treads, and the computational time required to generate random Treads and perform matches.

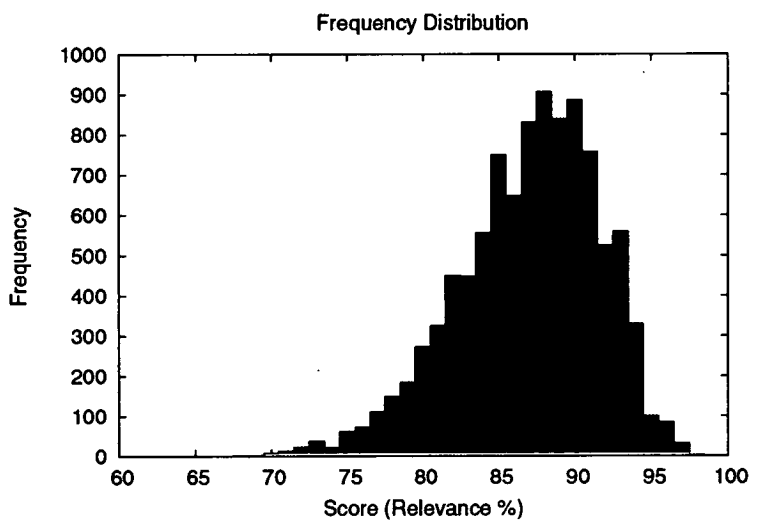


Figure 3.16: The heights of the bars in the histogram are proportional to the frequency of the occurrence of the corresponding similarity score range. The mean for the data is 87.49.

As the relevance score can be any value between 0 and 100%, a continuous random variable, can be summarised with a *frequency distribution* (Figure 3.16). The raw data are condensed into score ranges so any underlying patterns can easily be

identified. The main purpose of plotting the frequency distribution is to determine which, if any, distribution best approximates the data.

3.7.2 Determining the Distribution

One candidate for the distribution that best approximates the data is the *Normal Distribution.* If a set of data are said to be Normally distributed this means a frequency plot would adopt the classical bell shaped curve of the normal distribution. The curve is symmetrical, all frequency values are positive and the area under the curve is finite. A normally distributed set of data can be transformed to a Standard normal probability function (continuous probability function), where the mean of the distribution is zero and the variance one. From this Standard distribution it is then possible to calculate the probability of a Tread matching with a score above a specified value i.e a statistical significance score.

However before the significance of a score can be calculated, first it is necessary to determine whether the data is well approximated by a Normal distribution. This is achieved by plotting the data against an example of a Normal distribution. A linear relationship between the data and the Normal distribution indicates a good approximation. Figure 3.17 shows the data from the randomised Treads plotted against a Normal distribution, and the deviation of the data around the tails indicates that the Normal distribution is not a suitable approximation for the data. In addition to the plot the Anderson-Darling (Stephens, 1974) statistic, was calculated to be 52.008, confirming a poor fit. A good fit would be

somewhere below a value of 15. The Anderson-Darling statistic is considered to be distribution free, meaning their is no dependence on the distribution being compared to.

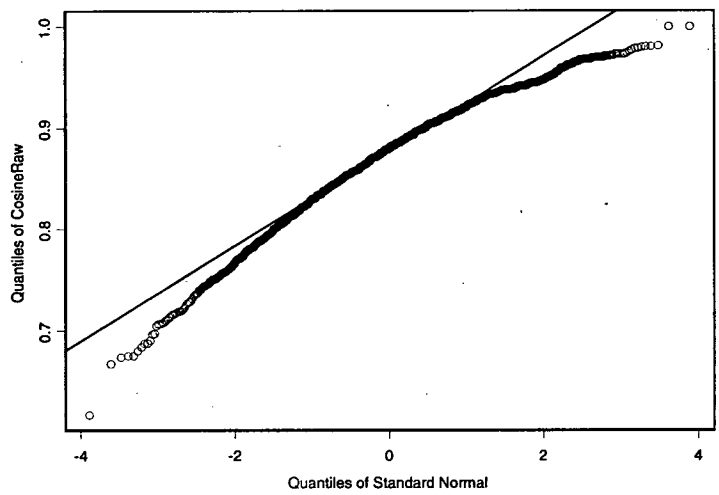


Figure 3.17: qqPlot showing the quantiles for the randon Tread comparison data plotted against quantiles for a standard normal distribution. Significant deviation from a linear relationship indicated the standard normal distribution would not be a good approximation for the Tread data. S-Plus commands used to generate this plot can be found in Appendix A

With the Normal distribution discounted it is necessary to look for another distribution likely to give a good approximation to the data. The *General Extreme Value* (GEV) distribution is a promising option. If the maximum value is always taken from a set of observations, as with each random Tread, this is likely to produce a distribution best approximated by the GEV distribution. In the case of the GEV is the tails of the distribution that are used to calculate the probability a score will be above a specified value.

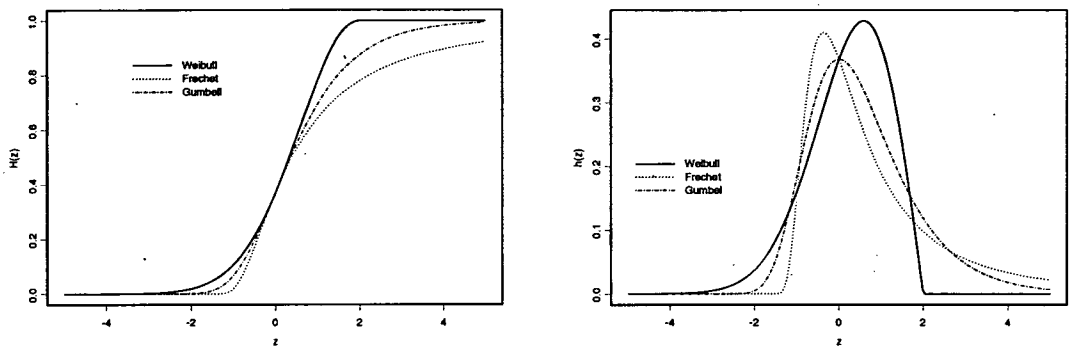
The GEV distribution is described to have *leptokurtosis*, referring to the distribution having *heavier* tail than the Normal distribution. The Fischer-Tippet theorem (Fischer & Tippet, 1928) states that if the Z value (Equation 3.5) converges to some non-degenerate distribution then it must be a GEV distribution of one of three forms, determined by the shape parameter[1] ξ (See Section 3.7.3). Type I is the *Gumbel* where $\xi = 0$ and the tail declines exponentially. The Gumbel is often referred to as being thin tailed. Type II is the *Fréchet* where $\xi > 0$[2] and the tail declines more slowly; described to be fat tailed. Type III is the *Weibull* where $\xi < 0$ and the tail is finite.

$$Z_n = \frac{M_n - \mu_n}{\sigma_n} \tag{3.5}$$

Figure 3.18 show examples of the three types of the GEV distribution. Figure 3.18A shows the cumulative distribution functions (CDF) for the three types and Figure 3.18B show the probability distribution functions (PDF). The differences in the tails mentioned above can clearly be seen in the CDF and PDF plots.

To determine if the data are well approximated by the GEV it is necessary to plot a *QQPlot of Residuals.* In this plot an example of the exponential distribution is used as a reference against the data. A linear relationship is a positive indication of a good fit by the GEV distribution. Figure 3.19 shows the random Tread data against an exponential distribution, indicating a good fit for the GEV. The data deviates slightly from the exponential distribution at the tail, however this is still

[1]It is worth noting that the mean, μ, is referred to as the location parameter of a distribution and the standard deviation, σ, as the scale parameter



A B

Figure 3.18: Generalised Extreme Value Distributions (Cumulative Distribution Functions and Probability Distribution Functions) for H_ξ for Fréchet (ξ = 0.5), Weibull (ξ = -0.5) and Gumbell (ξ = 0). S-Plus commands used to generate the example distributions can be found in Appendix A

within a reasonable level and still confirms a good fit for the GEV distribution.

3.7.3 Calculating Significance using GEV Theory

The GEV distribution is defined by three parameters, ξ, μ, σ, (the shape, location and scale parameters) which are estimated using parametric maximum likelihood estimation (MLE). The S-Plus statistics software package (S-Plus, 2005) and the S+FinMetrics/EVIS (Zivot & Wang, 2003) module fit the GEV distribution to data by MLE. The details of MLE will not be discussed here, but a good description can be found in the book accompanying the S+FinMetrics module (Zivot & Wang, 2003). The S-Plus commands used to fit the GEV to the randomised Tread data can be found in Appendix A.

$$\Phi(z) = Pr(M_n \leq z) = H_\xi(z) = e^{-(1+\xi z)^{\frac{-1}{\xi}}} \tag{3.6}$$

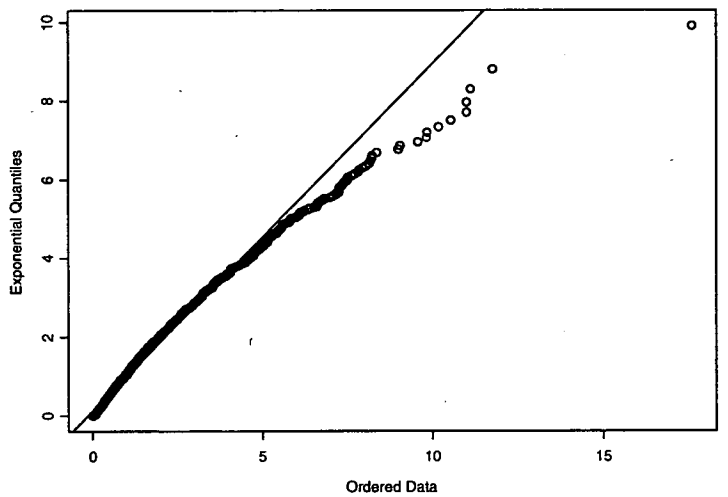


Figure 3.19: QQPlot showing the quantiles for the random Tread comparison data plotted against quantiles for an exponential distribution. The linear relationship indicates a good approximation to the GEV; as the exponential distribution itself is a good aproximaton to the GEV. S-Plus commands used to generate this plot can be found in Appendix A

$$\phi(z) = h_\xi(z) = \frac{1}{\sigma}(1+\xi z)^{\frac{-1-1}{\xi}} e^{-(1+\xi z)^{\frac{-1}{\xi}}} \tag{3.7}$$

Once calculated the three parameters (ξ, μ, σ) can be used in the CDF (Equation 3.6) and the PDF (Equation 3.7) equations. The plots of the CDF for the random Tread data can be seen in Figure 3.20 and the PDF in Figure 3.21.

The ξ parameter for the random Tread data was estimated to be -0.35, indicating the GEV distribution is Type III or Weibull. The CDF and PDF plots also match well to the example of a Weibull distribution in Figure 3.18.

The statistical significance score is calculated using the PDF. The area under the

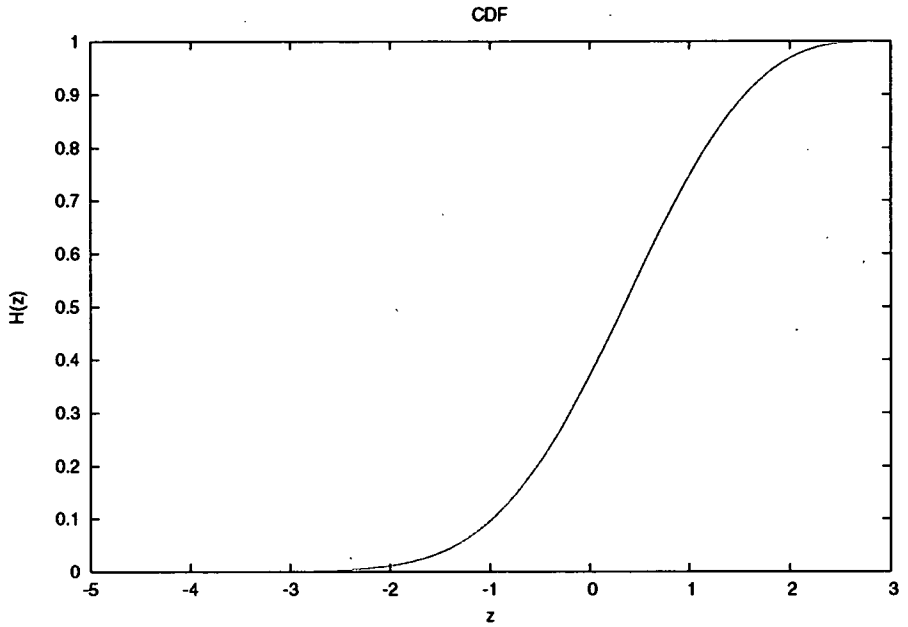


Figure 3.20: Cumulative Distribution Function (CDF)

entire curve represents the probability of all scores occurring. So by calculating the area under the curve to the left of a specific score $(1 - h_\xi(z))$ gives the probability of a higher score being achieved by chance. This score is called the probability value or P-Value.

However a statistical significance score becomes more useful when the size of the reference Tread database is taken into consideration. This is an expectation value or E-Value and is calculated by multiplying the P-Value by the number of reference Treads in the database.

3.7.4 Significance Levels

Even with the calculation of a statistical significance score this is insufficient to assess the quality of a Tread match. To test the hypothesis that a match

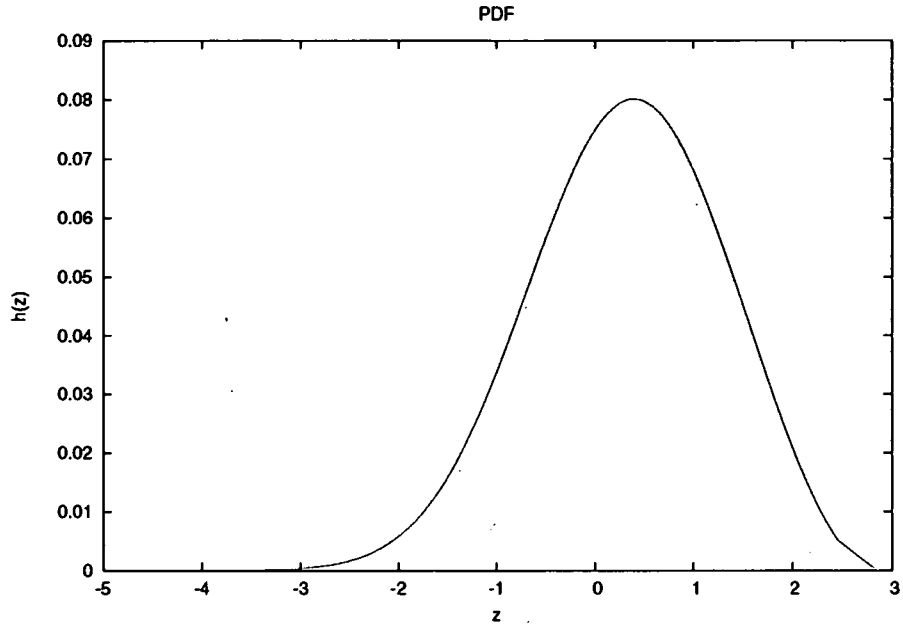


Figure 3.21: Probability Distribution Function (PDF)

could have occurred by chance a significance level must be defined. A commonly accepted standard is a 5% significance level, which accepts matches as being significant when they have a less than 5% chance of occuring by chance alone.

The DAROGAN method allows the user to select a significance level from a set of options (Section 4.2.2). A 1% significance level is the default option, this is more stringent than the commonly accepted 5% level to ensure the chances of random matches are kept to a minimum. To simplify the assessment of the significance levels the DAROGAN interface colours the matches by significance and assigns a qualitative assessment to the levels (Table 3.7).

P(Random Match)	P(Genuine Match)	Qualitative Assessment
1%	99%	Excellent
5%	95%	Good
10%	90%	Poor
10%	90%	None

Table 3.7: Statistical Significance Levels

3.8 Cofactor Prediction

Two methods are employed to try and predict the cofactor utilised, if any, by the query enzyme. In the current version of DAROGAN this feature is still in the development stage, so is provided as an option rather than as an incorporated part of the main function prediction method.

The first method is based on the concept of *General* Treads. All the reference Treads for enzymes utilising a particular cofactor are amalgamated into one *General* Tread for that cofactor. The result is a set of *General* Treads, one per cofactor. A comparison using the same measure for comparing the query Tread to reference Treads is utilised for comparing the query Tread to the *General* Treads. A relevance score is given for matching each of the *General* Treads to the query Tread, the higher the score the more similar the Treads. No significance score is assigned to the matches due to the limited number of *General* Treads.

The second method simply takes the matches between the query Tread and the reference Treads at a specified significance threshold (Section 3.7.4) and calculates the proportions of cofactor utilisation in the matches. The results are presented as percentages of the top matches utilising particular cofactors.

3.9 Concluding Remarks

This chapter has presented a novel method for enzyme function prediction utilising conserved functional residues, Treads, extracted from multiple sequence alignments. Details for creating optimal sequence alignments with the required fifteen conserved residues by tailoring E-values cut offs in BLAST searches and the creation of the actual Treads are presented. From the creation of the Treads the method employed to make comparisons between the Treads and their statistical significance are also presented.

An in-depth evaluation of the method is provided in Chapter 5. In the chapter the results of utilising each of the different Tread scoring methods as well as the significance levels selected are investigated. The proceeding chapter details the actual computational implementation of the DAROGAN method.

Chapter 4

DAROGAN Implementation

4.1 Overview

This chapter discusses the computational implementation of the function prediction method as outlined in Chapter 3. DAROGAN is the name given to the suite of software developed as part of this project and is freely accessible over the Internet (www.darogan.co.uk).

darogan*("d'rogan") [da-RO-gan-(DRO-gan)](verb)(North Wales) to predict*

DAROGAN was designed to develop and test the function prediction method, so has been coded to allow changes to the method to be easily implemented as it was developed. The DAROGAN database currently contains only a small subset of all known enzymes, however it has been made freely available for researchers to perform function predictions within this small subset.

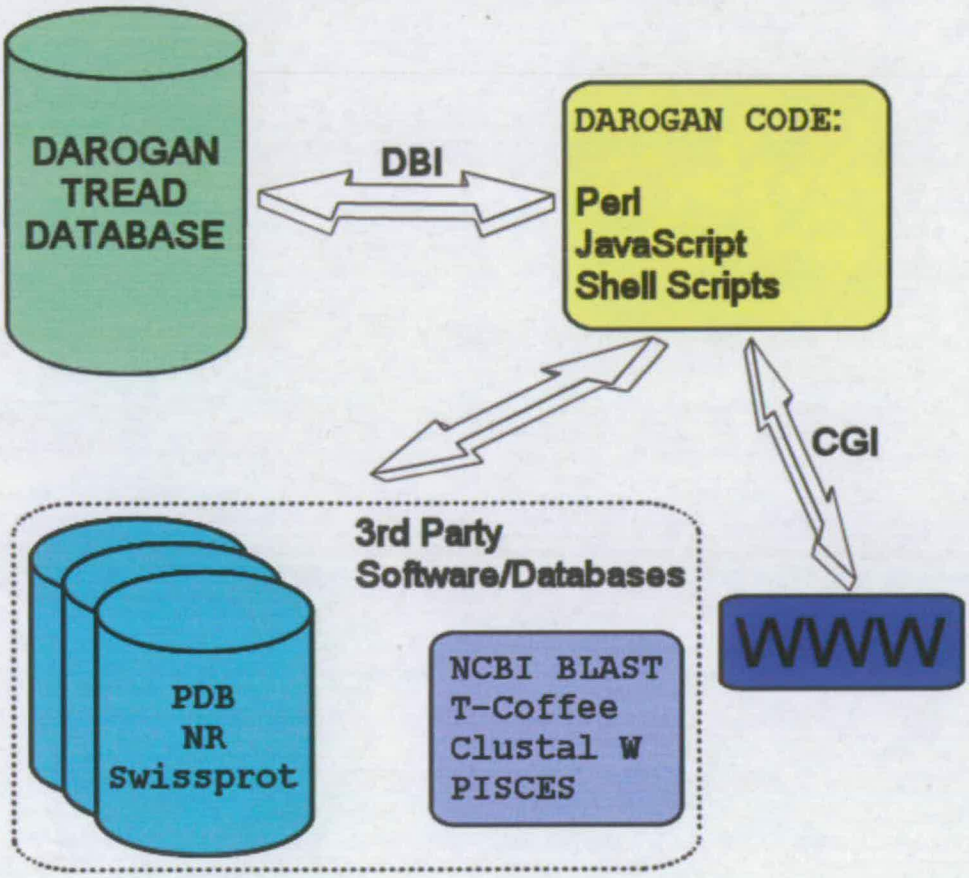


Figure 4.1: A schematic highlighting the different components of the DAROGAN software package. The Tread database is connected to the main code via the Perl DBI (Perl Module: DBI, 2005). Third party software and databases are also accessed by the main DAROGAN code. The function prediction service is accessed over the Internet though CGI (Perl Module: CGI, 2005).

The core of DAROGAN (Figure 4.1) consists of the Reference Tread Database (Section 4.3), which stores the actual Treads as well as a wealth of additional information relating to each of the Treads. This is a relational database utilising the MySQL Relational Database Management System (RDMS)(Section 4.3.1). The DAROGAN interface (Section 4.2) is divided into three sections (Input, Results and Tread viewing) and is accessed over the Internet with the help of commonly available Internet browsing software. The code for the actual function

prediction method is written almost entirely in Perl (Section 4.4.1) and made freely available as a web service through the use of the Common Gateway Interface (CGI)(Section 4.4.2)(Perl Module: CGI, 2005). JavaScript (Section 4.4.4) is utilised to handle user input errors, ensuring all input is validated before being sent to the DAROGAN server. In addition to the code for the function prediction dynamic images are created *on-the-fly* for the display of the function prediction results and the viewing of the actual Reference Treads (Section 4.4.3).

4.2 DAROGAN User Interface

The DAROGAN server is accessed over the Internet and has been tested on all the most popular Internet browsers (Mozilla, Netscape and Internet Explorer). The web address for DAROGAN is:

`http://www.darogan.co.uk`

During the development of DAROGAN importance was placed on the aesthetics and usability of the interface. The typical user was assumed to be a bench biologist with a modest level of computer experience. Navigation through the interface, during the course of performing a function prediction, has been designed to be as intuitive as possible; a User Guide and help pages are provided on the web site. The layout of the various pages of the interface are presented in a clear and logical manner, to ensure a user can easily locate required input fields and understand the results being presented. Future improvements to the interface are discussed in Section 7.

4.2.1 Preliminary Requirements

The first step in performing a function prediction on a query enzyme using DAROGAN is to conduct a sequence database search (e.g. BLAST) to retrieve a set of similar sequences to the query sequence. This set of sequences, including the query sequence, should then be aligned using a multiple sequence alignment program (e.g. ClustalW, T-Coffee or MUSCLE). Ideally there should be approximately 15 functionally conserved residues (KRENDYCHQST) in the resulting alignment. This can be achieved by altering the E-value threshold in the BLAST search, which has the effect of increasing the number of functionally conserved residues as the E-value decreases. Once a multiple sequence alignment has been created then a function prediction can be made using DAROGAN (N.B. Multiple sequence alignments must be provided in Clustal format).

4.2.2 User Input Page

The first page viewed when utilising DAROGAN is the user input page (Figure 4.2). This page provides a form for the user to input all the required data to perform a function prediction. The main data to be inputted is the users multiple sequence alignment for the query enzyme, uploaded from the users local file system to the DAROGAN server. Once a prediction has been made this file is deleted and not used for any purpose other than the function prediction so confidentiality of the input data is ensured. In addition to the alignment file it is necessary to input the sequence identifier for the query enzyme in the alignment. This should appear exactly as it does in the input alignment file.

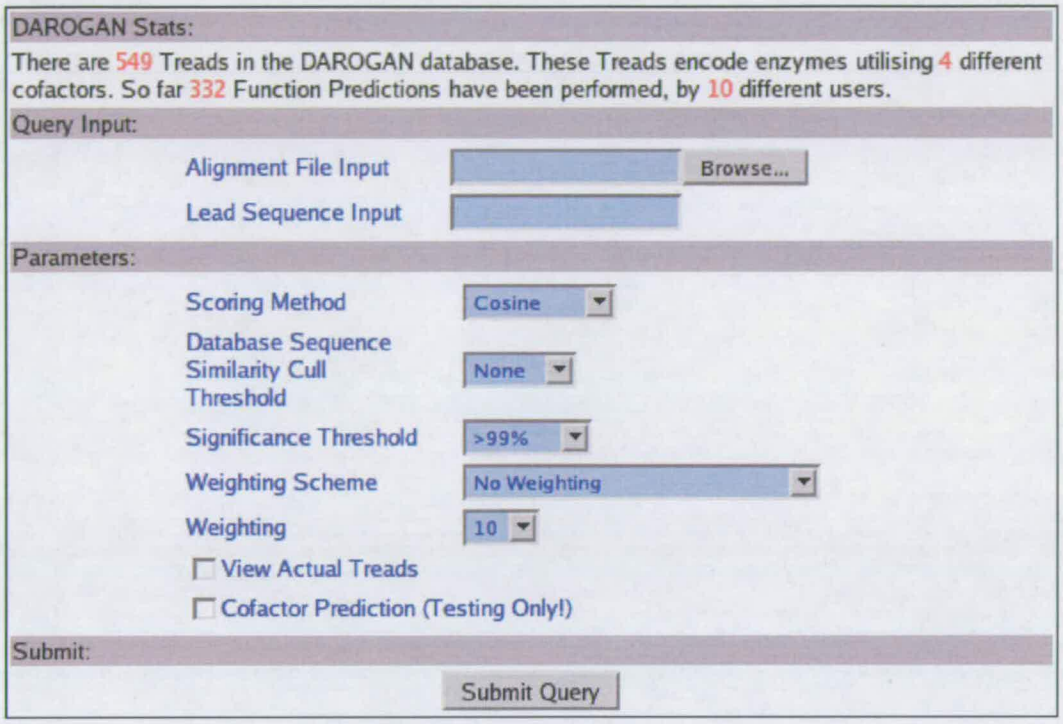


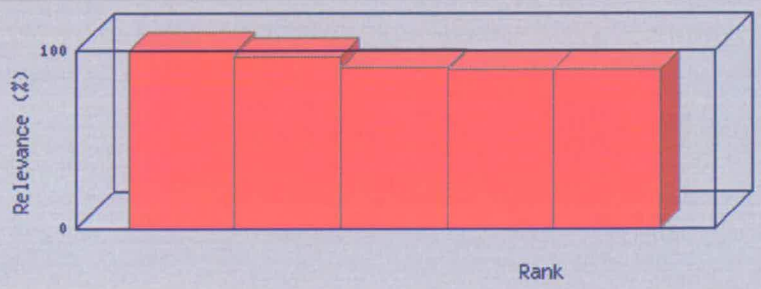
Figure 4.2: The input page is the front end of the DAROGAN interface. A summary of the number of Treads currently in the Reference Tread database is shown. The user submits a multiple sequence alignment containing their query putative enzyme along with a small number of parameters which can be altered from the recommended default parameter settings.

