UNIVERSITY COLLEGE LONDON

UCL Research Department of Structural and Molecular Biology

Post-genomic structural analysis of single amino acid polymorphisms

Lisa E M McMillan

A dissertation submitted to University College London for the degree of Doctor of Philosophy

Declaration

I, Lisa McMillan, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Lisa McMillan

February 10, 2009

Abstract

Inherited genetic variation is critical in defining disease susceptibility. PDs, or pathogenic deviations, are mutations reported to be disease-causing, while SNPs, or single nucleotide polymorphisms, are understood to have a negligible effect on phenotype. With recent developments in biotechnology—most relevant being increased reliability and speed of sequencing—a wealth of information regarding SNPs and PDs has been acquired. Quite apart from the analytical challenge of analysing this information with a view to identifying novel therapies and targets for disease, the challenge of simply storing, mapping and processing these data is significant in itself.

This thesis describes the development of a large-scale, automated pipeline that provides hypotheses as to what the structural effects of these genomic variations might be. This includes the development of nine new analyses. Eight of these new methods are structural, identifying mutations that disrupt various aspects of protein structure, including the interface, binding sites, folding mechanics and stability. The final new analysis is a novel method of identifying highly conserved residues from sequence. Here, the distribution of conservation scores from a multiple sequence alignment (MSA) is analysed to generate an MSA-specific threshold for high conservation. In order to construct MSAs for the sequence analysis, a novel method for identifying functionally equivalent proteins has been developed.

Further, PDs and SNPs are characterised with respect to these structural analyses, and with respect to basic sequence and structural features. The findings support trends elsewhere in the literature: PDs are more often found in the core of proteins and at highly conserved sites; they most often affect the stability of protein structures; and they more often are between very different amino acids. In addition to the implications for disease therapies, these findings are informative in the more general context of protein structure.

Abbreviations

ASA	:	**A**ccessible **S**urface **A**rea
CDF	:	**C**umulative **D**istribution **F**unction
DNA	:	**D**eoxyribo**n**ucleic **A**cid
FN	:	**F**alse **N**egative
FOSTA	:	**F**unctional **O**rthologues from **S**wissprot **T**ext **A**nalysis
FP	:	**F**alse **P**ositive
ImPACT	:	**Im**proved **P**rotein **A**lignment **C**onservation **T**hreshold
KS	:	**K**olmogorov-**S**mirnov
LSMDB	:	**L**ocus-**S**pecific **M**utation **D**ata**b**ase
MCC	:	**M**atthews **C**orrelation **C**oefficient
MSA	:	**M**ultiple **S**equence **A**lignment
NW	:	**N**eedleman & **W**unsch
PD	:	**P**athogenic **D**eviation
PDB	:	**P**rotein **D**ata**b**ank
PQS	:	**P**rotein **Q**uaternary **S**tructure database
ROC	:	**R**eceiver **O**perating **C**haracteristic
SAAP	:	**S**ingle **A**mino **A**cid **P**olymorphism
SNP	:	**S**ingle **N**ucleotide **P**olymorphism
TN	:	**T**rue **N**egative
TP	:	**T**rue **P**ositive

Acknowledgements

First and foremost, I'd like to thank Andrew Martin, my supervisor. Andrew has never failed to challenge me or trust me over the last three years, and I thank him for his unwavering support.

All PhD students should have a post-doctoral colleague like Jacob Hurst. At those inevitable times when I felt completely out of my depth, frustrated or bewildered, Jake was always available for the necessary encouragement and help.

Jesse Oldershaw, Tom Knight, Jahid Ahmed and Duncan McKenzie were always on hand to fix something that I had broken, retrieve something that I had accidentally deleted or investigate some IT irregularity that I didn't understand. I thank them for all their help.

Thanks are due to David Jones, my mentor, and Steven Perkins, my thesis committee chair. Their advice throughout the last couple of years has been invaluable.

I have been very fortunate to work with many fantastic people during my time at UCL. Thanks to everyone at UCL (Room 636 in particular), past and present, for their help, company and laughs over the last few years. I will remember you all fondly and wish you all the best of luck.

And finally, the most immense amount of thanks to Michael Hopcroft: ♡

Contents

List of Figures

List of Tables

Chapter 1

An introduction to mutations

The human body is a complex machine. Occasionally, individuals inherit very slight modifications that have a significant impact on their health. Other inherited modifications have little or no effect on health. This chapter will explore the biology of inherited disease-associated mutations, define the investigative scope to be pursued and introduce the scientific questions that will be asked throughout this thesis.

1.1 Deoxyribonucleic acid: the blueprint for life

Deoxyribonucleic acid (DNA) is the blueprint for many living organisms. It encodes proteins and other functional molecules that are necessary for the organism throughout its lifespan and is the vehicle through which offspring inherit information from their parents. It is a double stranded helical structure, comprising a sugar-phosphate backbone and a sequence of nucleotides or 'bases': adenine (A), cytosine (C), guanine (G) and thymine (T). Complementary base pairing between purine (A/G) and pyrimidine (C/T) bases (specifically between A/T and C/G) bases hold the two helical strands together (Figure 1.1).

It is the complete sequence of these four nucleotides—the genome—that defines the organism. In *Homo Sapiens*, the genome is approximately 3.2 billion base pairs long (International Human Genome Sequencing Consortium, 2004). Approximately 1.5-2% of the genome encodes

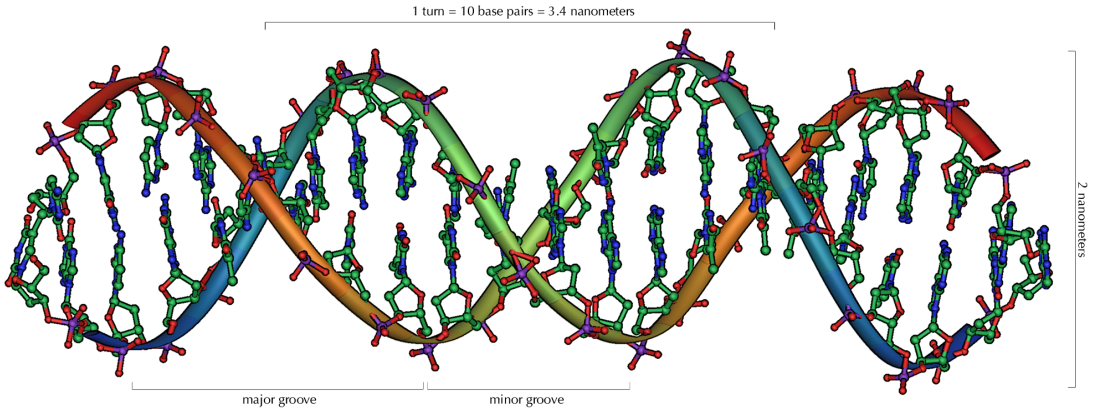


Figure 1.1: Deoxyribonucleic acid: DNA
The nucleotides are shown here, attached to the sugar-phosphate backbone. Figure obtained from Creative Commons.

proteins (Lander *et al.*, 2001), biomolecules that do much of the 'operational' work in the organism. These 'coding' regions of the genome are organised into 'genes', distinct protein encoding units that define individual proteins. The remaining ≈98% of the genome is 'non-coding'. Previously erroneously referred to as 'junk' DNA, some of these non-coding regions are now known to be well conserved (Bejerano *et al.*, 2004; Prabhakar *et al.*, 2006). Further, the recent ENCODE project suggests that the genome is exhaustively transcribed (The ENCODE Project Consortium, 2007). It is now commonly accepted that the non-coding regions of the genome perform other essential tasks in the cell, for example, the regulation of protein expression (Sandelin *et al.*, 2004; Couzin, 2002; Biémont and Vieira, 2006).

1.2 Variation in the genome

The genome of individual organisms within a species varies; it is variation between organisms that allows species to evolve via differental responses to external stimuli. The various different forms of a gene that exist in a population are described as 'alleles'.

There are various kinds of genomic variation and various mechanisms by which genomic variation can arise. This thesis will investigate one kind of small-scale genomic variation: the point mutation, where a single nucleotide is exchanged for another. Specifically, it will look at point

mutations in coding regions of DNA that lead to a single amino acid mutation at the protein level (see Section 1.4).

Point mutations can be 'germline' or 'somatic'. Germline mutations are transmitted to offspring via germ cells (in humans, the egg and sperm). Somatic mutations however are acquired by the organism during its lifespan and are not transmitted to offspring. Somatic mutations, therefore, are a failure of the DNA repair mechanisms to identify errors in DNA replication.

In sexual reproduction, offspring inherit half of their DNA from their mother and half from their father. The inherited genomic information is combined to create the offspring genome, and as such, children may inherit mutations from their parents. Rare alleles can come to persist in the population if they offer some advantage to the individuals that carry them or if they are completely neutral in their effect. Some pathogenic alleles may persist if their effects do not come into play until after the reproductive lifespan of the parent (e.g., propensity towards cancer or heart disease).

Throughout this thesis, the word 'native' will be used to describe the genotype containing the most common allele, while the word 'mutant' will describe the genotype containing mutations.

1.3 The vast phenotypic spectrum of mutations

The term 'genotype' describes the specific genomic information that defines an individual organism, i.e., the sequence of nucleotides in the organism's genome. The term 'phenotype' refers to the observable manifestation of the genotype. For example, human eye colour has been shown to be associated with genomic variation near the OCA2 gene (Duffy *et al.*, 2007); in this example, the genotype is the specific genomic variation near the OCA2 locus and the phenotype is the resulting eye colour.

Genomic abberations do not have a consistent effect on phenotype: some have negligible or no effect on phenotype, some introduce variation in phenotype without compromising health, some result in increased susceptibility to disease, some are directly causitive of disease and some are fatal. It is important to appreciate that, in disease research, the most severe, fatal mutations will never be observed in a patient: any cell encoding a fatal mutation will die without

being replicated.

In this thesis, point mutations that have been shown to be causitive of disease are described as **Pathogenic Deviations** or **PDs** and point mutations that have not been shown to be associated with disease are described as **Single Nucleotide Polymorphisms** or **SNPs**. The term SNP (The International Hapmap Consortium, 2005) is often used to refer to any point mutation, but strictly SNPs are defined as alleleic variants where the least common allele occurs in $\geq$1% of a normal population. So, while they may be associated with a complex disease, they cannot be involved in high penetrance Mendelianly inherited disease states. The most conservative estimates suggest that SNPs occur once every 1000 base pairs (Collins *et al.*, 1998; Taillon-Miller *et al.*, 1998) although others suggest that SNPs may occur as often as once every 100-300 bases (Wang *et al.*, 2006).

1.4 Genomic mutations manifest at the protein level

Figure 1.2 gives a broad overview of protein synthesis. The double stranded DNA helix is unwound to expose a single strand of DNA. Complementary base pairing forms messenger ribonucleic acid (or mRNA) in a process called *transcription* (note that thymine is replaced by uracil at the mRNA level). The mRNA is then *translated* into a series of amino acids, using the genetic code. The protein is the resulting sequence of amino acids.

The four letter alphabet of nucleotides encodes an alphabet of twenty amino acids. The amino acids vary in atom composition, size, charge, polarity, affinity for water (hydrophobicity) and so on. They are the building blocks of protein sequences. Further complexity at the protein level is afforded by post-translational modifications: chemical alterations of residues occurring after the protein sequence has been translated.

Given the redundancy in the genetic code and the structure of the genome, point mutations are differentially manifest at the protein level. The first differentiation to be made is between coding and non-coding mutations. **Coding** point mutations occur in coding areas of the genome. **Non-coding** point mutations occur *between* the coding areas of the genome (i.e., in the 'junk' DNA, see Section 1.1); see the T>C mutation shown in orange in Figure 1.2 for an example.

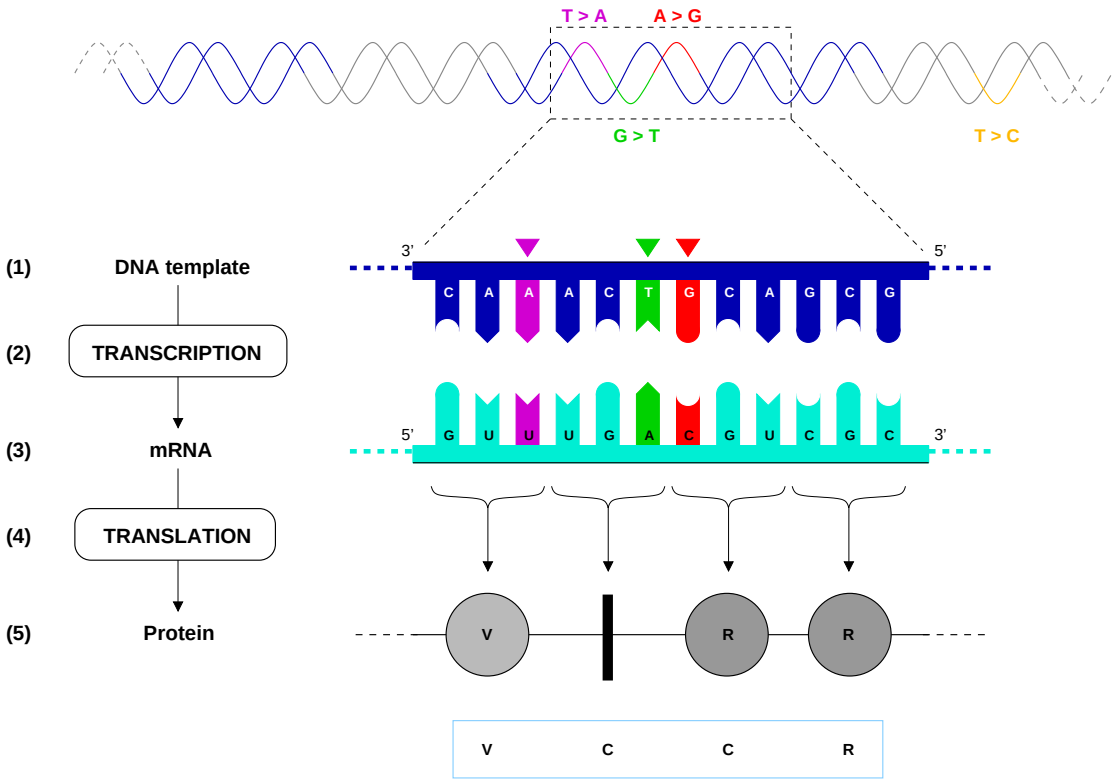


Figure 1.2: A broad overview of protein synthesis

A section of DNA is shown at the top of this figure with the coding regions (i.e., genes) in blue and the non-coding regions marked in grey. Proteins are synthesised from genes and proceeds as follows: **(1)**: The double stranded helix is broken to expose a DNA 'template'; **(2)**: The DNA is *transcribed* (using complementary base pairing) into RNA (ribonucleic acid), specifically **(3)**: mRNA 'messenger RNA' (note that thymine has become uracil); **(4)**: the mRNA is then *translated* according to the genetic code, where each three letter combination of RNA bases corresponds to an amino acid; **(5)**: the protein is formed by forming peptide bonds between the encoded amino acids (shown as grey circles). Four mutations are marked in purple, green, red and orange in the DNA. The respective base changes, at the DNA and mRNA levels are given in the corresponding colour. Coding mutations are marked with a triangle in the corresponding colour above the appropriate nucleotide at the single-stranded DNA level. The native protein sequence (i.e., the protein that would be synthesized without the mutations) is given below the mutant protein sequence in a light blue box. The purple T>A mutation is same-sense/synonymous/silent, inducing no change in the protein sequence (both GUU and GUA encode valine). The green G>T mutation is a nonsense mutation, introducing a premature stop codon (indicated with the thick vertical line). The red A>G mutation is a missense/non-synonymous mutation, that replaces the native cysteine residue (encoded by UGU) with an arginine (encoded by CGU). The orange T>C mutation is non-coding as it occurs outside of a gene.

Coding mutations can be further classified as synonymous, non-synonymous or nonsense. **Synonymous** mutations (also described as same-sense or silent mutations) do not alter the protein sequence (e.g., the purple T>A mutation in Figure 1.2). **Non-synonymous** mutations (also described as mis-sense mutations, nsSNPs (**n**on **s**ynonymous **SNP**s) or SAAPs (**single amino acid polymorphisms**)) induce a change in the amino acid sequence; see the red A>G mutation in Figure 1.2 for an example. Finally, **non-sense** mutations replace the native amino acid with a stop codon, resulting in an incomplete protein sequence.

The vast majority of proteins are **globular**: once the protein sequence has been synthesised, the string of amino acids fold together to adopt a three dimensional *globe-like* structure (as opposed to a fibrous structure) with a substantial buried 'core'. It is the folded structure that is functional.

There are four levels of protein structure. The first (or **primary**) level describes the one dimensional string of amino acids that comprises the sequence. Hydrogen bonds are formed between the backbone atoms of the amino acids to form helical (α) and sheet (β) 'secondary structures' in the **secondary** level. In the third (or **tertiary**) level, further bonds are formed between the residues to fold the entire protein chain into a single globular unit. In the fourth (**quaternary**) and final layer, multiple globular units (or 'chains') may be combined to create the functional protein.

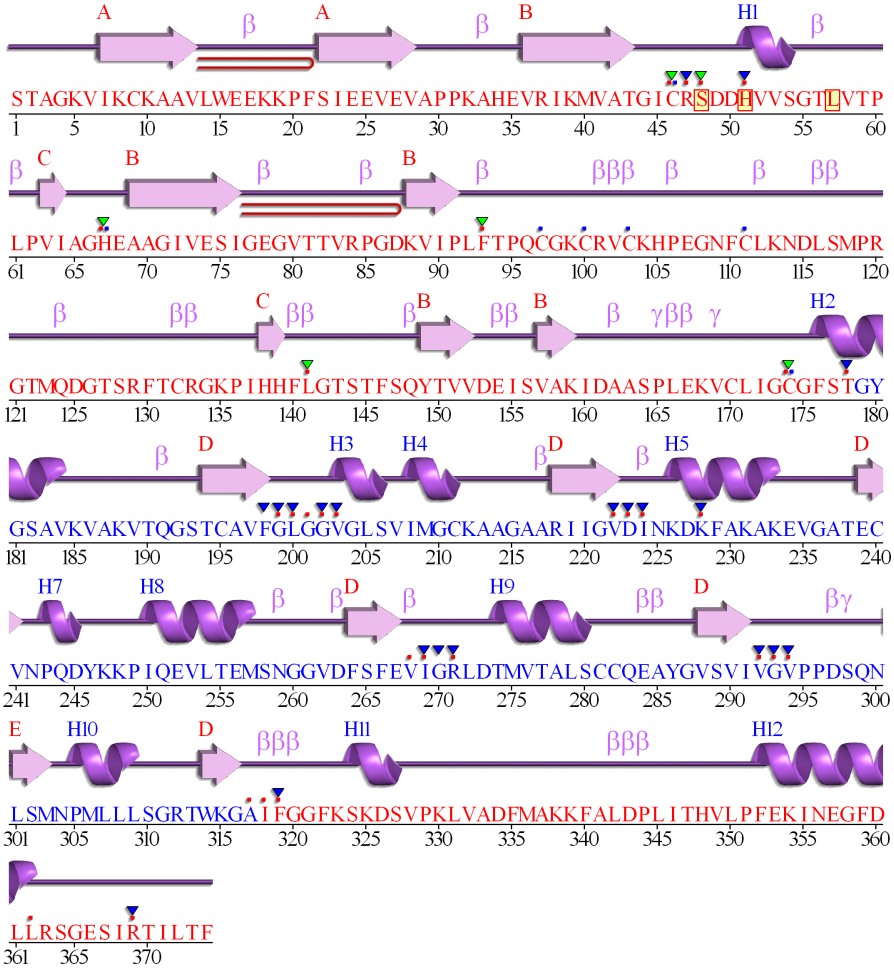
Figures 1.3 and 1.4 show the four levels for *Equus caballus* alcohol dehydrogenase (chain A) [UniProtKB:P00327/ADH1E_HORSE], an enzyme that breaks down otherwise potentially toxic alcohols to ketones and aldehydes[1]. Figure 1.3(a) shows the primary structure, the sequence of amino acids (obtained from UniProtKB/Swiss-Prot and shown in FASTA format). Figure 1.3(b) shows the same sequence annotated with the α (shown as purple helices) and β (shown as purple arrows) secondary structures. The tertiary structure of ADH1E_HORSE is shown in Figure 1.4(a), with the same α and β structures highlighted in pink and gold respectively (turns are shown in blue). The structure of ADH1E_HORSE is complete when two identical structures as shown in Figure 1.4(a) are combined to form a homodimer (i.e., a structure containing two copies of the same chain). The completed quaternary structure is shown in Figure 1.4(b), where the two chains are coloured different shades of blue.

Each chain in the homodimer is bound to the ligand NAD, or nicotinamide-adenine-dinucleotide, which is embedded within the structure (see Figure 1.4(b)). This demonstrates

[1] `http://www.expasy.org/enzyme/1.1.1.1`

```
>sp|P00327|ADH1E_HORSE Alcohol dehydrogenase E chain
MSTAGKVIKCKAAVLWEEKKPFSIEEVEVAPPKAHEVRIKMVATGICRSDDHVVSGTLVT
PLPVIAGHEAAGIVESIGEGVTTVRPGDKVIPLFTPQCGKCRVCKHPEGNFCLKNDLSMP
RGTMQDGTSRFTCRGKPIHHFLGTSTFSQYTVVDEISVAKIDAASPLEKVCLIGCGFSTG
YGSAVKVAKVTQGSTCAVFGLGGVGLSVIMGCKAAGAARIIGVDINKDKFAKAKEVGATE
CVNPQDYKKPIQEVLTEMSNGGVDFSFEVIGRLDTMVTALSCCQEAYGVSVIVGVPPDSQ
NLSMNPMLLLSGRTWKGAIFGGFKSKDSVPKLVADFMAKKFALDPLITHVLPFEKINEGF
DLLRSGESIRTILTF
```

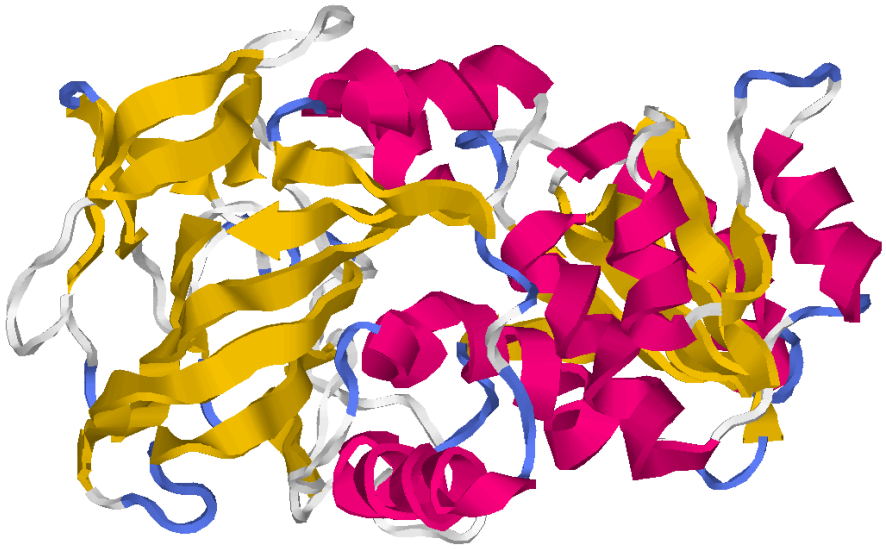
(a) Primary structure: the sequence of amino acids



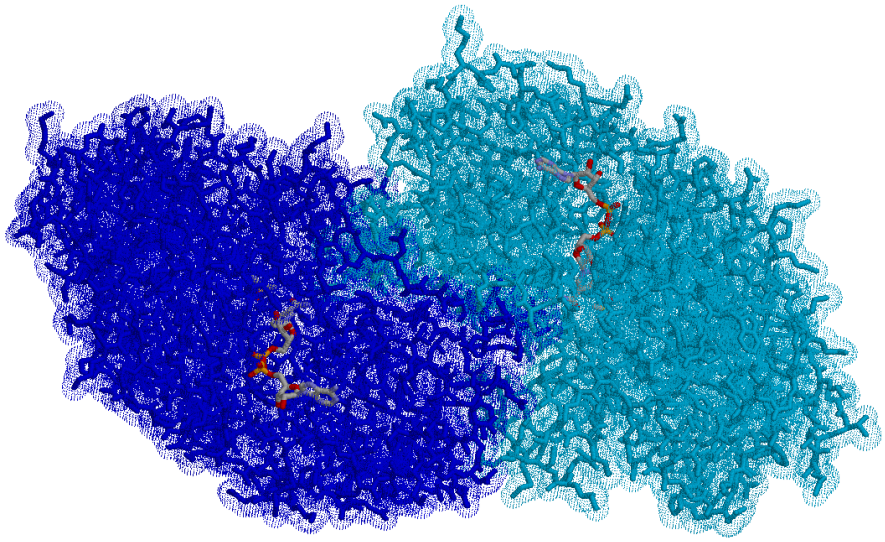
(b) Secondary structure: α and β structures are formed by hydrogen bonds

Figure 1.3: The primary and secondary structure of alcohol dehydrogenase

Figure 1.3(a) shows the UniProtKB/Swiss-Prot FASTA representation of ADH1E_HORSE, with the header line emboldened. Figure 1.3(b) shows a representation of secondary structure elements in the protein, as described in the PDB structure 6adh (diagram obtained from PDBSUM at the EBI). The α and β structures are shown as purple helices and arrows respectively (for the purposes of this discussion, other annotations can be ignored).



(a) Tertiary structure: globular chains are formed



(b) Quaternary structure: multiple chainsare combined to create the biologically relevant multimer

Figure 1.4: The tertiary and quaternary structure of alcohol dehydrogenase

Figure 1.4(a) shows the structure of 6adh, chain A. α secondary structure elements are shown as pink helices, β secondary structure elements are shown as gold arrows (an alternative secondary structure element, the turn, is shown in blue). Figure 1.4(b) shows the complete quaternary structure for ADH1E_HORSE. Here, two chains as described in Figure (shown in different shades of blue) 1.4(a) form a single structure, the biologically relevant multimer (specifically, a homodimer). A ligand (nicotinamide-adenine-dinucleotide or NAD) can be seen embedded in each chain.

the close structure/function relationship of proteins: if the structure were altered near the ligand binding site, NAD may not be able to bind to the protein. The specific biochemical action of alcohol dehydrogenase is to convert the bound NAD ligand into NADH by converting alcohol to ketones and aldehydes. Therefore, should the structure be mutated so as to inhibit binding to NAD, the creation of NADH may be inhibited or completely eradicated.

Given the process described in Figure 1.2 and the alcohol dehydrogenase example described above, it is clear that mutations at the genome level could induce a change in the protein sequence, thus altering the protein structure, and potentially affecting protein function.