Once the users data has been entered several customisable parameters are available to the user. In most cases, however, the default settings will be sufficient to perform a prediction. The Significance Threshold parameter determines the significance cut off for the results returned (Section 3.7). A 99% Threshold means that there is a 99% chance that the Tread match is genuine rather than a random match from the database. The user has the ability to customise the weighting scheme (Section 3.6.4) for the scoring of the Tread matches and the magnitude of the weightings used in the weighting scheme. Parameters are also provided to tailor the level of sequence culling preformed on the Reference Tread database

(Section 5.2.2). The View the Actual Tread option will toggle the display the raw reference Treads returned as matches at the chosen Significance Threshold level, this option is primarily for debugging but maybe of interest to expert users. The final option is to toggle the Cofactor Prediction facility (Section 3.8), currently in development stage. Once the input form has been completed the data can be submitted to the DAROGAN server to perform a prediction.

4.2.3 Results Page

Function Prediction Results:



Rank	Tread ID	EC	Enzyme	DB	Relevance	P-Value
1	1BJ40451T	2.1.2.1	SERINE HYDROXY...	SP	100	0.000e+00
2	1EQB0401T	2.1.2.1	SERINE HYDROXY...	SP	97	0.000e+00
3	1L8A81T	1.2.4.1	PYRUVATEDEHYDR...	SP	91	6.661e-16
4	1QIP0352T	4.4.1.5	LACTOYLGLUTATH...	NR	90	6.944e-10
5	1BOP1402T	1.2.7.1	PYRUVATE-FERRE...	NR	90	6.944e-10

Significance	99%	95%	90%	<90%
Significance Rating	Excellent	Good	Poor	None

Figure 4.3: The main section (1 of 2) on the results page of the DAROGAN interface. Matching Treads are ranked in order of their Relevance score for the match. The P-Value gives an indication of the statistical significance of the match and are coloured according to their significance rating (Section 3.7). Individual Tread IDs link directly to a detailed Tread Report.

The results of the function prediction are displayed as a list of the best matching reference Treads, within the user defined significance threshold (Figure 4.3) are

ordered by Relevance score. The unique identification code for the Tread is given, which can be clicked to access further information for the Tread (Section 4.2.4). The EC number of the Tread is displayed, and links directly to the ExPASy (ExPASy, 2005) web site entry containing detailed information relating to the EC number. The next piece of information given is the common name for the enzyme the Tread describes. This is followed by the identity of the database from which the sequences used in the creating of the Tread were derived. Lastly the Relevance score (Section 3.6.3) and the E-Value (Section 3.7) relate how well each tread matches the query sequence. If the *View Actual Treads* and/or *Cofactor Prediction* parameters were selected then the data for these options is displayed (Figure 4.4).

In addition to the actual function prediction results an assessment is made regarding the quality of the users multiple sequence alignment (Section 3.4). The results of the function prediction are entirely reliant on the quality of the users alignment, so guides to possible deficiencies in the alignment are given via three separate means. The first is through a numerical view of the measurements made on the users alignment. The second method is to display a more visually appealing representation of the measurements in the form of dials, these dials are analogous to pressure gauges where red signals a warning. The third method is a schematic view of the sequences appearing in the alignment. Sequences are displayed in blue inter-dispersed with gaps with the aim to highlight any unusual distribution of gaps in the alignment.

Cofactor Prediction (Testing Only!):

Method 1:

General Cofactor Tread	Relevance
PLP	91
THIAMIN	80
GLUTATHIONE	82

[How is this calculated?]

Method 2:

There is a 100% chance the input enzyme utilises **PLP** as a cofactor

A [How is this calculated?]

Actual Treads:

AA	[C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W]	
Query:	[0	2	4	0	0	0	3	2	2	0	0	3	1	0	0	0	0	0	3	0]	
Ref: 1	[0	2	4	0	0	0	3	2	2	0	0	3	1	0	0	0	0	0	3	0]	100
Ref: 2	[0	2	3	0	0	0	4	3	2	0	0	3	1	0	0	0	0	0	2	0]	96
Ref: 3	[0	1	4	0	0	0	3	3	2	0	1	2	1	0	0	0	0	0	3	0]	96
Ref: 4	[0	1	3	0	0	0	2	3	2	1	0	3	1	0	0	0	0	0	4	0]	95
Ref: 5	[0	1	3	0	0	0	2	3	2	1	0	3	1	0	0	0	0	0	4	0]	95
Ref: 6	[0	2	3	0	0	0	1	3	2	1	0	3	1	0	0	0	0	0	4	0]	93
Ref: 7	[1	3	3	0	0	0	1	2	1	1	1	4	1	0	0	0	0	0	2	0]	89

B

Figure 4.4: DAROGAN Results (2 of 2). **A.** Cofactor Prediction (Section 3.8) the query putative enzyme is assessed against each of the cofactors in the Reference Tread database. The option of performing a cofactor prediction is selected on the input screen. **B.** Actual Tread View. Displays the Treads for the top matches, useful for expert users and for debugging, this view is also optional.

4.2.4 Tread View Page

If the unique identifier for a Tread is clicked in the list of matching Treads in the results section, then a page containing further information for that particular Tread is displayed. At the top of the page is a table containing all the files associated with the Treads that are available for download to the users local file system. These files include the *Tread Report* (Section 3.2.4), a modified PDB file in which conservation scores replace B-factor values, the raw text version of the alignment file used to create the Tread and images appearing further down on the Tread view page.

The main part of the page consists of the database view of the Tread (Figure

A

Alignment	Lead	Residue	Score	1	2	3	4	5	6	7	8
50	50	E	100	0	0	0	0	0	0	0	0
52	52	D	100	0	0	0	0	0	0	0	0
55	55	T	100	0	0	0	0	0	0	0	0
75	74	Y	100	0	0	0	0	0	0	0	0
129	127	K	100	0	0	0	0	0	0	0	0
140	138	Y	100	0	0	0	0	0	0	0	0
189	185	N	100	0	0	0	0	0	0	0	0
218	211	D	100	0	0	0	0	0	0	0	0
221	216	Y	100	0	0	0	0	0	0	0	0
249	242	N	100	0	0	0	0	0	0	0	0
253	246	K	100	0	0	0	0	0	0	0	0
258	251	T	100	0	0	0	0	0	0	0	0
261	254	R	100	0	0	0	0	0	0	0	0
297	284	Q	100	0	0	0	0	0	0	0	0
348	328	Y	100	0	0	0	0	0	0	0	0
401	365	R	100	0	0	0	0	0	0	0	0
403	367	S	100	0	0	0	0	0	0	0	0
179	175	T	97	0	0	0	0	0	0	0	0
324	305	R	92	0	0	0	0	0	0	0	0
128	126	D	90	0	0	0	0	0	0	0	0

B

Key	Description
Alignment	Alignment Numbering
Lead	Lead Sequence Numbering
Residue	Amino acid appearing at position in Lead Sequence
Score	Conservation Score
1	General substrate binding
2	General Cofactor binding
3	Covalent binding of substrate/cofactor
4	General acid/base
5	Positive charge stabilising leaving group
6	Metal binding residue
7	Proton shuttle partners (charge network)
8	Unknown Function or Non functional role

Figure 4.5: **A.** A detailed view of an individual Tread entry in the Reference Tread Database. Each of the conserved functional residues is listed along with its position in the alignment and lead sequence. The conservation score and functional role(s) of each of the residues is also shown. **B.** A key for the columns in **A.**.

4.5A) and a key (Figure 4.5B) to the column headings in the Tread table. The database view includes numbering for the functionally conserved residues in the Tread both with respect to the alignment file and the ungapped sequence of the enzyme used to create the Tread. The residue type and conservation score for each position are also given. The remaining eight columns in the table relate to the functional role assignments described in Section 3.3.

Number	LPos	APos	Amino Acid	ConsScore	Assignment
1	80	77	D	100	1 2 3 4 5 6 7 8
2	92	89	K	100	1 2 3 4 5 6 7 8
3	108	105	K	100	1 2 3 4 5 6 7 8
4	112	109	Q	100	1 2 3 4 5 6 7 8
5	113	110	S	100	1 2 3 4 5 6 7 8
6	118	115	R	100	1 2 3 4 5 6 7 8
7	133	130	E	100	1 2 3 4 5 6 7 8
8	138	135	D	100	1 2 3 4 5 6 7 8
9	146	143	D	100	1 2 3 4 5 6 7 8
10	148	145	R	100	1 2 3 4 5 6 7 8
11	166	163	K	100	1 2 3 4 5 6 7 8
12	189	186	N	100	1 2 3 4 5 6 7 8
13	197	194	K	100	1 2 3 4 5 6 7 8
14	202	199	D	100	1 2 3 4 5 6 7 8
15	211	208	Y	100	1 2 3 4 5 6 7 8
16	111	108	H	96	1 2 3 4 5 6 7 8
17	68	65	R	93	1 2 3 4 5 6 7 8
18	82	79	R	93	1 2 3 4 5 6 7 8
19	181	178	K	93	1 2 3 4 5 6 7 8
20	248	244	R	93	1 2 3 4 5 6 7 8

Submit Reset

Figure 4.6: The Tread Database Interface is designed aid the assignment, and entry into the database, of functional roles to each of the residues in a Reference Tread. The interface bypasses the need for any knowledge of SQL syntax.

4.2.5 Tread Database Interface (TDI)

The Tread Database Interface is designed aid the assignment, and entry into the database, of functional roles to each of the residues in a Reference Tread. The interface bypasses the need for any knowledge of SQL syntax (Figure 4.6). The positions of the conserved residues in the Tread being processed are displayed relative to the original sequence and the alignment to simplify finding the residues in the structure. Radio buttons allow the functional roles of each of the residues to be selected with ease. Once all the residues have been assigned functional roles the submit button updates the entry in the Reference Tread database.

4.3 Database Architecture

Relational databases are now ubiquitous in bioinformatics resources as they provide the best method of collating, managing and accessing data. There are many advantages of using databases, mainly concerned with data integrity, and controlling data redundancy and consistency. Other advantages include the ease of sharing data held with in a database and the ability to easily back-up the data. Databases however have the disadvantage of being complex, requiring a level of expertise to create, populate and maintain a database. The structure, or schema, of a database can be modelled with Entity Relationship (ER) modelling, employed to simplify the process of designing a database schema (Connolly *et al.*, 1999). ER models comprise of *entities*, *attributes* and *relationships*. An entity is characterised as an object or concept having an independent existence (e.g. a student or degree course). Attributes are properties associated with entities (e.g. address or lecturer) and separate entities are linked together by their relationships (e.g. students *enrolled on* degree course).

A simplified ER model for the DAROGAN database is shown in Figure 4.7, showing the relationships between the enzyme, Tread and alignment entities and their associated attributes (primary keys only). A full database layout including all attributes for each of the entities is shown in Table 4.1

Databases are accessed through Database Management Systems (DBMS); software enabling the database to be queried and modified. The DAROGAN database utilises the MySQL DBMS.

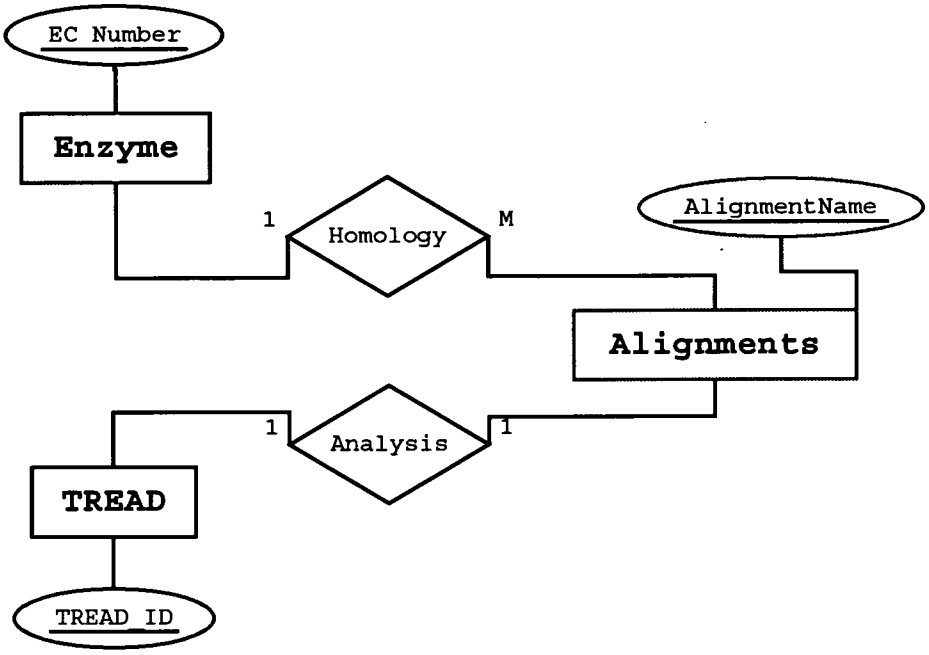


Figure 4.7: Entity Relationship Model (ERM), a high-level conceptual model describing the structure of the Reference Tread Database. Entites are represented as rectangles, attributes as ovals (limited to primary key attributes for simplicity) and relationships as diamonds.

4.3.1 MySQL

MySQL is a relational database management system (RDBMS), supporting the database Structured Query Language (SQL). Numerous RDBMS are available, but few offer the versatility and connectivity of MySQL. The distribution of MySQL is open source and is also the most popular RDBMS available. The MySQL RDBMS runs as a service on the host server and can be accessed via the command interpreter (a command line front end) or the DBI (Database Interface) module (a programming language interface).

Enzyme Table

Column Name	Description	Key Type
EnzymeName	Name of the enzyme	Primary Key
ECNumber	EC classification number	Foreign Key
Cofactor	Name of any cofactors utilised by enzyme	
Substrate	Name of any substrates utilised by enzyme	

Alignment Table

Column Name	Description	Key Type
AlignmentName	File name of the alignment	Primary Key
ECNumber	EC classification number	Foreign Key
EValue	E Value Cut off for BLAST search	
LeadSeqID	Identifier for the lead sequence	
DBID	Database BLAST conducted against	
ACons	Total number of conserved residues	
FCons	Total number of Functionally conserved residues	
LPos	Total number of residues in lead sequence	
APos	Total number of positions in the alignment	
NSeqs	Total number of sequences in the alignment	

TREAD Table

Column Name	Description	Key Type
TreadID	Unique identifier for TREAD and components	Primary Key
AlignmentName	File name of the alignment	Foreign Key
LPos	Position of FCons in the lead sequence	
APos	Position of FCons in the alignment file	
AA	Type of amino acid conserved	
ConScore	Conservation score for the conserved residue	
one	Functional catagory one	
two	Functional catagory two	
three	Functional catagory three	
four	Functional catagory four	
five	Functional catagory five	
six	Functional catagory six	
seven	Functional catagory seven	
eight	Functional catagory eight	

Global Values

Column Name	Description	Key Type
GVName	Name of the global value	
GVValue	Value stored	

Table 4.1: DAROGAN Reference Tread Database Structure

4.4 Programming Languages

4.4.1 Perl

The Practical Extraction Report Language version 1.0 was first released in 1987 by Larry Wall (Wall & Schwartz, 1991) and was designed as an alternative to Unix tools which could only perform specific tasks. The Perl interpreter is a hybrid of an interpreter and a compiler, code is parsed into an internal format before it is executed much like a compiler would do and the interpreter ensures there is no code using up disk space. Perl has a similar syntax to

other programming languages such as C and Java, provides intuitive error messages and does not require specific variable types. There is also a very large number of open source modules available through CPAN (*www.cpan.org*). Perl is highly portable and it is available on most popular operating systems including Linux, Unix, Windows and Macintosh. There is a rivalry between programming in Perl and C. The main argument concerned with the speed at which the two programs can run in. It is generally accepted that C is the faster programming language. The slowness of Perl comes from its flexibility of its data structures, however this is also one of the main strengths of Perl.

Perl was used for most of the coding in the DAROGAN suite of programs, from the creation of the Reference Treads to the user interface for performing predictions. The interface itself was coded in Perl and made available as a web service through the use of CGI.

4.4.2 Common Gateway Interface (CGI)

The Common Gateway Interface is most commonly used in conjunction with Perl, however it is possible to use almost any other programming language. CGI is an interface allowing a program to handle requests from a web server. This interfacing with a program allows the resulting web page to be a dynamic resource rather than static web pages. In fact the main advantage of CGI is that it has the ability to produce output that looks no different, than any other web page, to an end user. CGI can produce a wide variety of outputs including HTML, PDF files, plain text and various image formats.

The CGI Perl module written by Lincoln Stein allows the production of web based Perl scripts using CGI, as much of the difficult programming is taken care of by the module (Available in the standard release of Perl). Alternatives to CGI include ASP (Automatic Service Pages) and PHP (PHP: Hypertext Pre-processor); CGI however remains the most popular.

4.4.3 Perl GD

The GD (Perl Module: GD, 2005) module is the Perl interface to the GD graphics library written by Thomas Boutell for the C programming language (GD Graphics Library, 2005). The Perl GD module was developed by Lincoln Stein and is considered to be the default graphics package for Perl.

GD offers routines for reading, manipulating and writing images in a wide variety of formats and GD is particularly suited to creating *on-the-fly* graphics for CGI applications. GD images are hard coded and embedded into Perl programs, text and shapes (lines, rectangles, spheres and polygons) are placed on a user defined grid, allowing very accurate placement. The advantage of hard coding the images is that they can be implemented with variables to create dynamic images. The disadvantage is the time required to actually hard code the images. A GD extension is the GD::Graph (Perl Module: GD::Graph, 2005) module, written by Martien Verbruggen, allowing graphs and charts to be created. The graph/chart formats include lines, bars, points and pie charts. A

further extension to GD::Graph is GD::Graph3D (Perl Module: GD::Graph3D, 2005), with the ability to create graphs with an additional dimension. Most of the images found on the DAROGAN web interface were created using the GD family of modules.

4.4.4 JavaScript

JavaScript is a fully functional programming language although many consider it to be a lesser scripting language. JavaScript is interpreted rather than compiled but has all the features expected of a full programming language. A common misconception is that JavaScript is related to the Java programming language, this is not true and the name was developed as a marketing ploy (Flanagan, 1998).

Syntactically JavaScript resembles C, Java and Perl and is supported by the more recent Internet browsers (Netscape Navigator version 4+, MS Internet Explorer version 4+) and can be implemented either client-side or server-side. The most common usage is client-side where the code is interpreted by the clients web browser allowing the production of dynamic web pages. However due to possible security risks JavaScript is limited in the tasks it can perform, so it is commonly used for validating the content of HTML forms before they are sent of to the server. For all but the simplest of dynamic web pages alternatives such as CGI must be used.

JavaScript is used in DAROGAN primarily for checking user input. It is advan-

tageous for the error checking to be performed by the users Internet browser, rather than the DAROGAN server as this frees up the server from unnecessary calculations.

4.5 Hardware

The code developed for DAROGAN was written and tested on a modest performance laptop (Table 4.2). However all computationally intensive calculations were performed on the servers *Nova* and *Supernova* (Table 4.2) which were set up as an openMosix cluster for this purpose. The cluster allows large numbers of computationally intensive jobs to be run much more efficiently than on a standard set up, as described below.

Name	Processor	RAM(Mb)	Operating System
Nova	Athlon 1.8+GHz	256	RedHat 8.0 with openMosix
SuperNova	Athlon 2.4+GHz	512	RedHat 8.0 with openMosix
Laptop	Celeron 1.06GHz	256	RedHat 9.0

Table 4.2: DAROGAN Hardware

4.5.1 openMosix

OpenMosix (openMosix, 2005) is an extension to the standard Linux kernel which takes a network of computers and allows then to operate as one. It is important to point out that openMosix is not parallel processing, but rather distributed processing. The aim of an OpenMosix cluster is not to decrease the running time of a single process, but to optimise the processes running on a cluster as a whole. Programs do not have to be written specifically for

openMosix as they would for parallel processing. Any program that will run on a standard Linux setup will run on an OpenMosix setup.

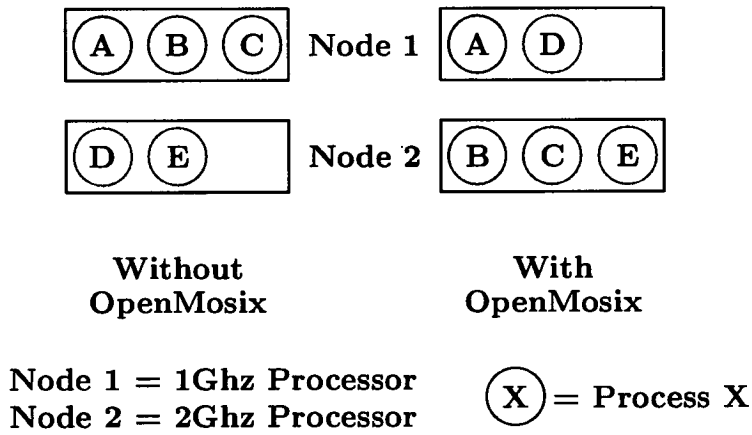


Figure 4.8: A cluster with two nodes is used to illustrate process migration with and without OpenMosix. Without OpenMosix processes are not migrated, they are run on the node they were initiated. With OpenMosix processes are migrated to ensure they are run as efficiently as possible. In the example more processes are migrated to Node 2 as this is a faster machine.

Installing openMosix on a network of computers turns them into a cluster of openMosix nodes. OpenMosix then continually optimises the work load by distributing the processes running over the nodes so they can run as efficiently as possible. An openMosix cluster is a Single Image System (SSI) cluster and consists of two parts. The first part is involved with Pre-emptive Process Migration (PPM) which migrates processes to available nodes, at any time during the processes running time. The second part consists of algorithms dedicated to adaptive resource sharing ensuring the users do not have to have any knowledge of the current resource usage of each of the cluster nodes. This means a process can be started on any of the nodes and the adaptive resource sharing algorithms

decide which node to migrate the process to (Figure 4.8). The resource sharing algorithms allow process migration between the nodes in the cluster and is done transparently, so the user would not notice the migration to a different node.

4.6 Concluding Remarks

DAROGAN as a publicly available web service was designed to be as user friendly as possible and has gained positive feedback from people who have used the service. in addition to being intuitive to use, a considerable effort was placed into the design of the software to allow modifications to be made as the method was developed. The flexibility of the Perl programming language and it ease of linking to a MySQL database made for a very suitable platform to develop DAROGAN.

Chapter 5

Method Evaluation

5.1 Overview

To evaluate the DAROGAN method, as described in the previous chapter, and compare it to an existing method used for function prediction several statistical test were employed. And in addition to the evaluation, the determination of the optimal parameters; Scoring method, PISCES sequence identity cut off, statistical significance level for the DAROGAN method was undertaken. The implementation and the results of these tests are described in the chapter below.

The evaluation of the method began with the selection of the seed sequences used to make the Reference Treads and the most appropriate E-Value to use in the intelligent alignment heuristic (See Section 3.2.2). This was then followed by a look into the amino acid preferences with in the Treads; for each of the cofactor utilising families used for the evaluation.

In order to assess the DAROGAN method, it was compared to another existing method: pHMMs. The Self-Consistency and Jack-Knife tests were used in this comparison, and the determination of the optimal parameters to utilise in real life predictions, as introduced in Chapter 6, were undertaken.

5.2 Reference Tread Creation and Composition

5.2.1 Selection of Seed Sequences for Reference Treads

Reference Treads were created for four families of enzymes (defined by the cofactor they utilise); pyridoxal-5'phosphate (PLP), thiamin (TPP), glutathione (GLU) and folic acid (FOL). The choices for these four families and further details are given in Section 1.6. Seed sequences were used as queries for BLAST searches to yield a set of similar sequences, then aligned using a multiple sequence alignment program. The alignment must contain approximately 15 conserved functional residues in order to be made into a Reference Tread; achieved by iteratively tailoring the E-Value of the BLAST searches. The method for creating a Tread from a seed sequence is discussed fully in Section 3.2, the process of the initial selection of the seed sequences is described below along with details of the sequences selected.

The role of the Reference Treads is to collate enough information to represent a set or family of enzymes (in this case not necessarily evolutionarily related; e.g. PLP utilising) to facilitate the prediction of cofactor usage for a putative enzyme. It would be redundant to collect a series of identical Treads for an

Pyridoxal 5'-phosphate Utilising Proteins
Number = 250
Average Sequence Identity = 16.41% (7.22 s.d.)