1.5 An introduction to protein structure

Although protein structures vary immensely, they adhere closely to the same basic principles. This section will introduce these underlying concepts of protein structure. At a very general level, protein structure must (1) be stable; (2) fold correctly and (3) function properly. This section will conclude with a brief description of the two major methods by which the structures of proteins are determined experimentally: X-ray crystallography and nuclear magnetic resonance (NMR).

1.5.1 Hydrogen bonding

Hydrogen bonds form between (i) an electronegative atom and (ii) a hydrogen atom bonded to an electronegative atom (Baker and Hubbard, 1984). In the context of amino acids, the electronegative atom is either oxygen or nitrogen. In this interaction, the hydrogen atom is described as the **donor** atom and the electronegative atom is described as the **acceptor** atom. The sidechains of Arginine, asparagine, glutamine, histidine, lysine, serine, threonine, tryptophan and tyrosine can act as hydrogen bond donors and the sidechains of asparagine, aspartic acid, glutamic acid, glutamine, uncharged histidine, serine, threonine and tyrosine can act as hydrogen bond acceptors; the sidechains of the nine remaining residues (alanine, cysteine, phenylalanine, glycine, isoleucine, leucine, methionine, proline and valine) do not participate in hydrogen bonding. The backbones of all residues are able to both accept and donate a hydrogen bond, save for proline, which may only accept a hydrogen bond, given its cyclic sidechain (Cuff

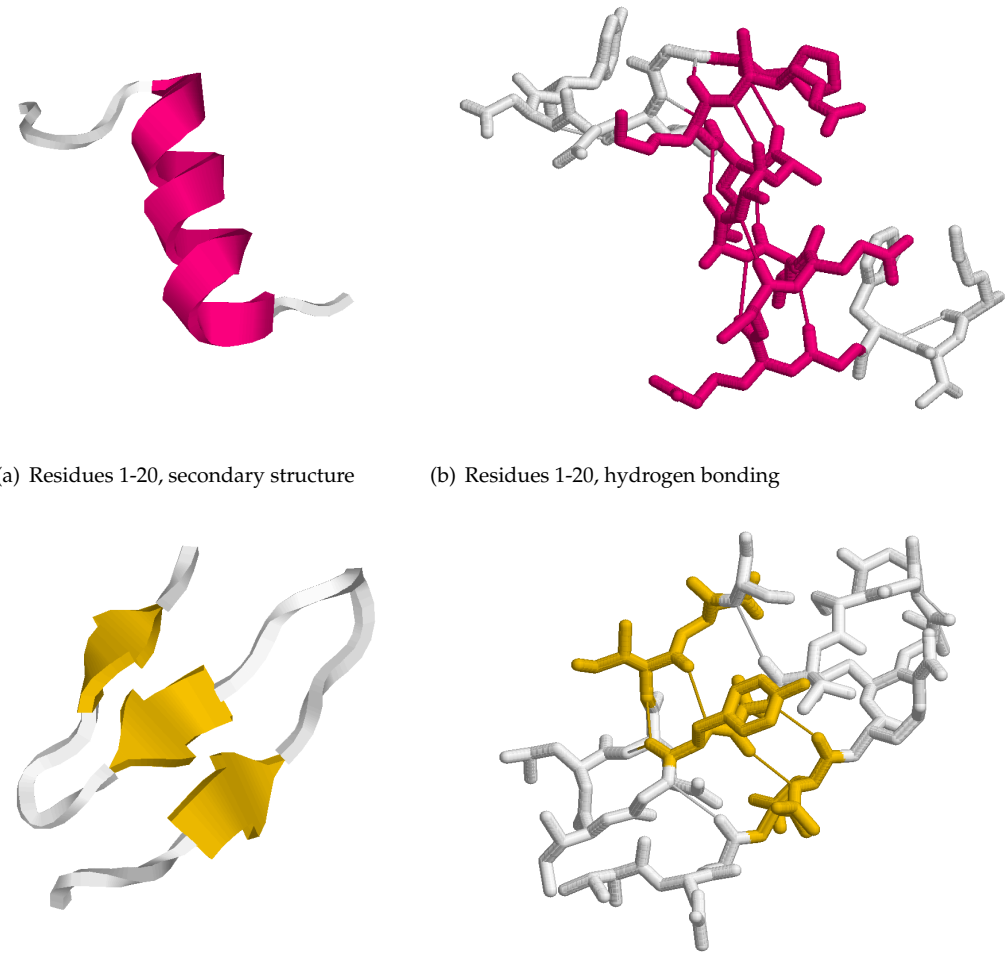
et al., 2006).

Most hydrogen bonds (68%) are formed between backbone atoms (Stickle *et al.*, 1992); secondary structure elements (described in Section 1.4) are maintained largely by a scaffold of backbone-backbone hydrogen bonding (see Figure 1.5). The remaining 32% of hydrogen bonds are formed between backbone-sidechain and sidechain-sidechain atoms. Hydrogen bonds are fundamental to the proper formation and stability of protein structure. It has been shown that most buried, hydrogen bonding capable sidechains do form hydrogen bonds (McDonald and Thornton, 1994).

1.5.2 Other important bonds

In addition to hydrogen bonds, protein structure is maintained by several other inter-residue bonds. **Salt bridges**, ionic bonds that are formed between the positively and negatively charged sidechains, also contribute to protein stability when found in the buried core of the structure (Torshin and Harrison, 2001). Other **Van der Waals** (non-electostatic, non-covalent) interactions, which arise from induced dipole-induced dipole interactions, also contribute to protein stability.

Several covalent bonds can form between residues. The most well known of these is the **disulphide bond**, a bond that can form between two sulphur atoms of cysteine residues of certain geometries (Hazes and Dijkstra, 1988); see Figure 1.6 for an example. Other covalent **crosslinks** that are formed between amino acids include 4-amino-3-isothiazolidinone-L-serine[2], a bond that forms between cysteine and serine residues (for example in protein-tyrosine phosphatase 1B (van Montfort *et al.*, 2003, see Figures 1(c) and 1(d) in manuscript)); N6-glycyl-L-lysine[3] a bond that forms between lysine and glycine residues (particularly important for interactions with small proteins like ubiquitin (Cripps *et al.*, 2006)), and N6-(L-isoglutamyl)-L-lysine[4], a bond that forms between lysine and glutamic acid residues (for example, between histone proteins H4 and H2B (Shimizu *et al.*, 1996, see Figure 1 in manuscript)).



(a) Residues 1-20, secondary structure

(b) Residues 1-20, hydrogen bonding

(c) Residues 40-60, secondary structure

(d) Residues 40-60, hydrogen bonding

Figure 1.5: Backbone hydrogen bonding generates α and β secondary structures

A α helix (residues 1-20, Figures 1.5(a)-1.5(b)) and β sheet (residues 40-60, Figure 1.5(c)-1.5(d)) from the structure of lysozyme (PDB ID 7lyz). Hydrogen bonds are indicated by thinner connections. Residues are coloured by structure (with gold indicating β structures and pink indicating α structures).

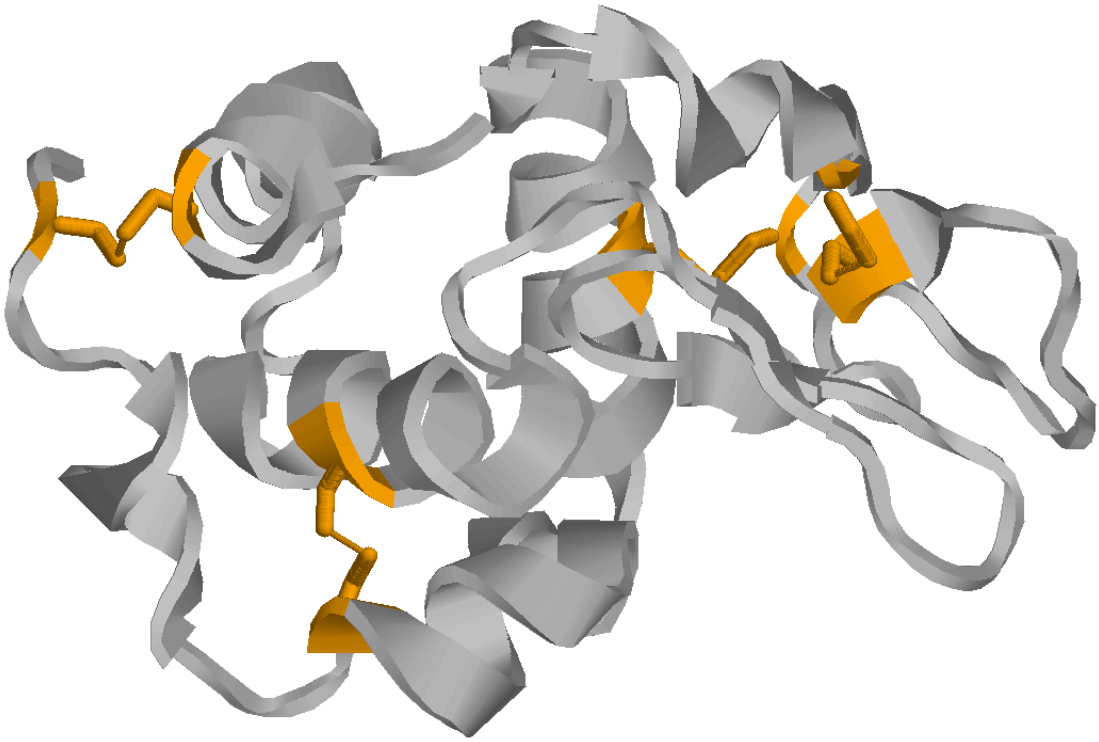


Figure 1.6: Disulphide bonding
Disulphide bonds in lysozyme (PDB ID 7lyz) are highlighted in orange. Four disulphide bonds are formed between eight cysteine residues (6-127, 30-115, 76-94 and 64-80).

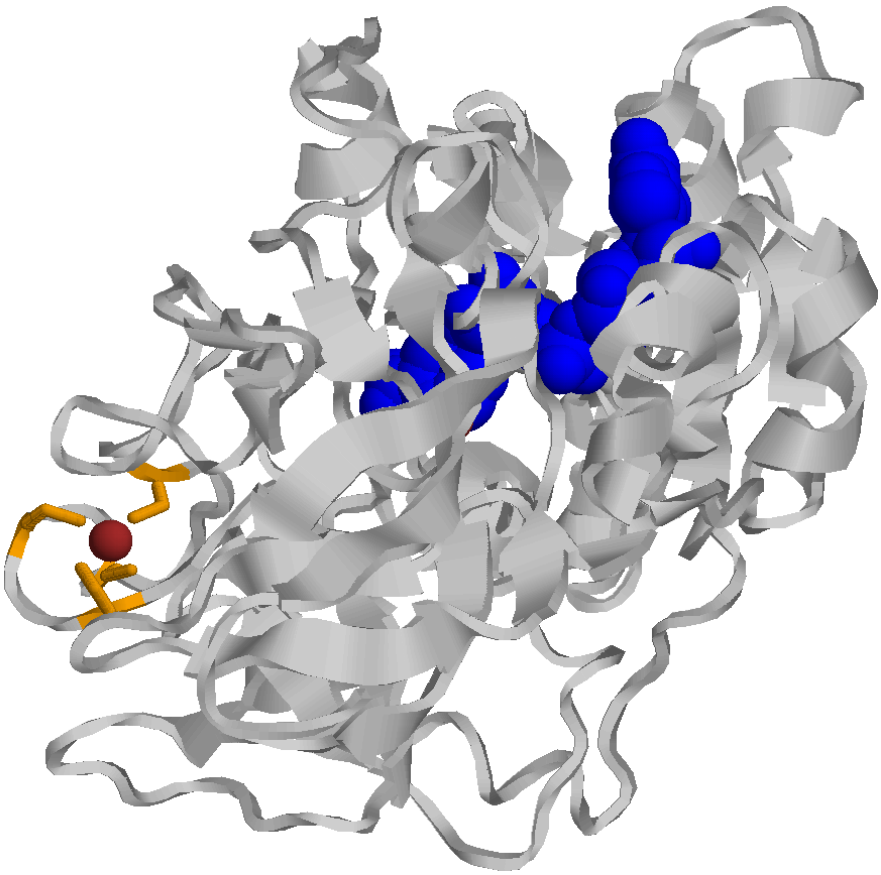


Figure 1.7: Ligand binding

The structure of alcohol dehydrogenase (PDB ID 6adh). The ligand nicotinamide-adenine-dinucleotide (NAD) is shown in blue; a zinc co-factor is shown in dark red, and residues within 4Å of the zinc co-factor (four cysteines at 97, 100, 103 and 111) are highlighted in orange (the rest of the protein is coloured grey, with secondary structure elements indicated). Note that the NAD is embedded in a binding pocket and that the zinc co-factor is supported by the sidechains of the four cysteine residues.

1.5.3 Ligand binding

Proteins act on, or complex with, each other and a vast array of other biomolecules or 'ligands'. The ligands interact with the protein by way of a **binding site**, or pocket. Intra-molecular forces (including ionic bonds, hydrogen bonds and Van der Waals interactions) secure the ligand to the protein. Proteins can also incorporate metal ions in their structure. The sidechains of histidine and aspartic and glutamic acids, and the sulfhydryl sidechain of cysteine bind to metal ions in 'metalloproteins'. See Figure 1.7 for examples of ligand and metal ion binding.

1.5.4 Hydrophobicity and folding

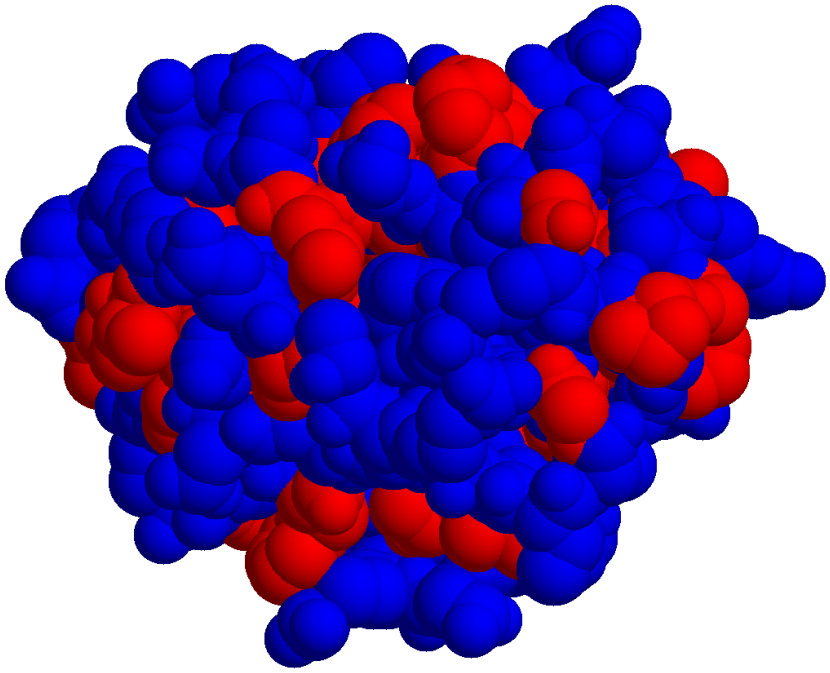
The composition of a residue's sidechain defines whether the residue is **hydrophobic** (repelled from water) or **hydrophilic** (attracted to water). Biochemically, hydrophilic residues are those that can form hydrogen bonds with water and are polar, while hydrophobic residues are non-polar and are unable to form hydrogen bonds with water. Many hydrophobicity 'scales' assess the hydrophobic properties of amino acids. They can be constructed using (i) experimental data that assess directly the behaviour of the residue in water (Yunger and Cramer, 1981) and/or (ii) structural data that identify residues commonly found in the protein core (Chothia, 1976). Others combine existing scales (Kyte and Doolittle, 1982). See Cornette *et al.* (1987) for a review of 38 hydrophobicity scales.

The driving force in protein folding is largely to bury hydrophobic residues in the protein core so as to limit their contact with water (compare Figures 1.8(a) and 1.8(b)). Where hydrophobic sidechains *do* occur in the hydrophilic core, their hydrogen bonding potential is always satisfied (McDonald and Thornton, 1994).

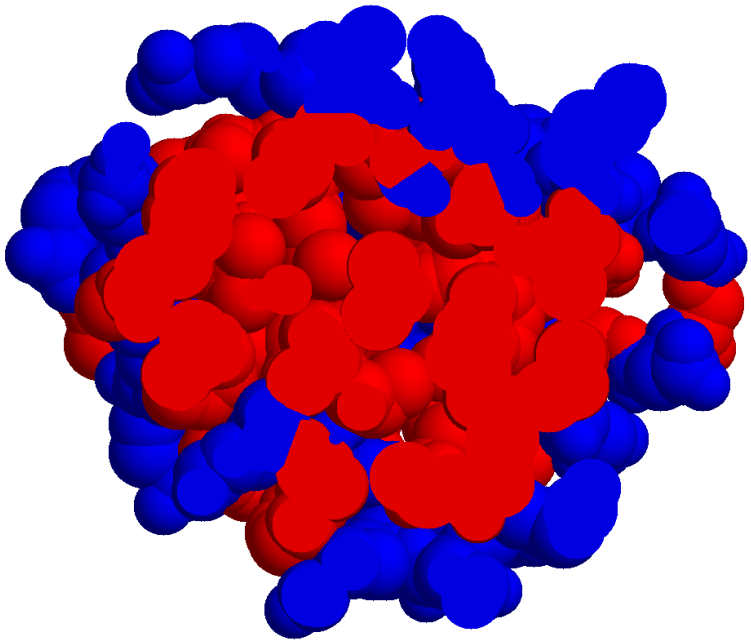
1.5.5 Protein structure determination

There are two common methods of protein structure determination. Most widely used is **X-ray crystallography**. Here, the protein of interest must be 'grown' as a crystal; that is, the

[2] http://www.ebi.ac.uk/ontology-lookup/?termId=MOD:00349
[3] http://www.ebi.ac.uk/ontology-lookup/?termId=MOD:00134
[4] http://www.ebi.ac.uk/ontology-lookup/?termId=MOD:00133



(a) Lysozyme (7lyz)



(b) Lysozyme (7lyz), sliced in half along the Z-axis

Figure 1.8: Hydrophobic residues are buried in the protein core

Hydrophobicity in lysozyme (PDB ID 7lyz). Blue indicates hydrophilic residues, red indicates hydrophobic residues. Figure 1.8(a) shows the whole protein; Figure 1.8(b) shows the same protein, sliced in half along the Z-axis, to expose the patterns of hydrophobicity in the core of the structure. Hydrophilic residues cluster on the surface, while hydrophobic residues predominantly form the core.

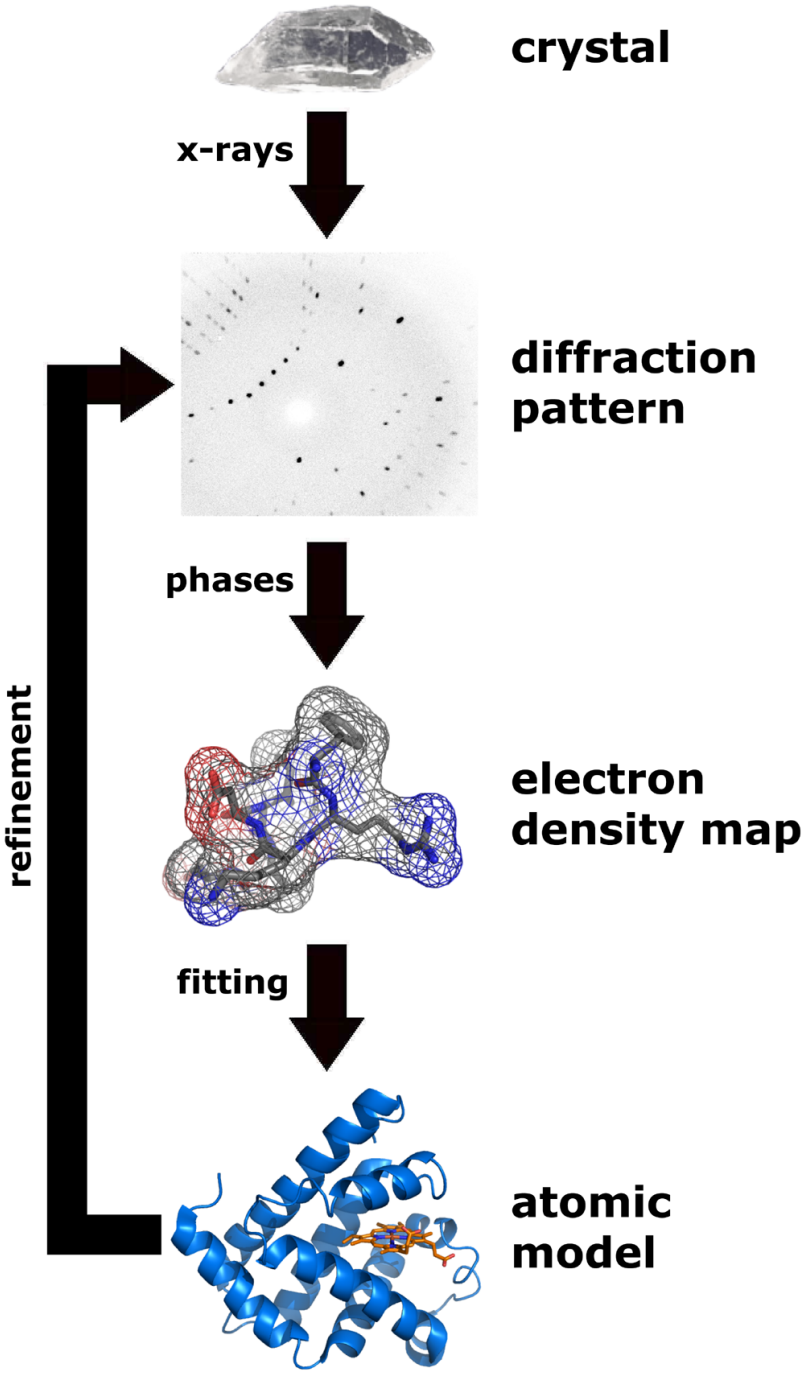


Figure 1.9: The process of X-ray crystallography

Crystalline forms of proteins and bombarded with a stream of X-rays. The resulting diffraction pattern can be interpreted as an electron density map, placing the electrons in 3D space. From this electron density map, a model can be constructed. A process of iterative refinement results in the best model for the diffraction pattern. Image obtained from Wikimedia commons.

same protein structure must be arranged in a repeating, symmetric array. The crystal is bombarded with a stream of X-rays; on colliding with electrons in the crystal, some of the X-rays are diffracted. The pattern and intensity of diffraction is recorded by an X-ray detector placed behind the protein crystal. The pattern of diffraction and the intensity of the diffracted X-rays can be mathematically transformed to yield an electron density map, to which model structures can be fitted (see Figure 1.9). A process of iterative refinement is applied to find the structure that corresponds most closely to the diffraction pattern. Although the structural representation is largely static (large scale motions are inhibited owing to the fact that the structure is in a crystal and where regions do undergo large movements, these are invisible as they are not described by the diffraction pattern) the B-factor, a measure of the electron density spread of each atom, can provide some idea of local mobility. An estimate of confidence, called the R-factor, can be made from the data, by comparing the eventual model with the diffraction pattern. The resolution of an X-ray structure indicates the angle at which the X-rays were diffracted by the crystal (the larger the angle of diffraction, the higher the quality of diffraction pattern and the higher resolution represented by a smaller number).

Nuclear magnetic resonance (**NMR**) spectroscopy is an alternative technology that exploits the 'magnetic moments' or 'spin' of certain atom isotopes. In NMR, a sample solution containing the protein of interest is subjected to a brief magnetic field to disturb the atoms. When the atoms return to their normal state, they emit radiation, which is measured. By comparing the frequency of this radiation to reference values, the 'chemical shift' of each of the atoms in the sample can be measured. Chemical shift data from different radio frequency pulses describe different kinds of interactions (e.g., interactions through bonds or interactions between spatially close atoms which may be distant with respect to sequence). By identifying 'cross-peaks' (patterns of chemical shift peaks that are generated by specific amino acids) and combining data from multiple chemical shift spectra, it is possible to define the interactions of each residue in the sequence. Distance constraints, derived from the peaks in the chemical shift data, are then used to generate possible structures. NMR structures are solved in solution and therefore contain more information about flexibility.

Structures described by X-ray crystallography and NMR are complementary, and each method has its strengths and weaknesses. For example, NMR is limited to small proteins, while crystallography can determine the structure of large proteins, should it be possible to crystallise them; crystallography requires that the protein structure be solid state, while NMR can more successfully model the flexibility of protein structures as the structure is determined in solution;

similarly, while NMR can capture dynamic processes such as protein folding, crystallography allows the more precise characterisation of protein surfaces.

At the time of writing (November 2008), 85.50% (46570/54466) of the protein structures described by the PDB are resolved using X-ray crystallography; 13.94% (7591/54466) are resolved by NMR, and 0.56% (305/54466) are resolved by other means (e.g., electron microscopy). Both kinds of structure are analysed by SAAPdb.

The structural information (including the secondary structure annotations in Figure 1.3(b)) shown in Figures 1.3-1.4 is derived from an X-ray crystal structure of the protein, described by Protein Data Bank record 6adh (the Protein Data Bank, or PDB (Berman *et al.*, 2000), will be described in greater detail in Section 2.1.4).

1.6 Mutating protein structure can affect phenotype

As summarised in Section 1.5, protein folding is a complicated, hierarchical process that relies on the proper formation of scaffolding bonds (particularly hydrogen bonds) between residues in the protein sequence. Should a point mutation arise that alters the protein sequence, the resulting protein structure may change, potentially affecting folding or function. In this thesis, such a mutation is predominantly descrived as a SAAP, or single amino acid polymorphism (see Section 1.4). A resulting functional change can either be (i) a *gain* of function, where the protein acquires a novel (toxic) function; (ii) a *loss* of function, where the protein can no longer perform its native function or (iii) *both* a gain and loss of function. Whether functionality is lost or gained, the functional change caused by the SAAP may compromise native function and the SAAP can be described as 'deleterious'.

The most commonly cited structurally disruptive disease example is that of sickle-cell anaemia. Here, a single mutation A>T replaces a glutamic acid (glu, codon GAG) with a valine (val, codon GTG) at position six in the protein sequence. This SAAP is highlighted in the haemoglobin structure in Figure 1.10. This mutation is distant from the ligand binding site and distant from the other chains in the protein. How then does it cause a disease phenotype?

Glutamic acid is a polar residue and as such is often found on the surface of proteins, where it

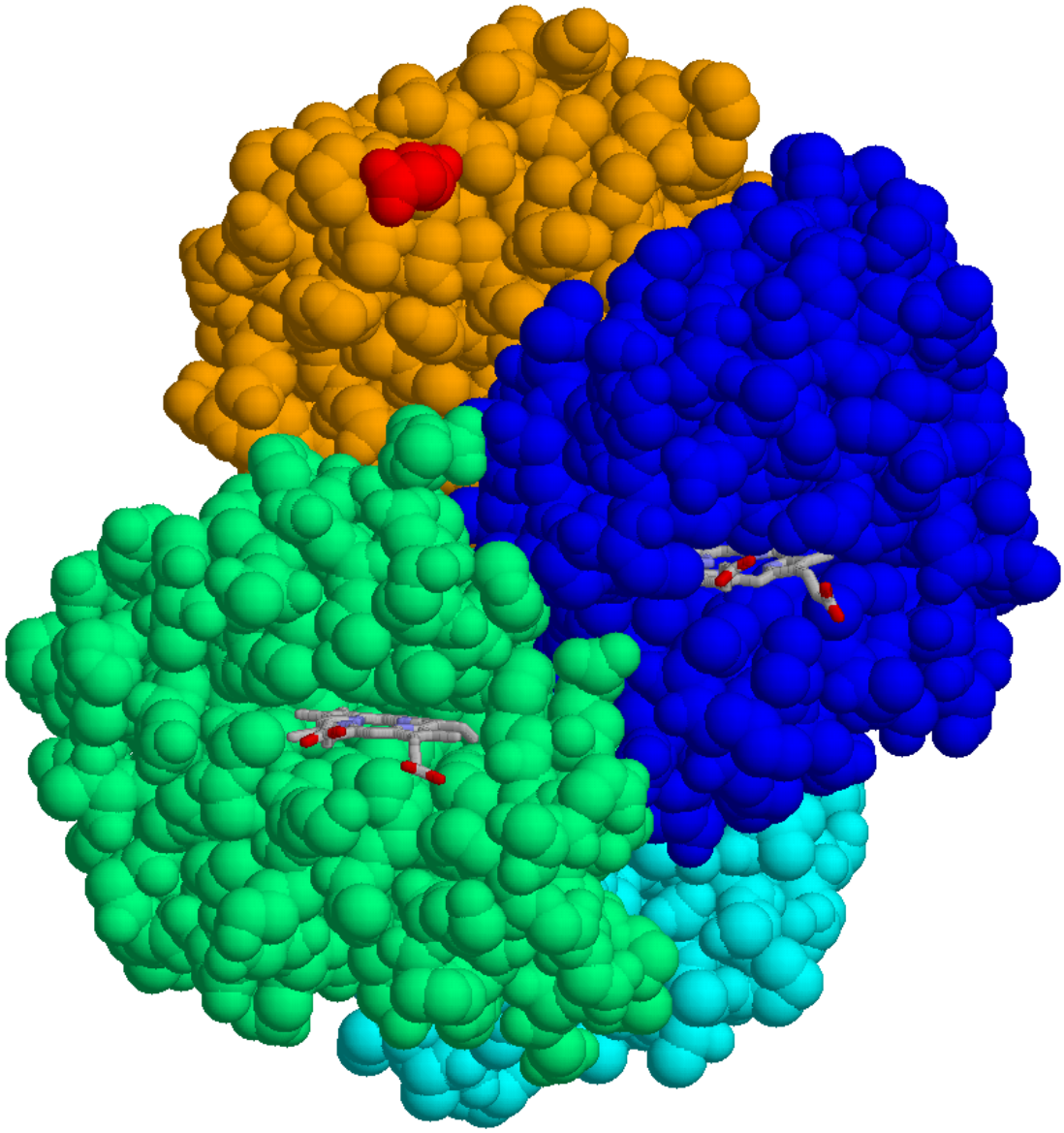


Figure 1.10: Sickle-cell anaemia: the glu6val mutation

The glu6bval mutation known to cause sickle-cell anaemia is highlighted in red in chain B of the haemoglobin structure (PDB ID 1bz0). Chain A, B, C and D are coloured dark blue, orange, light blue and green respectively. Haem ligands are coloured using the CPK colour scheme.

is solvated. Valine, however, is hydrophobic and is less often found on the surface of proteins. Introducing the hydrophobic residue on the surface of the protein causes the protein to aggregate; forming harmful fibrils (see Figure 1.11); deforming the erythrocytes and resulting in the disease phenotype. However, it is found that the sickle-cell phenotype also gives some protection against malaria potentially offering a selective advantage, explaining why the deleterious phenotype has persisted in areas where malaria is common.

However, it is also possible that a SAAP will not compromise function. Although the twenty amino acids are all different, many share similar characteristics (for example, isoleucine, leucine and valine are all small hydrophobic residues) and therefore may be able to replace each other without affecting protein structure and therefore function. As such, SAAPs may also be described as 'neutral'.

1.7 Quantifying the effect on protein structure

The work in this thesis rests upon the hypothesis that it will be possible to identify any structural effect of a deleterious SAAP. That is, where a pathogenic variation in the genome induces a change at the protein level, the deleterious phenotype will be attributable to some disruption of the protein structure and therefore the protein function.

To identify or quantify the structural effect of a particular mutation, the introduced residue must be considered in the context of the protein structure. It will then be possible to assess whether the mutation violates any of the underlying principles of protein structure, as described in Section 1.5. Throughout this thesis, the act of identifying the structural effect of a mutation is described as 'explaining' the mutation. **Note that there there is no** *deleterious* **effect to be explained in the case of SNPs, but the word 'explained' is used to keep the terminology consistent**.

The hypothesis is that disease-associated SAAPs will have a different impact on protein structure from neutral SAAPs. Should it be possible to quantify the structural effect(s) of disease-associated SAAPs and neutral SAAPs, this hypothesis can be tested by comparing the *kinds* of structural effects associated with disease-associated and neutral SAAPs.

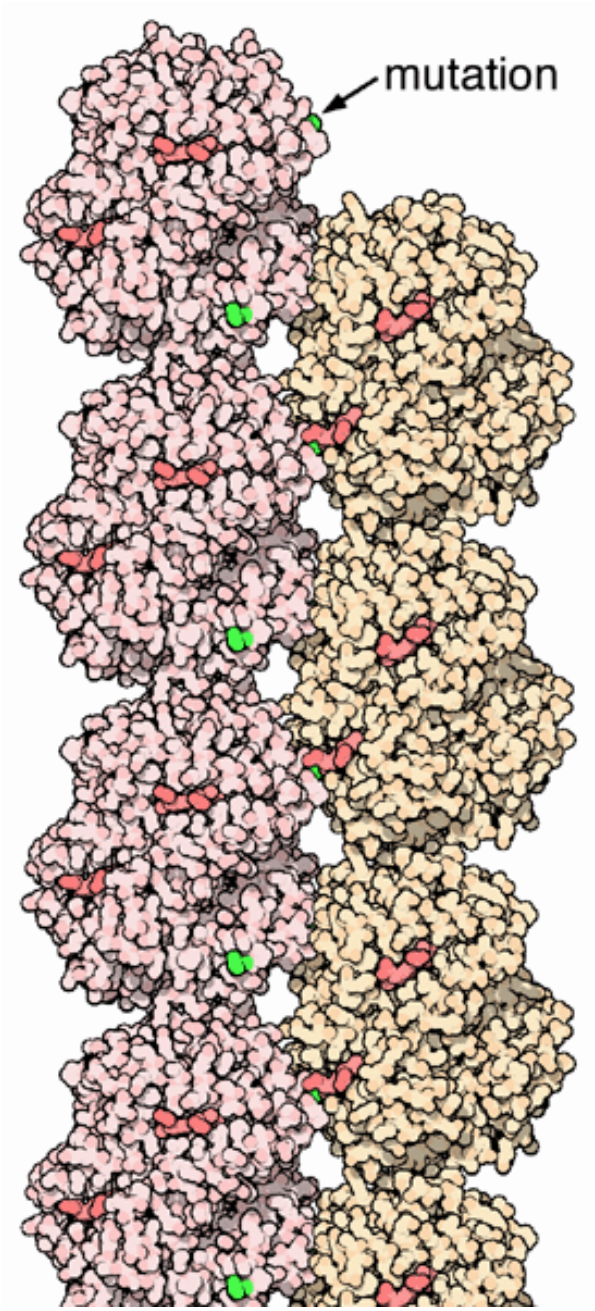


Figure 1.11: Deleterious fibrils in sickle-cell anaemia

A single mutation from glu to val at position 6 in the haemoglobin protein results in aggregation, fibril accumulation, malformed erythrocytes and, therefore, the disease phenotype. The image shows the mutated structure (PDB ID, 2hbs) with the mutant residue highlighted in green in each protein chain. Image taken from the PDB and available from Wikimedia Commons.

1.8 Learning from mutation data

Andrew Martin's group has previously concentrated on explaining mutations in individual proteins, including G6PD (Kwok *et al.*, 2002) and p53 (Martin *et al.*, 2002). In addition, sophisticated analyses have been developed to assess the more complex aspects of protein structure (Cuff and Martin, 2004; Cuff *et al.*, 2006).

One objective of the work described in this thesis is to build on this previous work, maintaining and further developing the existing structural analysis pipeline to process SAAPs automatically.

Should it be possible to hypothesize what the structural effect of an amino acid substitution might be, it may also be possible to predict whether previously unseen non-synonymous mutations will have a significant effect on protein structure or not, and therefore whether the mutation will be deleterious or not. Predictive methods can only work if there is a difference between PDs and SNPs. As such, it is important to characterise and compare both sets of data. Trends identified here could inform which machine learning approaches would be most successful and which data should be used to make predictions. The second objective of the work described in this thesis is to characterise both sets of SAAPs and identify significant differences between them.

Characterising PDs also contributes to the design of novel therapies: should the deleterious effect of the mutant protein structure be attributed to a specific structural abnormality, it may be possible to design a compound that nullifies the deleterious effect, thereby recovering native function. For example, should the mutation destabilise the protein structure, a compound that can either (i) stabilise the mutant form or (ii) chaperone the native structure may be a viable therapy. Two such targets have been reported for cancers caused by destabilisation of P53 (Boeckler *et al.*, 2008; Friedler *et al.*, 2002).

1.9 Characterising pathogenicity: existing work

Table 1.9 summaries 18 characterisations of pathogenic deviations (PDs) and single nucleotide polymorphisms (SNPs) that exist in the literature.

Table 1.1: Existing characterisations of PDs and SNPs
●: PDs were associated with this feature; ○: SNPs were associated with this feature. '-': no relationship was found. ⋆: the paper includes some prediction work. †: the paper considered only deleterious data. A blank cell denotes that the feature was not considered. **Datasets**: *A* = LacI repressor (Suckow *et al.*, 1996); *B* = T4 lysozyme (Rennell *et al.*, 1991); *C* = HIV protease (Loeb *et al.*, 1989); *D* = dbSNP (Smigielski *et al.*, 2000); *J* = uses natural residue variation across species to represent 'neutral' SAAPs; *M* = HMGD (Stenson *et al.*, 2003); *N* = HGVBase (Fredman *et al.*, 2002); *O* = OMIM (McKusick, 1998; Amberger *et al.*, 2009); *S* = UniProtKB/Swiss-Prot VARIANT (The UniProt Consortium, 2009); *X* = other LSMDB (various references). **Structural data used**: Y^* = where PDB structures were unavailable, models were used; Y^i = structural features were inferred from sequence. **AA**: amino acid.

Reference	Data			Seq features			Str features						Extreme changes	Other
	Dataset(s)	Sequence data used?	Structural data used?	AA preference	AA matrix	Conservation	Buried	Destabilising	Secondary structure	Interface	Binding	Buried charge		
Bao & Cui (2005) ⋆	S	Y	Y			●	●		●				●	●
Cai *et al.* (2004) ⋆	AB	Y	Y			●							●	
Chasman & Adams (2001) ⋆	AB	Y	Y			●○	●					●	●	
Clifford *et al.* (2004) ⋆	DX	Y	Y^*											●
Dobson *et al.* (2006) ⋆	S	Y	Y	●							●			
Ferrer-Costa *et al.* (2002)	S	Y	Y		●		●	●	●	●			●	
Ferrer-Costa *et al.* (2004) ⋆	AS	Y	Y^i		●									
Khan & Vihinen (2007) †	MX	n	Y	●					-				●	
Krishnan & Westhead (2003) ⋆	AB	Y	Y^i			●	●					●	●	
Needham *et al.* (2006) ⋆	AB	Y	Y				●		-			●	●	
Ng & Henikoff *et al.* (2001) ⋆	ABC	Y	n			●								
Saunders & Baker *et al.* (2002) ⋆	AC	Y	Y		-	●	●						●	●
Stitziel *et al.* (2003)	DO	Y	Y			●	-	●						●
Steward *et al.* (2003)	O	Y	Y			●	●						●	
Sunyaev *et al.* (2001) ⋆[a]	DNS	Y	Y				●		●		●	●	●	●
Torkamani & Schork (2007)	X	Y	Y	●○		●								●
Verzilli *et al.* (2005) ⋆	AB	Y	Y			●	●	●				●		●
Vitkup *et al.* (2003)	S	Y	Y	●		●	●						●	
Wang & Moult (2001) [b]	DM	n	Y					●						
Yue *et al.* (2005) ⋆	JM	n	Y^*				●	●						●

[a] `http://genetics.bwh.harvard.edu/pph/`

[b] `http://www.snps3d.org/`

Work has predominantly focussed on characterising PDs: compare the number of PD-feature annotations (•) with the number of SNP-feature annotations (◦) in Table 1.9. If the objective is to understand the molecular basis of disease and use that knowledge to design appropriate disease therapies, this is the most important perspective from which to consider the data. However, with a view to *predicting* whether a novel mutation will be deleterious or not, or with a view to extrapolating findings to protein structure in general, it is important to characterise *both* pathogenic and neutral polymorphisms. It may be the case that the most informative characterisation of disease-causing mutations is a characterisation that describes which features are *absent* rather than present. Indeed, using decision trees to build predictors for pathogenicity, Krishnan and Westhead (2003) found that rules predicting 'no effect' (i.e., neutral) were of higher confidence than those predicting 'effect' (i.e., deleterious).

Most commonly, PDs are found (i) in the protein core (i.e., buried); (ii) at sites of high conservation and (iii) to introduce extreme changes in amino acid properties (including hydrophobicity (Ferrer-Costa *et al.*, 2002; Sunyaev *et al.*, 2001; Saunders and Baker, 2002; Cai *et al.*, 2004) and volume (Ferrer-Costa *et al.*, 2002; Khan and Vihinen, 2007)). Other features associated with PDs are most commonly measurements of the structural environment of the mutation, including overpacking or C_β density (Saunders and Baker, 2002; Yue *et al.*, 2005), B-factor (Chasman and Adams, 2001; Verzilli *et al.*, 2005; Needham *et al.*, 2006) and UniProtKB/Swiss-Prot features (Saunders and Baker, 2002).

Only one result challenges the reasonably consistent characterisation that emerges from the other investigations: Stitziel *et al.* (2003) did not find PDs buried in the interior of the protein. However, this is owing to a slightly different definition of 'buried'. Stitziel *et al.* use a geometric analysis of protein structure to categorize residues as belonging to one of three classes: (1) on the surface (2) in a surface crevice or internal void or (3) completely buried in the interior of the protein (i.e., remote from any void). They found that very few PDs nor SNPs belonged to category (3). However, they did find that PDs are more than twice as likely to be found in an internal void or crevice. This is consistent with the characterisation that emerges from the other studies and draws attention to a subtlety of protein structure that has otherwise not been considered: maintainance of the protein core may be dependent on essential, stabilising voids.

In making predictions, structural information is largely found to be more valuable than sequence information (Bao and Cui, 2005; Needham *et al.*, 2006), although where structural information is unavailable or it is difficult to *measure* the structural feature (for example, flexibil-

ity), sequence information can complement structural data when discriminating between PDs and SNPs (Saunders and Baker, 2002). Further, significant differences have been found when considering combinations of features (e.g., native/mutant amino acid and secondary structure (Khan and Vihinen, 2007); B-factor and conservation (Verzilli *et al.*, 2005); accessibility and conservation (Stitziel *et al.*, 2003)).

Further, some associations may be more complex even within the same 'feature'. For example, Chasman and Adams (2001) identified conservation based features associated with *both* PDs and SNPs: PDs were found to be at sites of high conservation, whereas SNPs were found at sites where the introduced residues were identified as the native residue in another species (Chasman and Adams, 2001), explaining the •○ annotation in the corresponding cell in Table 1.9. Torkamani and Schork (2007) identified native and mutant amino acids associated with both kinds of SAAPs.

Two-thirds (12/18) of the methods include some attempt at pathogenicity prediction; as yet, no particular learning method has emerged as superior. Results vary with respect to overclassification—erroneously classifying neutral examples as deleterious (Bao and Cui, 2005)—and underclassification—erroneously classifying deleterious examples as neutral (Cai *et al.*, 2004; Krishnan and Westhead, 2003)—however, prediction accuracy is reasonably consistent at approximately 70-85%. Prediction peformance has peaked at MCC=0.50, where support vector machines were used to predict the pathogenicity of polymorphisms annotated in UniProtKB/Swiss-Prot (denoted with S in Table 1.9) using a 94-dimensional vector of sequence attributes (Tian *et al.*, 2007) (see Section 2.3.5 for a description of binary classification performance statistics, including MCC).

The characterisations in Table 1.9 portray PDs as mutations that primarily disrupt the stability of proteins by altering the protein core. Further, it appears that extreme changes in the amino acid property at the site of mutation are associated with pathogenicity. This motivates the inclusion of both sequence and structural features in an analysis pipeline.

This thesis aims to contribute to this body of work, first by collating data in a resource called **SAAPdb** (**S**ingle **A**mino **A**cid **P**olymorphism **d**ata**b**ase), then by either consolidating or challenging the existing characterisation of deleterious and neutral mutations. The analysis will be motivated by identifying features that will benefit future machine learning methods, and by an understanding of the basic underlying principles of protein structure.

The data generated by the methods described in this thesis have been made publicly available. Only two servers currently exist that provide some characterisaton of *both* SNPs and PDs, with respect to structure—SNP3D[5] (Wang and Moult, 2001) and PolyPhen[6] (Sunyaev *et al.*, 2001)—although many other servers exist that exclusively characterise SNPs (SNPeffect[7] (Reumers *et al.*, 2006), LS-SNP[8] (Karchin *et al.*, 2005), StSNP[9] (Uzun *et al.*, 2007)) and others that simply collate mutation data (MutDB[10] (Dantzer *et al.*, 2005), SNP@DOMAIN[11] (Han *et al.*, 2006), SNAP[12] (Li *et al.*, 2007), TopoSNP[13] (Stitziel *et al.*, 2004)), providing external links to genomic, proteomic and/or pathway data and records. Of these resources, TopoSNP, SNPeffect, LS-SNP, MutDB and StSNP are updated at least once or twice a year (no information is available for the other resources) (Uzun *et al.*, 2007). The SAAPdb resource described in this thesis aims to contribute to this field by providing regularly updated and extensive sequence and structural hypotheses as to the effect of disease and neutral mutations.

1.10 A summary of aims

This thesis will investigate the differential sequence and structural properties and effects of neutral and deleterious point mutations. At a general level, the work described aims to (A) expand the pre-existing suite of analyses which aim to 'explain' the structural effect of a SAAP and (B) compare and contrast the neutral and deleterious mutations with a view to developing predictive methods that will classify a novel point mutation as neutral or deleterious. In order to achieve this, the current SAAPdb pipeline has been extended, requiring the development of several new structural analyses; a method for identifying functionally equivalent proteins, and a method for identifying highly conserved residues. Very preliminary predictive work has shown that the work described in this thesis should contribute significantly to the problem of identifying deleterious mutations (see Appendix [A]).

[5] http://www.snps3d.org/
[6] http://genetics.bwh.harvard.edu/pph/
[7] http://snpeffect.vib.be/
[8] http://modbase.compbio.ucsf.edu/LS-SNP/
[9] http://glinka.bio.neu.edu/StSNP/
[10] http://mutdb.org/
[11] http://snpnavigator.net/
[12] http://snap.humgen.au.dk/
[13] http://gila.bioengr.uic.edu/snp/toposnp/

Chapter 2

An introduction to bioinformatics methods

This thesis describes a database of single amino acid polymorphisms (SAAPs) that have been mapped to structure and subsequently analysed to provide hypotheses as to their effect(s), if any, on protein structure. The resource, named SAAPdb, employs several well established data resources, data handling methods and data analysis methods. In this chapter, these are introduced to provide the context for later chapters.

2.1 Resources

SAAPdb requires data from several sources. SNPs are obtained from dbSNP (Sherry *et al.*, 1999; Smigielski *et al.*, 2000) and HGVBase (Fredman *et al.*, 2002); genomic information is taken from EMBL (Kulikova *et al.*, 2007) and Genbank (Benson *et al.*, 2008); PDs are extracted from OMIM (Amberger *et al.*, 2009) and several smaller locus-specific mutation databases (LSMDBs); protein data are taken from UniProtKB (The UniProt Consortium, 2009) (predominantly UniProtKB/Swiss-Prot), and protein structures are taken from the PDB (Berman *et al.*, 2000). Further, the PDBSWS resource (Martin, 2005) is used to map sequence data onto structural data. These resources and their contents are described in this section.

2.1.1 dbSNP and HGVBase

2.1.1.1 dbSNP

dbSNP is a central repository maintained by the NCBI that collates data about small-scale genomic variation, the vast majority of records (>95%) describing single nucleotide polymorphisms (SNPs) (Sherry *et al.*, 1999; Smigielski *et al.*, 2000). dbSNP accepts submissions of both disease-associated and 'neutral' SNPs and makes no assumptions about allele frequency. Mappings to protein sequence are provided.

The version of dbSNP currently described by SAAPdb is dbSNP build 129[1], which was made available in April 2008 and describes 14 708 752 records for *Homo Sapiens*. The dbSNP data analysed in Chapter 7 is dbSNP build 126 [2], which was made available in May 2006. This build includes 11 961 761 records for *Homo Sapiens* and 6 491 554 records for *Mus Musculus*.

2.1.1.2 HGVBase

The HGVBase (Human Genome Variation database) resource exclusively describes small-scale human genetic variation, the vast majority of which (>95%) are SNPs (Fredman *et al.*, 2002). Although much of the information in HGVBase is complementary to that in dbSNP, the focus here is to collate data relevant to *phenotype*, and stringent quality criteria are applied (Fredman *et al.*, 2002). Unfortunately, however, HGVBase has not been consistently maintained, with only sporadic updates since 2003 (note that a new resource, HGVBaseG2P (Thorisson *et al.*, 2009), has recently become available but was not used in this thesis).

2.1.2 OMIM and LSMDBs

The PD data in SAAPdb are derived from multiple sources.

[1] http://www.ncbi.nlm.nih.gov/projects/SNP/snp_summary.cgi?view+summary=view+summaryi&build_id=129

[2] http://www.ncbi.nlm.nih.gov/projects/SNP/snp_summary.cgi?view+summary=view+summaryi&build_id=126

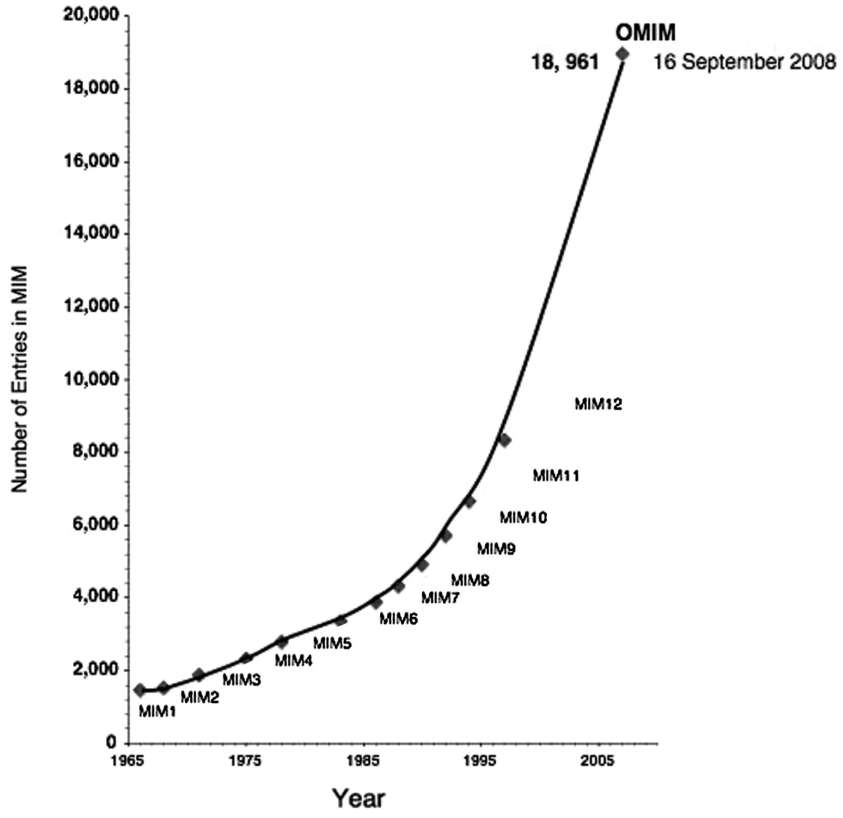


Figure 2.1: (O)MIM growth since 1965
The size of the (O)MIM versions are marked with diamonds; image taken from Amberger *et al.* (2009).

The first and largest of these resources is Online Mendelian Inheritance in Man, or OMIM[3] (Amberger *et al.*, 2009). OMIM is a central description of inherited, disease-associated genetic mutations that is updated on a daily basis and is the online version of the original book resource, MIM (McKusick, 1998). As of October 2008, OMIM contains 19 023 mutations of which 6 514 (34.24%) are characterised phenotypically. OMIM is based on peer-reviewed literature: journal contents are scanned to identify articles of relevance, with particular emphasis on disease phenotypes, genes with novel biology and genes currently absent in OMIM.

With genomic sequencing becoming cheaper and more reliable, the number of PDs being identified is increasing exponentially. Figure 2.1 shows the increase in MIM (1965-1998) and OMIM (2008) content: in the last ten years, the number of disease mutations has increased more than two-fold from approximately 8 000 in 1998 (McKusick, 1998) to almost 20 000 in 2008 (Amberger *et al.*, 2009).

[3] `http://www.ncbi.nlm.nih.gov/omim/`

OMIM provides a wealth of disease-associated information on which substantial bioinformatic analysis can be done. However, data are also available in locus-specific mutation databases or LSMDBs, which are maintained separately by research groups with an interest in a particular disease. Such resources potentially hold much more data, both with respect to quantity and quality (George *et al.*, 2008): such special interest resources may include detailed phenotypic information such as enzymatic function or prognosis. Bioinformatic analysis using these data could reveal more subtle effects on protein structure: rather than training classifiers on the coarse-grained binary classification problem of disease-causing or neutral, methods could learn to predict disease severity. As such, the PD dataset is augmented by seven other mutation datasets.

ADAbase ADAbase[4] is a mutations registry for inherited adenosine deaminase (ADA) deficiency mutations (OMIM:608958) (Piirilä *et al.*, 2006), which account for approximately half of severe combined immunodeficiency disorders (SCID).