O05321_E-15_SP	O08445_E-35_SP	O13326_E-5_SP	O14092_E-20_SP	O14370_E-5_SP	O26103_E-20_SP	O27390_E-15_SP
O30418_E-10_SP	O31461_E-20_SP	O31665_E-30_SP	O32148_E-15_SP	O33065_E-5_SP	O46560_E-10_SP	O50584_E-5_SP
O54694_E-35_SP	O58489_E-45_SP	O59828_E-35_SP	O67019_E-25_SP	O67262_E-20_SP	O67507_E-25_SP	O67687_E-25_SP
O67733_E-25_SP	O74267_E-5_SP	O74351_E-25_SP	O84693_E-30_SP	O94069_E-35_SP	O96567_E-20_SP	P00584_E-10_SP
P05031_E-20_SP	P05459_E-25_SP	P06655_E-35_SP	P08080_E-40_SP	P10725_E-35_SP	P11096_E-25_SP	P11603_E-45_SP
P14173_E-30_SP	P16524_E-30_SP	P16609_E-40_SP	P18285_E-35_SP	P18949_E-20_SP	P23279_E-20_SP	P24288_E-10_SP
P25269_E-40_SP	P27119_E-20_SP	P27718_E-20_SP	P28578_E-5_SP	P29012_E-35_SP	P32582_E-25_SP	P34899_E-5_SP
P36605_E-25_SP	P37303_E-5_SP	P37419_E-45_SP	P39643_E-35_SP	P40807_E-20_SP	P43089_E-40_SP	P44506_E-20_SP
P45837_E-10_SP	P47176_E-5_SP	P49361_E-60_SP	P49725_E-15_SP	P50134_E-20_SP	P50433_E-5_SP	P50554_E-20_SP
P52055_E-25_SP	P52056_E-15_SP	P52069_E-5_SP	P53206_E-20_SP	P54377_E-10_SP	P54687_E-5_SP	P54688_E-5_SP
P54691_E-20_SP	P56099_E-25_SP	P56129_E-15_SP	P56142_E-40_SP	P57289_E-5_SP	P58715_E-20_SP	P59237_E-40_SP
P60120_E-45_SP	P61000_E-35_SP	P63479_E-35_SP	P63482_E-15_SP	P63512_E-25_SP	P66803_E-5_SP	P66876_E-20_SP
P66899_E-5_SP	P66901_E-5_SP	P66985_E-40_SP	P69911_E-10_SP	P69912_E-10_SP	P70727_E-20_SP	P72039_E-35_SP
P75298_E-15_SP	P77690_E-15_SP	P77727_E-35_SP	P78599_E-15_SP	P78698_E-35_SP	P81893_E-20_SP	P87131_E-20_SP
P87187_E-30_SP	P91856_E-5_SP	P94967_E-25_SP	Q04792_E-10_SP	Q05174_E-25_SP	Q05567_E-15_SP	Q05683_E-25_SP
Q06086_E-30_SP	Q06965_E-35_SP	Q10349_E-5_SP	Q12198_E-5_SP	Q16773_E-30_SP	Q21890_E-5_SP	Q44004_E-15_SP
Q44686_E-60_SP	Q44688_E-10_SP	Q50723_E-10_SP	Q51687_E-15_SP	Q54899_E-35_SP	Q55128_E-40_SP	Q56346_E-35_SP
Q58414_E-35_SP	Q58466_E-15_SP	Q59169_E-35_SP	Q59447_E-40_SP	Q634V8_E-10_SP	Q64611_E-25_SP	Q6FUP6_E-5_SP
Q72VI2_E-5_SP	Q7MEH7_E-5_SP	Q7ND67_E-5_SP	Q7NL03_E-35_SP	Q7TUL2_E-35_SP	Q7U9J7_E-5_SP	Q7UQN2_E-5_SP
Q7V3M9_E-5_SP	Q7VLP0_E-5_SP	Q7VTF1_E-40_SP	Q7VUW7_E-5_SP	Q7VWL5_E-35_SP	Q7WD04_E-40_SP	Q81JY4_E-10_SP
Q81M08_E-10_SP	Q822W3_E-45_SP	Q82A82_E-40_SP	Q82AA5_E-30_SP	Q82JI0_E-5_SP	Q82WQ3_E-15_SP	Q83LP3_E-5_SP
Q84I53_E-30_SP	Q87JS8_E-5_SP	Q87ST5_E-20_SP	Q884R9_E-50_SP	Q88AD1_E-5_SP	Q88MO7_E-5_SP	Q89A48_E-10_SP
Q89AX7_E-25_SP	Q89WE5_E-40_SP	Q8A9S7_E-5_SP	Q8CTA4_E-35_SP	Q8CTG8_E-40_SP	Q8DOM6_E-35_SP	Q8D7G5_E-5_SP
Q8D8Q1_E-25_SP	Q8DB36_E-35_SP	Q8DCL0_E-40_SP	Q8DVF3_E-40_SP	Q8ECR2_E-35_SP	Q8EEH2_E-5_SP	Q8EFB2_E-25_SP
Q8FHG5_E-10_SP	Q8FLJ6_E-30_SP	Q8GYY0_E-20_SP	Q8K5Z8_E-35_SP	Q8K929_E-15_SP	Q8K9P2_E-5_SP	Q8KC36_E-5_SP
Q8KZ92_E-40_SP	Q8LOZ4_E-5_SP	Q8NT73_E-20_SP	Q8NWU2_E-40_SP	Q8P122_E-5_SP	Q8P5R4_E-25_SP	Q8PBK7_E-45_SP
Q8PCN4_E-5_SP	Q8PFQ5_E-5_SP	Q8PGD0_E-40_SP	Q8QZR1_E-30_SP	Q8QZR5_E-15_SP	Q8R860_E-35_SP	Q8TH25_E-35_SP
Q8U093_E-45_SP	Q8UG75_E-5_SP	Q8UGD4_E-30_SP	Q8UJB0_E-40_SP	Q8X4I9_E-40_SP	Q8X5V2_E-35_SP	Q8XTQ1_E-10_SP
Q8XV80_E-30_SP	Q8Y1G1_E-10_SP	Q8YD03_E-30_SP	Q8YU96_E-40_SP	Q8Z2Z9_E-5_SP	Q8Z4W1_E-15_SP	Q8Z688_E-30_SP
Q8ZCR1_E-5_SP	Q8ZF73_E-5_SP	Q8ZFX6_E-30_SP	Q8ZGB4_E-5_SP	Q8ZYF9_E-10_SP	Q91XF0_E-5_SP	Q92A83_E-40_SP
Q92B90_E-10_SP	Q92JD9_E-30_SP	Q92MG0_E-35_SP	Q97GV1_E-5_SP	Q98QM2_E-5_SP	Q9A671_E-35_SP	Q9CC54_E-40_SP
Q9CCE2_E-10_SP	Q9CG20_E-5_SP	Q9CHW5_E-5_SP	Q9CL60_E-5_SP	Q9CPD5_E-30_SP	Q9DBE0_E-25_SP	Q9F9L1_E-20_SP
Q9FEW2_E-40_SP	Q9HVX0_E-30_SP	Q9HZ66_E-5_SP	Q9JTH8_E-30_SP	Q9JVC0_E-40_SP	Q9JWA6_E-15_SP	Q9JX42_E-20_SP
Q9KMP4_E-5_SP	Q9KST6_E-40_SP	Q9KSX2_E-25_SP	Q9LE06_E-15_SP	Q9LPM9_E-15_SP	Q9PBC6_E-30_SP	Q9PET2_E-5_SP
Q9PH02_E-5_SP	Q9RAS9_E-25_SP	Q9VOL2_E-45_SP	Q9X191_E-30_SP	Q9XAY7_E-5_SP	Q9XB01_E-5_SP	Q9Y9H2_E-10_SP
Q9Z831_E-5_SP	Q9ZBH5_E-5_SP	Q9ZBY8_E-40_SP	Q9ZJU9_E-40_SP	Q9ZMW6_E-20_SP		

Table 5.1: A list of Swissprot accession codes for the PLP testing set, comprising of PLP Utilising Enzymes. Accession numbers are supplemented by the E-Value used to find optimal alignments (See Section 3.2); SP=Swissprot. Average sequence identity calculated by aligning each pair of sequences in an all-against-all comparison using MUSCLE (Edgar, 2004). Once aligned sequence identity is defined as the number of identical residues divided by the length of the longest of the two sequences being compared.

enzyme family, so care was taken to ensure the Treads represent the diversity of the enzymes in the family (See Section 5.2.2 below). The simplest way to achieve this was to utilise a non-redundant set of sequences to represent each family. The SwissProt sequence database (Release 47.4, July 2005) (SwissProt & TrEMBL, 2005) was used to generate a set of sequences per family, selected by keyword (e.g. *pyridoxal* for PLP enzymes). The SwissProt database was chosen as it is curated and has rich annotation for protein function; including cofactor

utilisation.

Glutathione Utilising Proteins
Number = 200
Average Sequence Identity = 14.32% (8.85 s.d.)

O04437_E-30_SP	O04885_E-15_SP	O04922_E-35_SP	O08709_E-15_SP	O09114_E-15_SP	O09131_E-10_SP	O15770_E-35_SP
O16115_E-20_SP	O22850_E-35_SP	O23970_E-25_SP	O24296_E-35_SP	O32770_E-35_SP	O35660_E-20_SP	O36032_E-10_SP
O48646_E-40_SP	O49069_E-40_SP	O59402_E-5_SP	O59858_E-35_SP	O65857_E-15_SP	O73888_E-25_SP	O82451_E-15_SP
P03019_E-15_SP	P04907_E-25_SP	P06610_E-30_SP	P09623_E-70_SP	P10299_E-20_SP	P10575_E-15_SP	P12711_E-50_SP
P12864_E-15_SP	P15626_E-20_SP	P16413_E-20_SP	P16635_E-5_SP	P17695_E-15_SP	P18425_E-15_SP	P18956_E-50_SP
P19440_E-60_SP	P19639_E-20_SP	P19854_E-60_SP	P20107_E-15_SP	P20135_E-10_SP	P21161_E-15_SP	P21765_E-20_SP
P23908_E-10_SP	P25373_E-10_SP	P26624_E-15_SP	P27014_E-20_SP	P27456_E-60_SP	P27457_E-5_SP	P28342_E-30_SP
P28801_E-25_SP	P30109_E-15_SP	P30111_E-20_SP	P30115_E-20_SP	P30116_E-20_SP	P30635_E-40_SP	P30710_E-20_SP
P30711_E-15_SP	P31577_E-15_SP	P32771_E-50_SP	P35666_E-20_SP	P36014_E-20_SP	P36267_E-40_SP	P36268_E-15_SP
P36969_E-30_SP	P36970_E-30_SP	P38143_E-35_SP	P39050_E-45_SP	P40581_E-35_SP	P41921_E-50_SP	P42761_E-30_SP
P42769_E-20_SP	P42770_E-60_SP	P43783_E-50_SP	P44638_E-15_SP	P44933_E-10_SP	P45382_E-45_SP	P45482_E-50_SP
P45522_E-5_SP	P45534_E-10_SP	P46418_E-20_SP	P46420_E-25_SP	P46423_E-15_SP	P46436_E-25_SP	P46439_E-15_SP
P47734_E-20_SP	P47791_E-45_SP	P48438_E-5_SP	P48638_E-60_SP	P48641_E-50_SP	P48774_E-10_SP	P50107_E-20_SP
P52033_E-20_SP	P52035_E-40_SP	P52036_E-35_SP	P54422_E-45_SP	P57108_E-30_SP	P57336_E-10_SP	P58580_E-5_SP
P59600_E-10_SP	P59601_E-10_SP	P63186_E-45_SP	P64290_E-35_SP	P64291_E-35_SP	P65510_E-10_SP	P65511_E-10_SP
P67878_E-20_SP	P68689_E-5_SP	P70619_E-50_SP	P72324_E-45_SP	P72933_E-5_SP	P73138_E-25_SP	P73493_E-5_SP
P78965_E-45_SP	P79764_E-10_SP	P80467_E-60_SP	P80572_E-60_SP	P80894_E-20_SP	P81065_E-5_SP	P81431_E-45_SP
P81601_E-60_SP	P83645_E-20_SP	P91254_E-25_SP	P91938_E-40_SP	P92980_E-10_SP	P99097_E-35_SP	Q03662_E-10_SP
Q05584_E-10_SP	Q06099_E-50_SP	Q08863_E-15_SP	Q09596_E-20_SP	Q12320_E-5_SP	Q16772_E-15_SP	Q16873_E-10_SP
Q21355_E-20_SP	Q29095_E-10_SP	Q29562_E-15_SP	Q42891_E-15_SP	Q43621_E-60_SP	Q56415_E-15_SP	Q64625_E-20_SP
Q6BPI1_E-40_SP	Q6C5H4_E-50_SP	Q6CM04_E-15_SP	Q6HA24_E-50_SP	Q7MYD5_E-10_SP	Q7TUG9_E-10_SP	Q7U3W8_E-5_SP
Q83PX9_E-10_SP	Q83PYO_E-10_SP	Q83SQ4_E-15_SP	Q873E8_E-50_SP	Q87LK1_E-5_SP	Q89WLO_E-5_SP	Q8CSR9_E-35_SP
Q8DCN1_E-15_SP	Q8FLA1_E-10_SP	Q8FXW6_E-10_SP	Q8UOB3_E-10_SP	Q8VUS5_E-5_SP	Q8X742_E-10_SP	Q8XA20_E-10_SP
Q8XA24_E-15_SP	Q8Z308_E-10_SP	Q8Z9KO_E-10_SP	Q8Z9K1_E-15_SP	Q8ZJC5_E-10_SP	Q8ZRW2_E-10_SP	Q8ZUG2_E-15_SP
Q93S39_E-25_SP	Q96266_E-15_SP	Q96324_E-15_SP	Q96533_E-60_SP	Q96SL4_E-25_SP	Q976K1_E-10_SP	Q99735_E-15_SP
Q99KB8_E-5_SP	Q99LJ6_E-25_SP	Q99MZ4_E-30_SP	Q9CPUO_E-10_SP	Q9ESH6_E-15_SP	Q9JYJ3_E-5_SP	Q9K4Z2_E-5_SP
Q9K4Z7_E-10_SP	Q9LYB4_E-35_SP	Q9N2J2_E-30_SP	Q9N4X8_E-20_SP	Q9QWE9_E-40_SP	Q9RUH3_E-10_SP	Q9SLM6_E-15_SP
Q9SRY5_E-15_SP	Q9TSM4_E-25_SP	Q9UJ14_E-25_SP	Q9V1I3_E-5_SP	Q9VG93_E-15_SP	Q9VG94_E-10_SP	Q9VNT5_E-35_SP
Q9X755_E-10_SP	Q9Z2A9_E-35_SP	Q9Z339_E-10_SP	Q9ZDW1_E-10_SP			

Table 5.2: A list of Swissprot accession codes for the GLU testing set, comprising of glutathione utilising enzymes. Sequence identity calculated as described in Table 5.1. Accession numbers are supplemented by the E-Value used to find optimal alignments (See Section 3.2) and the database identity (SP=Swissprot).

For PLP enzymes the keyword search yielded 2484 sequences. Unfortunately it would have been computationally prohibitive to iteratively create multiple sequence alignments with the required fifteen conserved functional residues for each of the sequences. A cut off of 250 sequences was therefore selected as a compromise between sequence coverage and CPU time required to create the alignments. Sequences were randomly selected, using a Perl script, from the 2484 sequences and alignments created until 250 optimal alignments were generated (See Table 5.1). The same approach was used for the GLU enzyme family (keyword: *glutathione*) yielding 754 sequences and a cut off of 200 sequences was

chosen (See Table 5.2).

Thiamin Utilising Proteins
Number = 43
Average Sequence Identity = 13.72% (8.56s.d.)

034292_E-5_SP	034293_E-40_SP	034294_E-15_SP	060779_E-35_TR	P06169_E-20_SP	P08559_E-20_SP	P0A6B7_E-30_SP
P11177_E-50_SP	P16467_E-25_SP	P20906_E-25_SP	P24031_E-20_TR	P25362_E-10_TR	P26263_E-25_SP	P29401_E-30_SP
P30137_E-15_TR	P30138_E-15_SP	P30636_E-10_TR	P31550_E-20_TR	P32318_E-25_SP	P32896_E-5_SP	P35202_E-5_TR
P38141_E-5_TR	P39594_E-25_SP	P40998_E-30_SP	P43544_E-15_TR	P43545_E-5_SP	P50970_E-80_SP	P51854_E-25_SP
P53823_E-15_TR	P53824_E-5_SP	P76422_E-30_SP	P76423_E-30_SP	Q01682_E-15_TR	Q05998_E-10_SP	Q06490_E-35_TR
Q07471_E-25_SP	Q08224_E-25_SP	Q08975_E-25_SP	Q9H3S4_E-25_TR	Q9ROM5_E-25_TR	Q9RGS4_E-20_SP	Q9RGS5_E-30_SP
Q9RGS6_E-35_SP						

Table 5.3: A list of Swissprot accession codes for the TPP testing set, comprising of thiamin utilising enzymes. Sequence identity calculated as described in Table 5.1. Accession numbers are supplemented by the E-Value used to find optimal alignments (See Section 3.2) and the database identity (SP=Swissprot).

Folic Acid Utilising Proteins
Number = 56
Average Sequence Identity = 20.39% (15.83 s.d.)

005701_E-5_SP	033724_E-5_SP	035409_E-10_SP	065355_E-20_TR	P00381_E-20_TR	P02702_E-15_TR	P05382_E-5_SP
P07807_E-20_SP	P0A578_E-5_SP	P0A579_E-5_SP	P0C002_E-5_SP	P0C003_E-5_SP	P0C004_E-5_SP	P11744_E-5_SP
P14207_E-15_TR	P15328_E-15_TR	P19465_E-40_SP	P19539_E-5_SP	P26282_E-5_SP	P28819_E-35_SP	P28820_E-25_SP
P28821_E-15_SP	P28822_E-5_SP	P28823_E-15_SP	P29251_E-30_SP	P29252_E-20_SP	P34044_E-15_SP	P35846_E-15_TR
P40099_E-10_SP	P41439_E-5_TR	P41440_E-30_TR	P43776_E-5_SP	P46812_E-5_SP	P51601_E-30_SP	P53848_E-20_SP
P57696_E-5_SP	P59655_E-5_SP	P64139_E-5_SP	P64140_E-5_SP	P64141_E-5_SP	P64142_E-5_SP	P73248_E-5_SP
Q05621_E-15_SP	Q05685_E-15_TR	Q12676_E-5_SP	Q51161_E-5_SP	Q59919_E-5_SP	Q5XCA8_E-5_SP	Q62867_E-5_TR
Q8CMT7_E-5_SP	Q8K7K8_E-5_SP	Q8NXZ2_E-5_SP	Q8P152_E-5_SP	Q9H2D1_E-25_SP	Q9JT70_E-5_SP	Q9ZOL8_E-5_TR

Table 5.4: A list of Swissprot accession codes for the FOL testing set, comprising of folic acid utilising enzymes. Sequence identity calculated as described in Table 5.1. Accession numbers are supplemented by the E-Value used to find optimal alignments (See Section 3.2) and the database identity (SP=Swissprot).

A slightly different approach was adopted for the TPP and FOL enzyme families. For TPP enzymes (keyword: *thiamin*) 60 sequences were retrieved and 63 sequences for the FOL enzymes (keyword: *folic*). In both cases the number of sequences was low compared to the PLP and GLU enzyme searches, meaning the CPU time for calculating the sequence alignments was less of a consideration. Unfortunately the process of taking a seed sequence and creating an alignment

with fifteen conserved functional residues does not have a 100% success rate, as it is not always possible to tailor the E-Value to yield the required fifteen conserved functional residues. Of the 63 FOL sequences only 56 were successfully made into Reference Treads (See Table 5.4) and out of the 60 TPP sequences 43 were successful (See Table 5.3).

5.2.2 PISCES Cull

During the selection of seed sequences for the optimum alignments there was no control over the sequence similarity of the sequences in each of the cofactor utilising families beyond the usage of the SwissProt database for sequences (a non-redundant sequence database). To investigate the impact of sequence similarity on the success of the function prediction methods the PISCES sequence culling server (See Section 2.7) was utilised. By culling the cofactor utilising families at various sequence identity cut offs (100%, 75%, 50%, 25%) the effect of the culling on the function predictions, can be determined (See Section 5.3.3 below for the results).

The PISCES culling removes similar sequences above a sequence identity threshold to leave a set of culled sequences. Figure 5.1 shows the results of the cull in respect to the number of sequences remaining in the enzyme families. The number of sequences can be seen to decrease with the lowering of the sequence identity cut offs for the culls.

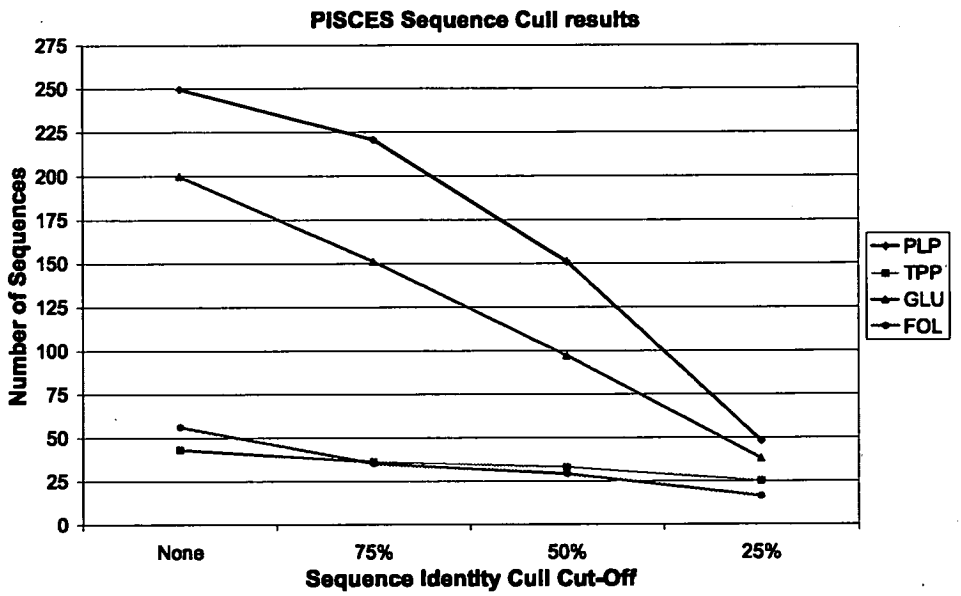


Figure 5.1: PISCES Sequence Cull Results.

5.2.3 E-Value Cut Off Preferences for Treads

In Section 3.2.1 the method of tailoring the E-Value in BLAST searches to automatically yield alignments with 15 conserved functional residues is discussed and in Section 3.2.2 particular attention is given to reducing the total number of alignments needed in the iterative process to produce an optimal alignment. By reducing the number of sequence alignments to be calculated, the CPU time required to go from seed sequence to optimal alignment is also reduced. The heuristic employed takes a starting guess for the E-Value cut-off most likely to lead to a optimal alignment. A preliminary study using PLP enzymes (data not shown) suggested an E-Value of 10^{-40} would be suitable. Once the Reference Treads had been created this cut-off value was then reassessed.

For each of the Reference Treads in the DAROGAN database the E-Value

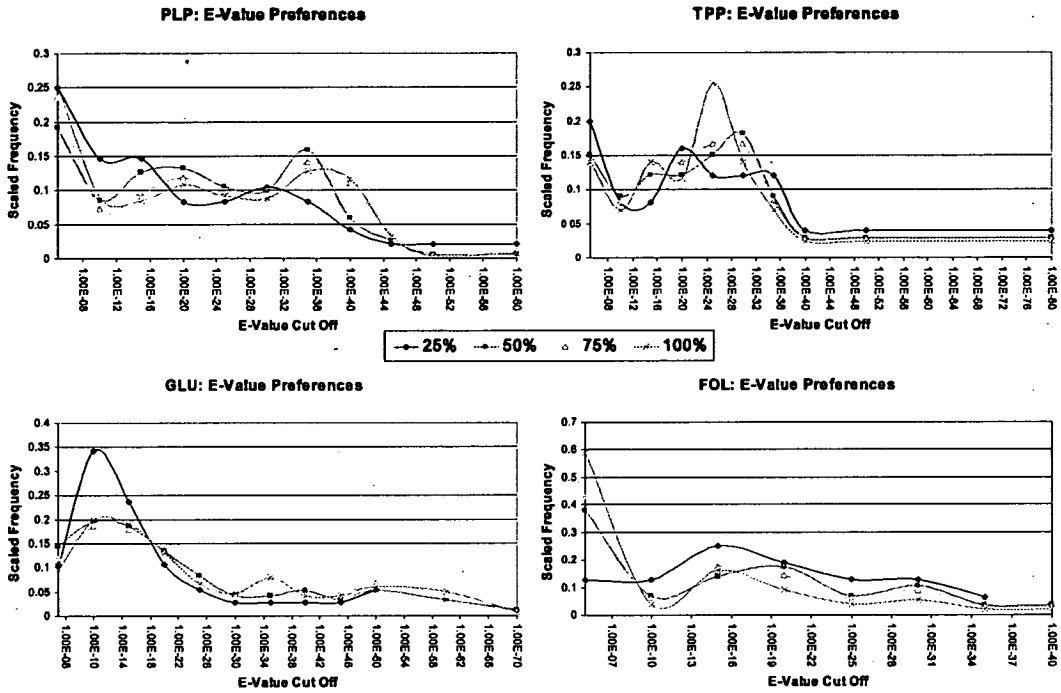


Figure 5.2: Each Tread in the testing set was created from a BLAST search at a specific E-Value (See Section 3.2). At each E-Value in the range (E^{-5} to E^{-150}) the proportion of Treads created at this E-Value is plotted for PLP, TPP, GLU, FOL and at a range for PISCES sequence identity cut off thresholds.

required to eventually yield an optimal alignment was recorded and are shown in Figure 5.2. The four cofactor utilising enzyme families were assessed individually for their E-Value preferences; the PISCES culled data sets were also included in this assessment to determine the impact, if any, of culling the sequences.

The PLP utilising enzymes do not show an obvious peak for the most likely E-Value cut off to utilise in creating optimum alignments. However there is a small peak at approximately 10^{-40} and a larger peak at 10^{-5}. The GLU utilising enzymes show a clear optimal E-Value of between 10^{-10} and 10^{-15} indicating this would be good starting guess for the alignment heuristic. The TPP utilising

enzymes show a broad peak between 10^{-15} and 10^{-35} indicating a likely starting point in this range for the heuristic. The FOL utilising enzymes show that an E-Value cut off at 10^{-5} would be optimal for the heuristic.

The results of the assessment show that it would be prudent to have starting guess cut offs specific to the cofactor utilised to improve the efficiency of the heuristic. Possible starting values for future version of DAROGAN would be PLP 10^{-35}, GLU 10^{-10}, TPP 10^{-25}, FOL 10^{-5}. However the small number of enzymes in the families, TPP and FOL in particular, may adversely affect the selection of good starting guess cut offs. The more members of a family there are, the more likely the starting guess can be accurately chosen.

The PISCES culling does not have a dramatic effect on the selection of a good starting guess E-Value cut off. Decreasing the sequence identity cut off in the cull has the effect of decreasing the starting guess E-Value cut off. However this effect is not large enough to justify using a starting guess specific to the PISCES cull sequence identity cut off.

5.2.4 Amino Acid Occurrence in Treads

To investigate the amino acid preferences for the Reference Treads utilising a particular cofactor a Perl script was written to calculate the average frequency of amino acid abundance by cofactor utilisation and PISCES cull threshold. The script was also used to calculate averaged Treads per cofactor, used in part of the DAROGAN prediction method (See Section 3.8).

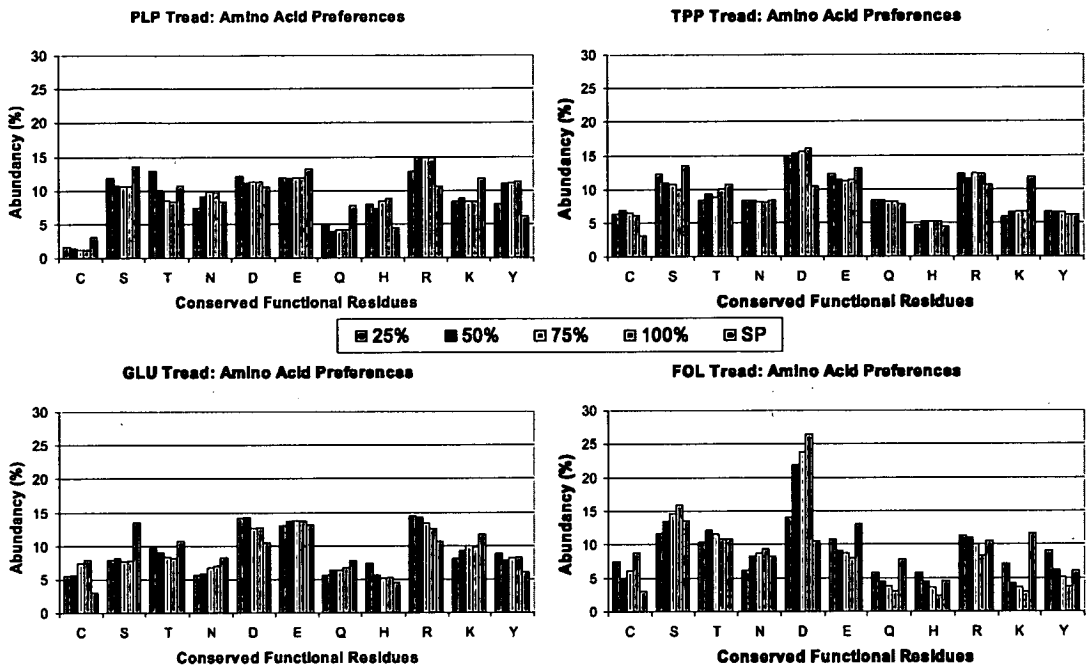


Figure 5.3: Average Tread Amino Acid Distribution. Treads for each of the enzyme cofactor utilising families (PLP,TPP,GLU,FOL) are assessed separately and for a range of PISCES sequence identity cut offs (25%,50%,75%,100%). An additional column showing the amino acid preferences of the SwissProt sequence database (unfiltered; see main text), is also shown. N.B. Only conserved functional residues (KRENDYCHQST) are encoded in Treads.

Each Reference Tread holds information for fifteen conserved functional residues specific to the enzyme used to make the Tread. It is these fifteen residues that are used to infer functional similarity between the Reference Treads and the Query Tread of a putative enzyme. For the DAROGAN prediction method to be successful Treads for different cofactor utilising enzyme families must be sufficiently different from the other families. If there was no discernible difference between Treads the prediction method would probably fail.

The average abundance of amino acids for the Reference Treads are shown in

Figure 5.3. The abundances were calculated for each of the cofactor utilising enzyme families and for each of the PISCES sequence identity culling thresholds (100%, 75%, 50%, 25%). However it is worth noting that the abundances are averages over the families, so the abundance for individual Treads can not be seen. To provide a background comparison for amino acid abundance data for the Swissprot sequence database was included. The Swissprot data is not ideal for the comparison as it contains many non-enzyme sequences and although non-redundant has had no similarity/homology filtering applied to it.

The PLP utilising enzymes have a relatively high abundance, as compared to the TPP and FOL families, of lysine residues as a lysine is required for covalently binding the PLP cofactor in the active site of the enzyme. The TPP Treads show a preference for glutamic acid (approx. 12%) and aspartic acid (approx. 15%) residues as there is a requirement for a general acid/base in the catalytic mechanism involving TPP as a cofactor. Unlike PLP enzymes there is no requirement for the covalent binding of the cofactor, so there is no preference of lysine residues (or any other amino acid for covalently binding the cofactor).

The GLU and FOL utilising enzymes have a wider variety of catalytic mechanisms than the well defined mechanisms for PLP and TPP utilising enzymes, so the preferences for particular amino acids are difficult to attribute to any particular mechanism. However in the GLU Treads there are examples of prostaglandin-D/E synthases, requiring a tyrosine residue to covalently bind the GLU cofactor (Kanaoka *et al.*, 1997; Yamada *et al.*, 2005) reflected in the

preference for tyrosines in the Treads. The involvement of a cysteine in the prostaglandin-D/E synthases, however is not reflected in the Tread preferences. The FOL utilising enzymes have the disadvantage of being the smallest of the enzyme families, so the underpopulation of this group is likely to have an adverse effect on the Tread amino acid preferences. As more enzymes are deposited in the sequences databases allowing more Treads to be created the more reliable the amino acid preference plots for the FOL enzymes will be.

The effect of the PISCES sequence culling on the amino acid preferences is not as pronounced as might be expected. The inclusion of highly similar sequences in an enzyme family will have the effect of causing certain amino acids to appear more prominently than they should be. However the effect of increasing the severity of the PISCES cull threshold is only really obvious in the FOL utilising enzymes. As this family is underpopulated the including of similar will have more of an observable effect on the amino acid preferences.

5.3 Scoring Method and HMM Comparison

In order to assess the performance of the DAROGAN method an existing, and established, function prediction method was used as a comparison. The DAROGAN prediction method is a novel approach to function prediction so there are no direct equivalents of the method to use for comparison. Profile hidden Markov models (pHMM) were chosen as the comparison method as the stages in creating a pHMM are similar to that of Treads, in that they are both

derived from a multiple sequence alignment. To reiterate: Treads are *unordered sets of conserved functional residues* and were primarily designed for predicting cofactor utilisation of a putative enzyme. With pHMMs the order of the residues encoded is maintained and all the residues in the alignment are included unlike Treads (See Section 2.6 for a more detailed description of pHMMs). Treads were designed to predict the functions of putative enzymes where their three dimensional structures are not necessarily similar, so residues important in the function of the enzyme are used. pHMMs are most commonly used for sequence similarity searching, where sequences are aligned against a pHMM and the similarity assessed. For function prediction, highly similar sequences are likely to indicate that the proteins are homologous and are likely to utilise the same cofactor.

To compare the DAROGAN method, with its six different scoring schemes, to the pHMM method it is necessary to utilise statistical testing methods; the methods chosen were the Self-Consistency and Jack-Knife re-substitution tests. The reasons for utilising these methods, how they work and the results of the tests are discussed below. As Treads and pHMMs have different applications in the field of function prediction any comparisons should be assessed with this caveat in mind.

In addition to comparing the DAROGAN method to pHMMs a more intuitive comparison was also made against random assignment of cofactor utilisation. As there are four enzyme families it might be expected that there would be a 25% chance of correctly guessing the cofactor utilised. However a more accurate

estimate is given by taking the number of enzymes in each of the reference enzyme families (Equation 5.1), the weighted random gives a 35.7% chance of correctly predicting cofactor utilisation. For a function prediction to be considered in anyway successful it must be able to out perform random assignment of cofactor utilisation.

$$\left(\frac{250}{549}\right)^2 + \left(\frac{43}{549}\right)^2 + \left(\frac{200}{549}\right)^2 + \left(\frac{56}{549}\right)^2 = 35.7\% \tag{5.1}$$

5.3.1 Self-Consistency Test

The Self-Consistency test is used to determine whether a data set describes an enzyme family in enough detail to enable a high success rate in a practical application of a prediction method. However the Self-Consistency test is not sufficient on its own to fully assess the success of a prediction method. For a complete assessment it must be accompanied by a cross-validation test (e.g. Jack-Knife re-substitution test as discussed in Section 5.3.2 below).

Figure 5.4 (A and B; Left hand side) describes the implementation of the Self-Consistency test for the DAROGAN and pHMM methods. Each entry (Tread or pHMM) from the DAROGAN Reference database is compared to all the other entries in the database, including itself.

A Self-Consistency test will show a high success rate and an underestimation of the error for a prediction method. The Self-Consistency test results are shown in Figure 5.5 (Raw data in Appendix A.2). The success rate is defined

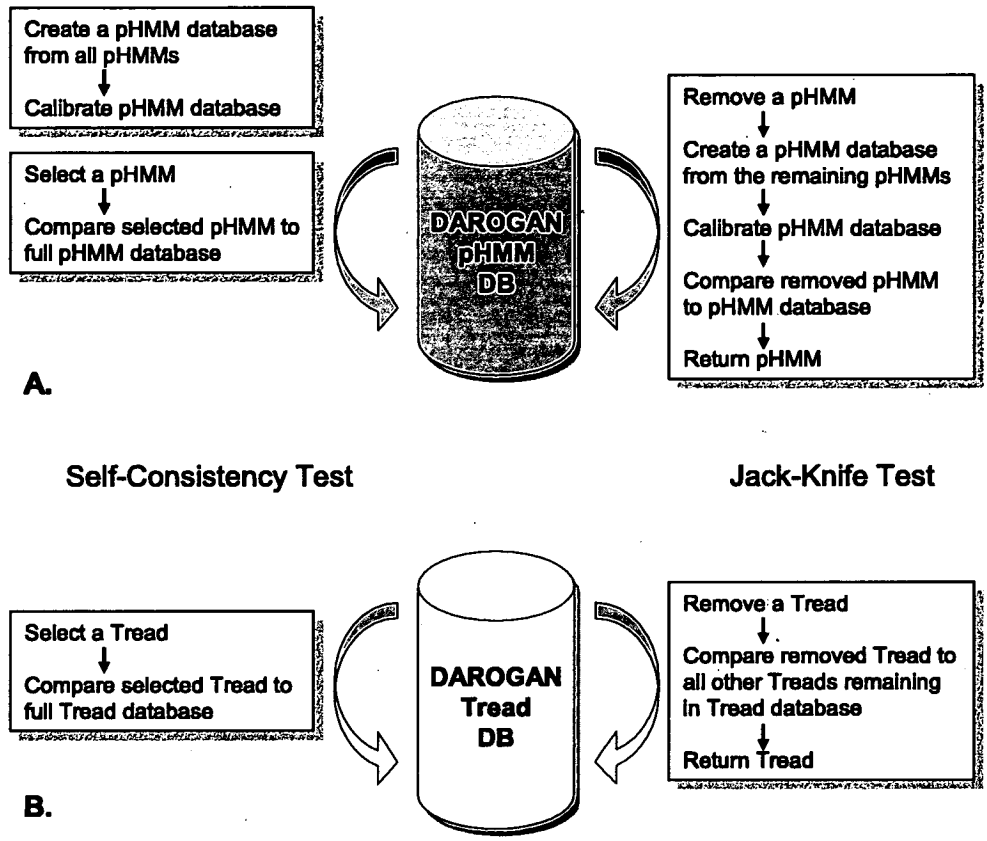


Figure 5.4: Self-Consistency and Jack-Knife Method Schematic. **A.** Flow diagram for the profile hidden Markov model Self-Consistency and Jack-Knife tests. **B.** Flow diagram for the DAROGAN Tread Self-Consistency and Jack-Knife tests.

to be whether the top scoring prediction utilises the same cofactor as the query (Defined in Table 5.5). As expected the success rates are artificially high with the pHMM and DAROGAN scoring methods showing success rates in excess of 95%.

A more detailed breakdown of the results is shown in Figure 5.6 (Raw data in Appendix A.2) where Accuracy, Error, Precision, Recall and the F1 statistic are used to describe the data (Defined in Table 5.5). The Accuracy of each of the meth-

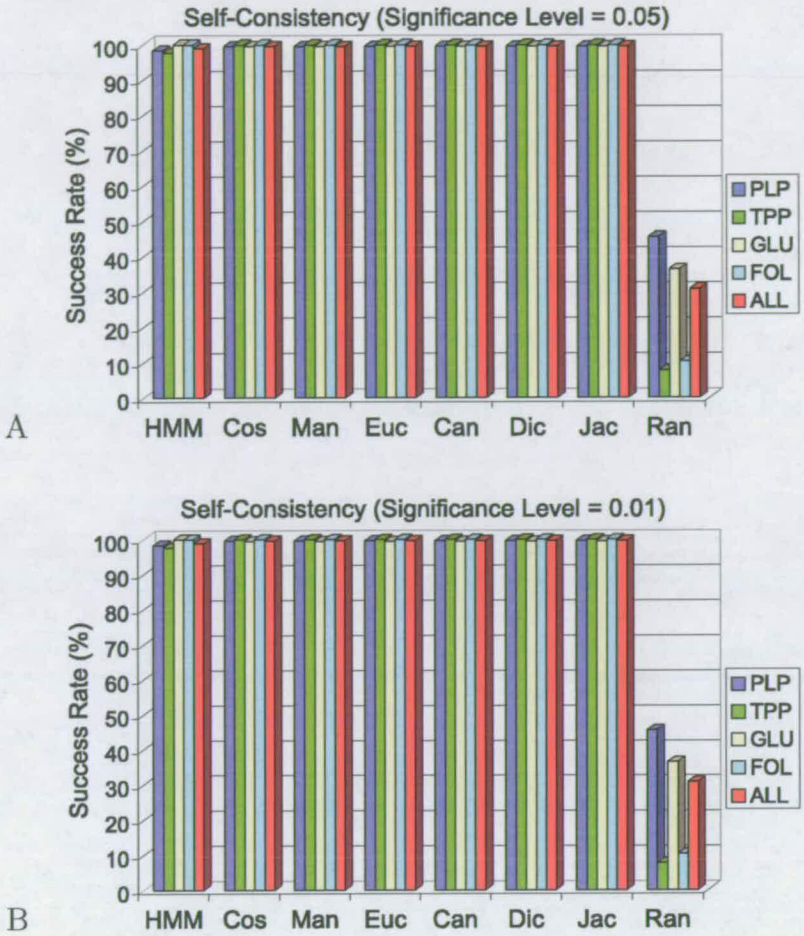


Figure 5.5: Success Rates for the Self-Consistency Tests. **A.** Assessment of the prediction success rates for pHMM, DAROGAN scoring methods and for Random (Ran). Significance level of 0.05 used. **B.** As **A** except at significance level of 0.01 (Key: HMM:hidden Markov Model; Cos:Cosine; Man:Manhattan Euc:Euclidean; Can:Canberra; Dic:Dice, Jac:Jaccard; Ran:Random).

ods, the overall proportion of correct predictions, are all within 5% of each other. The pHMM method showed the highest Accuracy (69.7%) with the Dice and Jaccard measures performing the least well (65.6%). The Error statistic reports the proportion of incorrect predictions and reflects the Accuracy results. The proportion of predictions made that were correct is assessed with the Precision statistic. The Cosine scoring method has the highest Precision level (99.3%), with the Euclidean, Jaccard and Dice measures slightly trailing (98.1%). The Recall statistic

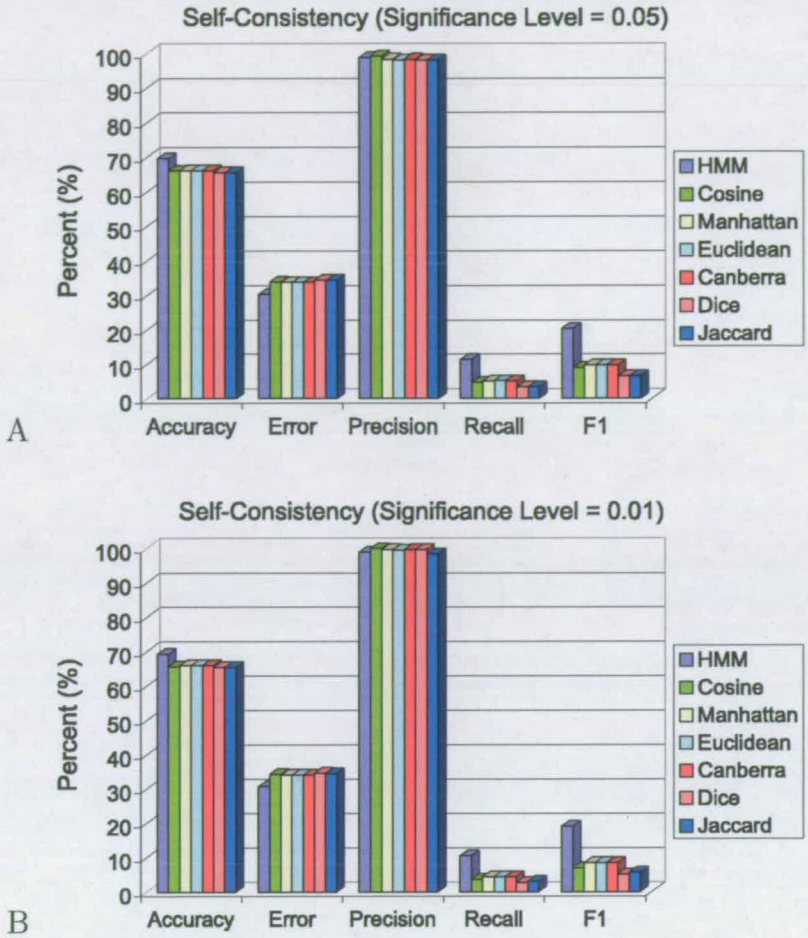


Figure 5.6: Self-Consistency Test Statistics. **A.** The accuracy, error, precision, recall and F1 statistics are shown for the pHMM method and the DAROGAN scoring methods. **B.** As for **A** except at 0.01 significance level.

measures the proportion of the predictions assigned the correct cofactor usage. If a prediction method is not stringent enough Recall will be significantly higher than the Precision; not the case with the pHMM and DAROGAN measures. The opposite is true with Precision significantly higher than Recall, indicating that the prediction methods could be too stringent. The fact that the pHMM and DAROGAN methods have similar Precision and Recall values could also indicate that the Reference database might not be optimal for assessing the methods. The inadequacy of the Reference data could lie with the inclusion of several highly

Correct	$100 * \frac{\text{correct prediction events}}{\text{total prediction events}}$
Accuracy	$\frac{TP + TN}{TP + FP + FN + TN}$
Error	$\frac{FP + FN}{TP + FP + FN + TN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1	$\frac{2*TP}{2*TP + FP + FN}$

Table 5.5: Equations for Resubmission Test Statistics. The F1 statistic is an equal measure of Precision and Recall (Key: TP:true positive; FP:false positive; TN:true negative; FN:false negative).

similar seed sequences used to create alignments for each enzyme family. This along, with other possibilities, is discussed in the chapter conclusions. The final statistic used in this assessment is the F1 measure, used to evaluate how successful predictions were when predictions of cofactor utilisation were made. The F1 measure is an equal measure of the Precision and Recall statistics. A perfect prediction method will make correct predictions and only correct predictions, reflected in high Precision and Recall values, therefore reflected in the F1 value. The pHMM method has the highest F1 measure (19.0%), followed by the Euclidean measure (8.5%). The lowest F1 values are shown by the Dice and Jaccard measures, consistant with all the statistics used.

5.3.2 Jack-Knife Test

The Jack-Knife is a Leave-One-Out-Cross-Validation (LOOCV) test, considered to be one of the best performing of the re-sampling methods available, especially for smaller data sets (R. Molinaro & Pfeiffer, 2005). Molinaro and Pfeiffer describe a comparison of several re-sampling methods (Jack-Knife, Split Sample, v-fold cross-validation, .632+ bootstrap and Monte Carlo cross-validation), with the Jack-Knife comparing favourably in terms of bias and mean squared error. Further to the comparisons the study also found that as the size of the data set increases, the differences in the performances of the methods are reduced. The mathematical principles and detailed discussions of the merits of the Jack-Knife and Self-Consistency tests are beyond the scope of this thesis can be found in the following papers (Mardia *et al.*, 1979; Zhou & Assa-Munt, 2001; Cai, 2001; Chou, 1995; Chou & Elrod, 2003).

The implementation of the Jack-Knife test for the DAROGAN and pHMM methods is outlined in Figure 5.4 (A and B; Right hand side). Each entry (Tread or pHMM) is removed from the Reference database, compared to the remaining entries and then returned to the database. This differs from the Self-Consistency test as each entry is not compared to itself.

The Jack-Knife success rates for the pHMM and DAROGAN scoring methods are shown in Figure 5.7 (Raw data in Appendix A.2). As would be expected the success rates are significantly different from those of the Self-Consistency test. The success rates for each of the methods are divided up by enzyme

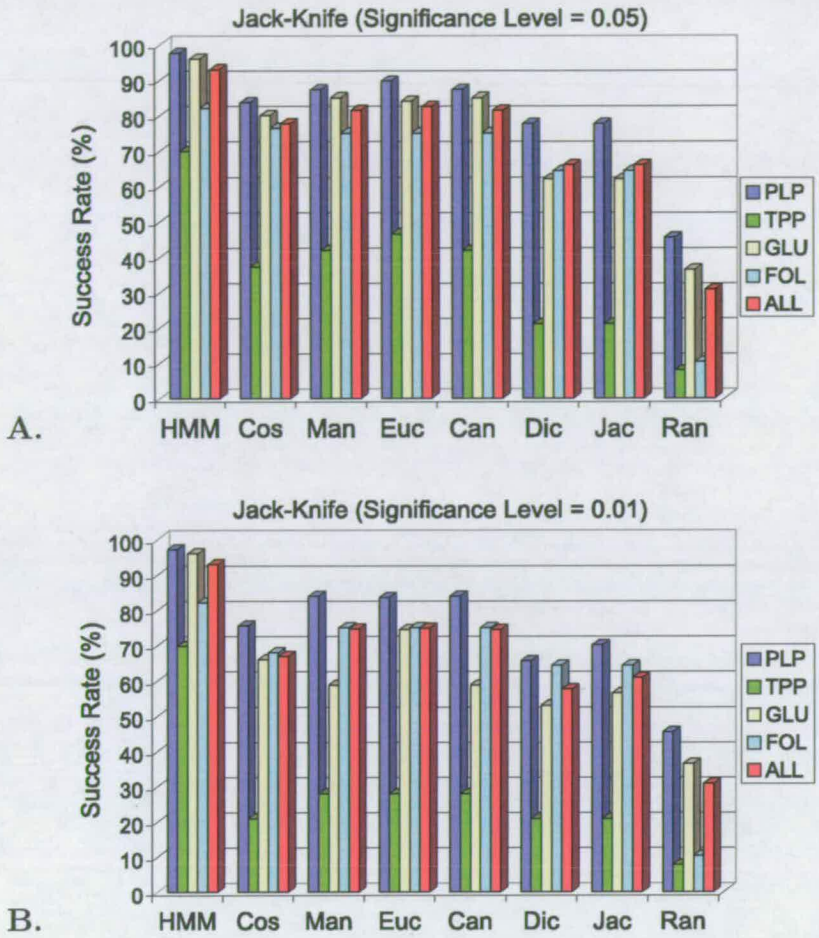


Figure 5.7: Success Rates for the Jack-Knife Tests. **A.** Assessment of the prediction success rates for pHMM, DAROGAN scoring methods and for Random (Ran). Significance level of 0.05 used. **B.** As **A** except at significance level of 0.01 (Key: HMM:hidden Markov Model; Cos:Cosine; Man:Manhattan Euc:Euclidean; Can:Canberra; Dic:Dice, Jac:Jaccard; Ran:Random).

family, showing marked differences between the families. The PLP enzymes are consistently the most successful predicted across all the different methods with the pHMM (97.6%) and Euclidean (89.6%) being the most successful. The GLU enzymes (pHMM; 96.0% and Manhattan; 85.0%) are the next most successfully predicted group, followed by the FOL enzymes (pHMM; 83.6% and Cosine; 74.6%). The TPP enzymes (pHMM; 69.8% and Euclidean; 46.5%) however trail the success rates of the other families by a significant margin. With some of

the methods, particularly Dice and Jaccard, the TPP success rate is below the level expected if they were predicted at random (See Equation 5.1). The TPP enzymes are one of the least popululated of the enzyme families in the Reference database and may not be sufficient to assess the prediction power of the pHMM and DAROGAN methods; Discussed further in the conclusions at the end of the chapter.

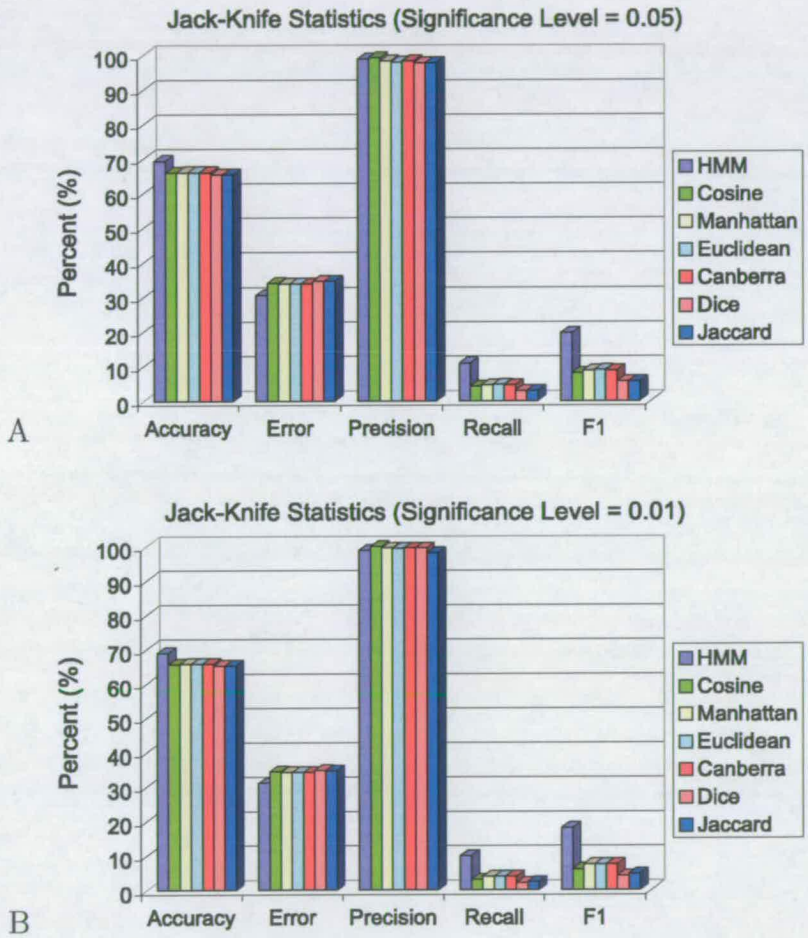


Figure 5.8: Jack-Knife Test Statistics. **A.** The accuracy, error, precision, recall and F1 statistics are shown for the pHMM method and the DAROGAN scoring methods. **B.** As for **A** except at 0.05 significance level.

In a more detailed analysis of the pHMM and DAROGAN methods using the Ac-

curacy, Error, Recall, Precision, and F1 statistics, shown in Figure 5.8 (Raw data in Appendix A.2), the differences to the Self-Consistency test are less evident. The pHMM method shows the highest Accuracy (69.5%), with the DAROGAN scoring methods closely following (Mahnattan, Euclidean, Canberra: 66.1%). The Precision statistic is highest in the Cosine method (99.2%) followed by the pHMM method (98.8%) and then the rest of the DAROGAN methods. The Recall statistic was highest in the pHMM method (10.9%), with the Euclidean (4.7%) method marginally out performing the other DAROGAN scoring methods. The F1 statistic, an equal measure of Precision and Recall, was again the highest in the pHMM method (19.6%), the DAROGAN methods performed less well, with Euclidean the highest scoring (8.9%). As with the Self-Consistency test the Dice and Jaccard methods were consistently the worst performing of the DAROGAN scoring methods.