ZAP70base A second SCID disease is represented in SAAPdb: the ZAP70base[5] resource describes a small number of mutations to the ZAP70 protein (Piirilä *et al.*, 2006) which cause a ZAP70 deficiency (OMIM:176947).

HAMSTeRS The Haemophilia A Mutation Structure, Test and Resource Site (or HAMSTeRS[6] resource) describes mutations to Factor VIII, the protein absent or defective in haemophilia A (OMIM:306700). These mutations are collated from peer-reviewed literature and electronic submissions.

G6PD The G6PDdb[7] resource was developed in a collaboration with the Martin group (Kwok *et al.*, 2002) and describes mutations to human glucose 6-phosphate dehydrogenase (or G6PD, OMIM:305900), which cause G6PD deficiency (Beutler *et al.*, 1968). This X-linked disease is characterised by abnormal breakdown of red blood cells (haemolysis), in response to infection, chemicals or particular foods; most famously, haemolysis is often induced by fava (broad) beans, explaining why G6PD deficiency is also known as 'favism'.

IARC TP53 The IARC TP53 mutation database[8] catalogues mutations to the gene TP53 and its protein product P53 (Olivier *et al.*, 2002; Petitjean *et al.*, 2007). Mutations to P53 occur

[4] http://bioinf.uta.fi/ADAbase/
[5] http://bioinf.uta.fi/ZAP70base/index2.html
[6] http://europium.csc.mrc.ac.uk/WebPages/Main/main.htm
[7] http://www.bioinf.org.uk/g6pd/
[8] http://www-p53.iarc.fr/

in approximately half of all human cancers (Greenblatt *et al.*, 1994; Sidransky and Hollstein, 1996; Lane and Fischer, 2004). IARC collate both germline (inherited) and somatic (acquired) mutations in P53 (OMIM:191170).

OTC OTC deficiency, a rare metabolic disorder, is caused by mutations to OTC (ornithine carbamoyltransferase, OMIM:300461). It is a disorder of the urea cycle which causes hyperammonemia, an excess of ammonia in the blood (Gilbert-Dussardier *et al.*, 1996). Tuchman *et al.* (2002) describe the dataset that is used in SAAPdb.

SOD1db Amyotrophic lateral sclerosis (ALS) or motor neuron disease (MND) is a progressive, often fatal, neurological disease characterised by the degeneration of motor neurons (OMIM:147450). ALSOD[9] at the Institute of Psychiatry, King's College London describes ALS-associated mutations deposited by registered users. In SAAPdb, only ALS-associated mutations to superoxide dismutase or SOD1 (a dataset that is referred to as SOD1db) are analysed.

At the time of writing (November 2008), over 700 LSMDBs are recorded on the Human Genome Variation Society's website (`http://www.hgvs.org/dblist/glsdb.html`). Although SAAPdb only includes a small fraction of these data, the system has been designed and implemented so that integrating more locus-specific data is straightforward.

2.1.3 UniProtKB and UniProtKB/Swiss-Prot

UniProtKB[10] is the world's most comprehensive resource of protein information (The UniProt Consortium, 2009). It is comprised of several smaller databases, including UniProtKB/Swiss-Prot and UniProtKB/trEMBL. These datasets differ in their level of curation and annotation. UniProtKB/Swiss-Prot is a manually annotated dataset, which aims to reduce redundancy, improve annotation and provide comprehensive cross-references to other resources. UniProtKB/trEMBL however is an automatic translation of the genome as described by EMBL. It therefore contains a great deal of redundancy and little annotation (where annotation does exist, it is transferred by homology and has not been experimentally verified).

SAAPdb almost exclusively uses UniProtKB/Swiss-Prot; UniProtKB in its entirety is only

[9] `http://alsod.iop.kcl.ac.uk/Als/index.aspx`
[10] `http://www.uniprot.org/`

```
 1:  ID   TACY_LISMO              Reviewed;         529 AA.
 2:  AC   P13128; Q48747; Q57096; Q57206;
     ...
 3:  DE   Listeriolysin O precursor (Thiol-activated cytolysin) (LLO).
     ...
 4:  DR   EMBL; X15127; CAA33223.1; -; Genomic_DNA.
 5:  DR   PIR; A43505; A43505.
     ...
 6:  FT   SIGNAL        1     25
 7:  FT   CHAIN        26    529       Listeriolysin O.
 8:  FT                                 /FTId=PRO_0000034102.
 9:  FT   SITE         484    484      Binding to cholesterol (By similarity).
10:  FT   VARIANT       35     35      S -> L (in strain: F4233 / Serotype 1/2b,
11:  FT                                F5782 / Serotype 4b, F6789 / Serotype 1/
12:  FT                                2b and 12067).
13:  FT   VARIANT      438    438      V -> I (in strain: F4233 / Serotype 1/2b,
14:  FT                                F5782 / Serotype 4b, F6789 / Serotype 1/
15:  FT                                2b and 12067).
16:  FT   VARIANT      523    523      K -> S (in strain: F4233 / Serotype 1/2b,
17:  FT                                F5782 / Serotype 4b, F6789 / Serotype 1/
18:  FT                                2b and 12067).
     ...
19:  //
```

Figure 2.2: An example of a UniProtKB/Swiss-Prot record

The above record is for [UniProtKB:TACY_LISMO/P13128], in UniProtKB/Swiss-Prot version 13.5/55.5;it has been edited only to include those data that are relevant for SAAPdb and FOSTA, i.e., ID (the **id**entifier), AC (the **ac**cession number), DE (the **de**scription field),DR (**d**atabase cross-**r**eference line) and FT (annotated **f**ea**t**ures);records are terminated with a \\; line numbers are given on the left for references in the text and '...' are used to indicate skipped lines.

used to construct mappings between accession codes (ACs, see Section 2.1.3.1 below) and to map protein records to gene records (see Section 6.2.2). All functional annotation is provided by UniProtKB/Swiss-Prot by means of the `uniprot_sprot.dat` flatfile, released with every new version of UniProtKB. The format of this file has recently changed (see `http://www.uniprot.org/docs/xml_news.htm`). However the UniProtKB/Swiss-Prot data used in SAAPdb at present is UniProtKB/Swiss-Prot version 55.5, dating from June 2008. This section describes UniProtKB/Swiss-Prot version 55.5 (the SAAPdb parsers have recently been updated to deal with the changed format).

Each UniProtKB/Swiss-Prot protein is described in a separate record using start-of-line, two-character keys to classify the fields. An example is shown in Figure 2.2. Records are separated by a line containing only the string '\\' (line #19 in Figure 2.2). The data that are relevant to this work are described in the remainer of this section (a full description of the UniProtKB/Swiss-Prot file format can be found at `http://www.expasy.ch/sprot/userman.html`).

2.1.3.1 The UniProtKB/Swiss-Prot identifier and accession number

Each UniProtKB record is described with *both* an identifier ID and an accession number (AC). The ACs are a string of 6 alphanumeric characters (currently beginning with A, P, Q or O). Once an AC has been assigned to a protein sequence, either in UniProtKB/Swiss-Prot or UniProtKB/trEMBL, it is guaranteed always to refer to that particular protein (although the sequence records may be amended). Should records be merged or deleted, the original AC will be retained as a 'secondary' AC to the new 'primary' AC. In the example, the ID is TACY_LISMO and the primary AC is P13128 (lines #1-2 in Figure 2.2). There are three secondary ACs: Q48747, Q57096 and Q57206 (the primary AC is simply the first AC provided, see line #2 in Figure 2.2).

The IDs are of the format PROTEIN_SPECIES, where PROTEIN is a string indicating what the protein is or does, and SPECIES is a string describing the species from which the sequence has been derived. The steadily expanding (and occassionally revised) vocabulary of species is described and made available at `http://www.uniprot.org/taxonomy/`. IDs are *not* guaranteed to remain the same. For example, in UniProtKB/Swiss-Prot version 4.0/46.0, human protein C had the identifier PRTC_HUMAN; in the successive version, the ID changed to PROC_HUMAN. It has, however, always been identifiable with the AC P00470.

When working with UniProtKB/Swiss-Prot data, it is important to ensure data integrity by always using *primary* accession numbers.

2.1.3.2 The description field

The description or DE field contains a description of the protein. Proteins may be described using any number of synonyms. In the example, there are three:"Listeriolysin O precursor", "Thiol-activated cytolysin" and "LLO" (line #3 in Figure 2.2). Also included in this line is an indication of whether the protein is a "(Fragment)" or not, and any relevant EC number(s).

2.1.3.3 The database cross-references

UniProtKB/Swiss-Prot provides cross-references between databases. These data are used later to construct datasets with which to benchmark a novel method of identifying functionally

equivalent proteins (FOSTA) against a similar existing method (Inparanoid) (see Section 3.3.7). In the example, TACY_LISMO is cross referenced to EMBL records X15127 and CAA33223.1, and PIR records A43505 and A43505 (lines #4-5 in Figure 2.2).

2.1.3.4 Annotated features

UniProtKB/Swiss-Prot provides annotations of sequence, structural and functional features that are found in the protein (see lines #4-5 in Figure 2.2). These may be transferred by homology, or there may be experimental evidence for the feature in the specific protein; however this information is not guaranteed to be included. The use of UniProtKB/Swiss-Prot FT information is described fully in Section 5.11.

2.1.4 PDB

The PDB is the largest publicly available repository for 3D data describing biological macromolecules (Berman *et al.*, 2000). Structures are primarily solved using X-ray crystallography (see Section 1.5.5). PDB files are plain text, most importantly describing the 3D coordinates of each atom. Residues are described simply by annotating each constituent atom with the same residue ID. In addition to the atomic coordinates, PDB files contain information regarding the method by which the structure was solved; references to the literature; cross-references to other resources (e.g., UniProtKB); specification of ligands, and so on.

Unfortunately, the format is not well structured and standards are not enforced consistently. Consequently, parsing is difficult. In Section 2.2.3 an alternative format, developed by Andrew Martin while at Inpharmatica, for the PDB that is more amenable to automated parsing is described.

2.1.5 PDBSWS

SAAPdb uses the PDB-to-UniProtKB mapping PDBSWS (Martin, 2005) to map sequence residues to their corresponding structural residues. It relies on accurate mapping between

UniProtKB IDs and ACs, and accurate mapping between primary and secondary accession numbers (see Section 2.1.3). These data are parsed from the UniProtKB release (from both UniProtKB/Swiss-Prot and UniProtKB/trEMBL). As of October 2008, 96.34% of PDB protein chains are successfully mapped to a UniProtKB sequence.

The PDBSWS database is populated as follows:

1. Extract UniProtKB cross-references from PDB files
2. Extract PDB cross-references from UniProtKB files
3. Brute-force scan the remaining PDB chains against UniProtKB
4. Align PDB sequence with UniProtKB sequence to generate the PDBSWS mapping

In Step 1, UniProtKB accession (AC) and identifier (ID) references are parsed from the `DBREF` field from all PDB files. If no UniProtKB cross-reference exists in the `DBREF` field, the `REMARK 999` field is parsed in an attempt to find UniProtKB AC or ID references. Invalid UniProtKB references (e.g., references to obsolete IDs) are flagged to be analysed later in the brute-force scan (Step 3). References to UniProtKB IDs are replaced with their corresponding AC and all AC references are updated to the current primary AC.

Next, PDB references are extracted from UniProtKB (Step 2). The UniProtKB sequence is aligned with each PDB chain in turn to identify which chain (or chains) are relevant. UniProtKB now includes chain information, but this protocol was designed at a time when it did not.

Steps 1 and 2 may yield multiple matches as a protein sequence can map to multiple PDB structures, and to several chains within a single PDB structure. All identified mappings are stored in the PDBSWS relational database implemented in PostgreSQL.

The final PDB/AC mapping step is a brute-force scan, which attempts to match all remaining PDBs to UniProtKB ACs. A PDB sequence is reconstructed from the ATOM records. This sequence is then searched for in UniProtKB (UniProtKB/Swiss-Prot and UniProtKB/trEMBL) using `fasta33` (Pearson and Lipman, 1988). The best match is identified and the mapping is recorded if (i) the residue overlap is $\geq$30 and the identity is at least 90% (ii) the residue overlap is $\geq$15 and the identity is at least 93%, or (iii) the entire chain is matched with 100% identity.

Once all possible PDB-UniProtKB record-to-record mappings are identified, the two sequences are aligned, using `ssearch33` (Pearson and Lipman, 1988), to generate the PDB-UniProtKB *residue-to-residue* mappings. These data are freely available via a webserver or in flatfile format at `www.bioinf.org.uk/pdbsws`.

2.1.6 EMBL and Genbank

EMBL (Kulikova *et al.*, 2007) and Genbank (Benson *et al.*, 2008) are two publicly available nucleotide sequence databases, EMBL being curated by the EBI and Genbank being curated by the NCBI. Data are derived from submissions from individual researchers, large-scale genome sequencing projects and patent records. Each record describes a particular section of DNA and includes annotations of coding regions, database cross-references, literature citations, biologically relevant features and so on. EMBL and Genbank exchange data on a regular basis, so the sequence content should be identical.

2.2 Data handling

It is essential that appropriate and robust data handling is employed in large-scale, automated systems such as SAAPdb to ensure data integrity. The vast quantities of information involved require that data are retrieved and processed quickly and reliably. Here, several of the fundamental data handling methods are introduced: relational databases (Section 2.2.1), XML and the associated XML translation specification XSLT (Section 2.2.2) and an alternative representation of the PDB, XMAS, which is based on a combination of ideas from XML and ASN.1 formats (Section 2.2.3).

2.2.1 Relational databases

Relational databases are a means of storing information. Data are structured as tables consisting of columns or 'fields' and contain data in unordered rows or 'tuples'. The 'relational' aspect of these data structures is in the use of 'foreign keys' and common attributes, which refer to equivalent data in different tables. This allows for potentially very large tables, with many

fields and much data redundancy, to be 'normalised' into smaller data structures describing individual concepts.

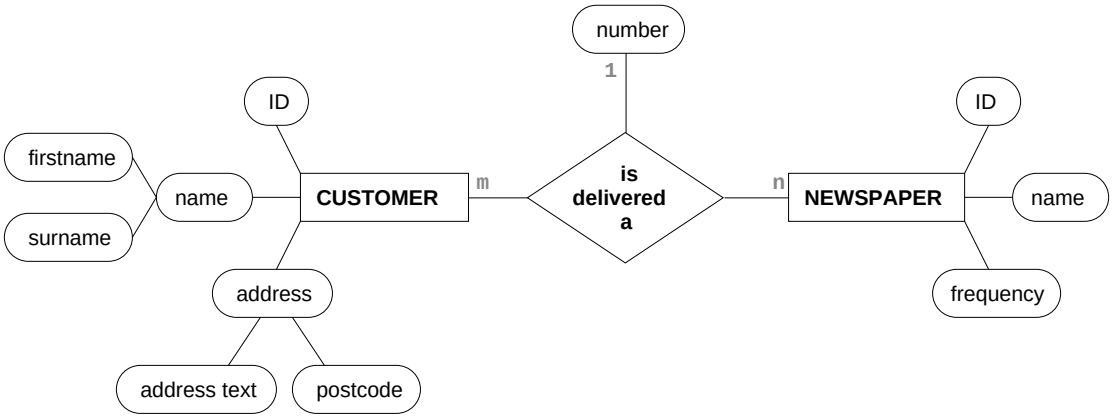
To illustrate the key concepts in a relational database, a small example dataset will be used throughout this section. The example being used is a list of newspaper deliveries, where customers can have any number of different papers delivered to an address. The first step in good database design is to decompose the problem into its constituent 'entities', 'relationships' and 'attributes'. Entites describe distinct objects in the dataset. Combining entities using relationships allows more abstract entities to exist. Further, both entities and relationships can have attributes which describe a corresponding object. Good database design often begins with an 'entity-relationship' (ER) diagram that clearly defines what the entities and relationships are in the data to be stored.

Figure 2.3(a) describes the entities, relationships and attributes in the newspaper delivery example. There are two entities: a customer and a newspaper. These entities are joined by the relationship 'is delivered a' which captures the more abstract or 'associative' delivery entity. Both entities and relationships can have attributes: a customer has a name and an address, a newspaper has a name and frequency (i.e., daily/weekly) and a delivery is defined by the number of papers that are to be delivered. In addition, each entity is given an 'ID' which will allow each example of an entity (i.e., each customer or newspaper) to be identified uniquely.

Often, it is useful to decompose an attribute into two or more further attributes. These attributes are 'multi-valued' attributes. In the example, a customer's name has been split into 'firstname' and 'surname' and the address is split into the 'address' text and the 'postcode'. This allows for direct access to the sub-attribute.

Relationships between entities should also be defined with respect to cardinality, which describes how entities are related to each other. The cardinality may be many-to-many, one-to-many or one-to-one. In the example, the relationship between the customer entity and the newspaper entity is many-to-many, as a customer may have more than one newspaper delivered and similarly a newspaper may be delivered to more than one customer; this is shown in Figure 2.3(a) as grey text attached to the relationship connectors.

With the ER diagram completed, the application of several rules will derive a suitable database design. In the newspaper example, the relevant rules are that (i) each entity should be repre-



(a) An example entity relationship (ER) diagram
Entities are in square-edged rectangles while attributes are in round-edged boxes. Lines join entities and relationships to their attributes. Grey monospace text indicates the cardinality of a relationship.

Customer	Address	Newspaper	Frequency	Number
Will Guthrie	18 Craigholme PA6 7DB	The Guardian	daily	1
Chris Guthrie	18 Craigholme PA6 7DB	The Sunday Herald	weekly	1
Ewan Tavendale	Craigielea by Quarriers PA11 3SX	The Local Gazette, The Guardian	daily, daily	1,2
Chae Strachan	48 Minerva Way G3 8GA	The Local Gazette, The Sun	daily, daily	1,1
Molly Douglas	88 Kent Road G3 4MH	The Guardian	daily	2

CUSTOMER

ID*	firstname	surname	address^
1	Chris	Guthrie	1
2	Ewan	Tavendale	4
3	Chae	Strachan	2
4	Molly	Douglas	3
5	Will	Guthrie	1

ADDRESS

ID*	address	postcode
1	18 Craigholme	PA6 7DB
2	48 Minerva Way	G3 8GA
3	88 Kent Road	G3 4MH
4	Craigielea by Quarriers	PA11 3SX

NEWPAPER

ID*	name	frequency
1	The Local Gazette	daily
2	The Guardian	daily
3	The Sun	daily
4	The Sunday Herald	weekly

DELIVERY

ID*	customer^	paper^	number
1	1	4	1
2	2	1	1
3	2	2	2
4	3	1	1
5	3	3	1
6	4	2	2
7	5	2	1

(b) An example relational database
The data to be represented is a list of newspaper deliveries, shown in the top half of Figure 2.3(b).These data can be decomposed into smaller entities (Customer, Newspaper, Delivery) and stored in separate tables as shown in the bottom half of Figure 2.3(b). Primary keys are annotated with an asterisk (*), foreign keys are annotated with a caret (^). Primary keys in the Customer, Address and Newspaper tables are highlighted in blue, red and green respectively. The same colours are used in the Delivery table to indicate where these primary keys are used as foreign keys.

Figure 2.3: Using a toy dataset of newspaper deliveries to illustrate database design

sented by a table (ii) each many-to-many relationship should be represented by a table and (iii) any multiple attribute for which there are dependencies between the sub-attributes (e.g., the address text and the postcode in the customer's address) should be factored out into a different table. The resulting database and its relationship to the original data are shown in Figure 2.3(b).

Two fundamental concepts in relational databases are primary and foreign keys. Primary keys are IDs that allow each example in a table to be identified uniquely. Most often they are arbitrary numbers applied to data as they are entered into the database. Foreign keys are references to external fields, that is, fields in other tables. In Figure 2.3(b) all primary keys are marked with an asterisk (*) and all foreign keys are annotated with a caret (^); further, all foreign keys and the data to which to they refer are highlighted with the same background colour to ease identification of inter-table referencing. Using foreign keys in a well designed database improves data integrity and facilitates administration as changes need only be made in one table.

Additional 'constraints' may be placed on fields in a table to improve data integrity and performance further. These can define whether a field must be unique, whether a field must be present and not 'null', or what range of values the field may take.

One final mechanism of relational databases that vastly improves performance is indexing. Indexing generates a secondary table that permits rapid look-up of the original data. Any field, or combination of fields, that are used frequently in constraining a search (i.e., often used as elements of a 'WHERE' clause, see below) should be indexed. In both FOSTA (Chapter 3) and SAAPdb (Chapter 6), indexes are used extensively for practicable use of the large datasets.

Once the database has been successfully designed, structured query language (SQL) can be used to build, populate and query the database. The PostgreSQL database management system is used throughout this thesis. Foreign keys are used to retrieve related data by 'joining' tables together using a common term or terms. An example query is shown in Figure 2.4, which requests the total number of each paper that is delivered daily. This query employs the basic `SELECT/FROM/WHERE` grammar, but also uses `GROUP BY` and `ORDER BY` to aggregate and sort the data respectively, and `SUM()`, one of many built-in, standard SQL functions. PostgreSQL also allows the user to define new functions.

```
mcmillan=> SELECT n.name, SUM(d.number)
FROM newspaper n, delivery d
WHERE d.paper = n.id
AND n.frequency = 'daily'
GROUP BY n.name
ORDER BY n.name;
       name        | sum
-------------------+-----
 The Guardian      |   5
 The Local Gazette |   2
 The Sun           |   1
(3 rows)
```

Figure 2.4: An example PostgreSQL query
Two tables (`newspaper` aliased to `n` and `delivery` aliased to `d`) are joined on `d.paper` and `n.id`; the data are constrained to those `newspapers/n` with a daily frequency (`n.frequency = 'daily'`); the aggregate function `SUM` is calculated for each `n.name` as defined by the `GROUP BY n.name` clause; results are sorted by n.name as defined by the `ORDER BY n.name` clause; all PostgreSQL commands and functions are given in capitals.

2.2.2 XML and XSLT

XML (eXtensible Markup Language) is a standard for document markup. By defining a restricted grammar of elements and attributes, the user can define a specialised framework for representation and storage of their data. A DTD (document type definition) defines such a framework.

By representing data in the same XML format, the same parser can be used to extract relevant data for processing, database population and so on. Figure 2.5 shows an excerpt from an XML file, which is taken from an RSS (version 2.0) feed of a website. The enclosing element is an 'rss', within which a 'channel' is described. Within a channel, there is one instance of the elements 'title', 'link', 'description' and 'language', followed by multiple 'item's that enclose further sub elements ('title', 'link', 'description' and 'guid'), demonstrating the hierarchical nature of the XML structure. The corresponding DTD that completely specifies RSS 2.0 is shown in Figure 2.6.

Extensible stylesheet language translation (XSLT) is a method for translating XML into another format. Commonly it is used to translate XML into HTML for display on a website. An example

```
<rss>
<channel>

    <title>
        Craigends of the 20th Century
    </title>
    <link>
        http://example.com/features/
    </link>
    <description>
        Updates on new featues at craigends.net.
    </description>
    <language>
        en-us
    </language>

    <item>
        <title>
            Johnstone Advertiser: 15th January 1960
        </title>
        <link>
            http://craigends.net/feature/26/
        </link>
        <description>
            The story of a German POW who painted the murals at Craigends Stables.
        </description>
        <guid>
            http://craigends.net/feature/26/
        </guid>
    </item>

    <item>
        <title>
            Driving Mrs Cuninghame
        </title>
        <link>
            http://craigends.net/feature/27/
        </link>
        <description>
            The work and duties of a Craigends chauffeur.
        </description>
        <guid>
            http://craigends.net/feature/27/
        </guid>
    </item>

</channel>
</rss>
```

Figure 2.5: An example of XML, here taken from an RSS feed from a website
See Figure 2.6 for the corresponding DTD.

```
<!ELEMENT rss (channel)>
<!ATTLIST rss version CDATA #FIXED "2.0">

<!ELEMENT channel ((item+)|
                   (title,link,description,(language|copyright|
                    managingEditor|webMaster|pubDate|lastBuildDate|
                    category|generator|docs|cloud|ttl|image|
                    textInput|skipHours|skipDays)*))>

<!ELEMENT item ((title|description)+,link?,
                (author|category|comments|enclosure|guid|pubDate|source)*)>

<!ELEMENT author (#PCDATA)>
<!ELEMENT category (#PCDATA)>
<!ATTLIST category domain CDATA #IMPLIED>
<!ELEMENT cloud (#PCDATA)>
<!ATTLIST cloud domain CDATA #IMPLIED
                port CDATA #IMPLIED
                path CDATA #IMPLIED
                registerProcedure CDATA #IMPLIED
                protocol CDATA #IMPLIED>
<!ELEMENT comments (#PCDATA)>
<!ELEMENT copyright (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT docs (#PCDATA)>
<!ELEMENT enclosure (#PCDATA)>
<!ATTLIST enclosure url CDATA #REQUIRED
                    length CDATA #REQUIRED
                    type CDATA #REQUIRED>
<!ELEMENT generator (#PCDATA)>
<!ELEMENT guid (#PCDATA)>
<!ATTLIST guid isPermaLink (true|false) "true">
<!ELEMENT height (#PCDATA)>
<!ELEMENT image (url,title,link,(width|height|description)*)>
<!ELEMENT language (#PCDATA)>
<!ELEMENT lastBuildDate (#PCDATA)>
<!ELEMENT link (#PCDATA)>
<!ELEMENT managingEditor (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT pubDate (#PCDATA)>
<!ELEMENT skipDays (#PCDATA)>
<!ELEMENT skipHours (#PCDATA)>
<!ELEMENT source (#PCDATA)>
<!ATTLIST source url CDATA #REQUIRED>
<!ELEMENT textInput (title,description,name,link)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT ttl (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT webMaster (#PCDATA)>
<!ELEMENT width (#PCDATA)>
```

Figure 2.6: An example of DTD for RSS 2.0

This DTD specifies the format of RSS 2.0: which elements with which attributes can exist; what the hierarchical relationships between the elements are; what data type the attributes are (PCDATA/CDATA) and whether data is required or may be omitted.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method='html' version='1.0' encoding='UTF-8' indent='yes'/>

<xsl:template match="/">

  <html>

  <head>
  <style type="text/css">
    body { width: 800px; margin: auto; }
    table { border-collapse: collapse; border-bottom: 2px solid black;
            border-top: 2px solid black; }
    td, th { text-align: left; padding: 10px; }
    td { vertical-align: top; border-bottom: 1px dotted black; }
    th { background: #CFE9FF; border-bottom: 2px solid black; }
    td.title { font-weight: bold; }
  </style>
  </head>

  <body>
  <h2>Craigends Updates</h2>
    <table>
      <tr>
        <th>Title</th>
        <th>Description</th>
      </tr>

      <xsl:for-each select="channel/item">
      <tr>
        <td class="title"><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="description"/></td>
      </tr>
      </xsl:for-each>