5.3.3 Culled Sequence Sets Vs. Prediction Results

Up to this point the DAROGAN and HMM prediction methods have been largely assessed against the un-culled enzyme families. This section will discuss how the utilisation of the culled sets of sequences for each of the cofactor utilising families affect the prediction results. The purpose of these analyses was to determine the most appropriate sequence identity cut offs for the PISCES cull, significance level and scoring method to utilise in the DAROGAN method.

Firstly the Accuracy, Recall, Precision and F1 statistics are applied to each of the PISCES culled sequence sets (100%, 75%, 50%, 25%). This analysis was

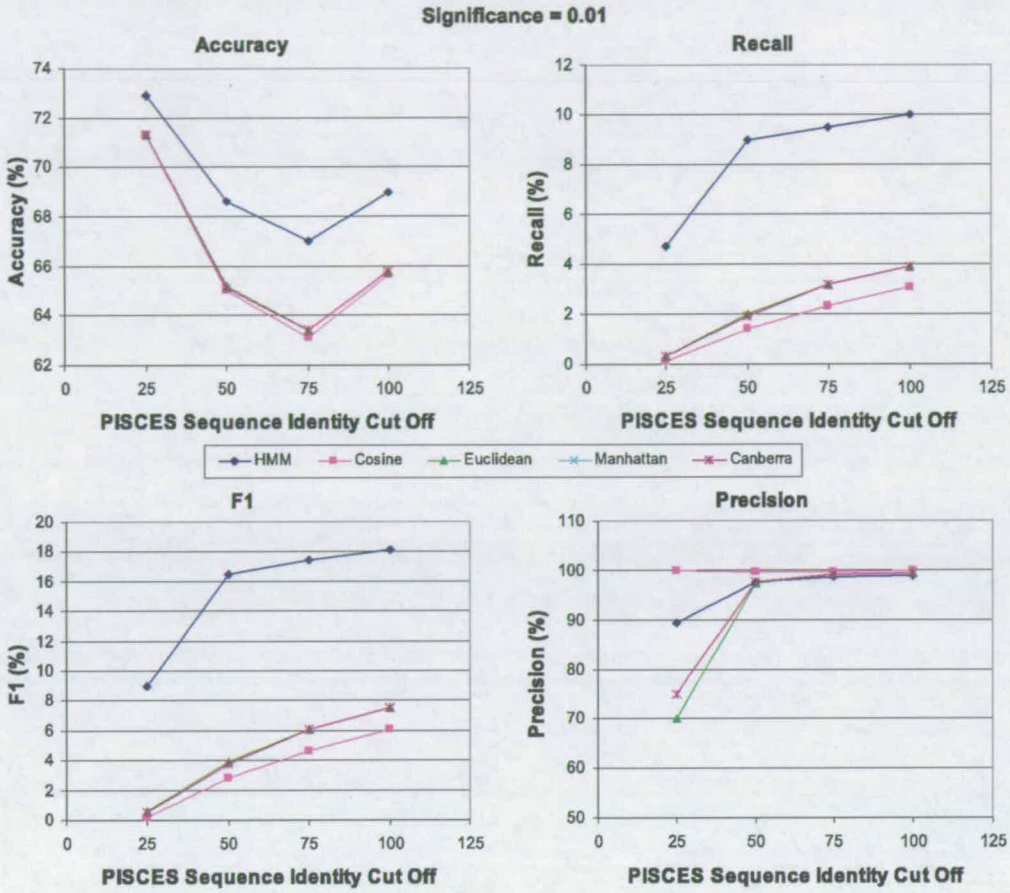


Figure 5.9: PISCES sequence identity cut off affect on the function prediction statistics; Accuracy, Precision, Recall and F1. N.B Error statistic was not included as it is represented by the Accuracy statistic (i.e. Error = 100-Accuracy).

run at two different significance levels; 0.01 and 0.05 (See Figures 5.9 and 5.10 respectively), to aid identification of the most suitable level.

The plots show, for both the 0.01 and 0.05 significance levels, that the Accuracy values decrease with increasing PISCES cut off value; with the 100% sequence identity value deviating from this trend. The three other statistical measures (Recall, Precision, F1) all show the opposite trend to the Accuracy measure. In these measures the trend is that each of the measures increases

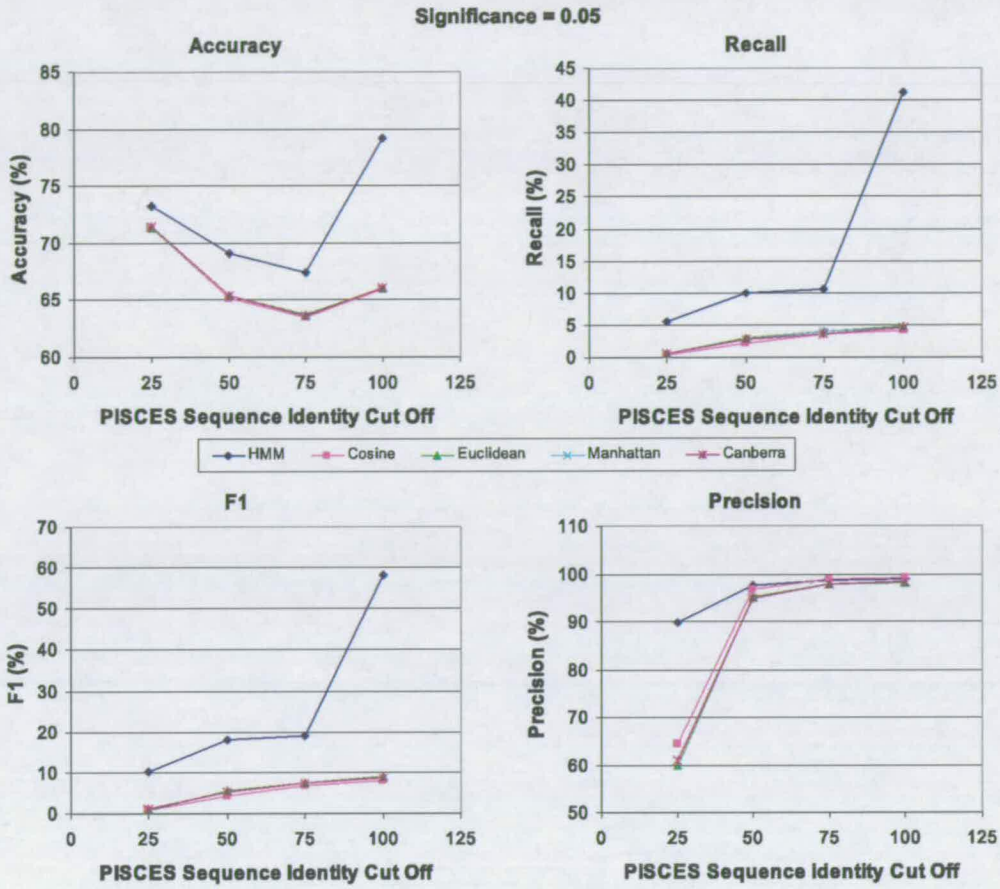


Figure 5.10: PISCES sequence identity cut off affect on the function prediction statistics; Accuracy, Precision, Recall and F1. N.B Error statistic was not included as it is represented by the Accuracy statistic (i.e. Error = 100-Accuracy).

with increasing PISCES cut off value. This indicates that the inclusion of more similar sequences aids in the maximisation of the Recall, Precision and F1 values.

Comparing the values for each of the DAROGAN scoring methods shows the Euclidean, Manhattan and Canberra measures to perform with similar Accuracy, Recall, Precision and F1 values. However the Cosine measure is seen to perform less well than the other measures; this is most obvious in the 0.01 significance level plots (Figure 5.9), but is also true in the 0.05 significance level plots (Figure 5.10).

The plots for the different significance levels show similar overall trends for each of the statistical measures, however the values for the PISCES 100% sequence identity cut off (i.e. no culling) are much higher in the 0.05 significance level plots. Also, for all the PISCES cut off values, the values for Recall, Precision and F1 are higher in the 0.05 significance level than the 0.01 plots. This suggests that the significance level to best suit the DAROGAN prediction is at 0.05 rather than 0.01. The improvement in the F1 statistic values alone are enough to support this choice as the maximisation of the Precision and Recall values can be achieved with out improving the predictive power of the method. At the 0.01 significance cut off fewer predictions are made as it is a more stringent cut off than the 0.05 level and these non-predictions are treated just as harshly as a false positive prediction. An increased number of sequences and increased sequence identity within the enzyme groups aids HMM and DAROGAN prediction methods.

ROC (Receiver-Operator Characteristic) Curves

To aid in the determination of the most suitable PISCES sequence identity cut off to utilise in DAROGAN predictions, ROC curves were plotted (See Figure 5.11). This analysis was run at the 0.05% significancy level as suggested as the optimal value in the previous section. The data for each of the DAROGAN scoring methods was plotted individually and for each of the different PISCES cut off values. In ROC plots a good prediction method is seen to *hug* the y-axis at $y = 0$ and the x-axis at $y = max$; also the plot should not fall below the 45°

line where Recall is equal to 1-Precision[1].

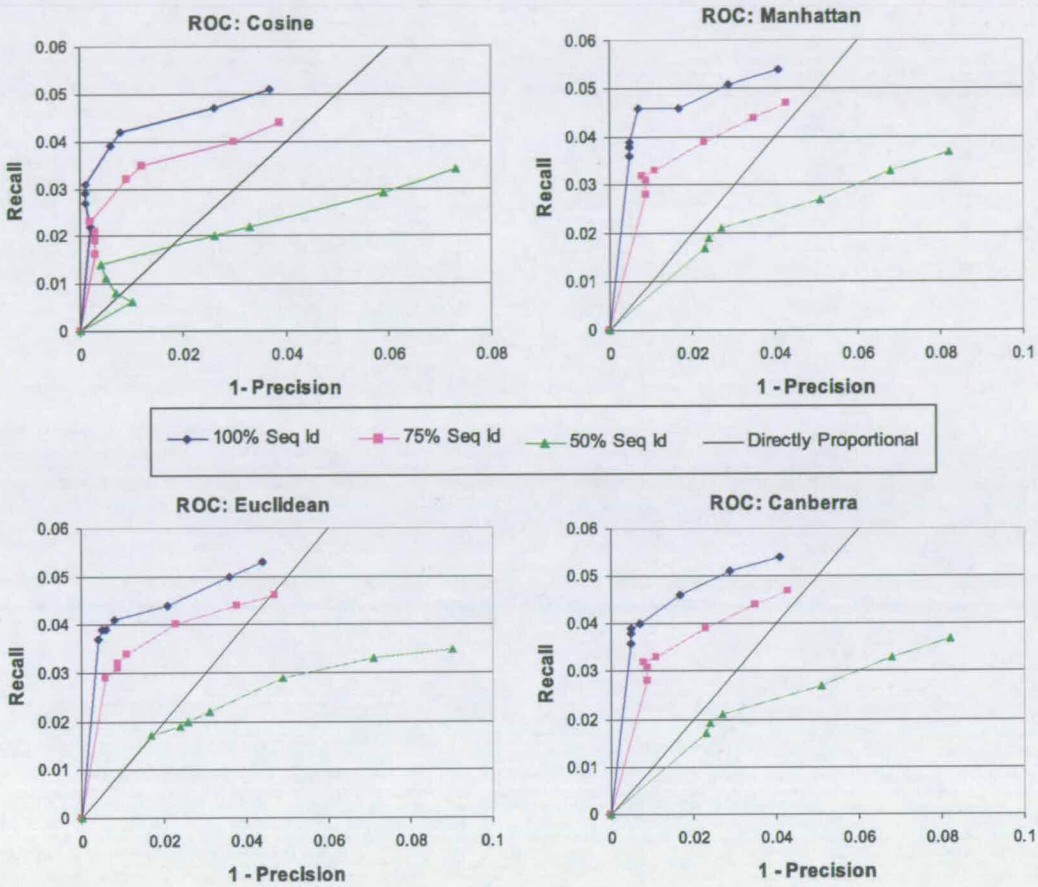


Figure 5.11: Receiver-Operator Characteristic (ROC) curves for the Cosine, Manhattan, Euclidean and Canberra scoring methods in the DAROGAN method. Recall and Precision values are plotted at the various PISCES cull cut off values (100%, 75%, 50%, 25%).

The ROC plots, in all cases show that the 50% PISCES cut off value is not suitable for use in predictions as it falls below the 45° line. This was also true for the 25% cut off value, but was so far below the line that is was not included in the plots. This leave the 75% and the 100% PISCES cut offs to be considered for their sutialbility for use in the DAROGAN method. The ROC plots indicate that both would be suitable, with the 100% cut off preferable over the 75%

[1]N.B. This line has been included in Figure 5.11 for reference

sequence identity cut off.

Comparisons between the different scoring methods indicate that the ordering of the methods by performance should be:

```
Canberra > Manhattan > Euclidean > Cosine >> Jaccard & Dice
```

5.4 Conclusions

The Self-Consistency and Jack-Knife tests have allowed the DAROGAN method, with its different scoring schemes to be compared to an established prediction method; pHMMs. The tests however, are biased towards the pHMM method as the preservation of the order of the amino acid in the pHMMs is an advantage here. The application of the DAROGAN method in the prediction of cofactor utilisation through use of conserved functional residues was designed to be order independent, and utilised in cases where sequence identity methods would fail (e.g. similar mechanisms of action, but different three dimensional fold). Putative enzymes not to have had their functions successfully predicted through sequence similarity methods, such as pHMMs, are the target for the DAROGAN method. Despite the different application areas of the pHMM and DAROGAN methods, meaningful conclusions can still be drawn from the comparisons.

The Self-Consistency test highlights the over stringency of both methods, shown in the Precision values being significantly higher than those of Recall. As previously mentioned this is likely to be due to limitations in the Reference data.

The Reference data are divided by the high level categories of cofactor utilising enzyme families. Further dividing the families into subclasses (e.g. cofactor usage type or structural relatedness) could improve the prediction success. Recall measures the proportion of an enzyme family predicted to be a member of that family in the predictions. The low value suggests that the prediction methods, to some extent are able to distinguish enzyme family subsets, where the Self-Consistency test does not.

The Jack-Knife test allows the selection of the best performing of the DAROGAN scoring methods. The Euclidean method has the highest F1 statistic value, from the DAROGAN methods, albeit by a small margin. Both the Self-Consistency and Jack-Knife tests, with the exception of some of the TPP enzyme predictions, have shown the DAROGAN method to have a higher prediction success than would be expected for weighted random. This suggests Treads are correlated with enzyme cofactor utilisation. Therefore if a good Reference set of Treads are available, successful predictions can be made for cofactor utilisation. If an enzyme is incorrectly predicted to utilise a cofactor, this may not indicate a failing of the prediction method. The enzyme in question could just not be well represented in the limited number of Reference Treads. This problem can be minimised by adding further enzymes to the Reference Treads, especially to the smaller enzyme families. However a Reference Tread set encompassing all enzyme cofactor utilisation space is unlikely to be achievable until the sequence databases are completed.

The effect of performing PISCES culls of sequence identity with in each of the enzyme family groups was explored in respect to the Accuracy, Recall, Precision and F1 statistical measures. The main conclusions from this analysis was that the less stringent statistical significance level of 0.05 should be utilised in predictions over the 0.01 level. This was most evident in the higher F1 statistic obtained using the 0.05 level rather than 0.01. To determine the most appropriate PISCES cull level to use in predictions, Receiver Operator Characteristic plots were generated for each of the DAROGAN scoring methods. The plots showed that the most appropriate PISCES cull threshold is the 100% sequence identity (i.e. no culling performed). In addition to this cull level the 75% level would also suitable for performing predictions, however not as sucessfully as the 100% level.

In the Jack-Knife tests the Euclidean scoring method was found to be marginally the best of the DAROGAN scoring methods, closely followed by the Manhattan and Canberra scores. The analyses with the PISCES cull levels and the ROC curves also found the same top three scoring methods although the Canberra method scored slightly higher. The Cosine method in all tests, was found to perform less well than the top three methods, however not by a large margin. With this in mind it would be prudent to utilise either the Euclidean or Canberra scoring methods for the predictions.

An additional, and more trivial, consideration in the selection of the best DAROGAN scoring method is the CPU time required to search a query Tread against

Measure	Jack-knife (Approx. Secs.)
Cosine	5
Manhattan	80
Euclidean	75
Canberra	150
Dice	230
Jaccard	215

Table 5.6: Similarity Measure Timings for the Jack-Knife test (150x149 comparisons)

the Reference database. Table 5.6 shows the approximate timings for a Jack-Knife test (150x149 comparisons). The timings show the Cosine method to be the fastest; approximately fifteen times faster than the Manhattan and Euclidean methods. Therefore for a fast prediction run, the Cosine method would be a more prudent choice. However the best performing, but slower, methods remain the Euclidean or Canberra methods.

Chapter 6

EcoCyc Application

6.1 Overview

The DAROGAN function prediction method has been applied to perform a *real life* prediction. At this stage this is done mostly as an example and to highlight a typical application area for the DAROGAN method once is has fully matured.

The EcoCyc resource, part of the BioCyc project, has provided a list of enzymes known to be present in *E. coli*, but have either no gene associated with them or no sequence associated with their gene. Several of these enzymes utilise pyridoxal-5'-phosphate (PLP) and it is to these enzymes the function prediction method has been applied. A set of *E. coli* sequences where there are known homologues, but with no clear function have been made available on the GeneQuiz web server. It is this set of 511 sequences that DAROGAN has used to propose candidates for the seven PLP utilising enzymes described by EcoCyc.

Firstly multiple sequence alignments containing approximately fifteen functionally conserved residues were generated for each of the GeneQuiz sequences. Each of the alignments was then run against the Reference Tread database to produce two lists of candidates for the seven PLP utilising enzymes; one using the Canberra scoring method and the other using the Euclidean scoring method.

6.2 BioCyc

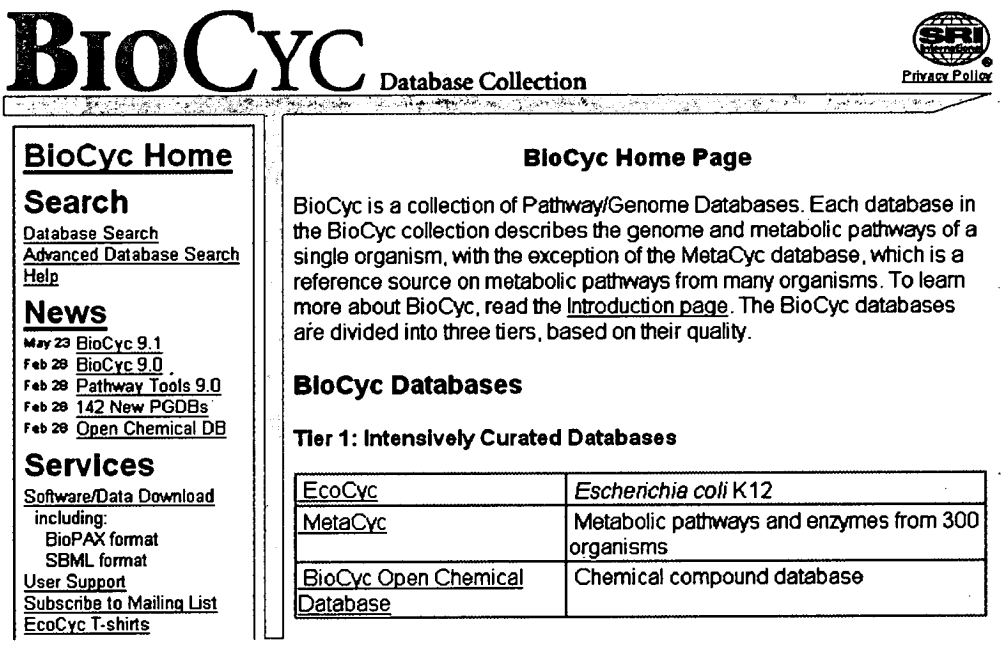


Figure 6.1: BioCyc Screen Shot from http://www.biocyc.org/

The BioCyc knowledge library (Karp *et al.*, 2005; BioCyc, 2005) is a collection of databases designed to provide a central resource for a wide range of individual microorganisms. The data stored within the databases are very diverse and range from gene location on chromosomes, to the structures of compounds

involved in biochemical reactions. The advantage of storing such a range of data is that it allows researchers to easily navigate through the data available through a single interface.

Within the BioCyc family of databases are the MetaCyc and Pathologic databases. MetaCyc databases are pathway/genome databases for particular organisms, with experimentally determined metabolic pathway information. In contrast to this the Pathologic databases contain genome data for specific organisms, where gaps in the metabolic information have been predicted computationally. The EcoCyc section of BioCyc is a MetaCyc database so only experimentally inferred data is included for the *E. coli* K12 genome.

6.3 EcoCyc

The goal of EcoCyc (Karp *et al.*, 1999; EcoCyc, 2005) is to provide a description of the *E. coli* organism in biochemical detail. The *E. coli* system has had a large proportion of its biochemical pathways determined experimentally, making it particularly suitable for a resource such as EcoCyc.

Of specific interest to this project, the EcoCyc databases contain descriptions of all biochemical reactions known to occur in the *E. coli* cell. In most cases the enzymes catalysing the reactions, and the genes encoding them, have been experimentally identified. However there are a small set of enzymes known to be present in *E. coli*, which have either no gene associated with them or have

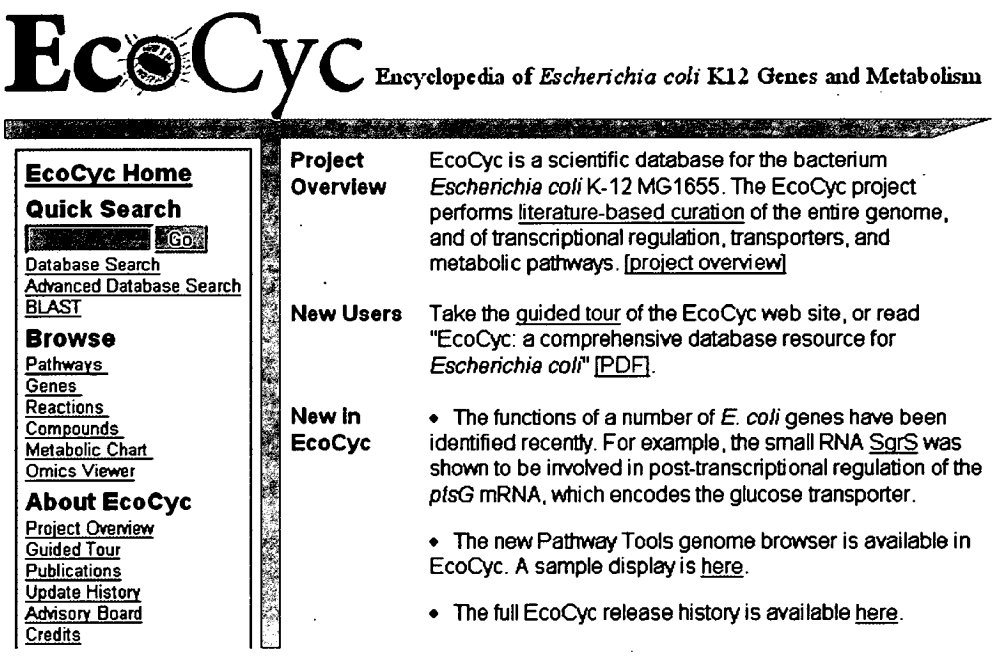


Figure 6.2: EcoCyc Screen Shot from http://ecocyc.PangeaSystems.com/ecocyc

no sequence associated with their gene. The list of enzymes is under constant revision, in January 2002 there were 64 of these enzymes, reduced to 55 in the most recent version, October 2003. The enzymes cover a wide range of biochemical reactions, and seven of them are thought to be PLP dependent[1] (See Table 6.1).

6.4 GeneQuiz

The GeneQuiz project (Hoersch *et al.*, 2000) aims to perform large scale assignment of biochemical functions to sequences from entire genomes. GeneQuiz contains data for several genomes including that of *E. coli*. Functional assign-

[1]N.B. Some of these reaction may involve multiple steps

Description	EC Number
N-succinyldiaminopimelate-aminotransferase	2.6.1.17
putrescine transaminase / diamine transaminase	2.6.1.29
pyridoxamine-oxaloacetate transaminase	2.6.1.31
histidine transaminase	2.6.1.38
pyridoxamine-phosphate transaminase	2.6.1.54
L-cysteine desulfhydrase / cystathione gamma-lyase / homoserine deaminase	4.4.1.1
D-cysteine desulfhydrase / 3-chloro-D-alanine dehydrochlorinase	4.4.1.15

Table 6.1: The seven PLP utilising enzymes thought to be present in *E. coli*, but either have no sequence associated with their gene or no gene associated with their sequence (29th October 2003 Release; most recent update as of January 2006).

ments have been made to the protein sequences predicted to be encoded by the *E. coli* genome. GeneQuiz sequences are either from expressed protein sequences or those predicted from open reading frames (ORFs). There are five classes of functional assignments for the 4289 *E. coli* sequences in GeneQuiz. The first is where there is a three dimensional structure for the protein, either determined experimentally or modelled by homology. The second class is where there is a clear function derived from probable close homologues. The third class is where the function has been assigned from tentative homologues. The fourth class is where there are homologues to the sequence, but the homologues have themselves got uncertain functional assignments. The final class is where there are no homologues, so no functional role assignment can be made by homology. The distribution of the 511 *E. coli* sequences over the five classes is shown in Figure 6.3.

It is the 511 sequences, for which there are homologues with uncertain function, that are suitable for function prediction using the DAROGAN method[2].

[2]The 511 sequences are freely available on the GeneQuiz web site (GeneQuiz, 2005)

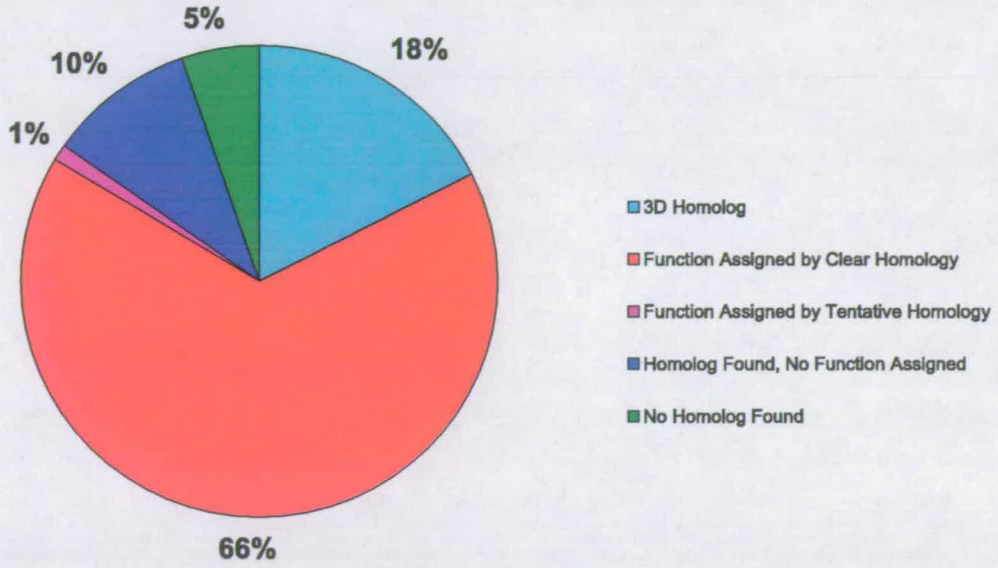


Figure 6.3: Pie chart showing the five classes of functional assignments for the 4289 *E. coli K12* sequences in GeneQuiz

6.5 Example Application of the Function Prediction Method

Before the actual function prediction could be performed on the 511 sequences downloaded from the GeneQuiz web site, first optimum alignments had to be generated for each of the sequences. The same intelligent alignment generating and selection heuristic as utilised in Tread creation was used to produce the optimum alignments (See Section 3.2.2). The definition of an optimum alignment is based on the number of functionally conserved residues (KRENDYCHQST) appearing in the alignment, with the optimum number being approximately fifteen. The number of functionally conserved residues in an alignment can be tailored by altering the E-value cut off in a BLAST search used to produce a set of sequences subsequently aligned using a multiple sequence alignment program.

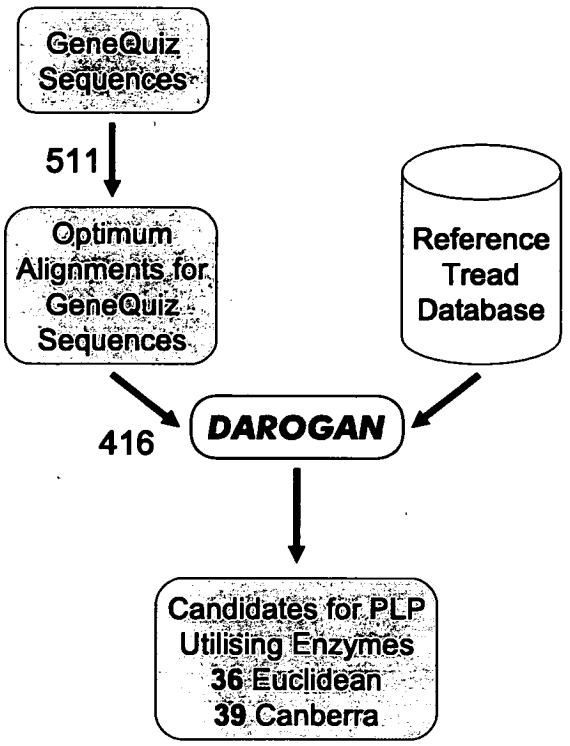


Figure 6.4: Flow diagram showing the steps involved in using the DAROGAN method to produce lists of candidates for the 7 missing EcoCyc enzymes

Figure 6.4 shows a flow diagram for the function prediction of the GeneQuiz sequences using DAROGAN. Unfortunately optimum alignments could not be generated for a small number of the 511 sequences. Despite trying a wide range of E-value cut offs and with a less stringent range of functionally conserved residue requirement it was not possible to generate optimum alignments. In all, 95 (approx. 18%) sequences could not have an optimum alignment generated for them. So the 511 set of sequences for function prediction was reduced to 416.

Once the optimum alignments had been generated, the DAROGAN function prediction method could be applied. The web version of DAROGAN is unsuitable

for the function prediction of batches of putative enzymes, so a command line version of the method was developed for processing large numbers of alignments. The method is exactly the same as for the web version of DAROGAN; except producing a single output file summarising the function prediction results for each of the input alignments. The statistical significance threshold for the Tread matches was set at 0.05, and the PISCES sequence identity threshold set at 100% as suggested by the analyses in Chapter 5. The predictions were also run for both the Canberra and Euclidean scoring methods, also suggested by the analyses in Chapter 5. Processing the batch of 416 alignments required approx. 27 mins for the Canberra run and approx. 13 mins for the Euclidean run[3].

6.6 Prediction Results

For the 416 GeneQuiz alignments there were 39 significant hits for the Canberra run and 36 for the Euclidean run to Reference Treads for PLP utilising enzymes. The candidate lists for the Canberra and Euclidean runs are shown in Table 6.2 and Table 6.3 respectively. The full outputs for the prediction runs can be found in Appendix B.

The purpose of running the DAROGAN function prediction method with both the Canberra and Euclidean scoring methods was to place more confidence in the predictions by assessing the overlap between the two scoring methods. The

[3]CPU time for 1GHz processor, 256Mb RAM

Canberra (Significance:0.05, PISCES:100%)

Rank	GeneQuiz Details			Tread	Score	Sig.	No.
	Seq ID	*E-Value*	*DB*	Top Hit			Hits
1	1790609	10^{-60}	NR	Q8P5R4	0.844	0.00012	5
2	1787254	10^{-50}	NR	Q8YU96	0.844	0.00012	10
3	1789878	10^{-10}	NR	Q8XV80	0.840	0.00016	3
4	1788458	10^{-15}	NR	Q7NL03	0.836	0.00021	9
5	1787395	10^{-35}	NR	P57289	0.820	0.00050	1
6	1789439	10^{-15}	SP	O54694	0.814	0.00068	4
7	1790663	10^{-10}	NR	O67687	0.810	0.00082	4
8	1790765	10^{-10}	NR	P63479	0.799	0.00134	7
9	1790442	10^{-30}	NR	Q44686	0.781	0.00258	1
10	1787408	10^{-25}	NR	Q84I53	0.688	0.02571	3
11	1790461	10^{-40}	NR	Q89AX7	0.688	0.02571	6
12	1787219	10^{-25}	NR	Q89AX7	0.688	0.02571	6
13	1788093	10^{-20}	NR	P52069	0.682	0.02851	1
14	1788636	10^{-30}	SP	P77690	0.680	0.02967	1
15	1788803	10^{-5}	SP	O13326	0.680	0.02967	1
16	1789591	10^{-25}	NR	Q8Z4W1	0.680	0.02967	1
17	1786795	10^{-30}	NR	P05459	0.676	0.03173	4
18	1787057	10^{-60}	NR	P29012	0.676	0.03161	7
19	1788490	10^{-50}	NR	Q8P5R4	0.675	0.03237	4
20	1787071	10^{-5}	SP	P47176	0.673	0.03361	1
21	1789925	10^{-15}	NR	P49725	0.671	0.03443	1
22	1788440	10^{-45}	NR	Q58466	0.670	0.03533	1
23	1790855	10^{-30}	NR	P53206	0.662	0.04017	1
24	1789985	10^{-10}	NR	Q8QZR5	0.662	0.04018	1
25	1790671	10^{-5}	SP	P23279	0.662	0.04018	2
26	1787831	10^{-15}	NR	Q8QZR5	0.662	0.04018	1
27	1786830	10^{-35}	NR	Q8NT73	0.662	0.04018	1
28	1788770	10^{-35}	NR	P14173	0.662	0.04018	3
29	1786990	10^{-35}	NR	O32148	0.662	0.04017	1
30	1789994	10^{-20}	NR	Q51687	0.662	0.04018	3
31	1789882	10^{-15}	NR	Q8GYY0	0.658	0.04283	1
32	1789251	10^{-20}	NR	O32148	0.658	0.04258	1
33	1789493	10^{-30}	NR	Q44688	0.657	0.04317	1
34	1790825	10^{-25}	NR	Q8XV80	0.656	0.04403	1
35	1786730	10^{-25}	NR	Q8GYY0	0.656	0.04403	1
36	1789158	10^{-35}	NR	O58489	0.655	0.04492	2
37	1790248	10^{-10}	NR	Q16773	0.652	0.04720	1
38	1790307	10^{-10}	NR	Q55128	0.652	0.04720	1
39	1787465	10^{-60}	NR	Q44004	0.650	0.04830	1

Table 6.2: Summary of the output for the Canberra scoring method scoring prediction run. A full output can be found in Appendix B.

two lists share 29 common hits, representing 74% of the Canberra and 81% of the Euclidean hits. Comparing the lists also reveals that there are 46 unique hits in total over both lists, the statistical significance values of the common hits are largely similar between the two lists. The differences in the two scoring methods lies with the Canberra method being more sensitive to small changes in the Tread vector values when they are near zero. This difference explains why the lists are not identical, but share approx. 75% of hits. However the 7 EcoCyc enzymes must have had their sequences experimentally verified before this assessment can

Euclidean (Significance:0.05, PISCES:100%)

Rank	GeneQuiz Details			Tread	Score	Sig.	No.
	Seq ID	*E-Value*	*DB*	Top Hit			Hits
1	1790609	10^{-60}	NR	Q8P5R4	0.779	0.00018	5
2	1787254	10^{-50}	NR	Q8YU96	0.779	0.00018	17
3	1790765	10^{-10}	NR	P63479	0.779	0.00018	10
4	1789878	10^{-10}	NR	Q8XV80	0.774	0.00034	3
5	1788458	10^{-15}	NR	Q7NL03	0.768	0.00060	15
6	1790663	10^{-10}	NR	O67687	0.766	0.00069	2
7	1787395	10^{-35}	NR	P57289	0.746	0.00269	1
8	1790442	10^{-30}	NR	Q44686	0.744	0.00303	1
9	1789439	10^{-15}	SP	O54694	0.737	0.00426	4
10	1788093	10^{-20}	NR	P52069	0.728	0.00673	1
11	1789994	10^{-20}	NR	Q51687	0.714	0.01153	3
12	1789251	10^{-20}	NR	O32148	0.709	0.01404	1
13	1787057	10^{-60}	NR	O67687	0.703	0.01745	4
14	1788490	10^{-50}	NR	O96567	0.702	0.01765	1
15	1787408	10^{-25}	NR	Q8QZR1	0.696	0.02182	2
16	1786830	10^{-35}	NR	P05459	0.696	0.02169	4
17	1786928	10^{-35}	NR	Q8PBK7	0.696	0.02169	1
18	1787219	10^{-25}	NR	P29012	0.692	0.02420	5
19	1790461	10^{-40}	NR	P29012	0.692	0.02420	5
20	1786795	10^{-30}	NR	P05459	0.691	0.02492	4
21	1789496	10^{-15}	NR	Q59447	0.685	0.03007	3
22	1786990	10^{-35}	NR	O32148	0.685	0.03017	1
23	1788440	10^{-45}	NR	O32148	0.684	0.03089	2
24	1789291	10^{-40}	NR	Q05998	0.684	0.03107	11
25	1788636	10^{-30}	SP	P77690	0.680	0.03461	1
26	1789591	10^{-25}	NR	Q8Z4W1	0.680	0.03461	1
27	1790825	10^{-25}	NR	Q8XV80	0.677	0.03768	1
28	1789493	10^{-30}	NR	Q44688	0.675	0.03881	1
29	1790589	10^{-50}	NR	Q8ZYF9	0.673	0.04187	29
30	1789925	10^{-15}	NR	P49725	0.671	0.04327	1
31	1790671	10^{-5}	SP	Q12198	0.668	0.04715	1
32	1789158	10^{-35}	NR	O58489	0.668	0.04715	3
33	1790837	10^{-30}	NR	Q8QZR5	0.668	0.04715	1
34	1787465	10^{-60}	NR	Q44004	0.668	0.04715	1
35	1786840	10^{-10}	NR	P63479	0.667	0.04858	1
36	1789176	10^{-25}	NR	Q87JS8	0.667	0.04858	1

Table 6.3: Summary of the output for the Euclidean scoring method scoring prediction run. A full output can be found in Appendix B.

be undertaken.

6.7 Concluding Remarks

DAROGAN has been implemented to produce two sets of candidates, using the Canberra and Euclidean scoring methods, for the 7 EcoCyc missing enzymes. However the candidates can only be verified by experimentally characterisation. It is unlikely all 7 missing EcoCyc functions will be present in the 511 GeneQuiz sequences, characterised as having homologues with uncertain function. Some of

the missing functions could be present in the set of 279 GeneQuiz sequences with no homologues (See Figure 6.3), not explored in this analysis as the DAROGAN method requires homologous sequences to create a query Tread. Also the missing functions may not actually be present in the *E. coli* cell at all, the functions deemed missing could be based on inaccurate experimental evidence. So searching for candidates for some of these missing functions could be a futile endeavour.

Experimentally determining the function of an enzyme, particularly if there are no clues to the function, is a time consuming and expensive exercise. A computational method capable of proposing small sets of candidates for a particular enzyme function from a large set of possible protein sequences would greatly aid wet lab biologists in their effort to identify missing enzyme functions. The DAROGAN method has been shown to be capable of producing such candidate lists, but the full potential of the method can only be determined when the functions of the candidate sequences have been experimentally determined. Comparison with the pHMMs in Chapter 5 showed that further refinement is required before predictions can be confidently made.

Chapter 7

Conclusions

7.1 Method Development

This thesis began with a review of the field of protein function prediction and introduced the main aim of the project; to develop a novel enzyme function prediction method and explore, through an example of the EcoCyc project, the feasibility of the method becoming an addition to the existing function prediction tools. The DAROGAN method was designed with a specific area of function prediction in mind; where there are examples of enzymes with different tertiary structures, but utilising the same functional residues in the catalytic mechanism. The method uses Treads, storing information on these conserved functional residues (KRENDYCHQST) independent of order and the overall fold of the enzyme. By comparing Reference Treads from known enzymes to a Tread for a query putative enzyme, functional similarity can be inferred.

On a more practical level an important goal was to ensure the method was

as fully automated as possible. Ultimately the method should be able to perform predictions with as little human intervention as possible. Where human intervention is required, as with the assignment of functional roles to the residues appearing in Treads, the method has been adapted to also be functional when bypassing these steps. Also solutions have been proposed to attempt to automate these steps and are discussed below in Section 7.4.3.

The implementation of a web service for the DAROGAN function was also presented to allow researchers to use the method on sequences before experimentally verifying the prediction.

7.2 Method Evaluation and Comparison to pHMMs

To evaluate the DAROGAN function prediction method an existing method used for enzyme function prediction was used. The choice for a comparison was complicated by the fact that the DAROGAN method was designed for a particular niche of function prediction not directly covered by other methods. However it was decided that pHMMs would be suitable as they, like DAROGAN, rely on multiple sequence alignments. This meant both methods could utilise the same alignments and be subject to the same limitations of the reference data (e.g. limited coverage).

Several statistical measures were employed in the evaluation including the

Self-Consistency and Jack-Knife tests. In addition to the comparison to the pHMMs these tests also allowed the optimum parameters for the DAROGAN method to be determined. The first parameter to be considered was the statistical significance level to utilise; determined to be the 0.05 level over 0.01, as the 0.01 level proved too stringent. The second parameter was the selection of the best scoring measure to use for Tread vector similarity; the Canberra and Euclidean measures were found to outperform the other similarity measures. Finally the Reference Tread data was subjected to a PISCES cull, where sequences (and hence Treads) with sequence identities above a certain threshold are culled. It was found that the predictions are best with no culling performed, followed by a cull at a threshold of 75% sequence identity. However the low number of Treads in the Reference database are likely to adversly affect this result.

The comparison with the pHMMs was not ideal as the pHMMs were expected to outperform the DAROGAN method as the Self-Consistency and Jack-Knife tests allowed the pHMMs to take full advantage of the order of the residues in the alignments. In the intended application of the DAROGAN method this would not be possible. However the DAROGAN method was able to perform reasonable well against the pHMMs, with comparable success rates, Accuracy, Error, Precision, Recall and F1 statistics. This is all the more impressive considering the level of information stored in each Tread compared to that of a pHMM.

7.3 EcoCyc Application

In Chapter 6 an example of the application area for the DAROGAN method using the EcoCyc and GeneQuiz web resources. In the EcoCyc project seven PLP utilising enzymes have been identified, in *E. coli K12*, that either have no gene associated with them or no sequence associated with their gene. The DAROGAN method was applied to the 511 GeneQuiz sequences characterised as having homologues with uncertain function. From the 511 sequences DAROGAN was able to produce candidates lists with 36 and 39 (for the Canberra and Euclidean scoring methods) sequences predicted to utilise pyridoxal-5'phosphate as a cofactor.

The experimental characterisation of proteins is a costly and time consuming endeavour for a biologist so computational method capable of proposing small sets of candidates for a particular enzyme function from a large set of possible protein sequences would greatly aid wet lab biologists in their effort to identify missing enzyme functions.

7.4 Future Directions

7.4.1 User Sequence Input

To perform a function prediction at present the user must submit a pre-calculated multiple sequence alignment for their putative enzyme sequence of interest. Ideally the user would submit just their sequence, without the need for performing

a BLAST search followed by the alignment of the sequences, to yield an alignment with approx. fifteen functionally conserved residues. Unfortunately this is computationally intensive, even with the intelligent alignment generation and selection heuristic (See Section 3.2.2). Future versions of DAROGAN will allow a user to submit their sequence along with their email address, allowing the results to be viewed once calculated. This would be preferably to the current system as the quality of the users input sequence alignment is crucial the the success of the success of the function prediction.

7.4.2 Viewing Treads

In future versions of DAROGAN, principal component analysis techniques could be used to reduce the dimensionality of Tread space, currently twenty dimensions, to two and three dimensions. Once the number of dimensions has been reduced it will be possible to visualise the clustering of Treads within Tread space giving a more intuitive view of the relationship between the Treads (Figure 7.1). The visualisation of the Treads in 2D/3D could be accomplished through the use of JavaApplets, having the advantage of being available over the Internet.

In addition to providing a visual representation of how the Reference Treads cluster in 2D/3D, the populations of each of the clusters can be studied. Underpopulated clusters could easily be identified and additional Reference Treads added to swell the numbers of Treads in these clusters.

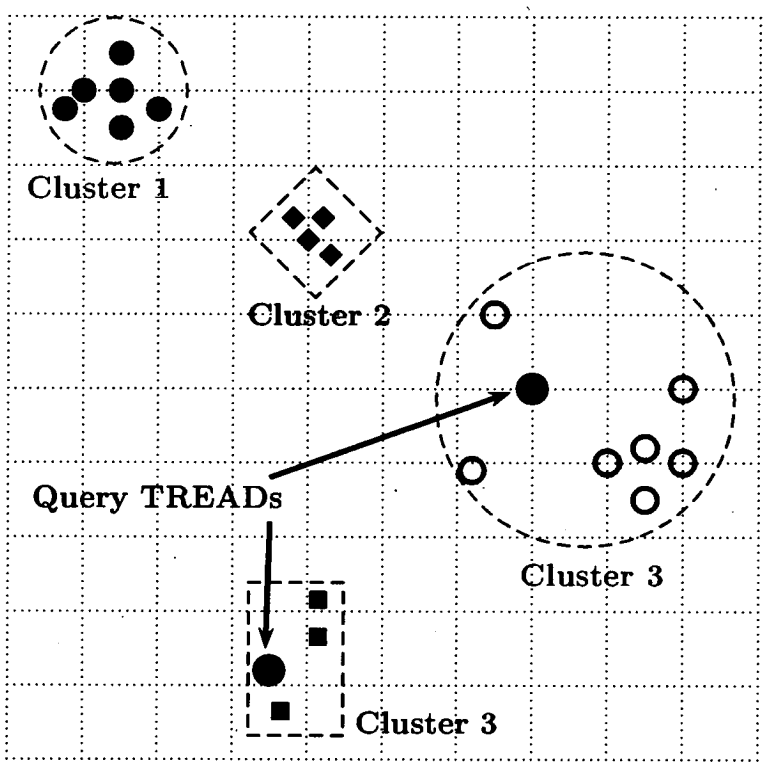


Figure 7.1: Tread Clustering

7.4.3 Automated Functional Role Prediction

At the moment the bottle neck in the Tread creation process is the assignment of functional roles to the residues in the Treads. This process is time consuming so an automated method of assigning functional roles would considerably speed up the creation of Treads. The manual system of assigning functional roles involves viewing the structure of the enzyme the Tread is being created for and going through conserved residues one-by-one in the structure, assigning the functional role. This process requires someone with sufficient knowledge of the relationship between structure and function. While this has the advantage of having an expert involved in the creation of a Tread, the time limitations mean an automated method would be necessary to create Treads for large numbers of

proteins.

The automated method, like the manual method would require the structure of the enzyme to have been determined. This would be done preferably by experimental methods (X-ray Crystallography or NMR) although it would also be possible to use structural models. It would also be possible to produce structural models for enzymes to be made into Treads. This will be a more viable option once the results of high-throughput experimental structure determination ensure the databases are filled with a diverse range of structures to act as templates for enzymes with unknown structure.

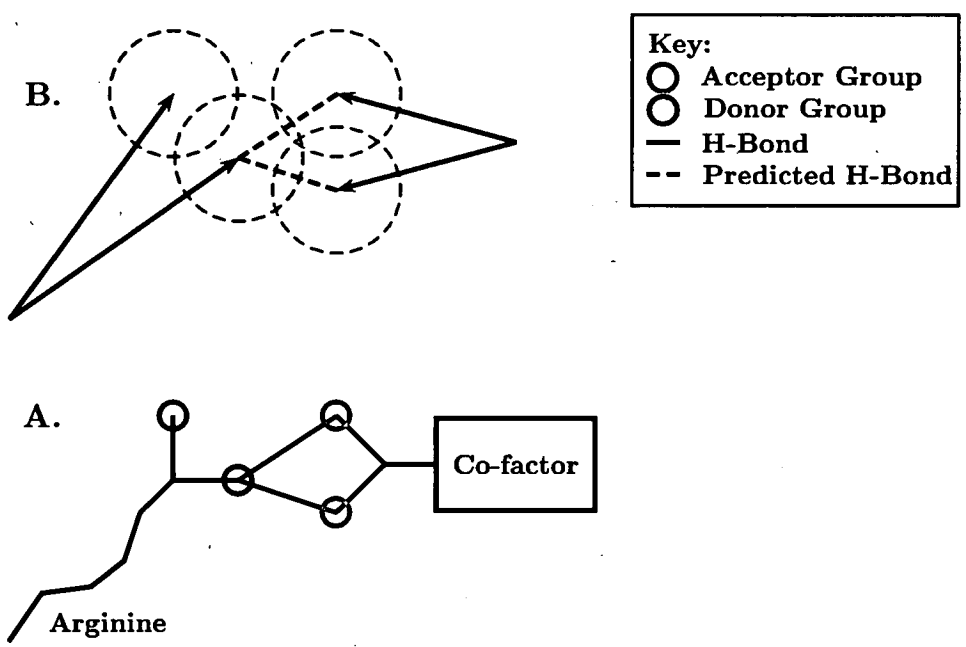


Figure 7.2: Predicting Active Sites

The goal of a heuristic to assign functional roles would be exactly the same as for an expert and the enzyme structure would ideally also have a substrate, reaction intermediate or cofactor bound. One method of assigning functional roles would be to represent functionally conserved residues as vectors. This representation would allow the direction of the residue to be taken into account. For example one could then postulate that if a functionally conserved residue is pointing toward the cofactor and is within 3Å, then the residue could be assigned the role of cofactor binding (Figure 7.2). In the case of second shell active site residues, where the residue is outside a nominal cut-off distance from the active site, these residues could also be assigned functional roles. If residues are ambiguous in their roles then they would be assigned either multiple roles or an unknown role.

Appendix A

Supplementary Data

A.1 Chapter 3 Data

A.1.1 $S-PLUS$® Commands

S-PLUS 6 requires the S+FinMetrics module for the commands in this section to work. A full guide to the S+FinMetrics can be found in the book by Zivot and Wang published by the S-PLUS company Insightful (Zivot & Wang, 2003). "CosineRaw" is the name of the datafile imported into S-Plus and contains the ordered scores for the highest scoring random Treads compared against the DAROGAN database.

Generalised Extreme Value Distribution CDFs

The plot for these commands can be found in Figure 3.18 A.

```
> z.vals = seq(-5,5,length=200)
> cdf.f  = ifelse((z.vals > -2), pgev(z.vals,xi=0.5),0)
> cdf.w  = ifelse((z.vals <  2), pgev(z.vals,xi=-0.5),1)
> cdf.g  = exp(-exp(-z.vals))
> plot(zvals, cdf.w, type="l", xlab="z", ylab="H(z)")
> lines(z.vals, cdf.f, type="l", lty=2)
> lines(z.vals, cdf.g, type="l", lty=3)
> legend(-5,1,legend=c("Weibull H(-0.5,0,1)",
```

```
+ "Frechet H(0.5,0,1)","Gumbell H(0,0,1"), lty=1:3)
```

Generalised Extreme Value Distribution PDFs

The plot for these commands can be found in Figure 3.18 B.

```
> pdf.f  = ifelse((z.vals > -2), dgev(z.vals,xi=0.5),0)
> pdf.w  = ifelse((z.vals <  2), dgev(z.vals,xi=-0.5),1)
> pdf.g  = exp(-exp(-z.vals))*exp(-z.vals)
> plot(zvals, pdf.w, type="l", xlab="z", ylab="h(z)")
> lines(z.vals, pdf.f, type="l", lty=2)
> lines(z.vals, pdf.g, type="l", lty=3)
> legend(-5.25,0.4,legend=c("Weibull H(-0.5,0,1)",
+ "Frechet H(0.5,0,1)","Gumbell H(0,0,1"), lty=1:3)
```

qqPlot

The plot for these commands can be found in Figure 3.17.

```
> class(CosineRaw)
> plot(-CosineRaw)
> qqPlot(CosineRaw,strip.text="",
> xlab="Quantiles of standard normal",
> ylab="Quantiles of CosineRaw")
> qqnorm(CosineRaw)
```

QQplot of Residuals

The plot for these commands can be found in Figure 3.19.

```
> plot(gev.fit.year)

Make a plot selection (or 0 to exit):
1: plot: Scatterplot of Residuals
2; plot: QQplot of Residuals
Selection:
```

Output GEV distribution fit

```
out <- gev(CosineRaw)
> out$par.ests
         xi       sigma         mu
 -0.3497692  0.04925293  0.8595927

> out$par.ses
          xi         sigma            mu
 0.002703265  0.0003529383  0.0005145767
```

A.2 Chapter 5 Data

Scoring Method	ξ	σ	μ
Cosine	-0.3497692	0.04925293	0.8595927
Manhattan Distance	-0.2564316	0.05443188	0.7459994
Euclidean Distance	-0.283455	0.09372719	0.4771651
Canberra	-0.2564802	0.136157	0.3650702
Dice	-0.2403048	0.07972698	0.6031941
Jaccard	-0.1554295	0.08086309	0.4329782

Table A.1: GEV Estimates for All Scoring Methods

	Enzyme Family				
Scoring method	**PLP** (250)	**TPP** (43)	**GLU** (200)	**FOL** (56)	**Overall** (550)
Significance Level = 0.05					
HMM	246(98.4%)	42(97.7%)	200(100%)	56(100%)	544(98.9%)
Cosine	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Manhattan	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Euclidean	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Canberra	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Dice	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Jaccard	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Significance Level = 0.01					
HMM	246(98.4%)	42(97.7%)	200(100%)	56(100%)	544(98.9%)
Cosine	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Manhattan	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Euclidean	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Canberra	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Dice	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)
Jaccard	249(99.6%)	43(100%)	199(99.5%)	56(100%)	548(99.6%)

Table A.2: Self-Consistency success rates for the HMM and the DAROGAN scoring methods. The success rates are divided up by the different cofactor utilising enzyme sets and for all enzymes in the tesing set.