    </table>
  </body>

  </html>

</xsl:template>
</xsl:stylesheet>
```

Figure 2.7: An example of XSLT
This XSLT converts the XML shown in Figure 2.5 into HTML, as shown in Figure 2.8; the `xsl:for-each` loop identifies each `item` in the `channel` tag and displays them in tabular format.

Craigends Updates

Title	Description
Johnstone Advertiser: 15th January 1960	The story of a German POW who painted the murals at Craigends Stables.
Driving Mrs Cuninghame	The work and duties of a Craigends chauffeur.

Figure 2.8: The browser's view of XML (Figure 2.5) translated to HTML using XSLT (Figure 2.7).

XSLT specification is shown in Figure 2.7. This XSLT converts the XML shown in Figure 2.5 into HTML (see Figure 2.8). Chapter 6 (specificially Section 6.2.6.4) describes how XSLT is used to translate XML into SQL statements when populating SAAPdb.

2.2.3 An alternative format for the PDB: XMAS

There are numerous flaws in the PDB data format, not least a lack of adherence to a standard format (see Section 2.1.4). In addition, there is a wealth of information implicit in the PDB files (for example, protein ligand bonds and accessibility) that is not explicitly stated and must be calculated for each individual PDB structure. In the context of large automated structural analysis systems such as SAAPdb, a standardised format containing all relevant PDB data, that is easily parsed is essential. Ideally programs for adding information on hydrogen bonds, accessibility and so on, may be run in any order, extending the data stored with the structure in a self-describing way.

The XMAS format of PDB structures was developed by Dr Andrew Martin while at Inpharmatica and represents PDB data using a hybrid XML/ASN.1 format (the XMAS name is derived from the first two letters of XML and ASN.1). XMAS files are used extensively in SAAPdb.

Conversion from PDB to XMAS format is as follows:

1. Convert raw PDB data to XMAS format
2. Calculate and add atom and residue solvent accessibility statistics
3. Calculate and add secondary structure assignments for each residue
4. Identify and add hydrogen bonds in the structure

Solvent accessibility (step 2) is calculated using the method of Lee and Richards (1971) and secondary structure assignments (step 3) are calculated using the method of Kabsch and Sander (1983) as modified by Smith and Thornton (unpublished). Protein-protein, protein-ligand and ligand-ligand hydrogen bonds are identified using the simple Baker and Hubbard (1984) criteria for defining a hydrogen bond (step 4). In addition, non-bonds (non-consecutive residue atom pairs 2.7-3.35Å apart that are not covalently bonded or hydrogen bonded, for example,

electrostatic interactions and Van der Waals contacts) and pseudo-Hbonds (atom pairs satisfying the constraints described in Baker and Hubbard (1984) for hydrogen bonding, where one or both atoms do not strictly form hydrogen bonds, for example, metal ions) are identified and annotated.

Locally, XMAS files are automatically generated for all new or updated structures from the PDB. XMAS formatted structures are easily generated for mutant structures where necessary using proprietary software.

All the desirable requirements for file formats would be achieved by using XML, allbeit with considerably larger files sizes than XMAS files. In fact, PDB data are available in XML format[11] (Westbrook *et al.*, 2005). However, no functionality exists for generating the additional data in XML and methods for handling the XMAS format were already implemented as part of the SAAPdb system; there are no plans to update these as yet.

2.3 Methods and tools

Several established bioinformatics methods and tools are referred to throughout this thesis. These are primarily methods of sequence alignment (MUSCLE, Needleman & Wunsch and amino acid substitution matrices) or sequence similarity searching (BLAST). These are described in this section.

2.3.1 BLAST

BLAST (Basic Local Alignment Search Tool) is a method by which similar sequences to a protein[12] of interest, or *query* sequence, may be retrieved from a database (Altschul *et al.*, 1990). BLAST identifies similar proteins by identifying smaller regions of high sequence similarity. The use of an index of these smaller protein 'words' makes the search of large protein sequence databases feasible.

BLAST decomposes the query sequence into its constituent set of words (for proteins, BLAST

[11] http://pdbml.pdb.org/

[12] BLAST may also be used for DNA sequences but is only used to identify similar *proteins* in this thesis

uses a default word length of three). These words (plus similar, neighbouring words) are searched for in a similarly decomposed, indexed database. Matches (including non-exact matches scoring above a threshold) to these words that are found in the decomposed database are expanded at both ends in an attempt to build ungapped alignments between the query sequence and the corresponding sequence in the database. Any expanded alignment that exceeds a pre-defined threshold is returned as a match or 'hit'. Each hit is scored on how similar it is to the query sequence.

Using statistical theory, the scores are compared with the distribution of scores generated from the entire database. BLAST provides an E-value (expectation value) for each hit. The P-value describes the probability that the score for an alignment is no better than random. The E-value describes how many equal-or-better scores are expected to be found by chance in this database when queried with this sequence. For example, should a hit have an E-value of 0.02, there is a one in fifty chance that an alignment of the same or better quality would occur by chance alone. This is dependent on the size and content of the database. Thus the E-value is the P-value multiplied by the database size. However, in practice, the E-value is calculated from integrating the tail of an extreme value distribution fitted to the data.

2.3.2 MUSCLE

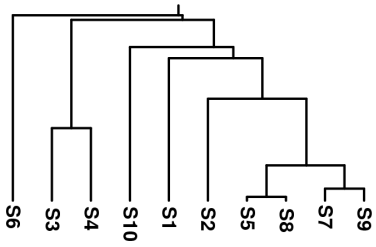
MUSCLE (Edgar, 2004a; Edgar, 2004b) is a method for generating multiple sequence alignments (MSAs) of proteins. The iterative approach is considered to be more accurate and is significantly faster than the current *de facto* standard multiple sequence alignment program, ClustalW (Thompson *et al.*, 1994a). The algorithm has three stages: (i) generating a draft alignment with which to start optimisation; (ii) improving this initial alignment; and finally (iii) refining the alignment.

Each stage employs the same general mechanism of generating a *progressive* alignment:

1. Construct a distance matrix (D) of the sequences to be aligned
2. Use D to construct a hypothesized phylogenetic tree, P
3. Build the progressive alignment by performing a pairwise alignment at each node of the binary tree P

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	0.00	0.48	0.66	0.67	0.64	0.71	0.63	0.49	0.60	0.63
S2		0.00	0.74	0.71	0.46	0.67	0.46	0.32	0.40	0.48
S3			0.00	0.28	0.70	0.74	0.70	0.68	0.70	0.71
S4				0.00	0.71	0.74	0.71	0.65	0.71	0.70
S5					0.00	0.72	0.12	0.01	0.12	0.68
S6						0.00	0.72	0.70	0.70	0.72
S7							0.00	0.16	0.05	0.67
S8								0.00	0.17	0.56
S9									0.00	0.63
S10										0.00

(a) Describing a set of ten sequences using a distance/dissimilarity matrix



(b) The phylogenetic tree generated from the matrix in Figure 2.9(a)

(c) Using the phylogenetic tree in Figure 2.9(b) to construct the progressive alignment

Figure 2.9: The underlying concept of MUSCLE: the progressive alignment

The similarity of ten sequences is shown as a distance matrix in Figure 2.9(a). These dissimilarity scores can be used to generate a phylogenetic tree in Figure 2.9(b), which here is constructed using the `kitsch` method of the PHYLIP (`http://evolution.genetics.washington.edu/phylip.html`) package. An alignment is made as each internal node of the tree is met when traversing from the leaf to the root node. Where two MSAs are required to be aligned, profile-profile alignment methods are used. A representative progressive alignment for the ten sequence example is shown in Figure 2.9(c), where progress is made from top-left to bottom-right and profile-profile alignment is indicated with a blue border while a red border represents pairwise alignment steps.

It is described as a *progressive* alignment because the process of merging pairwise or profile-profile alignments progresses from the leaf nodes to the root node of the tree. The process is shown in Figure 2.9. Here, an example set of ten sequences is represented by a distance matrix as shown in Figure 2.9(a) where the *dis*similarity of two sequences is scored between 0 and 1. This matrix can be used to construct a phylogenetic tree. The corresponding rooted tree for the dissimilarity scores of Figure 2.9(a) is shown in Figure 2.9(b) (this tree was constructed using the Fitch-Margoliash (1967)-based `kitsch` method from the PHYLIP[13] package of phylogenetic software).

A phylogenetic tree represents the hypothesized evolutionary relationship between the particular elements (sequences or species) being considered: at each branching of the tree, a differentiating evolutionary mechanism is hypothesized to have occured. As such, a phylogenetic tree can be used to define the MSA: where leaf nodes are individual sequences, a joining internal node represents their pairwise alignment.

Leaf nodes are aligned first. These pairwise alignments are then aligned using profile-profile sequence alignment methods, until the entire MSA has been constructed. In Figure 2.9(c) the alignment construction progresses from top-left to bottom-right. Simple pairwise alignments are bordered with red, while profile-profile methods are bordered with blue.

The method of constructing a progressive alignment is employed at each stage of the MUSCLE algorithm. These three stages are described briefly below; for full details, see Edgar (2004a).

The drafting stage To begin optimisation, an initial draft alignment is constructed using k-mer (specifically 6-mer) counting. k-mer counting generates a similarity score for each pair of sequences based on the prevalence of subsequences, or words, of length k. A phylogenetic tree P is constructed using the unweighted pair group method with arithmetic mean (UPGMA) (Sneath and Sokal, 1973).

The improvement stage The goal of this stage is to define and fix the phylogenetic tree to allow for further refinement of the MSA. A Kimura (1983) distance matrix using fractional identities is calculated from the mutual alignment of each pair of sequences in the context of the existing multiple alignment. UPGMA is used to generate a new phylogenetic tree, P', which is compared to P. Where the branching order of internal nodes has changed, new mutual alignments are made, and a new progressive alignment is constructed. This

[13] `http://evolution.genetics.washington.edu/phylip.html`

step can be iterated and completes when the set of changed internal nodes is empty. The phylogenetic relationship between the sequences is now fixed and the MSA can be refined.

The refinement stage The fixed tree is then subject to bi-partitioning (Hirosawa *et al.*, 1995) to generate pairs of profiles which are then realigned. Improved MSAs are identified by comparing the existing MSA to the new, realigned MSA using the sum-of-pairs metric (the average pairwise alignment score of every pair of sequences in the alignment).

Figure 2.10 shows a MUSCLE alignment of P53 proteins at each stage of iteration (no improvement is made after the fifth iteration). Figure 2.10 demonstrate that the first draft alignment does successfully align the highly conserved section of sequence internal to the protein, but performs poorly on the more sparsely populated start and end regions; alignment in these regions appears to improve with each iteration. Most clearly, the optimisation procedure eliminates many unecessary gaps present in the draft alignment (first iteration, Figure 2.10(a)).

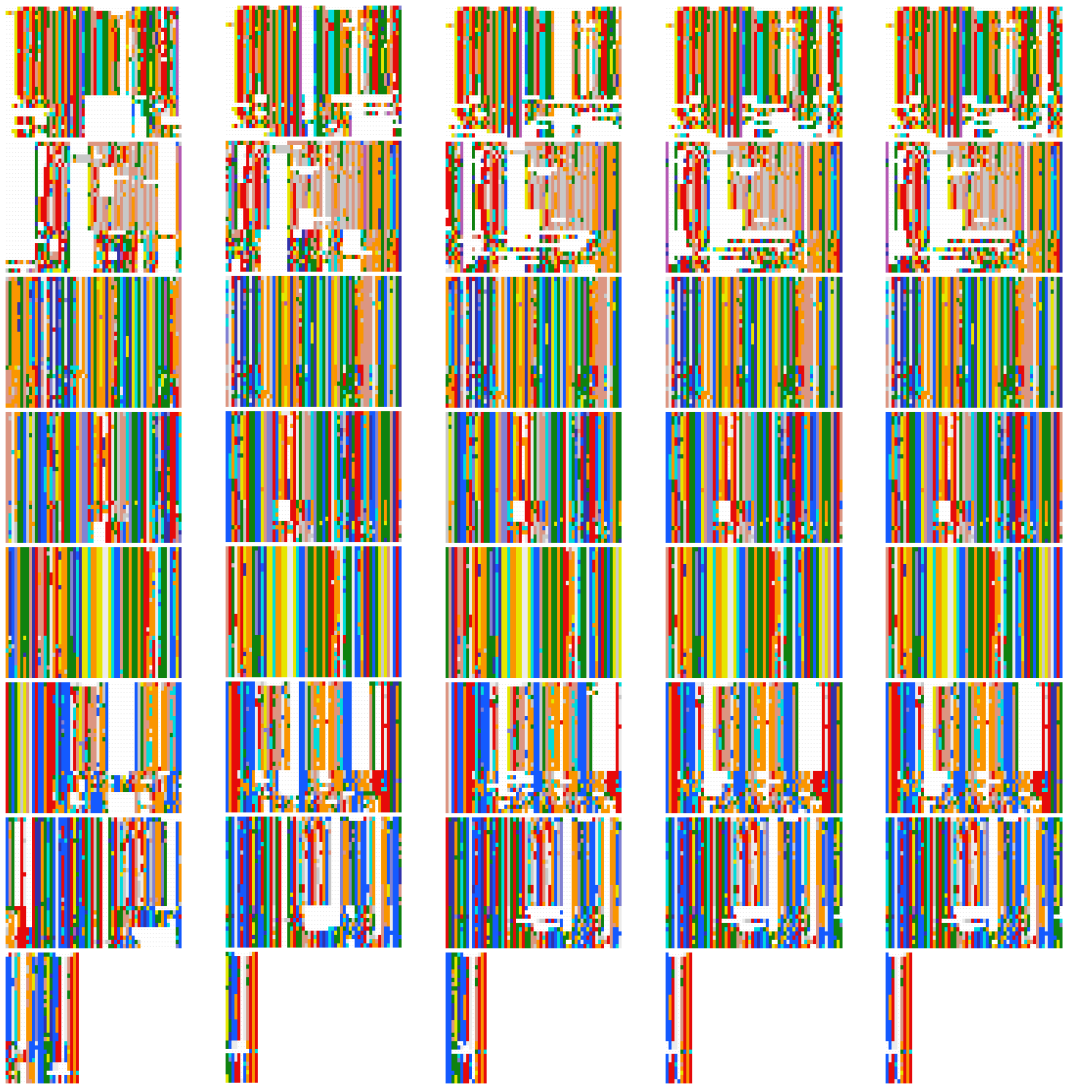
2.3.3 Needleman & Wunsch

The Needleman and Wunsch (1970) algorithm is a method for globally aligning two sequences. It employs dynamic programming to identify the optimal global alignment between two sequences. Dynamic programming is an algorithm by which the optimal procedure of decisions can be deduced by scoring all possible decisions at each step.

The method for aligning two sequences X and Y proceeds as follows (as described in Taylor and Orengo (1989)):

1. Initialise matrix by plotting X against Y
2. Populate matrix with alignment scores for each X/Y residue pair
3. Propagate scores through the matrix from bottom-right to top-left
4. Trace back the final alignment

In Step 1, X and Y are plotted against each other to derive an m by n matrix where every possible residue-pair alignment is represented and m and n are the lengths of the two sequences. The



(a) Iteration (1) (b) Iteration (2) (c) Iteration (3) (d) Iteration (4) (e) Iteration (5)

Figure 2.10: Aligning P53 proteins using MUSCLE

MUSCLE converges on an alignment after five iterations of the algorithm. The alignment from each iteration is shown continuing down the page in each column above, from Figure 2.10(a)-2.10(e). Residues are coloured as shown in Appendix [B.i], gaps are represented with white space. Significant alignment change can be seen towards the more sparsely populated ends of the alignment. The optimisation procedure eliminates many unecessary gaps in the first iteration (compare the length of the alignment in Figure 2.10(a) to that of the alignment in Figure 2.10(e)).

first sequence defines the rows and the second sequence defines the columns. Each cell is initialised with a score describing how well the two corresponding amino acids match each other. Various scoring systems exist. Most simply, identical residues score 1 and all non identical residues score 0; more representative scoring schemes, such as the amino acid scoring matrices described in Section 2.3.4, can be used to represent the frequency with which pairs of residues replace each other (from this, it is possible to summize that they have similar physicochemical properties).

Step 2 propagates scores from the bottom right hand corner of the matrix to generate scores that will be used in the final step to define the alignment. Matrix population proceeds from bottom-to-top row-wise and right-to-left column-wise, simultaneously. That is, row n and column n are completed before row $n-1$ and column $n-1$.

When considering each possible X/Y residue-pair alignment, three operations are possible:

1. the two residues are aligned
2. a gap is inserted in the first sequence
3. a gap is inserted in the second sequence

To represent this choice in the matrix, the score for the current cell in the matrix can be inherited from the diagonal (when the residues are aligned, operation 1) or from the off-diagonal (when gaps are inserted, operations 2-3). Each operation is scored and the highest score for each cell is entered in the matrix. The score for cell (i,j) can be formalised as follows:

$$D_{i,j} = s(i,j) + \max \begin{cases} D_{i+m,j+1} & -\delta \\ D_{i+1,j+m} & -\delta \\ D_{i+1,j+1} & \end{cases} \tag{2.1}$$

where D is the dynamic programming matrix; $D_{i+1,j+i}$ is the score of a diagonal move from cell $i+1, j+1$; $D_{i+m,j+1}$ defines a score from the j_1^{th} row and $D_{i+1,j+m}$ defines a score from the $i+1^{th}$ column. Inheriting from the j_1^{th} row or $i+1^{th}$ column requires that a gap be inserted in the appropriate sequence. Gaps are penalised using the penalty term δ, which is calculated as follows:

$$\delta = g_i + ng_e \tag{2.2}$$

where g_i is the gap initialisation penalty, g_e is the gap extension penalty and n is the length of the gap being inserted. Appropriate values for g_i and g_e are critical in obtaining reasonable alignments. Throughout this thesis, $g_i = 10$ and $g_e = 2$.

Once the scores have been propagated through the matrix, the highest scoring path is traced back from top-left to bottom right through the matrix to generate the optimal alignment of the two sequences.

2.3.4 Amino acid substitution matrices

Amino acid (AA) substitution matrices describe how similar a pair of residues are to each other. Matrices vary in the assumed mutation rate and scoring range. In general however, the approach is the same. Representative proteins are aligned and mutation rates (i.e., how often residue X is aligned with residue Y) for each pair of amino acids are recorded. These are the *observed* mutation rates between pairs of amino acids. The *expected* mutation rate is calculated using the frequency of the amino acids. A final log-odds value (S_{ij}) is calculated as follows:

$$S_{ij} = \log\left(\frac{obs_{ij}}{exp_{ij}}\right) \tag{2.3}$$

where obs_{ij} is the observed mutation rate between residue i and j and exp_{ij} is the expected mutation rate between residue i and j.

In this thesis, three matrices are used (i) to score conservation (see Chapter 4) and (ii) to characterise PDs and SNPs (see Chapter 7).

2.3.4.1 PAM30

PAM matrices (Dayhoff *et al.*, 1978) use the PAM (Percent Accepted Mutation) as a unit of sequence divergence: if two protein sequences are 1 PAM apart, they share 99% of their amino acids. Various PAMX matrices are available, where X describes the PAM distance between the protein sequences that are aligned to derive the observed and expected mutation rates. All PAM matrices are constructed by successive multiplications of the basic PAM1 matrix with itself; for example, the PAM30 $=$ PAM1^{30} (the PAM30 matrix is used in Chapter 7 and given in Appendix [C.i]).

The PAM1 matrix gives an estimate of the probability of residue b replacing residue a over a period of time t. These conditional probabilites are calculated from the phylogenetic analysis of evolutionarily related sequences, all with 85% or better sequence identity to each other. Using these observed probabilities, defining the unit of time as 1PAM and scaling the matrix such that rows and columns equal to one, the PAM1 matrix is derived (Durbin *et al.*, 1998).

2.3.4.2 BLOSUM62

BLOSUM matrices (Henikoff and Henikoff, 1992) were an attempt to update the older PAM matrices by exploiting the much increased wealth of protein sequence data and facilitate identification of very distantly related proteins. BLOSUM matrices use the BLOCKS resource (Henikoff *et al.*, 1999; Henikoff *et al.*, 2000) (since integrated into InterPro (Mulder *et al.*, 2007)) to generate multiple alignments and calculate observed and expected scores from which log-odds values are derived. Where PAM matrices use the basic unit of 1 PAM to construct matrices, BLOSUM matrices are based on alignments of proteins of varying levels of sequence identity. For example, the commonly used BLOSUM62 matrix (see Appendix [C.iii]) is derived from alignments of sequences that are $\geq$62% identical.

2.3.4.3 PET91

In 1991, the PAM250 matrix was updated by Jones *et al.* (1992) to create the PET91 (Pairwise Exchange Table 1991) matrix. The method used to derive the PET91 matrix is virtually identical to that of the PAM250 matrix, save for the construction of the raw PAM matrix: Dayhoff *et*

al. took the approach of inferring the common ancestral sequences and comparing this with the observed present-day sequences, whereas Jones *et al.* use the pairwise distances between present-day sequences to construct the PAM matrix.

Throughout this thesis, where the PET91 matrix is used, it has been normalised such that all values on the diagonal (i.e., the residue identity scores) are maximal and equal, as follows:

$$M(a,b) = \frac{m(a,b) - \min(m)}{\max(m) - \min(m)} \tag{2.4}$$

where m is the PET91 matrix; M is the normalised matrix, and a, b are amino acids. See Appendix [C.ii] for this normalised matrix.

2.3.5 Performance evaluation

Binary classification performance methods evaluate performance where the task is to assign each example to one of two classes, for example, present/absent, disease-causing/neutral or active/inactive. These three examples can all be generalised to positive/negative and the present discussion will continue using these generalised class names.

It is possible to assess the predictions made by the method being evaluated by comparing them with known answers, or with a gold standard dataset, which provides the best approximation to the correct answers that is available. In such a comparison, a binary classification method can make two kinds of errors: a type I error occurs when a result known to be negative is classified as positive and a type II error occurs when a results known to be positive is incorrectly classified as negative. A type I error is also known as a **false positive** (**FP**) and a type II error is also known as a **false negative** (**FN**); similarly, genuinely positive or negative results that are correctly classified are known as **true positives** (**TP**s) and **true negatives** (**TN**s) respectively. A successful prediction algorithm will minimise the number of incorrect results (i.e., FPs/FNs) and maximise the number of correct results (i.e., TPs/TNs). In this section and throughout this thesis, the four terms TP, TN, FP and FN will be used to describe the results of classification evaluation.

These four counts may be combined in many ways to assess how well the prediction method has performed. In this thesis, four measurements have been used to evaluate performance: sensitivity, specificity, positive predictive value (PPV) and Matthew's Correlation Coefficient (MCC). Their formulae for these statistics are shown below in Equations 2.5-2.8:

$$sensitivity = \frac{TP}{TP + FN} \tag{2.5}$$

$$specificity = \frac{TN}{FP + TN} \tag{2.6}$$

$$PPV = \frac{TP}{TP + FP} \tag{2.7}$$

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP + FP).(TP + FN).(TN + FP).(TN + FN)}} \tag{2.8}$$

The **sensitivity** is the fraction of results that are known to be positive that are correctly classified as positive by the prediction method; i.e., the number of positive examples that the method classifies correctly (Equation 2.5). A complementary measure is **specificity**, which describes the fraction of known negative examples that are correctly classified as negative by the algorithm (Equation 2.6). It is desirable to maximise both specificity and sensitivity such that $sensitivity - specificity = 0$. These measurements can be combined in a receiver operating curve or **ROC** plot which plots sensitivity (also called the **True Positive Rate** or **TPR**) against 1-specificity (also called the **False Positive Rate** or **FPR**); see Figure 2.11 for an example. If sensitivity=1-specificity (i.e., TPR=FPR), results are essentially random (indicated by a dashed blue line in Figure 2.11). As sensitivity increases and 1-specificity decreases (i.e., specificity *increases*), the results gravitate to the top-left of the ROC plot and results improve; a perfect result (indicated by a green circle in Figure 2.11) occurs when FPR = 0 and TPR = 1. It is possible to evaluate varying thresholds in a predictive model by plotting sensitivity against 1-specificity (FPR against TPR) and assessing which threshold maximises the sensitivity to 1-specificity ratio; i.e., which threshold is found furthest from the sensitivity=1-specificity line.