Scoring method	Accuracy	Error	Precision	Recall	F1
Significance Level = 0.05					
HMM	69.7%	30.3%	98.9%	11.4%	20.4%
Cosine	66.1%	33.9%	99.3%	4.7%	9.0%
Manhattan	66.2%	33.8%	98.4%	5.1%	9.8%
Euclidean	66.2%	33.8%	98.1%	5.2%	9.8%
Canberra	66.2%	33.8%	98.4%	5.1%	9.8%
Dice	65.6%	34.4%	98.1%	3.4%	6.6%
Jaccard	65.6%	34.4%	98.1%	3.4%	6.6%
Significance Level = 0.01					
HMM	69.3%	30.7%	98.9%	10.5%	19.0%
Cosine	65.7%	34.3%	99.9%	3.7%	7.0%
Manhattan	66.0%	34.0%	99.6%	4.4%	8.4%
Euclidean	66.0%	34.0%	99.5%	4.4%	8.5%
Canberra	66.0%	34.0%	99.6%	4.4%	8.4%
Dice	65.4%	34.6%	99.6%	2.7%	5.2%
Jaccard	65.5%	34.5%	98.4%	3.0%	5.7%

Table A.3: Self-Consistency statistics for the HMM and the DAROGAN scoring methods. Accurracy, Error, Precision, Recall and F1 statistics are divided up by the different cofactor utilising enzyme sets and for all enzymes in the tesing set.

	Enzyme Family				
Scoring method	**PLP** (250)	**TPP** (43)	**GLU** (200)	**FOL** (56)	**Overall** (550)
Significance Level = 0.05					
HMM	244(97.6%)	30(69.8%)	192(96.0%)	46(82.1%)	512(93.0%)
Cosine	209(83.6%)	16(37.2%)	160(80.0%)	42(76.4%)	427(77.6%)
Manhattan	218(87.2%)	18(41.9%)	170(85.0%)	42(75.0%)	448(81.5%)
Euclidean	224(89.6%)	20(46.5%)	168(84.0%)	42(75.0%)	454(82.5%)
Canberra	218(87.2%)	18(41.9%)	170(85.0%)	42(75.0%)	448(81.4%)
Dice	194(77.6%)	9(20.9%)	124(62.0%)	36(64.3%)	363(66.0%)
Jaccard	194(77.6%)	9(20.9%)	124(62.0%)	36(64.3%)	363(66.0%)
Significance Level = 0.01					
HMM	243(97.2%)	30(69.8%)	192(96.0%)	46(82.1%)	511(92.9%)
Cosine	189(75.6%)	9(20.9%)	132(66.0%)	38(67.9%)	368(66.9%)
Manhattan	210(84.0%)	12(27.9%)	147(58.8%)	42(75.0%)	411(74.7%)
Euclidean	209(83.6%)	12(27.9%)	149(74.5%)	42(75.0%)	412(74.9%)
Canberra	210(84.0%)	12(27.9%)	147(58.8%)	42(75.0%)	411(74.5%)
Dice	164(65.6%)	9(20.9%)	106(53.0%)	36(64.3%)	317(57.6%)
Jaccard	177(70.1%)	9(20.9%)	113(56.5%)	36(64.3%)	335(60.9%)

Table A.4: Jack-Knife success rates for the HMM and the DAROGAN scoring methods. The success rates are divided up by the different cofactor utilising enzyme sets and for all enzymes in the tesing set.

Scoring method	Accuracy	Error	Precision	Recall	F1
Significance Level = 0.05					
HMM	69.4%	30.6%	98.8%	10.9%	19.6%
Cosine	65.9%	34.1%	99.2%	4.2%	8.1%
Manhattan	66.1%	33.9%	98.3%	4.6%	8.8%
Euclidean	66.1%	33.9%	97.9%	4.7%	8.9%
Canberra	66.1%	33.9%	98.3%	4.6%	8.8%
Dice	65.4%	34.6%	97.8%	2.9%	5.6%
Jaccard	65.4%	34.6%	97.8%	2.9%	5.6%
Significance Level = 0.01					
HMM	69.0%	31.0%	98.9%	10.0%	18.1%
Cosine	65.6%	34.4%	99.9%	3.1%	6.1%
Manhattan	65.8%	34.2%	99.5%	3.9%	7.5%
Euclidean	65.8%	34.2%	99.4%	3.9%	7.5%
Canberra	65.8%	34.2%	99.5%	3.9%	7.5%
Dice	65.2%	34.8%	99.5%	2.1%	4.2%
Jaccard	65.3%	34.7%	98.1%	2.4%	4.8%

Table A.5: Jack-Knife statistics for the HMM and the DAROGAN scoring methods. Accurracy, Error, Precision, Recall and F1 statistics are divided up by the different cofactor utilising enzyme sets and for all enzymes in the tesing set.

Appendix B

BioCyc

B.1 Canberra (Signficance 0.05, PISCES Cull Threshold 100%)

```
-------------------------------------------------------------------------------
Query:1790307_E-10_NR.Taln                QID:  5/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     Q55128_E-40_SP.Maln   0.652   0.0471978707     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1787408_E-25_NR.Taln                QID:  6/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     Q84I53_E-30_SP.Maln   0.688   0.0257104956     PLP
         2     Q8D8Q1_E-25_SP.Maln   0.688   0.0257104956     PLP
         3     P54691_E-20_SP.Maln   0.662   0.0401716974     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1790855_E-30_NR.Taln                QID:  8/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     P53206_E-20_SP.Maln   0.662   0.0401716974     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1790671_E-5_SP.Taln                 QID: 17/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     Q03662_E-10_SP.Maln   0.662   0.0401786689     GLU
         2     P23279_E-20_SP.Maln   0.662   0.0401786689     PLP
         3     P30711_E-15_SP.Maln   0.650   0.0483003852     GLU
         4      Q12198_E-5_SP.Maln   0.650   0.0483003852     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1788636_E-30_SP.Taln                QID: 56/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     P77690_E-15_SP.Maln   0.680   0.0296711973     PLP
    +-------+----------------------+-------+--------------+----------+
```

```
-------------------------------------------------------------------------------
Query:1787831_E-15_NR.Taln              QID: 58/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
        1      Q8QZR5_E-15_SP.Maln   0.662   0.0401786689     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1787395_E-35_NR.Taln              QID: 68/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
        1       P57289_E-5_SP.Maln   0.820   0.0004992333     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1786795_E-30_NR.Taln              QID: 73/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
        1      Q8Z308_E-10_SP.Maln   0.680   0.0296711973     GLU
        2      P05459_E-25_SP.Maln   0.676   0.0317317291     PLP
        3      Q8DB36_E-35_SP.Maln   0.676   0.0317317291     PLP
        4      Q8ECR2_E-35_SP.Maln   0.676   0.0317317291     PLP
        5      Q884R9_E-50_SP.Maln   0.674   0.0329187076     PLP
        6      P23908_E-10_SP.Maln   0.652   0.0470959742     GLU
        7      P59600_E-10_SP.Maln   0.652   0.0470959742     GLU
        8      Q8X742_E-10_SP.Maln   0.652   0.0470959742     GLU
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1790461_E-40_NR.Taln              QID: 75/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
        1      Q89AX7_E-25_SP.Maln   0.688   0.0257104956     PLP
        2      Q8YU96_E-40_SP.Maln   0.688   0.0257104956     PLP
        3      Q9KSX2_E-25_SP.Maln   0.688   0.0257104956     PLP
        4      P29012_E-35_SP.Maln   0.678   0.0303510078     PLP
        5      P94967_E-25_SP.Maln   0.678   0.0303510078     PLP
        6      Q54899_E-35_SP.Maln   0.678   0.0303510078     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1789878_E-10_NR.Taln              QID:105/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
        1      Q8XV80_E-30_SP.Maln   0.840   0.0001561285     PLP
        2      P61000_E-35_SP.Maln   0.680   0.0296711973     PLP
        3      Q9JTH8_E-30_SP.Maln   0.680   0.0296711973     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1788770_E-35_NR.Taln              QID:112/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
        1      P14173_E-30_SP.Maln   0.662   0.0401786689     PLP
        2      P81893_E-20_SP.Maln   0.662   0.0401786689     PLP
        3      Q06086_E-30_SP.Maln   0.662   0.0401786689     PLP
    +-------+----------------------+-------+--------------+----------+

-------------------------------------------------------------------------------
Query:1789591_E-25_NR.Taln              QID:119/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
```

1	Q8Z4W1_E-15_SP.Maln	0.680	0.0296711973	PLP

Query:1786990_E-35_NR.Taln QID:125/416

rank	RTread Match	score	significance	cofactor
1	O32148_E-15_SP.Maln	0.662	0.0401716974	PLP

Query:1788440_E-45_NR.Taln QID:128/416

rank	RTread Match	score	significance	cofactor
1	Q58466_E-15_SP.Maln	0.670	0.0353261568	PLP

Query:1789158_E-35_NR.Taln QID:150/416

rank	RTread Match	score	significance	cofactor
1	O58489_E-45_SP.Maln	0.655	0.0449232919	PLP
2	Q9V0L2_E-45_SP.Maln	0.655	0.0449232919	PLP

Query:1789994_E-20_NR.Taln QID:156/416

rank	RTread Match	score	significance	cofactor
1	Q51687_E-15_SP.Maln	0.662	0.0401808139	PLP
2	P61000_E-35_SP.Maln	0.658	0.0425825070	PLP
3	Q9JTH8_E-30_SP.Maln	0.658	0.0425825070	PLP

Query:1790248_E-10_NR.Taln QID:167/416

rank	RTread Match	score	significance	cofactor
1	Q16773_E-30_SP.Maln	0.652	0.0471978707	PLP

Query:1790609_E-60_NR.Taln QID:171/416

rank	RTread Match	score	significance	cofactor
1	Q8P5R4_E-25_SP.Maln	0.844	0.0001173924	PLP
2	P14173_E-30_SP.Maln	0.656	0.0445555180	PLP
3	P81893_E-20_SP.Maln	0.656	0.0445555180	PLP
4	Q06086_E-30_SP.Maln	0.656	0.0445555180	PLP
5	Q8NT73_E-20_SP.Maln	0.656	0.0445555180	PLP

Query:1790765_E-10_NR.Taln QID:173/416

rank	RTread Match	score	significance	cofactor
1	P63479_E-35_SP.Maln	0.799	0.0013414042	PLP
2	Q56346_E-35_SP.Maln	0.688	0.0257104956	PLP
3	P29012_E-35_SP.Maln	0.678	0.0303510078	PLP
4	P94967_E-25_SP.Maln	0.678	0.0303510078	PLP
5	Q54899_E-35_SP.Maln	0.678	0.0303510078	PLP
6	O67687_E-25_SP.Maln	0.667	0.0369011244	PLP
7	Q8R860_E-35_SP.Maln	0.667	0.0369011244	PLP

```
    +-------+----------------------+-------+--------------+----------+

--------------------------------------------------------------------------------
Query:1786730_E-25_NR.Taln              QID:180/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     Q8GYYO_E-20_SP.Maln   0.656   0.0440320656     PLP
    +-------+----------------------+-------+--------------+----------+

--------------------------------------------------------------------------------
Query:1788803_E-5_SP.Taln               QID:189/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1      O13326_E-5_SP.Maln   0.680   0.0296711973     PLP
    +-------+----------------------+-------+--------------+----------+

--------------------------------------------------------------------------------
Query:1788458_E-15_NR.Taln              QID:196/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     Q7NLO3_E-35_SP.Maln   0.836   0.0002078647     PLP
         2     Q51687_E-15_SP.Maln   0.671   0.0344260273     PLP
         3     Q8ZFX6_E-30_SP.Maln   0.671   0.0344260273     PLP
         4     Q9PBC6_E-30_SP.Maln   0.671   0.0344260273     PLP
         5     P61000_E-35_SP.Maln   0.656   0.0440320656     PLP
         6     Q9JTH8_E-30_SP.Maln   0.656   0.0440320656     PLP
         7     P77727_E-35_SP.Maln   0.655   0.0449232919     PLP
         8     Q89AX7_E-25_SP.Maln   0.651   0.0480910228     PLP
         9     Q9KSX2_E-25_SP.Maln   0.651   0.0480910228     PLP
    +-------+----------------------+-------+--------------+----------+

--------------------------------------------------------------------------------
Query:1789925_E-15_NR.Taln              QID:207/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     P49725_E-15_SP.Maln   0.671   0.0344260273     PLP
    +-------+----------------------+-------+--------------+----------+

--------------------------------------------------------------------------------
Query:1787254_E-50_NR.Taln              QID:210/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     Q8YU96_E-40_SP.Maln   0.844   0.0001173924     PLP
         2     O59828_E-35_SP.Maln   0.688   0.0257104956     PLP
         3     P06655_E-35_SP.Maln   0.688   0.0257104956     PLP
         4     Q8DCLO_E-40_SP.Maln   0.688   0.0257104956     PLP
         5     Q8PGDO_E-40_SP.Maln   0.688   0.0257104956     PLP
         6     Q8X5V2_E-35_SP.Maln   0.688   0.0257104956     PLP
         7     P29012_E-35_SP.Maln   0.678   0.0303510078     PLP
         8     P94967_E-25_SP.Maln   0.678   0.0303510078     PLP
         9     Q54899_E-35_SP.Maln   0.678   0.0303510078     PLP
        10     P10725_E-35_SP.Maln   0.677   0.0311438026     PLP
        11     P10299_E-20_SP.Maln   0.651   0.0480910228     GLU
    +-------+----------------------+-------+--------------+----------+

--------------------------------------------------------------------------------
Query:1789985_E-10_NR.Taln              QID:217/416

      rank     RTread Match          score   significance   cofactor
    +-------+----------------------+-------+--------------+----------+
         1     Q8QZR5_E-15_SP.Maln   0.662   0.0401786689     PLP
    +-------+----------------------+-------+--------------+----------+

--------------------------------------------------------------------------------
Query:1790663_E-10_NR.Taln              QID:229/416
```

rank	RTread Match	score	significance	cofactor
1	O67687_E-25_SP.Maln	0.810	0.0008204525	PLP
2	Q8R860_E-35_SP.Maln	0.810	0.0008204525	PLP
3	P10725_E-35_SP.Maln	0.672	0.0340076267	PLP
4	P63482_E-15_SP.Maln	0.662	0.0401786689	PLP

Query:1790442_E-30_NR.Taln QID:231/416

rank	RTread Match	score	significance	cofactor
1	Q44686_E-60_SP.Maln	0.781	0.0025781532	PLP
2	P19440_E-60_SP.Maln	0.762	0.0046428402	GLU

Query:1789882_E-15_NR.Taln QID:236/416

rank	RTread Match	score	significance	cofactor
1	Q8GYY0_E-20_SP.Maln	0.658	0.0428253135	PLP

Query:1788490_E-50_NR.Taln QID:253/416

rank	RTread Match	score	significance	cofactor
1	Q8P5R4_E-25_SP.Maln	0.675	0.0323668348	PLP
2	P14173_E-30_SP.Maln	0.654	0.0458667741	PLP
3	P81893_E-20_SP.Maln	0.654	0.0458667741	PLP
4	Q06086_E-30_SP.Maln	0.654	0.0458667741	PLP

Query:1788093_E-20_NR.Taln QID:263/416

rank	RTread Match	score	significance	cofactor
1	P52069_E-5_SP.Maln	0.682	0.0285129470	PLP

Query:1789493_E-30_NR.Taln QID:279/416

rank	RTread Match	score	significance	cofactor
1	Q44688_E-10_SP.Maln	0.657	0.0431747823	PLP

Query:1787465_E-60_NR.Taln QID:280/416

rank	RTread Match	score	significance	cofactor
1	Q44004_E-15_SP.Maln	0.650	0.0483003852	PLP

Query:1787071_E-5_SP.Taln QID:301/416

rank	RTread Match	score	significance	cofactor
1	P47176_E-5_SP.Maln	0.673	0.0336056576	PLP
2	P26624_E-15_SP.Maln	0.666	0.0376381137	GLU

Query:1789439_E-15_SP.Taln QID:337/416

rank	RTread Match	score	significance	cofactor
1	O54694_E-35_SP.Maln	0.814	0.0006753816	PLP
2	P60120_E-45_SP.Maln	0.814	0.0006753816	PLP
3	P78698_E-35_SP.Maln	0.814	0.0006753816	PLP
4	O74351_E-25_SP.Maln	0.770	0.0036113563	PLP

Query:1786830_E-35_NR.Taln QID:352/416

rank	RTread Match	score	significance	cofactor
1	P40998_E-30_SP.Maln	0.662	0.0401786689	TPP
2	Q8NT73_E-20_SP.Maln	0.662	0.0401786689	PLP

Query:1787219_E-25_NR.Taln QID:363/416

rank	RTread Match	score	significance	cofactor
1	Q89AX7_E-25_SP.Maln	0.688	0.0257104956	PLP
2	Q8YU96_E-40_SP.Maln	0.688	0.0257104956	PLP
3	Q9KSX2_E-25_SP.Maln	0.688	0.0257104956	PLP
4	P29012_E-35_SP.Maln	0.678	0.0303510078	PLP
5	P94967_E-25_SP.Maln	0.678	0.0303510078	PLP
6	Q54899_E-35_SP.Maln	0.678	0.0303510078	PLP

Query:1789251_E-20_NR.Taln QID:398/416

rank	RTread Match	score	significance	cofactor
1	O32148_E-15_SP.Maln	0.658	0.0425825070	PLP

Query:1790825_E-25_NR.Taln QID:404/416

rank	RTread Match	score	significance	cofactor
1	Q8XV80_E-30_SP.Maln	0.656	0.0440320656	PLP

Query:1787057_E-60_NR.Taln QID:409/416

rank	RTread Match	score	significance	cofactor
1	P29012_E-35_SP.Maln	0.676	0.0316082831	PLP
2	P94967_E-25_SP.Maln	0.676	0.0316082831	PLP
3	Q54899_E-35_SP.Maln	0.676	0.0316082831	PLP
4	Q56346_E-35_SP.Maln	0.663	0.0396206005	PLP
5	O67687_E-25_SP.Maln	0.648	0.0499264443	PLP
6	P63479_E-35_SP.Maln	0.648	0.0499264443	PLP
7	Q8R860_E-35_SP.Maln	0.648	0.0499264443	PLP

B.2 Euclidean (Signficance 0.05, PISCES Cull Threshold 100%)

Query:1787408_E-25_NR.Taln QID: 6/416

rank	RTread Match	score	significance	cofactor
1	Q8QZR1_E-30_SP.Maln	0.696	0.0218186952	PLP
2	P54691_E-20_SP.Maln	0.685	0.0301725108	PLP

Query:1790671_E-5_SP.Taln QID: 17/416

rank	RTread Match	score	significance	cofactor
1	P30711_E-15_SP.Maln	0.668	0.0471457338	GLU
2	Q12198_E-5_SP.Maln	0.668	0.0471457338	PLP

Query:1788636_E-30_SP.Taln QID: 56/416

rank	RTread Match	score	significance	cofactor
1	P77690_E-15_SP.Maln	0.680	0.0346136118	PLP

Query:1787395_E-35_NR.Taln QID: 68/416

rank	RTread Match	score	significance	cofactor
1	P57289_E-5_SP.Maln	0.746	0.0026936675	PLP

Query:1786795_E-30_NR.Taln QID: 73/416

rank	RTread Match	score	significance	cofactor
1	P05459_E-25_SP.Maln	0.691	0.0249161033	PLP
2	Q8DB36_E-35_SP.Maln	0.691	0.0249161033	PLP
3	Q8ECR2_E-35_SP.Maln	0.691	0.0249161033	PLP
4	Q884R9_E-50_SP.Maln	0.685	0.0301725108	PLP

Query:1790461_E-40_NR.Taln QID: 75/416

rank	RTread Match	score	significance	cofactor
1	P29012_E-35_SP.Maln	0.692	0.0242037374	PLP
2	P94967_E-25_SP.Maln	0.692	0.0242037374	PLP
3	Q54899_E-35_SP.Maln	0.692	0.0242037374	PLP
4	Q8YU96_E-40_SP.Maln	0.688	0.0277411790	PLP
5	P63479_E-35_SP.Maln	0.684	0.0310689313	PLP

Query:1790589_E-50_NR.Taln QID: 80/416

rank	RTread Match	score	significance	cofactor
1	Q8ZYF9_E-10_SP.Maln	0.673	0.0418704902	PLP
2	P34899_E-5_SP.Maln	0.667	0.0485838219	PLP
3	P50433_E-5_SP.Maln	0.667	0.0485838219	PLP
4	P66803_E-5_SP.Maln	0.667	0.0485838219	PLP

5	Q6FUP6_E-5_SP.Maln	0.667	0.0485838219	PLP
6	Q7MEH7_E-5_SP.Maln	0.667	0.0485838219	PLP
7	Q7ND67_E-5_SP.Maln	0.667	0.0485838219	PLP
8	Q7U9J7_E-5_SP.Maln	0.667	0.0485838219	PLP
9	Q7UQN2_E-5_SP.Maln	0.667	0.0485838219	PLP
10	Q7VUW7_E-5_SP.Maln	0.667	0.0485838219	PLP
11	Q81JY4_E-10_SP.Maln	0.667	0.0485838219	PLP
12	Q82JIO_E-5_SP.Maln	0.667	0.0485838219	PLP
13	Q88AD1_E-5_SP.Maln	0.667	0.0485838219	PLP
14	Q8A9S7_E-5_SP.Maln	0.667	0.0485838219	PLP
15	Q8D7G5_E-5_SP.Maln	0.667	0.0485838219	PLP
16	Q8K9P2_E-5_SP.Maln	0.667	0.0485838219	PLP
17	Q8KC36_E-5_SP.Maln	0.667	0.0485838219	PLP
18	Q8P122_E-5_SP.Maln	0.667	0.0485838219	PLP
19	Q8PCN4_E-5_SP.Maln	0.667	0.0485838219	PLP
20	Q8UG75_E-5_SP.Maln	0.667	0.0485838219	PLP
21	Q8XTQ1_E-10_SP.Maln	0.667	0.0485838219	PLP
22	Q8Y1G1_E-10_SP.Maln	0.667	0.0485838219	PLP
23	Q8Z2Z9_E-5_SP.Maln	0.667	0.0485838219	PLP
24	Q97GV1_E-5_SP.Maln	0.667	0.0485838219	PLP
25	Q9KMP4_E-5_SP.Maln	0.667	0.0485838219	PLP
26	Q9PET2_E-5_SP.Maln	0.667	0.0485838219	PLP
27	Q9XAY7_E-5_SP.Maln	0.667	0.0485838219	PLP
28	Q9XB01_E-5_SP.Maln	0.667	0.0485838219	PLP
29	Q9Z831_E-5_SP.Maln	0.667	0.0485838219	PLP

Query:1786840_E-10_NR.Taln QID: 92/416

rank	RTread Match	score	significance	cofactor
1	P63479_E-35_SP.Maln	0.667	0.0485838219	PLP

Query:1789878_E-10_NR.Taln QID:105/416

rank	RTread Match	score	significance	cofactor
1	Q8XV80_E-30_SP.Maln	0.774	0.0003367370	PLP
2	P61000_E-35_SP.Maln	0.680	0.0346136118	PLP
3	Q9JTH8_E-30_SP.Maln	0.680	0.0346136118	PLP

Query:1789591_E-25_NR.Taln QID:119/416

rank	RTread Match	score	significance	cofactor
1	Q8Z4W1_E-15_SP.Maln	0.680	0.0346136118	PLP

Query:1786990_E-35_NR.Taln QID:125/416

rank	RTread Match	score	significance	cofactor
1	O32148_E-15_SP.Maln	0.685	0.0301725108	PLP

Query:1788440_E-45_NR.Taln QID:128/416

rank	RTread Match	score	significance	cofactor
1	O32148_E-15_SP.Maln	0.684	0.0308883968	PLP
2	Q58466_E-15_SP.Maln	0.666	0.0497107030	PLP

Query:1789158_E-35_NR.Taln QID:150/416

rank	RTread Match	score	significance	cofactor
1	O58489_E-45_SP.Maln	0.668	0.0471457338	PLP
2	Q9VOL2_E-45_SP.Maln	0.668	0.0471457338	PLP
3	P52069_E-5_SP.Maln	0.668	0.0471457338	PLP

Query:1789994_E-20_NR.Taln QID:156/416

rank	RTread Match	score	significance	cofactor
1	Q51687_E-15_SP.Maln	0.714	0.0115282314	PLP
2	P61000_E-35_SP.Maln	0.709	0.0140441643	PLP
3	Q9JTH8_E-30_SP.Maln	0.709	0.0140441643	PLP

Query:1790609_E-60_NR.Taln QID:171/416

rank	RTread Match	score	significance	cofactor
1	Q8P5R4_E-25_SP.Maln	0.779	0.0001796635	PLP
2	P14173_E-30_SP.Maln	0.685	0.0300710568	PLP
3	P81893_E-20_SP.Maln	0.685	0.0300710568	PLP
4	Q06086_E-30_SP.Maln	0.685	0.0300710568	PLP
5	Q8NT73_E-20_SP.Maln	0.685	0.0300710568	PLP

Query:1790765_E-10_NR.Taln QID:173/416

rank	RTread Match	score	significance	cofactor
1	P63479_E-35_SP.Maln	0.779	0.0001763872	PLP
2	P29012_E-35_SP.Maln	0.692	0.0242037374	PLP
3	P94967_E-25_SP.Maln	0.692	0.0242037374	PLP
4	Q54899_E-35_SP.Maln	0.692	0.0242037374	PLP
5	P36605_E-25_SP.Maln	0.689	0.0269682018	PLP
6	P77727_E-35_SP.Maln	0.685	0.0300710568	PLP
7	Q8YD03_E-30_SP.Maln	0.685	0.0300710568	PLP
8	Q92JD9_E-30_SP.Maln	0.685	0.0300710568	PLP
9	O67687_E-25_SP.Maln	0.684	0.0310689313	PLP
10	Q8R860_E-35_SP.Maln	0.684	0.0310689313	PLP