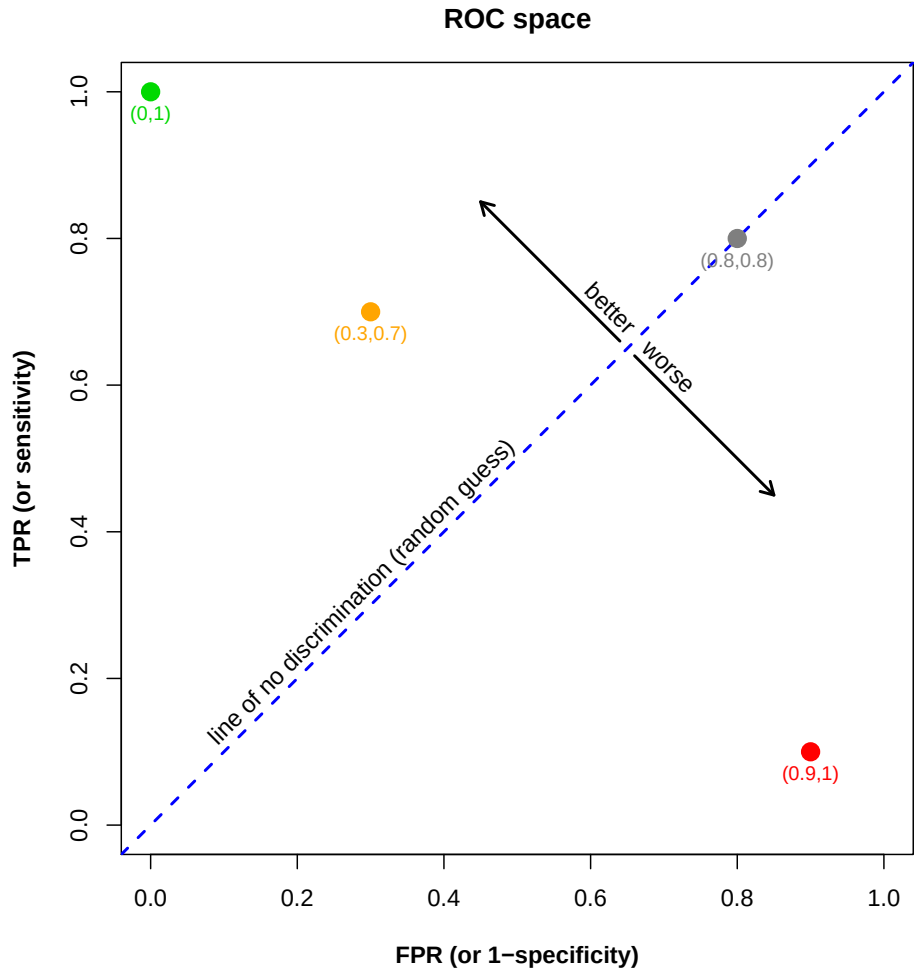


Figure 2.11: An example receiver-operating curve (ROC) plot

The true positive rate (TPR) is plotted against the false positive rate (FPR). Where TPR=FPR, results are no better than random (indicated with a dashed blue line); an example result is indicated with a grey circle. As the TPR increases and the FPR decreases, performance improves and results gravitate towards the top left hand corner of the plot, away from the line of no discrimination. Perfect prediction performance is achieved where TPR = 1 and FPR = 0; this is indicated with a green circle. Where the results drop below the line of no discrimination, towards the bottom right hand corner of the plot (as indicated by the red circle), the method more often predicts the incorrect result (i.e., predicts positive where the correct result is negative, and vice versa). The orange circle indicates a more average performance result.

An alternative measure is **positive predictive value** or **PPV**. This describes the fraction of those results that are *predicted* as being positive that are correct (Equation 2.7). This measure is particularly useful when the aim is to make conservative positive predictions at the expense of more false negatives.

Finally, the most comprehensive measurement of performance in a binary classification system is **Matthews Correlation Coefficient** or **MCC** (Equation 2.8). Where the previous measurements have considered a single 'dimension' of the performance, the MCC score incorporates all measures (TP, TN, FP and FN) into one value. The MCC can range from -1 to 1, where 1 indicates perfect performance, 0 indicates random performance and -1 indicates that performance is precisely the opposite of what is expected.

In addition to providing a comprehensive summary of evaluation performance, the MCC score is robust to class size inequality. This is not the case for the three other performance measures described above. For example, in computational biology it is often the case that the TN examples vastly outnumber the TP examples, and as such, high levels of specificity can be achieved simply by assigning everything to the negative class.

In reality, the appropriate performance statistics (the equations for which are summarised below) will depend on the aims of the method, on aspects of the data being examined and on an understanding of the gold standard dataset against which the method is being benchmarked. In Chapters 3 and 4, FOSTA and ImPACT are evaluated variously using sensitivity, specificity, ROC plots, PPV and MCC. The choice of the performance metrics is justified in each chapter.

2.3.6 Statistics and data representation

Chapter 7 analyses the data in SAAPdb and their hypothesised structural effects (see Chapters 5 and 6). In that chapter, several statistics are used to compare pathogenic deviations (PDs) and single nucleotide polymorphisms (SNPs) with respect to several sequence and structural features. In this section, the statistics used are summarised. Log ratios, as a means for *graphically* comparing datasets rather than statistically comparing datasets, are also described.

2.3.6.1 Log ratios

Log ratios compare the observed prevalence of a feature with the expected prevalence of a feature, as shown in Equation 2.9. Throughout this theses, unless otherwise stated, log ratios are calculated using $\log_2$. A value of 0 indicates that the observed and expected values are the same. A value of 1 would indicate that the observed value is double (2^1) what is expected, a value of 2 would indicate that the observed value is four times (2^2) what is expected, and so on. Similarly, a value of -1 would indicate that the observed value is half (2^{-1}) of what is expected.

$$logratio = \log_2 \left(\frac{observed}{expected} \right) \tag{2.9}$$

Log ratios are not a statistical test from which a p-value can be derived, but a way of representing the difference between an observed value and an expected value.

2.3.6.2 Kolmogorov-Smirnov

The Kolmogorov-Smirnov or KS test (Conover, 1971) is a non-parametric method for comparing distributions. In the one-sample test, an observed sample distribution is compared with a reference distribution (e.g., a normal distribution), while in the two-sample test, two observed sample distributions are compared (in Chapter 7, only the two-sample test has been used).

The null hypothesis of the KS test is that the distributions being compared are drawn from the same distribution. Therefore, should $p < \alpha$ (α is variously set at 0.05 or 0.01), the null hypothesis is rejected and it is concluded that the distributions were *not* drawn from the same distribution. The test statistic is the maximal vertical distance (D) between the two cumulative distribution functions (CDFs), that is:

$$D = \max |F(x) - G(x)| \tag{2.10}$$

where F and G are the test CDF and reference CDF respectively in the one-sample test, or the two test CDFs in the two-sample test. The CDF of the distribution function f is calculated as

follows:

$$F(x) = \int_{-\infty}^{x} f(t)dt \tag{2.11}$$

where x is the point at which the total is to be calculated. In this thesis, the CDFs are calculated from data, rather than from probability distributions; in this case, the *empirical* cumulative distribution of data x at point i is calculated as:

$$C_i(x) = \frac{1}{N} \sum_{j=1}^{j \leq i, i \leq D} x_j \tag{2.12}$$

where i, j are bins of data; D is the number of bins; x_j is the number of data points in x that belong in bin j; and N is the number of datapoints in x.

Where distributions are being compared in the presence of ties (i.e., there are many repeated values in the dataset), a bootstrapping method (n=1000) is carried out using the `ks.boot()` [14] method in `R`. This more accurately estimates the p-value when comparing discontinuous distributions (Abadie, 2002).

2.3.6.3 χ^2 test

Where data are nominal counts, the χ^2 test (Mood *et al.*, 1974) will indicate whether there is a difference between two datasets. Note that where χ^2 results are reported with percentages in this thesis, raw counts have been used to conduct the χ^2 test. The χ^2 statistic is calculated as shown in Equation 2.13. Expected values may not always be available; they can however be estimated using the observed data (Figure 2.12 shows an example). Where possible, known expected values have been used throughout this thesis, rather than estimated values; i.e., expected values are calculated from known data rather than being estimated from the observed data.

[14] `http://sekhon.berkeley.edu/matching/ks.boot.html`
`http://sekhon.berkeley.edu/papers/MatchingJSS.pdf`

	P		Q		Total	
Male	100	86.4	80	93.6	180	180
Female	20	33.6	50	36.4	70	70
Total	120	120	130	130	250	250

Figure 2.12: Calculating χ^2 expected values
Males and females have been classified as P and Q. Observed counts are in black, expected counts are in grey. Expected values can be calculated as follows: $e = (X * Y)/N$ where X is the row total, Y is the column total and N is the number of examples in the dataset.

$$\chi^2 = \sum_{i=1}^{k} \frac{(observed - expected)^2}{expected} \tag{2.13}$$

Throughout this thesis, the χ^2 test is Yates corrected where a χ^2 test is carried out on a 2x2 contingency table. This prevents low p-values being derived where there is only one degree of freedom by subtracting 0.5 from each observed value in the contingency table, as shown in Equation 2.14.

$$\chi^2_{Yates} = \sum_{i=1}^{k} \frac{(|observed - expected| - 0.5)^2}{expected} \tag{2.14}$$

2.3.6.4 Fisher exact test

The χ^2 test becomes unreliable where the contingency table is sparsely populated (i.e., where any cell has a value of ≤ 10) and where counts are distributed unevenly throughout the contingency table. However, the statistical theory upon which the Fisher exact test is based allows robust comparison of datasets of disparate sizes and is able to consider contingency tables with empty cells (Fisher, 1935). The probability of obtaining the set of values a, b, c, d as shown in the contingency table in Figure 2.13 can be calculated using the hypergeometric distribution as described in Equation 2.15.

	P	**Q**	
Male	a	b	a+b
Female	c	d	c+d
	a+c	b+d	n

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!} \tag{2.15}$$

Figure 2.13: Fisher exact test

Naming the values in the contingency table as a, b, c, d as shown, the probability of that particular combination of values can be calculated using the hypergeometric distribution as shown in Equation 2.15 where ! is the factorial operator.

Chapter 3

FOSTA: Functional Orthologues from Swiss-Prot Text Analysis

As introduced in Section 1.9, SAAPdb is a database of mutation data, which aims to identify the structural effect, if any, of single amino acid polymorphisms (SAAPs). It is an on-going project: the suite of structural analyses is expected to evolve to include a broader and more sophisticated range of structural analyses (the current structural analyses will be described in detail in Chapter 5). As such, while the pipeline is being developed, it may not always be possible to explain a mutation with respect to its structural annotation. However, it may be possible to *infer* functional relevance from sequence data. Residues that have been maintained across evolution, and therefore have been subject to selection pressure, are likely to be important to the native structure and/or function of the protein. The first step in identifying such residues is to construct an alignment of functionally equivalent proteins or FEPs. Proteins that have diverged in function (either by gaining or losing functionality) will show differences at key functional residues. To incorporate such an analysis in SAAPdb, a reliable, automatic method for extracting groups of FEPs is required.

This chapter describes a novel method for identifying FEPs by analysing functional annotations in UniProtKB/Swiss-Prot. This method and its evaluation have been published in McMillan and Martin (2008).

3.1 Introduction

To generate an informative multiple sequence alignment (MSA), the 'same' protein in different species should be aligned. In this chapter, the 'same' protein is defined as an orthologue that performs an equivalent function or functions. Proteins that have diverged in function (either by gaining or losing functionality) will show differences at key functional residues; aligning such proteins will obscure patterns of functionally-relevant conservation. Two entities are homologous if they have a common evolutionary origin. An *orthologous* relationship denotes that this common origin was a speciation event, whereas *paralogues* are related by a gene duplication (Koonin, 2005).

Consider the HOX family of genes: a large family of transcription factor proteins containing the well characterised homeobox motif. These proteins are well conserved across species and are believed to be critical in embryogenesis, oncogenesis and differentiation processes such as haematopoiesis (Yaron *et al.*, 2001; Lill *et al.*, 1995). HOX proteins are representative of large protein families in that there are several paralogues within a species—thirteen in the case of the human HOX family (Yaron *et al.*, 2001)—and each paralogue can be involved in several distinct aspects of the same biological process. A sequence alignment of such evolutionarily related, but functionally different, proteins (i.e., paralogues) would contain significant noise, and obscure much of the genuine functional conservation between true FEPs.

Paralogues, having been derived via a mechanism for functional divergence, are likely to perform different functions (Fitch, 2000). While orthologues generally perform the same function, it is possible for the function to diverge, particularly when orthologues are evolutionarily distant (Koonin, 2005). For example, Shibata *et al.* (2006) showed that although the general function of exportin-5 proteins (nuclear export of miRNAs and tRNAs) is conserved across different species, substrate specificity varies. Further, the *AGAMOUS* gene in Arabidopsis is involved in carpel and stamen development, but the two orthologues in maize have specialised: *ZAG1* is highly expressed during carpel development and *ZMM2* is expressed during stamen development (Wagner, 2002). It is clear then that orthology need not imply functional equivalence and it follows that sets of orthologues, defined by methods such as Inparanoid (O'Brien *et al.*, 2005), C/KOG (Tatusov *et al.*, 2001; Tatusov *et al.*, 2003) and TOGA (Lee *et al.*, 2002), are not appropriate as lists of FEPs. Further, these methods are computationally intensive and as such are often limited to small species sets.

While homology does not imply functional equivalence, it is also not possible to use functional data alone to identify FEPs. Proteins can converge on similar functions without being evolutionarily related. For example, subtilisin (EC 3.4.21.62) and trypsin (EC 3.4.21.4) have evolved separately in bacteria and vertebrates respectively. They differ significantly in protein sequence, structure and fold, yet the same three amino acids form the catalytic triad in both proteins (Akindahunsi and Chela-Flores, 2005). Aligning such functionally similar, but evolutionarily unrelated, proteins is meaningless: only proteins which are both homologous and functionally equivalent will generate an informative alignment.

The identification of true FEPs requires consideration of features such as functional assays, interaction networks, expression data and so forth. UniProtKB/Swiss-Prot is a carefully annotated databank of protein sequences that includes functional annotations (The UniProt Consortium, 2009). While many of these are transferred through orthology, where there is experimental evidence for function, it will be included. Thus, short of conclusive experimental studies, the most reliable way of identifying families of FEPs is first to identify families of homologues in UniProtKB/Swiss-Prot and then to examine the annotations to find a set of proteins that are annotated as performing the same function or functions. It is, of course, possible that annotations in UniProtKB/Swiss-Prot will be incorrect, but as UniProtKB/Swiss-Prot is updated on a regular basis, it is expected that these annotations will represent the most up-to-date knowledge of protein function and errors in annotations will be corrected with future releases.

While it is perfectly possible to perform this analysis on an individual basis by searching UniProtKB/Swiss-Prot for homologues and comparing the annotations manually, there is a pressing need for an automatically updated resource that simply lists families of FEPs in UniProtKB/Swiss-Prot. Several methods exist that exploit database annotations to identify related proteins (Artamonova *et al.*, 2005; Kretschmann *et al.*, 2001; Yu, 2004; Kunin and Ouzounis, 2005), however there has been no resource that very simply provides sets of FEPs annotated as having the same function in UniProtKB/Swiss-Prot in an easily-accessible format, with extensive coverage of multiple proteomes.

FOSTA (Functional Orthologues from Swiss-Prot Text Analysis) has been developed to automate the process that one would perform manually to extract a family of FEPs from UniProtKB/Swiss-Prot. It considers UniProtKB/Swiss-Prot proteins for inclusion in groups of FEPs (FOSTA families) rooted around human proteins. It refines an initial candidate list of homologues on the basis of functional annotation similarity to distinguish FEPs from

functionally diverged homologues (FDHs). To assess functional annotation similarity, FOSTA employs simple text-mining techniques to compare UniProtKB/Swiss-Prot description fields.

3.2 Method

3.2.1 Obtaining the data

Figure 3.1 describes the flow of data in the FOSTA system. FOSTA exploits data in two UniProtKB/Swiss-Prot files in forming families of FEPs: the FASTA formatted version of UniProtKB/Swiss-Prot sequences and the UniProtKB/Swiss-Prot `.dat` flatfile, from which the functional annotations are extracted. These files are automatically mirrored from Expasy (`ftp.expasy.org/databases/uniprot/` and `ftp.expasy.org/databases/swiss-prot/` specifically). The first step in populating FOSTA is to clone the most recent relevant UniProtKB/Swiss-Prot data and extract the desired information from them. All FOSTA analyses (one for each human protein) are then distributed across a local compute farm (using Sun GridEngine), with each individual process updating the FOSTA database. All data required by the distributed processes are available in the FOSTA database.

3.2.2 The FOSTA method

As input, FOSTA takes an entire UniProtKB/Swiss-Prot release; results presented in this chapter are based on UniProtKB/Swiss-Prot v53.0. FOSTA roots families of FEPs (FOSTA families) around human proteins of length $\geq$100 using the three stage filtering processes shown in Figure 3.2. Candidates rejected at filtering stages (2) and (3) are retained and recorded as functionally diverged homologues (FDHs).

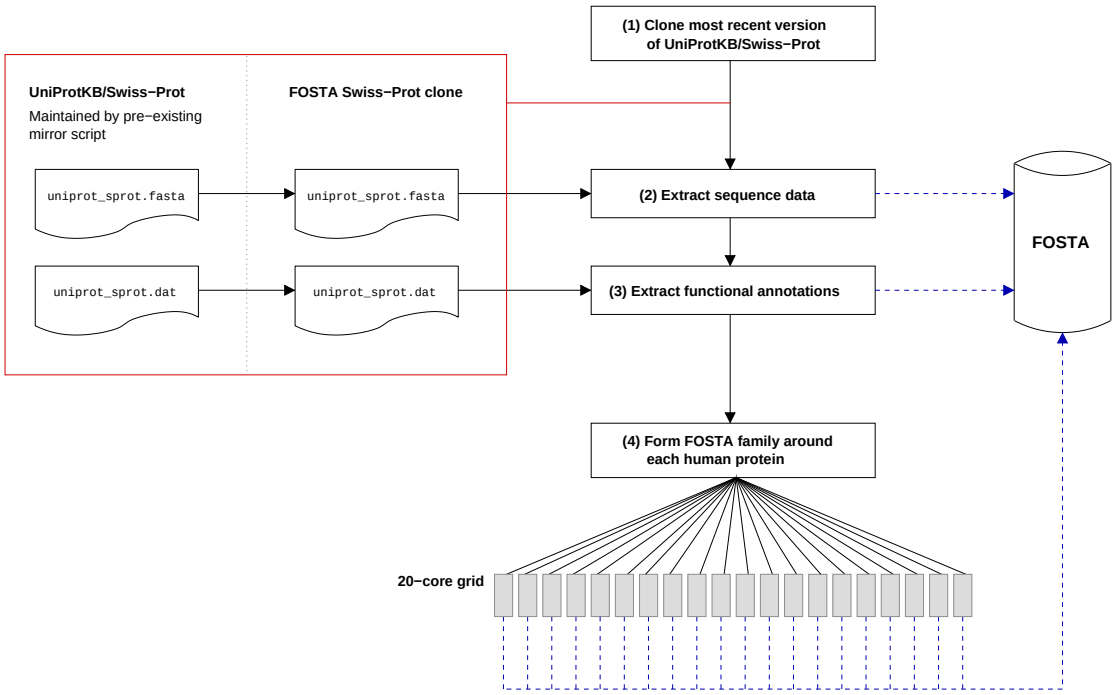


Figure 3.1: The workflow of FOSTA

STEP (1): the `uniprot_sprot.fasta` and `uniprot_sprot.dat` files from the most recent version of UniProtKB/Swiss-Prot are cloned (this process is highlighted in red); **STEP (2)**: sequence data are extracted from `uniprot_sprot.fasta` and stored in the FOSTA database; **STEP (3)**: functional annotations are extracted from `uniprot_sprot.dat` and stored in the FOSTA database; **STEP (4)**: FEPs for each human protein are identified (for details, see Figure 3.2 and Sections 3.2.2.1-3.2.2.3). Solid black lines indicate the direction of data flow, dashed blue lines indicate where data are stored in the FOSTA database. A pre-exising mirror script ensures that the external UniProtKB/Swiss-Prot data (separated from the cloned FOSTA data with a dashed grey line) are kept up to date. The analysis for each human protein is distributed across the local 20-core grid (each core is represented above with a grey rectangle). Note that FOSTA is updated by each inidividual process.

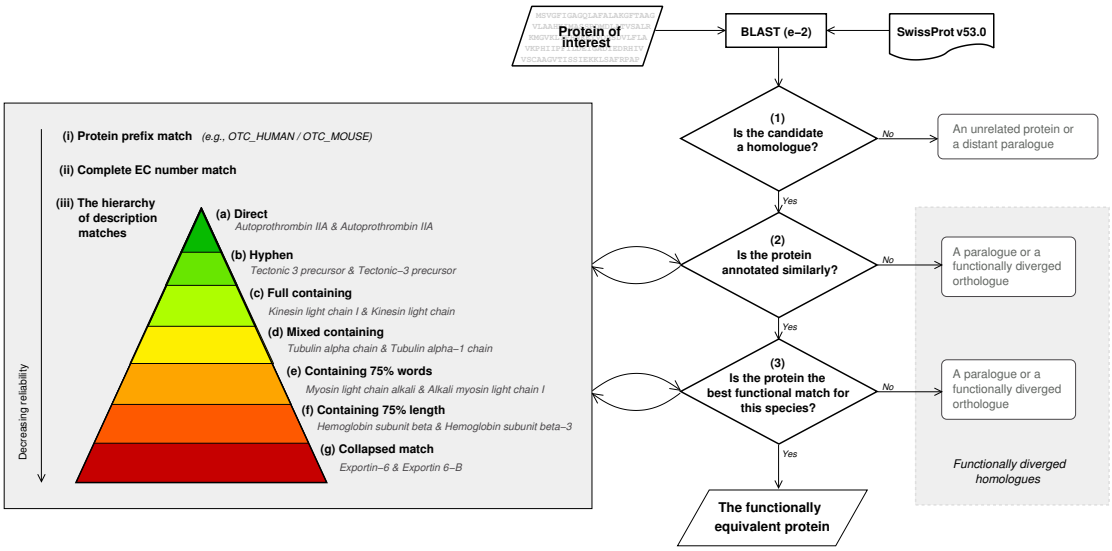


Figure 3.2: A schematic of the FOSTA method

The protein of interest is BLASTed against UniProtKB/Swiss-Prot (in the current version of FOSTA, this is UniProtKB/Swiss-Prot v53.0). **STEP (1)** any BLAST matches ($e \leq 10^{-2}$) are retained, non matches are unrelated proteins or distant paralogues; **STEP (2)** the annotations of each BLAST match are compared to those of the root protein of interest, matches are retained, non-matches are described as paralogues or functionally diverged homologues; **STEP (3)** the best match for each species is identified and described as the functionally equivalent protein, the rest are identified as paralogues or functionally diverged homologues. The inset box on the left hand side describes the functional annotation match hierarchy. See Section 3.2.2.2 for detailed discussion.

3.2.2.1 The sequence filter

The first stage identifies sequence homologues using a BLAST (Altschul *et al.*, 1990) e-value threshold of 10^{-2}. In stage (2) of Figure 3.2, this list of candidate FEPs is refined using the filters described in Sections 3.2.2.2 and 3.2.2.3.

3.2.2.2 The functional filter

This stage, shown in the functional match 'pyramid' in Figure 3.2, aims to 'read' the UniProtKB/Swiss-Prot annotations. The homologues obtained in the previous stage are filtered on function using information from the UniProtKB/Swiss-Prot 'Description' (DE) field and the UniProtKB/Swiss-Prot ID itself. Each homologue identified by the BLAST search will survive the functional filter if it matches the root protein in at least one of three levels (I-III); the DE field text matches compare synonyms at seven further levels of specificity (a-g):

(I) by the protein prefix element of the UniProtKB/Swiss-Prot ID

(II) by an EC number

(III) by matching synonyms at further multiple levels of specificity from the DE field

- (a) a **'direct' match**, where the two proteins share an intact synonym (e.g., *Autoprothrombin IIa* and *Autoprothrombin IIa*)
- (b) a **'hyphen' match**, where the proteins share a synonym after hyphen placement is mirrored across both strings (e.g., *Tectonic 3 precursor* and *Tectonic-3 precursor*)
- (c) a **'full containing' match**, where one synonym is completely contained within another (e.g., *Kinesin light chain I* and *Kinesin light chain*)
- (d) a **'mixed containing' match**, where one synonym is contained within another synonym, but the words can be in a different order or they may be interrupted by additional annotation (e.g., *Tubulin alpha chain* and *Tubulin alpha-1 chain*)
- (e) a **'containing 75% words' match**, where 75% of the words of the shorter synonym are also in the longer synonym (e.g., *Myosin light chain alkali* and *Alkali myosin light chain I*)

(f) a **'containing 75% length' match**, where 75% of the words in terms of *length* of the shorter synonym are also in the longer synonym (e.g., *Hemoglobin subunit beta* and *Hemoglobic subunit beta-3*)

(g) a **'collapsed' match**, where one synonym is a substring of another, after spaces and punctuation have been removed (e.g., *Exportin-6* and *Exportin 6-B*)

When any two synonyms are compared, three common, functionally irrelevant words—'protein', 'fragment' and 'precursor'—are removed so as to avoid matches on functionally irrelevant terms. All text comparisons are case insensitive.

The level (I) protein prefix match is considered the most reliable functional match (given that all candidates have survived the homology screen and are therefore known to be evolutionarily related) and the level (III) description match is considered to be the least reliable functional match. Within the description field match, reliability reduces from (a) the direct match to (g) the collapsed match. Although the choice of the 75% threshold is somewhat arbitrary, it is unlikely that false matches will be made, as all candidates have already been screened for homology. In text comparison (g), the smaller synonym that is contained in the other must be at least four characters long. The string 'inhibit' is treated as a special case: in synonym comparisons (c)-(g), both or neither of the synonyms can contain the string 'inhibit' for a match to be possible; for example, there is no match if one of the synonyms contains the word 'inhibitor' while the other does not.

3.2.2.3 The FEP filter