Query:1789291_E-40_NR.Taln QID:177/416

rank	RTread Match	score	significance	cofactor
1	Q05998_E-10_SP.Maln	0.684	0.0310689313	TPP
2	P91856_E-5_SP.Maln	0.668	0.0469619855	PLP
3	Q10349_E-5_SP.Maln	0.668	0.0469619855	PLP
4	Q72VI2_E-5_SP.Maln	0.668	0.0469619855	PLP
5	Q7VLPO_E-5_SP.Maln	0.668	0.0469619855	PLP
6	Q83LP3_E-5_SP.Maln	0.668	0.0469619855	PLP
7	Q88M07_E-5_SP.Maln	0.668	0.0469619855	PLP
8	Q8EEH2_E-5_SP.Maln	0.668	0.0469619855	PLP
9	Q8ZGB4_E-5_SP.Maln	0.668	0.0469619855	PLP
10	Q9CHW5_E-5_SP.Maln	0.668	0.0469619855	PLP
11	Q9HZ66_E-5_SP.Maln	0.668	0.0469619855	PLP

Query:1788458_E-15_NR.Taln QID:196/416

rank	RTread Match	score	significance	cofactor
1	Q7NL03_E-35_SP.Maln	0.768	0.0005968595	PLP
2	Q89AX7_E-25_SP.Maln	0.684	0.0310689313	PLP
3	Q9KSX2_E-25_SP.Maln	0.684	0.0310689313	PLP
4	P61000_E-35_SP.Maln	0.677	0.0376755220	PLP
5	Q9JTH8_E-30_SP.Maln	0.677	0.0376755220	PLP
6	Q51687_E-15_SP.Maln	0.671	0.0432731410	PLP
7	Q8ZFX6_E-30_SP.Maln	0.671	0.0432731410	PLP
8	Q9PBC6_E-30_SP.Maln	0.671	0.0432731410	PLP
9	P56099_E-25_SP.Maln	0.668	0.0469619855	PLP
10	P63482_E-15_SP.Maln	0.668	0.0471457338	PLP
11	P77727_E-35_SP.Maln	0.668	0.0471457338	PLP
12	Q8EFB2_E-25_SP.Maln	0.668	0.0471457338	PLP
13	P36605_E-25_SP.Maln	0.667	0.0485838219	PLP
14	Q82AA5_E-30_SP.Maln	0.667	0.0485838219	PLP
15	Q8TH25_E-35_SP.Maln	0.667	0.0485838219	PLP

Query:1789925_E-15_NR.Taln QID:207/416

rank	RTread Match	score	significance	cofactor
1	P49725_E-15_SP.Maln	0.671	0.0432731410	PLP

Query:1787254_E-50_NR.Taln QID:210/416

rank	RTread Match	score	significance	cofactor
1	Q8YU96_E-40_SP.Maln	0.779	0.0001796635	PLP
2	P10725_E-35_SP.Maln	0.698	0.0200063889	PLP
3	P29012_E-35_SP.Maln	0.692	0.0242037374	PLP
4	P94967_E-25_SP.Maln	0.692	0.0242037374	PLP
5	Q54899_E-35_SP.Maln	0.692	0.0242037374	PLP
6	O59828_E-35_SP.Maln	0.688	0.0277411790	PLP
7	P06655_E-35_SP.Maln	0.688	0.0277411790	PLP
8	Q8DCL0_E-40_SP.Maln	0.688	0.0277411790	PLP
9	Q8PGD0_E-40_SP.Maln	0.688	0.0277411790	PLP
10	Q8X5V2_E-35_SP.Maln	0.688	0.0277411790	PLP
11	Q9TSM4_E-25_SP.Maln	0.685	0.0300710568	GLU
12	P10299_E-20_SP.Maln	0.684	0.0310689313	GLU
13	P48774_E-10_SP.Maln	0.684	0.0310689313	GLU
14	Q21355_E-20_SP.Maln	0.684	0.0310689313	GLU
15	O67687_E-25_SP.Maln	0.684	0.0310689313	PLP
16	Q8R860_E-35_SP.Maln	0.684	0.0310689313	PLP
17	P18425_E-15_SP.Maln	0.676	0.0378775165	GLU

Query:1790663_E-10_NR.Taln QID:229/416

rank	RTread Match	score	significance	cofactor
1	O67687_E-25_SP.Maln	0.766	0.0006851983	PLP
2	Q8R860_E-35_SP.Maln	0.766	0.0006851983	PLP

Query:1790442_E-30_NR.Taln QID:231/416

rank	RTread Match	score	significance	cofactor
1	P19440_E-60_SP.Maln	0.754	0.0016440248	GLU
2	Q44686_E-60_SP.Maln	0.744	0.0030314608	PLP

Query:1790837_E-30_NR.Taln QID:233/416

rank	RTread Match	score	significance	cofactor
1	Q8QZR5_E-15_SP.Maln	0.668	0.0471457338	PLP
2	O60779_E-35_TR.Maln	0.667	0.0485838219	TPP

Query:1788490_E-50_NR.Taln QID:253/416

rank	RTread Match	score	significance	cofactor
1	O96567_E-20_SP.Maln	0.702	0.0176480414	PLP
2	P05031_E-20_SP.Maln	0.702	0.0176480414	PLP
3	P27718_E-20_SP.Maln	0.702	0.0176480414	PLP
4	Q8P5R4_E-25_SP.Maln	0.692	0.0242037374	PLP

Query:1788093_E-20_NR.Taln QID:263/416

rank	RTread Match	score	significance	cofactor
1	P52069_E-5_SP.Maln	0.728	0.0067294029	PLP

Query:1789493_E-30_NR.Taln QID:279/416

rank	RTread Match	score	significance	cofactor
1	Q44688_E-10_SP.Maln	0.675	0.0388082757	PLP

Query:1787465_E-60_NR.Taln QID:280/416

rank	RTread Match	score	significance	cofactor
1	Q44004_E-15_SP.Maln	0.668	0.0471457338	PLP

Query:1786928_E-35_NR.Taln QID:308/416

rank	RTread Match	score	significance	cofactor
1	Q8PBK7_E-45_SP.Maln	0.696	0.0216850881	PLP
2	Q9K4Z7_E-10_SP.Maln	0.692	0.0241274706	GLU
3	Q9K4Z2_E-5_SP.Maln	0.691	0.0249161033	GLU

Query:1789176_E-25_NR.Taln QID:315/416

rank	RTread Match	score	significance	cofactor
1	Q6C5H4_E-50_SP.Maln	0.671	0.0432731410	GLU
2	Q87JS8_E-5_SP.Maln	0.667	0.0485838219	PLP

Query:1789439_E-15_SP.Taln QID:337/416

rank	RTread Match	score	significance	cofactor
1	O54694_E-35_SP.Maln	0.737	0.0042640587	PLP
2	P60120_E-45_SP.Maln	0.737	0.0042640587	PLP
3	P78698_E-35_SP.Maln	0.737	0.0042640587	PLP

4	O74351_E-25_SP.Maln	0.735	0.0047827686	PLP

Query:1786830_E-35_NR.Taln QID:352/416

rank	RTread Match	score	significance	cofactor
1	P05459_E-25_SP.Maln	0.696	0.0216850881	PLP
2	Q8DB36_E-35_SP.Maln	0.696	0.0216850881	PLP
3	Q8ECR2_E-35_SP.Maln	0.696	0.0216850881	PLP
4	P26624_E-15_SP.Maln	0.685	0.0300710568	GLU
5	Q884R9_E-50_SP.Maln	0.685	0.0300710568	PLP

Query:1787219_E-25_NR.Taln QID:363/416

rank	RTread Match	score	significance	cofactor
1	P29012_E-35_SP.Maln	0.692	0.0242037374	PLP
2	P94967_E-25_SP.Maln	0.692	0.0242037374	PLP
3	Q54899_E-35_SP.Maln	0.692	0.0242037374	PLP
4	Q8YU96_E-40_SP.Maln	0.688	0.0277411790	PLP
5	P63479_E-35_SP.Maln	0.684	0.0310689313	PLP

Query:1789496_E-15_NR.Taln QID:366/416

rank	RTread Match	score	significance	cofactor
1	Q59447_E-40_SP.Maln	0.685	0.0300710568	PLP
2	O67507_E-25_SP.Maln	0.668	0.0471457338	PLP
3	P11096_E-25_SP.Maln	0.668	0.0471457338	PLP

Query:1789251_E-20_NR.Taln QID:398/416

rank	RTread Match	score	significance	cofactor
1	O32148_E-15_SP.Maln	0.709	0.0140441643	PLP

Query:1790825_E-25_NR.Taln QID:404/416

rank	RTread Match	score	significance	cofactor
1	Q8XV80_E-30_SP.Maln	0.677	0.0376755220	PLP

Query:1787057_E-60_NR.Taln QID:409/416

rank	RTread Match	score	significance	cofactor
1	O67687_E-25_SP.Maln	0.703	0.0174491632	PLP
2	P63479_E-35_SP.Maln	0.703	0.0174491632	PLP
3	Q8R860_E-35_SP.Maln	0.703	0.0174491632	PLP
4	Q56346_E-35_SP.Maln	0.698	0.0200063889	PLP

Bibliography

Altschul, S., Gish, W., Miller, W., Myers, E. & Lipman, D. (1990) Basic local alignment search tool. *J. Mol. Biol.,* **215**, 403–410.

Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. (1997) Gapped blast and psi-blast: a new generation of protein database search programs. *Nucl. Acids Res.,* **25**, 3389–3402.

AMAS (2005). `http://www.compbio.dundee.ac.uk`.

Armon, A., Graur, D. & Ben-Tal, N. (2001) Consurf: an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. *J. Mol. Biol.,* **307**, 447–463.

Attwood, T., Flower, D., Lewis, A., Mabey, J., Morgan, S., Scordis, P., Selley, J. & Wright, W. (1999) Prints prepares for the new millennium. *Nucl. Acids. Res.,* **27**, 220–225.

Bartlett, G., Porter, C., Borkakoti, N. & Thornton, J. (2002) Analysis of catalytic residues in enzyme active sites. *J. Mol. Biol.,* **324**, 105–121.

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

Benner, S., Badcoe, I., Cohen, M. & Gerloff, D. (1994) Bona fide prediction of aspects of protein confirmation. *J. Mol. Biol.,* **235**, 926–958.

Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I. & Bourne, P. (2000) The protein data bank. *Nucl. Acids. Res.,* **28**, 235–242.

BioCyc (2005). `http://www.biocyc.org/`.

BLOCKS (2005). `http://www.blocks.fhcrc.org/`.

Cai, Y. (2001) Is it a paradox or misinterpretation? *Proteins: Struct. Func. Genet.,* **43**, 336–338.

Cammer, S., Hoffman, B., Spier, J., Canaday, M., Nelson, M., Knutson, S., Gallina, M., Baxter, S. & Fetrow, J. (2003) Structure-based active site profiles for genome analysis and functional family subclassification. *J. Mol. Biol.,* **334**, 387–401.

Casari, G., Sander, C. & Valencia, A. (1995) A method to predict functional residues in proteins. *Nat. Struct. Biol.,* **2**, 171–178.

Chou, K. (1995) A noval approach to predicting protein structural classes in a (20-1) amino acid composition space. *Proteins: Struct. Func. Genet.,* **21**, 319–344.

Chou, K. & Elrod, D. (2003) Prediction of enzyme family classes. *J. Proteome Research,* **2**, 183–190.

COG (2005). `http://www.ncbi.nih.gov/COG`.

Connolly, T., Begg, C. & Strachan, A. (1999) *Database Systems: A Practical Approach to Design, Implementation, and Management.* Addison Wesley Longman Limited, Harlow, Essex, UK.

ConSurf & Rate4Site (2005). `http://consurf.tau.ac.il/`.

Dayhoff, M., Schwartz, R. & Orcutt, B. (1978) *Atlas of Protein Sequence and Structure* vol. 3,. Washington, DC, USA: National Biomedical Research Foundation pp. 345–352.

Dunathan, H. & Voet, J. (1974) Stereochemical evidence for the evolution of pyridoxal phosphate enzymes of various function from a common ancestor. *Proc. Natl. Acad. Sci.,* **71**, 3888–3891.

Dunbrack, R., Gerloff, D., Bower, M., Chen, X., Lichtarge, O. & Cohen, F. (1997) Meeting review: the second meeting on the critical assessment of techniques for protein structure presiction (casp2), asilomar, california, december 13-16, 1996. *Fold Des.,* **1**, 27–42.

EcoCyc (2005). `http://ecocyc.PangeaSystems.com/ecocyc`.

Eddy, S. (1998) Profile hidden markov models. *Bioinformatics,* **14**, 755–763.

Edgar, R. (2004) Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res.,* **32**, 1792–1797.

Erlandsen, H., Abola, E. & Stevens, R. (2000) Combining structural genomics and enzymology: completing the picture in metabolic pathways and enzyme active sites. *Curr. Opin. Struct. Biol.,* **10**, 719–730.

ExPASy (2005). `http://www.expasy.ch`.

Fetrow, J., Godzik, A. & Skolnick, J. (1998) Functional analysis of the escherichia coli genome using the sequence-to-structure-to-function paradigm: identification of proteins exhibiting the glutaredoxin/thioredoxin disulphide oxidoreductase activity-. *J. Mol. Biol.,* **282**, 703–711.

Fetrow, J. & Skolnick, J. (1998) Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and t-1 ribonucleases. *J. Mol. Biol.,* **281**, 949–968.

Fischer, R. & Tippet, L. (1928) Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Proc. Camb. Phil. Soc.,* **24**, 180–190.

Flanagan, D. (1998) *JavaScript: A definitive Guide.* O'Reilly and Associates, Sebastopol, USA.

GD Graphics Library (2005). `http://www.boutell.com/gd`. Author: Thomas Boutell.

GeneQuiz (2005). `http://jura.ebi.ed.ac.uk:8765/ext-genequiz/genomes/ec0005/`.

Gerloff, D., Cohen, F., Korostensky, C., Turcotte, M., Gonnet, G. & Benner, S. (1997) A predicted consensus structure for the n-terminal fragment of the heat shock protein hsp90 family. *Proteins: Struct. Func. Genet.,* **27**, 450–458.

Greenacre, M. (1984) *Theory and Applications of Correspondence Analysis.* Academic Press, New York, USA.

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

Henikoff, S. & Henikoff, J. (1991) Automated assembly of protein blocks for database searching. *Nucl. Acids Res.,* **19**, 6565–6572.

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

Hoersch, S., Leyroy, C., Brown, N., Andrade, M. & Sander, C. (2000) The genequiz web server: protein functional analysis through the web. *Trends Biol. Sci.,* **25**, 33–35.

Hofmann, K., Bucher, P., Falquet, L. & Bairoch, A. (1999) The prosite database, its status in 1999. *Nucl. Acids. Res.,* **27**, 215–219.

Jaccard, P. (1912) The distribution of flora in the alpine zone. *New Phytologist,* **11**, 37–50.

Jeanmougin, F., Thompson, J., Gouy, M., Higgins, D. & Gibson, T. (1998) Multiple sequence alignment with clustal x. *Trends Biol. Sci.,* **23**, 403–405.

John, R. (1995) Pyridoxal phosphate-dependent enzymes. *Biochim. Biophys. Acta,* **1248**, 81–96.

Kanaoka, Y., Ago, H., Inagaki, E., Nanayama, T., Miyano, M., Kikuno, R., Fujii, Y., Eguchi, N., Toh, H., Urade, Y. & Hayaishi, O. (1997) Cloning and crystal structure of hematopoietic prostaglandin d synthase. *Cell,* **90**, 1085–1095.

Karlin, S. & Brocchieri, L. (1996) Evolution conservation of reca genes in relation to protein structure and function. *J. Bacteriol.,* **178**, 1881–1894.

Karp, P., Ouzounis, C., Moor-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahren, D., Tsoka, S., Darzentas, N., Kunin, V. & Lopez-Bigas, N. (2005) Expansion of the biocyc collection of pathway/genome databases to 160 genomes. *Nucl. Acids Res.,* **33**, 6083–6089.

Karp, P., Riley, M., Paley, S. & M. Pellegrini-Toole, A. (1999) Eco cyc: encyclopedia of escherichia coli genes and metabolism. *Nucl. Acids Res.,* **27**, 55–58.

Karplus, K. & Hu, B. (2001) Evaluation of protein multiple alignments by sam-t99 using the balibase multiple alignment test set. *Bioinformatics,* **17**, 713–720.

Kern, D., Kern, G., Neef, H., Tittmann, K., Killenberg-Jabs, M., Wikner, C., Schneider, G. & Hubner, G. (1997) How thiamin diphosphate is activated in enzymes. *Science,* **275**, 67–70.

Kirsch, J., Eichele, G., Ford, G., Vincent, M. & Jansonius, J. (1984) Mechanism of action of aspartate aminotransferase proposed on the basis of its spatial structure. *J. Mol. Biol.,* **174**, 497–525.

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

Lichtarge, O. & Sowa, M. (2002) Evolutionary predictions of binding surfaces and interactions. *Curr. Opin. Struct. Biol.,* **12**, 21–27.

Livingstone, C. & Barton, G. (1993) Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. *CABIOS,* **9**, 745–756.

Madabushi, S., Yao, H., Marsh, M., Kristensen, D., Philippi, A., Sowa, M. & Lichtarge, O. (2002) Structural clusters of evolutionary trace residues are statistically significant and common in proteins. *J. Mol. Biol.,* **316**, 139–154.

Mardia, K., Kent, J. & Bibby, J. (1979) *Multivariate Analysis.* London, UK: Academic Press pp. 322,381.

Martin, A., Orengo, C., Hutchinson, E., Jones, S., Karmirantzou, M., Lasowski, R., Michell, J., Taroni, C. & Thornton, J. (1998) Protein folds and functions. *Structure,* **6**, 875–884.

Needleman, S. & Wunsch, C. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.,* **48**, 443–453.

Nelder, J. & Mead, R. (1965) A simplex method for function minimization. *Computer Journal,* **7**, 308–313.

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

Obermann, W., Sondermann, H., Russo, A., Pavletich, N. & Hartl, F. (1998) In vivo function of hsp90 is dependent on atp binding and atp hydrolysis. *Journal of Cell Biology,* **143**, 901–910.

openMosix (2005). `http://openmosix.sourceforge.net`.

Orengo, C., Jones, D. & Thornton, J. (2003*a*) *Bioinformatics: Genes, Proteins and Computers.* BIOS Scientific Publishers.

Orengo, C., Jones, D. & Thornton, J. (2003*b*) *Bioinformatics: Genes, Proteins and Computers.* Oxford, UK: BIOS Scientific Publishers pp. 49–64.

P. Vingron, M. (1989) A fast sensitive multiple sequence alignment algorithm. *CABIOS,* **5**, 115–121.

Pawlowski, K., Jaroszewski, L., Rychlewski, L. & Godzik, A. (2000) Sensitive sequence comparison as protein function predictor. *Pac. Symp. Biocomput.,* **5**, 42–53.

Pearson, W. & Lipman, D. (1988) Imporved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA,* **85**, 2444–2448.

Peisach, D. (1998). *Crystallographic Study of the Mechanism of D-amino Acid Transferase.* PhD thesis, Brandeis University.

Pellegrini, M., Marcotte, E., Thompson, M., Eisenberg, D. & Yeates, T. (1999) Assigning protein functions by comparative genome analysis: protein phylogenic profiles. *Proc. Natl. Acad. Sci.,* **96**, 4285–4288.

Perl Module: CGI (2005). `http://search.cpan.org/~lds/CGI.pm-3.05/CGI.pm`. Author: Lincoln D. Stein.

Perl Module: DBI (2005). `http://search.cpan.org/~timb/DBI-1.47/DBI.pm`. Author: Tim Bunce.

Perl Module: GD (2005). `http://search.cpan.org/~lds/GD-2.21/GD.pm.PLS`. Author: Lincoln D. Stein.

Perl Module: GD::Graph (2005). `http://search.cpan.org/~mverb/GDGraph-1.43/Graph.pm`. Author: Martien Verbruggen.

Perl Module: GD::Graph3D (2005). `http://search.cpan.org/~wadg/GDGraph3d-0.56/lib/GD/Graph3d.pm`. Author: Jeremy Wadsack.

Perl Module: Math::Amoeba (2005). `http://search.cpan.org/~jarw/Math-Amoeba-0.01/Amoeba.pm`. Author: John A. R. Williams.

Pfam (2005). `http://www.sanger.ac.uk/Software/Pfam/`.

PISCES (2005). `http://www.fccc.edu/research/labs/dunbrack/pisces`.

PRINTS (2005). `http://www.biochem.ucl.ac.uk/bsm/dbbrowser/PRINTS/PRINTS.html`.

PROCAT & TESS (2005). `http://www.biochem.ucl.ac.uk/bsm/PROCAT/PROCAT.html`.

Prodromou, C., Roe, M., OBrien, R., Ladbury, J., Piper, P. & Pearl, L. (1997) Identification and structural characterization of the atp/adp-binding site in the hsp90 molecular chaperone. *Cell,* **90**, 67–75.

PROSITE (2005). `http://www.expasy.ch/prosite/`.

ProtoMap (2005). `http://www.protomap.cs.huji.ac.il`.

Pupko, T., Bell, R., Mayrose, I., Glaser, F. & 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**, 71–77.

R. Molinaro, A. & Pfeiffer, R. (2005) Prediction error estimation: a comparison of resampling. *Bioinformatics,* **21**, 3301–3307.

R.A. Wallace, A. & Thornton, J. (1995) Ligplot: a program to generate schematic diagrams of protein-ligand interactions. *Protein Eng.,* **8**, 127–134.

S-Plus (2005). `http://www.insightful.com/products/splus`.

Salton, G. (1970) Automatic text analysis. *Science,* **168**, 335–342.

Schultz, J., Milpetz, F., Bork, P. & Ponting, C. (1998) Smart, a simple modular architecture research tool: identification of signaling domains. *Proc. Natl. Acad. Sci.,* **95**, 5857–5864.

Shah, I. & Hunter, L. (1997) Predicting enzyme function from sequence: a systematic appraisal. *ISMB,* **5**, 276–283.

Skolnick, J. & Fetrow, J. (2000) From genes to protein structure and function: novel applications of computational approaches in the genomic era. *Trends Biotechnol.,* **18**, 34–39.

SMART (2005). `http://www.bork.embl-heidelberg.de/Modules/sinput.shtml`.

Smith, T. & Waterman, M. (1981) Identification of common molecular subsequences. *J. Mol. Biol.,* **147**, 195–197.

Stephens, M. (1974) Edf statistics for goodness of fit and some comparisons. *J. Amer. Stat. Assoc.,* **69**, 730–737.

SwissProt & TrEMBL (2005). `http://ca.expasy.org/`. Swiss-SwissProt Release 45.5, TrEMBL Release 28.5 (04-Jan-2005).

Tai, C. & Cook, P. (2001) Pyridoxal 5-phosphate-dependent alpha,beta-elimination reactions: mechanism of o-acetylserine sulfhydrylase. *Acc. Chem. Res.,* **34**, 49–59.

Tatusov, R., Natale, D., Garkavtsev, I., Tatusova, T., Shankavaram, U., Rao, B., Kiryutin, B., Galperin, M., Fedorova, N. & Koonin, E. (2001) The cog database: new developments in phylogenetic classification of proteins from complete genomes. *Nucl. Acids Res.,* **29**, 22–28.

Thompson, J., Gibson, T., Plewniak, F., Jeanmougin, F. & Higgins, D. (1997) The clustalx windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucl. Acids Res.,* **25**, 4876–4882.

Thompson, J., Higgins, D. & Gibson, T. (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., Plewniak, F. & Poch, O. (1999*a*) Balibase: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics,* **15**, 87–88.

Thompson, J., Plewniak, F. & Poch, O. (1999*b*) A comprehensive comparison of multiple sequence alignment programs. *Nucl. Acids Res.,* **27**, 2682–2690.

Todd, A., Orengo, C. & Thornton, J. (1999) Evolution of protein function, from a structural perspective. *Curr. Opin. Chem. Biol.,* **3**, 548–556.

Todd, A., Orengo, C. & Thornton, J. (2001) Evolution of function in protein superfamilies, from a structural perspective. *J. Mol. Biol.,* **307**, 1113–1143.

Valdar, W. (2001). *Residue Conservation in the Prediction of Protein-Protein Interactions.* PhD thesis, University College London.

Valdar, W. (2002) Scoring residue conservation. *Proteins: Struct. Func. Genet.,* **48**, 227–241.

Valdar, W. & Thornton, J. (2001) Protein-protein interfaces: analysis of amino acid conservation in homodimers. *Proteins: Struct. Func. Genet.,* **42**, 108–124.

Vingron, M. & Sibbald, P. (1993) Weighting in sequence space: a comparison of methods in terms of generalised sequences. *Proc. Natl. Acad. Sci.,* **90**, 8777–8781.

Wall, L. & Schwartz, R. L. (1991) *Programming Perl.* O'Reilly & Associates, Inc., Sebastopol, USA.

Wallace, A., Borkakoti, N. & Thornton, J. (1997) Tess: a geometric hashing algorithm for deriving 3d coordinate templates for searching structural databases. application to enzyme active sites. *Protein Sci.,* **6**, 2308–2323.

Wang, G. & Dunbrack, R. (2003) Pisces: a protein sequence culling server. *Bioinformatics,* **19**, 1589–1591.

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

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

Wu, T. & Kabat, E. (1970) An analysis of the sequences of the variable regions of bence jones proteins and myeloma light chains and their implications for antibody complimentarity. *J. Exp. Med.,* **132**, 211–249.

Yamada, T., Komoto, J., Watanabe, K., Ohmiya, Y. & Takusagawa, F. (2005) Crystal structure and possible catalytic mechanism of microsomal prostaglandin e synthase type 2 (mpges-2). *J. Mol. Biol.,* **348**, 1163–1176.

Yona, G., Linial, N. & Linial, M. (1999) Protomap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins: Struct. Func. Genet.,* **37**, 360–378.

Zhang, B., Rychlewski, L., Pawlowski, K., Fetrow, J., Skolnick, J. & Godzik, A. (1999) From fold predictions to function predictions:automation of functional site conservation analysis for functional genome predictions. *Protein Sci.,* **8**, 1104–1115.

Zhou, G. & Assa-Munt, N. (2001) Some insights into protein structural class prediction. *Proteins: Struct. Func. Genet.,* **44**, 57–59.

Zivot, E. & Wang, J. (2003) *Modeling Financial Time Series with S-PLUS.* Insightful, (Springer), New York, NY, USA.

Zvelebil, M., Barton, G., Taylor, W. & Sternberg, M. (1987) Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J. Mol. Biol.,* **195**, 957–961.