If a protein survives both the sequence and functional filtering stages, it is either the FEP for that species or a homologue that has undergone some (small) degree of functional divergence. To eliminate the functionally diverged homologues (FDHs), only the best functional match from each species (as defined by the functional match reliability hierarchy described in Section 3.2.2.2 and in the match hierarchy pyramid shown in Figure 3.2) is assigned to the FOSTA family. If two or more proteins cannot be discriminated *functionally* (i.e., they match the root human protein at the same level of specificity), the protein with the highest sequence identity is chosen. Note that sequence identity is used only as a last resort as highest sequence identity does not guarantee functional equivalence even amongst close homologues (Notebaart *et al.*, 2005; Koski and Golding, 2001).

3.2.2.4 Unreliable proteins

UniProtKB/Swiss-Prot employs non-experimental qualifiers to describe proteins that have not been characterised directly using experimental procedures[1]: 'probable' (there is some experimental evidence, perhaps from a close homologue), 'putative' (there is some evidence, but not enough to describe the function as 'probable') and 'hypothetical' (the sequence is automatically translated from known genes). In addition, some proteins are not complete (fragments) and others are described as 'homologues' or contain the '-like' string. Further, as described in Section 3.2.2.2, the FOSTA methodology uses sequence identity as the last resort match should two candidate FEPs be indistinguishable with respect to function.

Such proteins and FEP assignments are less 'reliable' than others. FOSTA marks these assignments as such, so that the user may choose to remove them from the FOSTA family.

3.3 Results and Discussion

3.3.1 An overview of FOSTA

Before presenting the evaluation of the method, it is helpful to present an overview of the dataset. In this section, FOSTA is described with respect to proteome coverage and family size.

To appreciate the coverage of FOSTA, the UniProtKB/Swiss-Prot proteome coverage for each species has been calculated as N_F/N_{SP}, where N_{SP} is the number of proteins from that species that are described in UniProtKB/Swiss-Prot (i.e., the size of the 'UniProtKB/Swiss-Prot proteome') and N_F is the number of proteins from that species described in FOSTA. Therefore, a species that has all of its UniProtKB/Swiss-Prot proteins assigned to a FOSTA family will have a UniProtKB/Swiss-Prot proteome coverage of 100%, while a species with none of its UniProtKB/Swiss-Prot proteins represented in FOSTA would have a UniProtKB/Swiss-Prot proteome coverage of 0%.

It is clear from Figure 3.3 that many species have 2% or less of their UniProtKB/Swiss-Prot pro-

[1] See `http://www.uniprot.org/manual/non_experimental_qualifiers` and `http://www.uniprot.org/docs/annbioch`

teomes represented and a sizable number of species that have >98% of their UniProtKB/Swiss-Prot proteome described in FOSTA. The high number of poorly represented species will reduce over time as annotations are resolved across species.

Figure 3.4(a) shows the distribution of family sizes for all FOSTA families of 660 members or less (only one human protein is assigned more than 660 FEPs: FOSTA identifies 1787 FEPs for human Cytochrome b [UniProtKB:P00156/CYB_HUMAN]). FOSTA families with a membership of <100 are shown in more detail in Figure 3.4(b). The most common family size is two, which usually corresponds to an exclusively human/murine FOSTA family. These are not only the most well represented species in UniProtKB/Swiss-Prot v53.0, they are also the most extensively and similarly annotated.

There are increasingly fewer FOSTA families as family size increases beyond two. This is in part due to annotations being less consistent across species, however it is also owing to genuine functional divergence: as more species are added to a species set S, it is less likely that the function F is common to every species in S.

With respect to how FEP relationships are formed, most FOSTA families are formed exclusively using the protein prefix match, i.e., all members share the same protein prefix. However, 42.10% (6 266/14 884) of FOSTA families contain at least two different protein prefixes. Furthermore, of the 22 871 protein prefixes recorded in FOSTA, 5.42% are found to exist in more than one FOSTA family. This indicates that, although UniProtKB/Swiss-Prot protein prefixes are very often reliable, incorporating additional information derived from the description field is beneficial in identifying FEP relationships.

3.3.2 Difficulties in benchmarking

Evaluating FOSTA is difficult because no gold-standard dataset exists. In addition, it is difficult to design an evaluation procedure to isolate the performance of the FOSTA method from the quality of the UniProtKB/Swiss-Prot annotations that FOSTA interprets. To assess the FOSTA *method*, it is necessary to assess whether FOSTA is clustering proteins correctly given the functional annotations, rather than assessing whether the functional annotations are of sufficient detail to infer genuine functional equivalence. However, it is also very important to assess the latter, as FOSTA is dependent on the UniProtKB/Swiss-Prot annotations.

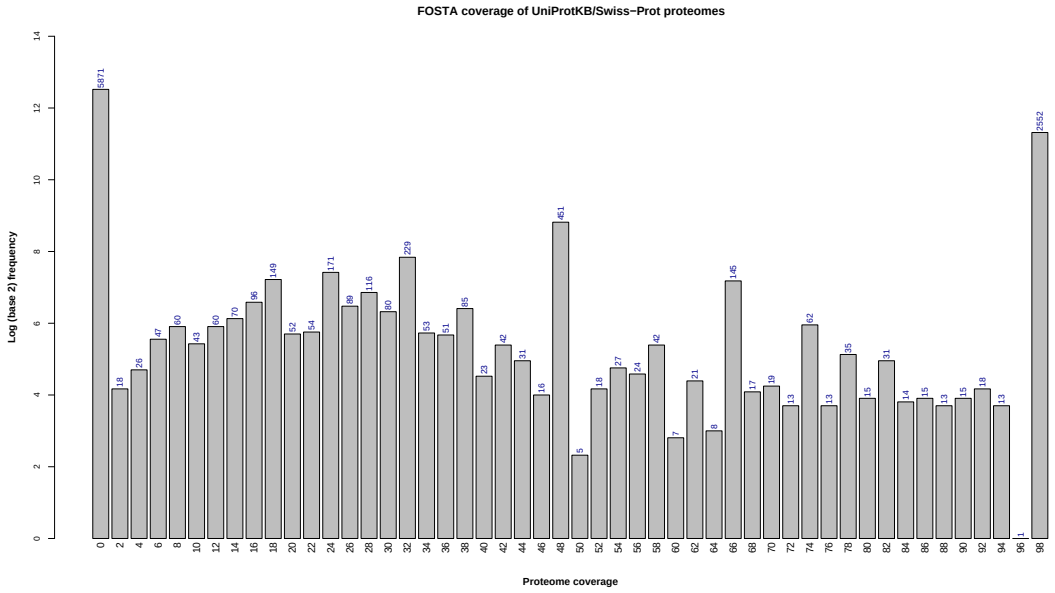


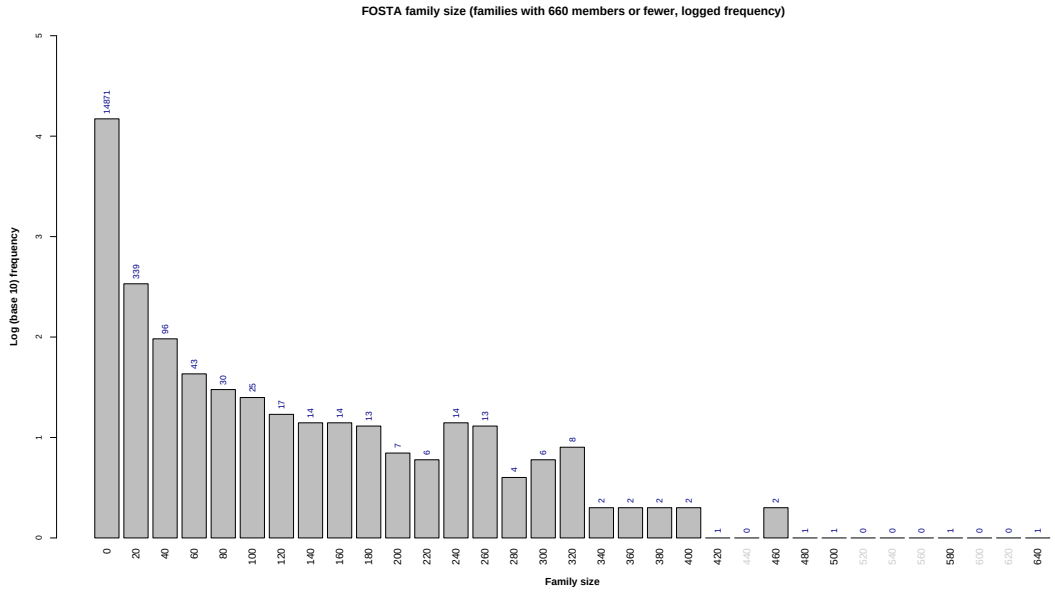
Figure 3.3: Proteome coverage in FOSTA

The proteome coverage for each species represented in FOSTA, calculated as $N_{X,F}/N_{X,SP}$, where $N_{X,SP}$ is the number of UniProtKB/Swiss-Prot proteins for species X and $N_{X,F}$ is the number of proteins from species X described in FOSTA. Frequency is logged (base 2). Blue text gives the raw, unlogged frequency.

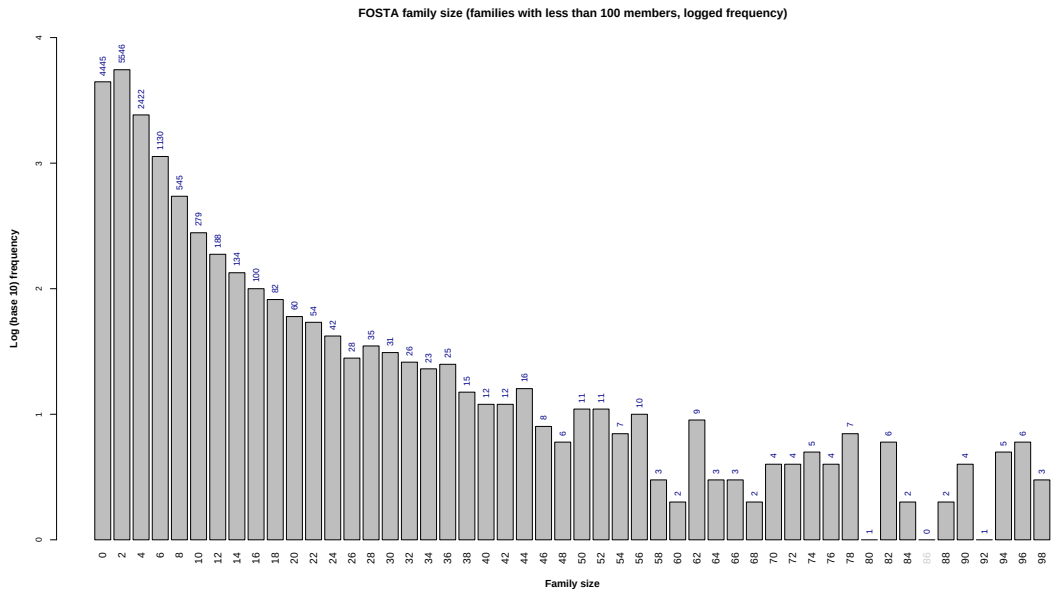
As such, FOSTA has been evaluated in three phases. The first involves manual interpretation of the results of several large protein families, some chosen at random and some chosen as known problematic cases. This phase assesses how well FOSTA can interpret functional annotations and infer functional equivalence compared with manual interpretation. The second phase benchmarks FOSTA against a fully manually annotated dataset and a larger partially annotated dataset. This phase not only indicates whether FOSTA performs well, but also assesses whether the annotations are good enough to infer functional equivalence. The final phase of evaluation involves comparing FOSTA to Inparanoid, a popular method for identifying orthologues. Note that the aim of FOSTA is not the same as that of the datasets used in the second and third evaluation phases. Nevertheless, some interesting comparisons can be made.

3.3.3 HOX proteins

The family of homeobox (HOX) proteins was introduced in Section 3.1. In this section, the performance of FOSTA is assessed when assigning the zebrafish (*Danio rerio*) FEP to *Homo Sapi-*



(a) FOSTA family size (families with 660 members or fewer)



(b) FOSTA family size (families with less than 100 members)

Figure 3.4: The distribution of FOSTA family size

To characterise the FOSTA dataset, the distribution of family sizes (i.e., the number of FEPs) is shown above. Family size is logged (base 10). Blue text gives the raw, unlogged count. Family size bins without data (having a raw count of 0) are indicated in grey. The plot in Figure 3.4(a) shows results for those families with 660 or fewer members; only one FOSTA family (that of CYB_HUMAN) has more than 660 members (it has 1787).

Table 3.1: Zebrafish candidates for the FOSTA family of HXB7_HUMAN
Protein: The UniProtKB/Swiss-Prot ID; **ID**: The sequence identity of the **Protein** to HXB7_HUMAN; **Description**: The UniProtKB/Swiss-Prot description (DE) field.

Protein	ID	Description
HXB7_HUMAN	100	Homeobox protein Hox-B7; Hox-2C; HHO.C1
HXB7A_DANRE	54	Homeobox protein Hox-B7a; Hox-B7
HXA1A_DANRE	63	Homeobox protein Hox-A1a; Hox-A1
HXA3A_DANRE	68	Homeobox protein Hox-A3a
HXA4A_DANRE	65	Homeobox protein Hox-A4a; Zf-26; Hoxx4
HXA5A_DANRE	75	Homeobox protein Hox-A5a
HXA9B_DANRE	62	Homeobox protein Hox-A9b
HXB1A_DANRE	64	Homeobox protein Hox-B1a; Hox-B1
HXB1B_DANRE	64	Homeobox protein Hox-B1b; Hox-A1
HXB2A_DANRE	57	Homeobox protein Hox-B2a; Hox-B2
HXB3A_DANRE	67	Homeobox protein Hox-B3a; Hox-B3
HXB4A_DANRE	62	Homeobox protein Hox-B4a; Hox-B4; Zf-13
HXB5A_DANRE	75	Homeobox protein Hox-B5a; Hox-B5; Zf-21
HXB5B_DANRE	75	Homeobox protein Hox-B5b; Hox-B5-like; Zf-54
HXB6A_DANRE	78	Homeobox protein Hox-B6a; Hox-B6; Zf-22
HXB6B_DANRE	75	Homeobox protein Hox-B6b; Hox-A7
HXB8B_DANRE	60	Homeobox protein Hox-B8b; Hox-A8
HXC1A_DANRE	62	Homeobox protein Hox-C1a
HXC3A_DANRE	61	Homeobox protein Hox-C3a; Hox-114; Zf-114
HXC5A_DANRE	72	Homeobox protein Hox-C5a; Hox-C5; Hox-3.4; Zf-25
HXC6A_DANRE	63	Homeobox protein Hox-C6a; Hox-C6; Zf-61
HXC6B_DANRE	77	Homeobox protein Hox-C6b
HXC8A_DANRE	73	Homeobox protein Hox-C8a
HXD4A_DANRE	62	Homeobox protein Hox-D4a; Hox-D4
HXD9A_DANRE	65	Homeobox protein Hox-D9a; Hox-D9
HXDAA_DANRE	61	Homeobox protein Hox-D10a; Hox-D10; Hox-C10

ens homeobox protein Hox-B7. There is a body of literature on the problem of elucidating HOX gene evolution, which is difficult in zebrafish given the extensive polyploidy in its evolutionary history (Amores *et al.*, 1998; Meyer, 1998; Stellwag, 1999).

The BLAST search identifies 83 zebrafish candidate FEPs and the filtering process assigns HXB7A_DANRE to the FOSTA family of HXB7_HUMAN. There are 24 zebrafish FDHs that have higher sequence similarity to HXB7_HUMAN than the assigned FEP. These proteins, the FEP and the root human protein are listed in Table 3.1, along with their UniProtKB/Swiss-Prot annotations and their sequence identity to HXB7_HUMAN. It is clear that HXB7A_DANRE is the FEP given the similarity of its description and its protein prefix to that of HXB7_HUMAN; this would be selected in a manual analysis of these candidates, despite its lower sequence identity.

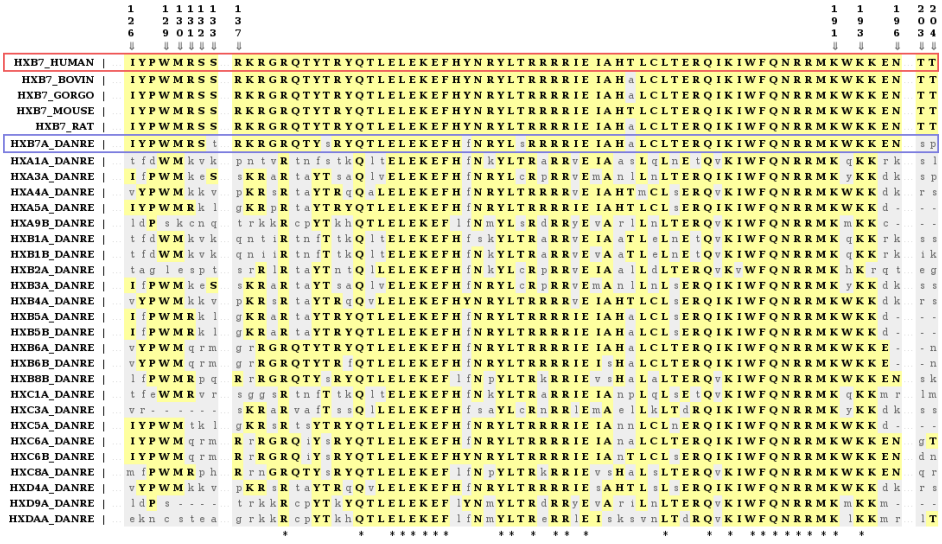
Several sites of functional relevance have been identified for HXB7_HUMAN (Table 3.2).

Table 3.2: Functional sites in HXB7_HUMAN
Functional site: a description of the functional site; **Location**: the residue number in HXB7_HUMAN; **Reference**: The source of the annotation.

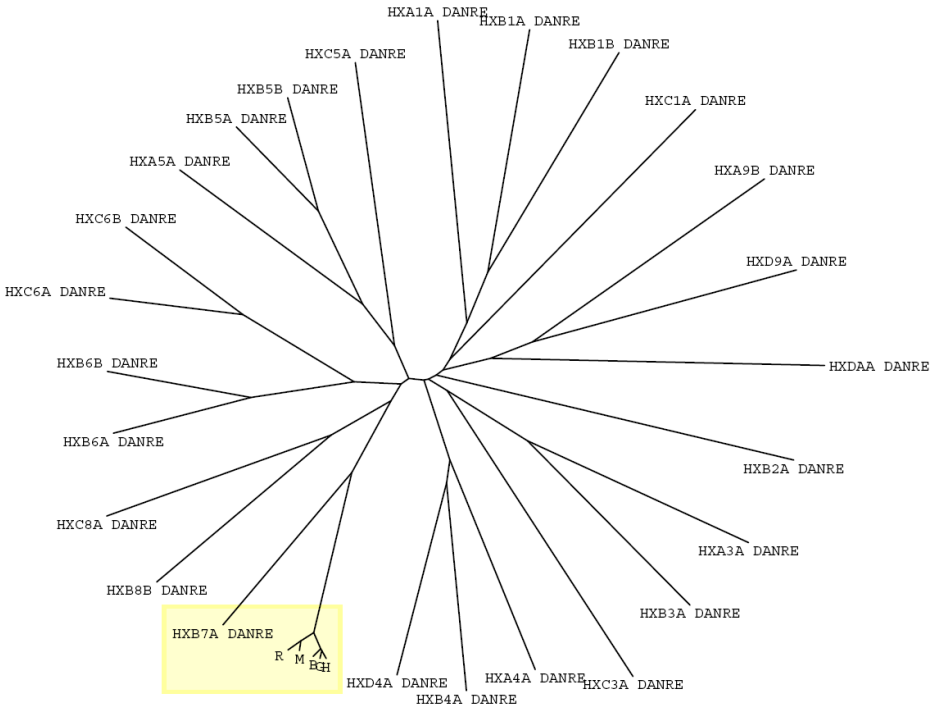
Functional site	Location	Reference
DNA binding (homeobox)	137 - 197	UniProtKB/Swiss-Prot/FT DNA_BIND
Crosslink (glycyl lysine isopeptide)	191 & 193	UniProtKB/Swiss-Prot/FT CROSSLNK
Motif (Antp-type hexapeptide)	126 - 131	UniProtKB/Swiss-Prot/FT MOTIF
Hypothesized binding to PBX	129 - 130	Yaron *et al.* (2001)
Putative CKII target	132 - 133	Yaron *et al.* (2001)
Putative CKII target	203 - 204	Yaron *et al.* (2001)

These functional sites have been extracted from UniProtKB/Swiss-Prot annotations and a mutagenesis study by Yaron *et al.* (2001). Figure 3.5(a) shows the alignment of HXB7_HUMAN and four confidently assigned FEPs (i.e., those assigned to the FOSTA family on the basis of protein prefix match) with HXB7A_DANRE and the other 24 *Danio rerio* candidates in the functionally relevant areas. Despite *globally* having the lowest sequence identity to HBX7_HUMAN of all the zebrafish proteins shown in Figure 3.5(a), it is clear that HXB7A_DANRE has the highest conservation at functionally critical sites. Across residues 126 to 133, HXB7A_DANRE only differs from HXB7_HUMAN at the position of a putative PBX binding site, unlike all but one (HXB5A_DANRE) of the other *Danio rerio* proteins which differ in a known sequence motif (residues 126-131). The homeobox region (which also includes crosslinking sites) is highly conserved across all of the zebrafish proteins, and again, conservation is highest in HXB7A_DANRE. None of the zebrafish proteins shows conservation at residues 203 and 204, which describe a putative CKII target site (Yaron *et al.*, 2001). It is possible that this functional site has been wrongly predicted; however, this is unlikely as it is absolutely conserved across the five mammalian species. It is more likely that this region is no longer functional in the *Danio rerio* lineage or that this is a recently acquired functionality in the mammalian clade.

Figure 3.5(b) shows an alternative approach to verifying the *Danio rerio* assignment of HXB7A_DANRE. Here, a phylogenetic tree has been constructed to characterise the relationships between the same proteins that were aligned in Figure 3.5(a). Again, it is clear that the protein selected by FOSTA (HXB7A_DANRE) is most closely related to the human protein and the four confidently assigned FEPs than any of the other zebrafish candidates, despite being of lower sequence identity.



(a) Verifying the FOSTA assignment by way of annotated functional regions
Residues identical to that of HXB7_HUMAN are in bold capitals and highlighted yellow, mismatching residues are non-captials and highlighted in light grey. The root human protein (HXB7_HUMAN) is indicated in the red box and the FOSTA-assigned zebrafish protein is highlighted in the blue box. The position relative to HXB7_HUMAN is given on the top line. The asterisks on the bottom line highlight fully conserved columns.



(b) Verifying the FOSTA assignment by way of phylogeny
Tree drawn using PHYLIP (fitch/drawtree), the FEPs identified by FOSTA are highlighted in yellow; HXB7_HUMAN, HXB7_BOVIN, HXB7_GORGO, HXB7_MOUSE and HXB7_RAT are shortened to H, B, G, M and R respectively.

Figure 3.5: Verifying the zebrafish assignment to the HXB7_HUMAN FOSTA family

3.3.4 A solved annotation problem: PROC_HUMAN

The UniProtKB/Swiss-Prot ID consists of a protein name followed by an underscore and the species name. The initial assumption was that the protein name part of the ID was a unique name used to label FEPs (Hulsen, 2004). However, while analysing human protein C (PROC_HUMAN) using the earlier UniProtKB/Swiss-Prot v50.6[2], it was evident that this approach was unreliable. The 'PROC' prefix was used in forty different species to describe three different proteins: Procalin in one species (PROC_TRIPT), protein C in 11 species (e.g., PROC_HUMAN), and pyrroline-5-carboxylate reductase in the remaining 28 species (e.g., PROC_ECOLI). FOSTA was successful in correctly assigning only true examples of protein C to the FEP group when analysing UniProtKB/Swiss-Prot v50.6, and analysis of human pyrroline-5-carboxylate reductase results highlighted the inconsistencies in UniProtKB/Swiss-Prot naming conventions.

Several of the FEPs in the FOSTA families of P5CR1_HUMAN (pyrroline-5-carboxylate reductase 1) and PROC_HUMAN (protein C) have had multiple protein prefix changes. However, after notifying UniProtKB/Swiss-Prot of the discrepancies, all the misnamed proteins were corrected for the release of UniProtKB/Swiss-Prot v51.2: pyrroline-5-carboxylate reductase proteins prefixed with PROC or PROH are now prefixed with P5CR or P5CR1 and PROC_TRIPT (procalin) is now called PRCLN_TRIPT.

UniProtKB/Swiss-Prot makes no guarantee that the protein prefix is a unique identifier, instead describing it as a 'mnemonic code', but it is stressed that work is ongoing to standardize protein nomenclature: *"Ambiguities regarding gene/protein names are a major problem in the literature and it is even worse in the sequence databases which tend to propagate the confusion...UniProt is constantly striving to further standardize the nomenclature for a given protein across related organisms"*[3]. Although this standardisation is discussed only with respect to protein names, and not the protein prefix elements of the UniProtKB/Swiss-Prot IDs, it is evident from the timings of prefix updates for protein C and pyrroline-5-carboxylate reductase proteins since UniProtKB/Swiss-Prot v50.6 that UniProtKB/Swiss-Prot does aim to standardize protein prefixes. If the protein prefix ID was used consistently across all proteins in UniProtKB/Swiss-Prot there would be no need for FOSTA.

[2]UniProtKB/Swiss-Prot v50.6 was released on 5th September 2006, UniProtKB/Swiss-Prot v53.0 was released on 29th May 2007; note that all results in this chapter are based on UniProtKB/Swiss-Prot v53.0, but older versions (including UniProtKB/Swiss-Prot v50.6 and UniProtKB/Swiss-Prot v51.2) are discussed in the evalution

[3]`http://www.expasy.org/cgi-bin/lists?nameprot.txt`

3.3.5 Manual analysis of five protein families

To evaluate FOSTA, a manual analysis of five protein families was carried out. The focus was the description fields and whether the description matches by FOSTA were appropriate. The first was trypsin-1, which was chosen because it belongs to the large serine protease family of proteins. The remaining four—glucose-6-phosphate isomerase, aminopeptidase N, ATP-dependent RNA helicase DDX51 and protoheme IX farnesyltransferase—were chosen at random.

3.3.5.1 Trypsin-1

FOSTA identifies eighteen FEPs for human trypsin-1 [UniProtKB:P07477/TRY1_HUMAN]. Of these, fifteen are clearly trypsin molecules. Some have additional non-functional qualifications (cationic, anionic and alkaline) and demonstrate that FOSTA can make correct FEP assignments despite extraneous information. There are five trypsin FEPs that are not described as trypsin-1 in the second species: TRYB_MANSE, TRYB_DROME, TRYDG_DROER, TRY5_AEDAE and TRYA3_LUCCU. TRYB_DROME and TRYDG_DROER are assigned in favour of several other trypsin proteins in *Drosophila melanogaster* and *Drosophila erecta* respectively. TRYB_MANSE is assigned in favour of TRYA_MANSE and TRYC_MANSE; in *Manduca sexta* (MANSE) it may be that trypsin-B corresponds to human trypsin-1. It is not clear whether the annotations are misleading or whether the FOSTA results are incorrect without specific information about how and why these proteins were annotated by the respective species annotation communities. There are only two trypsin proteins of adequate sequence similarity found in *Aedes aegypti*: TRY5_AEDAE and TRY3_AEDAE. TRY3_AEDAE is equivalent to TRY3_HUMAN and there is no human trypsin-5 protein in UniProtKB/Swiss-Prot v53.0, so the assignment here appears sensible.

In *Lucilia cuprina*, two trypsin proteins are of sufficient sequence similarity: TRYA3_LUCCU, which has been identified as the FEP of TRY1_HUMAN, and TRYA4_LUCCU which has been identified as the FEP of TRY3_HUMAN. This is a difficult assignment to assess, particularly as TRYA3_LUCCU is a fragmented protein. It is worth noting that these five questionable trypsin proteins are derived from insect species: LUCCU, DROME and DROER are flies, AEDAE is a mosquito and MANSE is a moth. It may be that trypsin genes have duplicated and diverged in insect species.

In addition to the trypsin molecules, FOSTA identifes GRAG_MOUSE, VSP1_BOTJR, VSP1M_TRIST as FEPs because they are described as serine proteases, as is TRY1_HUMAN. All mouse proteins explicitly described as trypsin belong to other FOSTA families, with protein prefix matches. A manual text search of UniProtKB/Swiss-Prot v53.0 reveals that there are no trypsin proteins for *Bothrops jararacussu* (BOTJR) or *Trimeresurus stejnegeri* (TRIST) described in UniProtKB/Swiss-Prot v53.0, but again it is unclear whether the assignment is correct or not.

3.3.5.2 Aminopeptidase N

FOSTA identifies 25 fully sequenced FEPs and two FEP fragments for aminopeptidase N [UniProtKB:P15144/AMPN_HUMAN]. There are seven assignments that do not match with respect to protein prefix: AAP1_YEAST, AMP11_ENCCU, AMP1_PLAFQ, AMPM_HELVI, AMPN1_LACLA, APE1_SULSO, APE1_SULTO. AMPN1_LACLA is assigned over the one other *Lactococcus lactis subsp. lactis* candidate (AMPN2_LACLA) as it is of higher sequence identity to AMPN_HUMAN. AMPN2_LACLA is assigned to the FOSTA family of PSA_HUMAN, another aminopeptidase. AMPN1_LACLA matches AMPN_HUMAN with respect to EC number, and contains the same synonym 'Aminopeptidase N'. AMPM_HELVI is the only protein *Heliothis virescens* protein found by the BLAST search and has a good description field match with AMPN_HUMAN; this appears to be the correct FEP for AMPN_HUMAN in *Heliothis virescens*. APE1_SULSO, APE1_SULTO, AAP1_YEAST, AMP11_ENCCU and AMP1_PLAFQ are the five least reliable assignments, although they are clearly aminopeptidases. Four of the five are flagged as unreliable (see Section 3.2.2.4) by FOSTA.

3.3.5.3 ATP-dependent RNA helicase DDX51

The ATP-dependent RNA helicase DDX51 [UniProtKB:Q8N8A6/DDX51_HUMAN] is assigned four full FEPs and four fragmented FEPs by FOSTA. The identification of FEPs for DDX51_HUMAN is a formidable task: DDX51_HUMAN belongs to a large family of 'DEAD box helicases', described by UniProtKB/Swiss-Prot family classifications[4]. All four of the fully sequenced proteins (DDX51_DANRE, DDX51_MOUSE, RH1_ARATH and RH1_ORYSJ) belong to the same subfamily as DDX51_HUMAN (the DDX51/DBP6 subfamily). The fragments IF413_TOBAC, DDX6_CAVPO, DDX1_DROVI and IF4A1_RABIT belong to the eIF4A,

[2]http://expasy.org/cgi-bin/get-similar?name=DEAD%20box%20helicase%20family

DDX6/DHH1, DDX1 and eIF4A subfamilies respectively. All proteins assigned to a different subfamily may be misassigned. The UniProtKB/Swiss-Prot family/domain classifications are manually confirmed, which suggests that in the case of DDX51_HUMAN, the candidate FEPs are so similar that FOSTA finds it difficult to discriminate between them. It should be stressed that a manual analysis of UniProtKB/Swiss-Prot entries for this family is no more effective than FOSTA, and that where FOSTA is incorrect in the DDX51_HUMAN assignments, the proteins are fragments and flagged as potentially unreliable.

3.3.5.4 Glucose-6-phosphate isomerase

The results for human glucose-6-phosphate isomerase [UniProtKB:P06744/G6PI_HUMAN] appear very robust: 309 FEPs are identified, of which two are fragments. All of these proteins are glucose-6-phosphate isomerases. Only eighteen of the 309 assignments are made on the basis of sequence (where sequence matching is required to differentiate between G6PI1-4 or G6PIA-B proteins) and 287 (92.88% of these are protein prefix matches). As already discussed, without explanation of how these proteins were named, it is not clear whether FOSTA is generating the correct pairs or whether the sequence matching is misleading.

3.3.5.5 Protoheme IX farnesyltransferase

FOSTA identifies 34 FEPs for human protoheme IX farnesyltransferase [UniProtKB:Q12887/COX10_HUMAN], all of which are fully sequenced proteins. These results appear very reliable, with only one FEP chosen from all candidates on the basis of sequence identity, where COXX_BACSU is chosen over CTAO_BACSU. Given that these two proteins are annotated identically in UniProtKB/Swiss-Prot, it is reasonable to resort to sequence similarity to discriminate between them. The results are particularly encouraging given that, unlike most of the G6PI_HUMAN FEPs for example, protoheme IX farnesyltransferases have different UniProtKB/Swiss-Prot protein prefixes in different species.

3.3.6 Further benchmarking

In this section, FOSTA is benchmarked against two datasets: the large, partially manually annotated PIRSF dataset (Wu *et al.*, 2004) and a refinement of Hulsen *et al.*'s manually curated dataset of six protein families that has been used previously to evaluate orthologue identification methods (Hulsen *et al.*, 2006). Manual inspection of the Hulsen *et al.* dataset identified the true one-to-one pairings in the one-to-many pairings.

FOSTA is designed to be conservative in the FEP assignments it makes: it is more important to minimise the number of false positives than to minimise the number of false negatives. Therefore, the most appropriate performance statistic with which to evaluate FOSTA is the positive predictive value (PPV): the proportion of positive predictions that are correct, $TP/(TP + FP)$. However, where possible, all performance statistics have been calculated to assess FOSTA (the performance statistics are described fully in Section 2.3.5). Within the context of FOSTA, sensitivity assesses what proportion of the FEPs that should be identified are identified, while specificity assesses what proportion of the FEPs that should be rejected are rejected.

3.3.6.1 Defining the negative examples

A true negative (TN) is a result that is correctly identified as negative. In the context of FOSTA, this is the number of non-FEPs that are correctly identified as non-FEPs. When benchmarking FOSTA against other datasets, there are several ways to count the number of genuinely negative examples. For example, all proteins identified by BLAST could be used regardless of the species. However, this would artifically boost the performance of FOSTA as measured by performance statistics that include true negatives (including specificity and MCC). A more appropriate count of negative examples would only consider the proteins from the same species identified by the BLAST search. For example, consider the assignment of a mouse FEP to the FOSTA family of human protein A: if the BLAST search returned two mouse proteins, a rat protein and a bovine protein, the count of negative examples could be two (the two mouse proteins) or four (all proteins identified by the BLAST search). Choosing the less conservative method (where the number of negative examples is 4), performance will be artificially boosted. In this chapter, the number of negative examples is calculated as the number of FDH proteins from that species.

Table 3.3: Benchmarking FOSTA against the PIRSF dataset
Set: the identifier for each curation set [A='Full/Desc.', B='Full', C='Preliminary', D='None', N=aNnotated (A+B+C), $*$=All (N+D)]; **Curation string**: the string that defines the curation set; **Families**: the number of discrete protein families in the curation set; **Pairings**: the number of discrete pairings across all families to be tested in FOSTA; **Basic statistics**: the basic counts of true positives (TP), false positives (FP), true negatives (TN), false negatives (FN); **Evaluation statistics**: the **spec/specificity** ($TN/(FP+TN)$), the **sens/sensitivity** ($TP/(TP+FN)$), the **PPV** (positive predictive value, $TP/(TP+FP)$), and the **MCC** (Matthews Correlation Coefficient), all rounded to 2dp.

Set	Fams	Pairs	Basic statistics				Evaluation statistics			
			TP	FP	TN	FN	spec	sens	PPV	MCC
A	122	2127	1744	2	3717	383	99.95	81.99	99.89	0.86
B	1095	18865	12967	23	34656	5898	99.93	68.74	99.82	0.77
C	474	11221	9146	62	11819	2075	99.48	81.51	99.33	0.83
D	339	5287	3674	16	4938	1613	99.68	69.49	99.57	0.72
N	1691	32213	23857	87	50192	8356	99.83	74.06	99.64	0.79
$*$	2020	37500	27531	103	55130	9969	99.81	73.42	99.63	0.79

3.3.6.2 PIRSF evaluation

The Protein Information Resource (PIR) is a widely used, publicly available resource, and is part of the UniProtKB consortium. With a view to the standardization of accurate propagation of protein annotations, PIR has developed the PIRSF (PIR super family) classification system for UniProtKB proteins (Wu *et al.*, 2004). However, unlike FOSTA, it does not identify FEPs as it contains many-to-many orthologous pairings.

FOSTA was benchmarked against all one-to-one orthologous relationships between UniProtKB/Swiss-Prot proteins that are listed in PIRSF families as 'regular' members ('associate' members can be alternative splice variants, which should not be FEPs), at all four levels of curation, where PIRSF families with a curation status of *'Full/Desc'* have the highest level of manual curation and families with a curation status of *'None'* have not been manually curated.

It is evident from Table 3.3 that FOSTA performs extremely well on the PIRSF protein families according to the PPV and specificity metrics that are particularly important. However, it also demonstrates reasonably high sensitivity and very high MCC scores.

Table 3.4: Benchmarking FOSTA against the refined Hulsen *et al.* dataset
Family: the protein family being examined; **TO pairings**: the number of TO pairs in the Hulsen dataset (including many-to-many orthologous pairings and non-UniProtKB/Swiss-Prot proteins); **Refined pairings**: the number of one-to-one TO pairings tested after refinement of Hulsen TO dataset; **Basic statistics**: the basic counts of true positives (TP), false positives (FP), true negatives (TN), false negatives (FN); **Evaluation statistics**: the **spec/specificity** ($TN/(FP + TN)$), the **sens/sensitivity** ($TP/(TP + FN)$), the **PPV** (positive predictive value, $TP/(TP + FP)$), and the **MCC** (Matthews Correlation Coefficient), all rounded to 2dp.

Family	Refined (TO)	Basic statistics				Evaluation statistics			
		TP	FP	TN	FN	spec	sens	PPV	MCC
HBB	2 (9)	2	0	17	0	100.00	100.00	100.00	1.00
HOX	30 (41)	30	0	3853	0	100.00	100.00	100.00	1.00
SMm	12 (17)	12	0	22	0	100.00	100.00	100.00	1.00
SMc	6 (6)	6	0	5	0	100.00	100.00	100.00	1.00
NR	4 (29)	1	1	327	3	99.70	25.00	50.00	0.35
All	54 (102)	51	1	4224	3	99.98	94.44	98.08	0.96

3.3.6.3 Refined Hulsen evaluation

Hulsen *et al.* (2006) recently evaluated the performance of several orthologue identification methods: BBH (bidirectional best hit), Inparanoid (O'Brien *et al.*, 2005), KOG (Tatusov *et al.*, 2003), OrthoMCL (Chen *et al.*, 2006), PhyloGeneticTree (van Noort *et al.*, 2003) and Z 1 hundred (estimating statistical significance of alignment scores). The benchmarking included comparison with manually annotated 'true-orthologue' (TO) pairs of six protein families. For human-mouse (*Homo sapiens* and *Mus musculus*) pairings, the protein families used were the homeobox proteins (HOX), haemoglobins (HBB), and Sm and Sm-like proteins (SMm). For human and worm (*Caenorhabditis elegans*) TO pairs, the families used were nuclear receptors (NR), toll-like receptors (TLR), and Sm and Sm-like proteins (SMc).

These methods all aim to identify orthologues and do not consider functional equivalence. Since they have different goals, it is not possible to compare FOSTA directly with the methods evaluated by Hulsen *et al.*, but FOSTA can be evaluated using a subset of the TO data.

The TO dataset supports many-to-many orthologous pairings where a human protein can map to one or more proteins in another species and vice versa. To evaluate FOSTA, these data were manually refined to include only those TO pairings that can be confidently identified as true one-to-one orthologous pairings, where *both* proteins can be mapped to UniProtKB/Swiss-Prot (compare the 'Refined' and 'TO' counts in Table 3.4). This refinement process removes the TLR dataset from the analysis as no definitive one-to-one orthologous pairings could be identified

through manual inspection.

The results are summarised in Table 3.4. FOSTA demonstrates perfect performance in the HBB, HOX, SMm and SMc families, identifying all refined true-orthologue pairings.

However, FOSTA identified only one of the four refined human/worm nuclear receptor (NR) TO pairs (NHR67_CAEEL). On closer inspection, it is evident that these three failures of FOSTA in the NR dataset are a result of widely varying formats of the UniProtKB/Swiss-Prot description field across the two species. For example, the Homo sapiens proteins often have multiple synonyms, which vary in format and content, whereas the Caenorhabditis elegans proteins are more consistently named as "Nuclear hormone receptor family member nhr-N" proteins (see Table 3.5). These primary protein names or descriptions are defined by the species-specific annotation communities (for example, Human Genome Nomenclature Committee, FlyBase and Caenorhabditis Genetics Centre/Wormbase for Homo sapiens, Drosophila melanogaster and Caenorhabditis elegans respectively) with additional synonyms obtained by UniProtKB/Swiss-Prot from the literature. Therefore, it is not possible to attribute the lack of annotation consistency to problems in UniProtKB/Swiss-Prot, as UniProtKB/Swiss-Prot is merely reflecting the differing practices of the annotation communities and the content of the literature. Nevertheless, the lack of consistent description field formatting within UniProtKB/Swiss-Prot limits the extent to which text-mining methods such as FOSTA can exploit the data.

It is encouraging to note that FOSTA makes only one false positive assignment in the refined Hulsen dataset. Furthermore, FOSTA does not eliminate any of the one-to-one TO pairs: where a FEP relationship is missed, the TO is retained as a FDH, indicating that the BLAST threshold is not too conservative.

3.3.7 A comparison with Inparanoid

Inparanoid is a well-known method of constructing sets of orthologous proteins (O'Brien *et al.*, 2005). It uses BBH (best bidirectional hit) pairs in different species as a 'seed' around which a cluster of orthologues can be formed. Other orthologues—or specifically other inparalogues—can be added to this pairing if they are more similar to one of the seed orthologues than they are to any other protein in another species.

Table 3.5: UniProtKB/Swiss-Prot annotations of human and worm nuclear receptor (NR) proteins.

NR Human proteins	
HNF4A_HUMAN	Hepatocyte nuclear factor 4-alpha; HNF-4-alpha; Transcription factor HNF-4; Transcription factor 14
HNF4G_HUMAN	Hepatocyte nuclear factor 4-gamma; HNF-4-gamma
NR1D1_HUMAN	Orphan nuclear receptor NR1D1; V-erbA-related protein EAR-1; ReverbA-alpha
NR1D2_HUMAN	Orphan nuclear receptor NR1D2; Rev-erb-beta; EAR-1R; Orphan nuclear hormone receptor BD73
NR1I3_HUMAN	Orphan nuclear receptor NR1I3; Constitutive androstane receptor; Constitutive activator of retinoid response; Constitutive active response; CAR; Orphan nuclear receptor MB67
NR2E1_HUMAN	Orphan nuclear receptor NR2E1; Nuclear receptor TLX; Tailless homolog; Tll; hTll
NR4A1_HUMAN	Orphan nuclear receptor NR4A1; Orphan nuclear receptor HMR; Early response protein NAK1; TR3 orphan receptor; ST-59
NR4A2_HUMAN	Orphan nuclear receptor NR4A2; Orphan nuclear receptor NURR1; Immediate-early response protein NOT; Transcriptionally-inducible nuclear receptor
NR4A3_HUMAN	Orphan nuclear receptor NR4A3; Nuclear hormone receptor; NOR-1; Neuron-derived orphan receptor 1; Mitogen-induced nuclear orphan receptor
NR5A2_HUMAN	Orphan nuclear receptor NR5A2; Alpha-1-fetoprotein transcription factor; Hepatocytic transcription factor; B1-binding factor; hB1F; CYP7A promoter-binding factor; Liver receptor homolog 1; LRH-1
NR6A1_HUMAN	Orphan nuclear receptor NR6A1; Germ cell nuclear factor; GCNF; Retinoid receptor-related testis-specific receptor; RTR
RORA_HUMAN	Nuclear receptor ROR-alpha; Nuclear receptor RZR-alpha
RORB_HUMAN	Nuclear receptor ROR-beta; Nuclear receptor RZR-beta
RORG_HUMAN	Nuclear receptor ROR-gamma; Nuclear receptor RZR-gamma
STF1_HUMAN	Steroidogenic factor 1; STF-1; SF-1; Adrenal 4-binding protein; Steroid hormone receptor Ad4BP; Fushi tarazu factor homolog 1
TR2_HUMAN	Orphan nuclear receptor TR2; Testicular receptor 2
TR4_HUMAN	Orphan nuclear receptor TR4; Orphan nuclear receptor TAK1
VDR_HUMAN	Vitamin D3 receptor; VDR; 1,25-dihydroxyvitamin D3 receptor
NR Worm proteins	
NHR6_CAEEL	Nuclear hormone receptor family member nhr-6; Steroid hormone receptor family member cnr8
NHR8_CAEEL	Nuclear hormone receptor family member nhr-8
CNR14_CAEEL	Steroid hormone receptor family member cnr14
NHR23_CAEEL	Nuclear hormone receptor family member nhr-23; Steroid hormone receptor family member chr-3
NHR25_CAEEL	Nuclear hormone receptor family member nhr-25
NHR41_CAEEL	Nuclear hormone receptor family member nhr-41
NHR48_CAEEL	Nuclear hormone receptor family member nhr-48
NHR64_CAEEL	Nuclear hormone receptor family member nhr-64
NHR67_CAEEL	Nuclear hormone receptor family member nhr-67
NHR69_CAEEL	Nuclear hormone receptor family member nhr-69
NHR85_CAEEL	Nuclear hormone receptor family member nhr-85
NHR91_CAEEL	Nuclear hormone receptor family member nhr-91

Inparanoid does not perform the same task as FOSTA: FOSTA is specifically interested in identifying *functionally* equivalent orthologous proteins, whereas Inparanoid is more interested in identifying the correct phylogenetic relationships between proteins in different species. As such, where Inparanoid detects one-to-one orthologous pairs, the results will be largely complementary, but need not be identical. Therefore, FOSTA cannot be 'benchmarked' against Inparanoid: it is not the gold standard dataset. However, by identifying one-to-one orthologous pairings in the Inparanoid dataset that FOSTA rejects, a dataset of more difficult test cases with which FOSTA can be evaluated can be constructed.

The XML datafiles for Inparanoid v6.1 were obtained by ftp from `http://inparanoid.sbc.su.se` and parsed in Perl using XML::DOM. All human/X one-to-one orthologues described by Inparanoid were extracted. For convenience, these extracted human/X one-to-one Inparanoid orthologue pairs will be referred to as the '**I**nparanoid **P**airs' or **IPs**. There were fifteen species in which no IPs were found (*Aedes aegypti*, *Anopheles gambiae*, *Arabidopsis thaliana*, *Caenorhabditis briggsae*, *Caenorhabditis elegans*, *Caenorhabditis remanei*, *Candida glabrata*, *Cryptococcus neoformans*, *Debaryomyces hansenii*, *Entamoeba histolytica*, *Escherichia coliK12*, *Kluyveromyces lactis*, *Schizosaccharomyces pombe*, *Takifugu rubripes* and *Yarrowia lipolytica*), leaving nineteen species with at least one IP to compare with FOSTA.

As FOSTA groups UniProtKB/Swiss-Prot pairings, all extracted IPs must be mapped to UniProtKB/Swiss-Prot. Inparanoid proteins are described using various database IDs, including Ensembl[5] (*Apis mellifera*, *Bos taurus*, *Canis familiaris*, *Ciona intestinalis*, *Gallus gallus*, *Gasterosteus aculeatus*, *Macaca mulatta*, *Monodelphis domestica*, *Pan troglodytes*, *Rattus norvegicus*, *Tetraodon nigroviridis*, *Xenopus tropicalis*), TAIR[6] (*Arabidopsis thaliana*), Zfin[7] (*Danio rerio*), Dictybase[8] (*Dictyostelium discoideum*), Flybase[9] (*Drosophila melanogaster* and *Drosophila pseudoobscura*), MGI[10] (*Mus musculus*), Gramene[11] (*Oryza sativa*) and SGD[12] (*Saccharomyces cerevisiae*). All relevant cross-references were extracted from UniProtKB/Swiss-Prot v53.0; any conflicting or multiple cross-references (e.g., X→Y, X→Z) were not used.

Using the UniProtKB/Swiss-Prot cross-references to map from the Inparanoid Ensembl protein

[5] `http://www.ensembl.org`
[6] `http://www.arabidopsis.org/`
[7] `http://zfin.org`
[8] `http://dictybase.org/`
[9] `http://flybase.org/`
[10] `http://www.informatics.jax.org/`
[11] `http://www.gramene.org/`
[12] `http://www.yeastgenome.org/`

IDs to UniProtKB/Swiss-Prot sequences does result in a biased dataset: the UniProtKB/Swiss-Prot sequences with explicit cross-references are likely to be well-annotated. Nevertheless, it is reassuring that where the Inparanoid dataset does identify one-to-one pairings between UniProtKB/Swiss-Prot proteins, FOSTA confirms 96.23% of these pairings in a large dataset (27 069 protein pairs) with significant protein coverage (*Apis mellifera*, *Bos taurus*, *Conis familiaris*, *Ciona intestinalis*, *Danio rerio*, *Dictyostelium discoideum*, *Drosophila melanogaster*, *Drosophila pseudoobscura*, *Gallus gallus*, *Gasterosteus aculeatus*, *Macaca mulatta*, *Monodelphis domestica*, *Mus musulus*, *Oryza sativa*, *Pan troglodytes*, *Rattus norvegicus*, *Saccharomyces cerevisiae*, *Tetraodon nigroviridis* and *Xenopus tropicalis*).

Columns 1-3 in Table 3.6 describe how many IPs from each species were successfully mapped to UniProtKB/Swiss-Prot IDs, and therefore how many IPs from each species can be compared with FOSTA. As described above, 27 069 IPs are extracted from Inparanoid v6.1, of which 26 073 (96.32%) are verified by FOSTA. Of the 996 IPs that are not found in FOSTA, 125 are rejected in favour of another UniProtKB/Swiss-Prot protein from the non-human species (these IPs will be described as 'contested' IPs). In the remaining 871 IPs, FOSTA fails to assign any FEP from the non-human species to the human protein (these IPs will be described as 'uncontested' IPs).

These datasets can be further 'cleaned' to remove those IPs that either (i) cannot be found by FOSTA or (ii) are clearly correct in FOSTA. Tables 3.7 and 3.8 describe this further refinement process. 43 of the contested Inparanoid pairs (IPs) appear to be wrong, since the FEP that FOSTA assigns matches the human protein confidently using the protein prefix match. For example, Inparanoid identifies SFH1_YEAST as the *Saccharomyces cerevisiae* partner to SNF5_HUMAN, while FOSTA identifies the more plausible FEP SNF5_YEAST. A further five appear to be wrong, as the non-human protein is assigned as a FEP using a protein prefix match elsewhere in FOSTA. For example, Inparanoid partners CAN9_HUMAN with CAN3_BOVIN, while FOSTA identifies the FEP CAN2_BOVIN; although it is not clear which result is correct, FOSTA identifies the Inparanoid partner CAN3_BOVIN as the FEP of CAN3_HUMAN. It is therefore unlikely that CAN3_BOVIN is the true equivalent protein of CAN9_HUMAN. Eliminating these examples, 77 IPs remain as test cases for FOSTA.

36.74% (320) of the 871 uncontested IPs cannot be identified by FOSTA: 26.98% are not found using a BLAST threshold of 10^{-2} and 1.15% involve short human proteins that FOSTA does not analyse (see Section 3.2). This is not strictly a failure of the FOSTA method but an inevitable result of implementing a conservative method. A further 75 (8.61%) are found to be wrong:

Table 3.6: Comparing FOSTA with Inparanoid

Code: The species code as used by Inparanoid; **Species**: The full species name; **Pairs**: The number of one-to-one orthologue pairs described by Inparanoid between **Species** and human; **Matches**: The number of one-to-one Inparanoid orthologue pairs (IPs) that are also found by FOSTA; **Mismatches**: The number of IPs pairs that are *not* found by FOSTA; **% match**: The percentage of IPs that are also found by FOSTA; **Contested**: The number of IPs where FOSTA assigns a different protein from the **Species** to the FOSTA family of the human protein; **Uncontested**: The number of IPs where FOSTA does not assign any protein from the **Species** to the FOSTA family of the human protein.

Code	Species	Pairs	Matches	Mismatches	% match	Contested	Uncontested
APIME	*Apis mellifera*	1	1	0	100.00%	-	-
BOSTA	*Bos taurus*	3508	3451	57	98.38%	1	56
CANFA	*Canis familiaris*	533	520	13	97.56%	1	12
CIOIN	*Ciona intestinalis*	6	5	1	83.33%	0	1
DANRE	*Danio rerio*	1246	1192	54	95.67%	21	33
DICDI	*Dictyostelium discoideum*	85	69	16	81.18%	0	16
DROME	*Drosophila melanogaster*	878	712	166	81.09%	14	152
DROPS	*Drosophila pseudoobscura*	73	67	6	91.78%	0	6
GALGA	*Gallus gallus*	1360	1297	63	95.37%	12	51
GASAC	*Gasterosteus aculeatus*	1	1	0	100.00%	-	-
MACMU	*Macaca mulatta*	214	207	7	96.73%	0	7
MONDO	*Monodelphis domestica*	22	21	1	95.45%	0	1
MUSMU	*Mus musculus*	12063	11960	103	99.15%	18	85
ORYSA	*Oryza sativa*	1	0	1	0.00%	0	1
PANTR	*Pan troglodytes*	412	408	4	99.03%	1	3
RATNO	*Rattus norvegicus*	5076	5005	71	98.60%	6	65
SACCE	*Saccharomyces cerevisiae*	1213	787	426	64.88%	49	377
TETNI	*Tetraodon nigroviridis*	6	6	0	100.00%	-	-
XENTR	*Xenopus tropicalis*	371	364	7	98.11%	2	5
-	All species	27069	26073	996	96.32%	125	871

Table 3.7: Identifying a 'clean' dataset of contested IPs to consider in FOSTA
Contested IPs are pairs of human/X Inparanoid protein pairs where FOSTA assigns a *different* protein from the species X as the FEP of the human protein. Here, 'confident' FEP assignments are defined as those based on a protein prefix match.

125	100.00%	**contested IPs**
43	34.40%	IPs are wrong: FOSTA assigns the non-human protein confidently
5	4.00%	IPS are wrong: FOSTA assigns the non-human protein confidently elsewhere
77		**contested IPs remain to be tested**

Table 3.8: Identifying a 'clean' dataset of uncontested IPs to consider in FOSTA
Uncontested IPs are pairs of human/X Inparanoid protein pairs where FOSTA does not assign *any* protein from the species X as the FEP of the human protein. Here, 'confident' FEP assignments are defined as those based on a protein prefix match. Some IPs will not be identified as a FEP by FOSTA because (i) the IP protein does not exceed the e-value BLAST threshold or (ii) the human protein is too short (see Section 3.2).

871	100.00%	**uncontested IPs**
10	1.15%	IPs will not be FEPs because the human protein is too short
235	26.98%	IPs will not be FEPs because they aren't found by BLAST
75	8.61%	IPs are wrong: FOSTA assigns the non-human protein confidently elsewhere
551		**uncontested IPs remain to be tested**

FOSTA assigns the non-human protein elsewhere on the basis of a protein prefix match. For example, Inparanoid identifies xxx_yyyyy as the partner to zzz_HUMAN, but FOSTA describes xxx_yyyyy as the FEP to xxx_HUMAN. This leaves 551 IPs to test as potential errors in FOSTA.

This leaves a 'clean' dataset of 77 contested IPs and 551 uncontested IPs with which to test FOSTA. In a random sample of ten of the contested IPs (see Table 3.9), three FOSTA assignments and one Inparanoid assignment appear to be correct. There is not enough evidence in the six remaining contested IPs to ascertain which assignment might be correct; however, four of the six remaining IPs are flagged as less reliable sequence matches by FOSTA and could therefore be removed from the dataset.

A random sample of 28 IPs (approximately 5%) were selected from the uncontested dataset (see Table 3.10). Note that the IPs described in this dataset are not necessarily correct; they can, however, be used as examples of difficult test cases. Most of the IPs are not assigned a FEP by FOSTA owing to uninformative or sparsely annotated DE fields. A significant number arise from large, densely populated protein families in which functional relationships are hard to

Table 3.9: A random sample of ten contested IPs (Inparanoid pairs)
These are contested by FOSTA (i.e., FOSTA assigns a different FEP from that species).

Human	**Inparanoid**	**FOSTA**
PLCH_HUMAN	PLCHB_DANRE	PLCHA_DANRE
MOXD1_HUMAN	MOX11_DROME	PHM_DROME
EPN4_HUMAN	ENT3_YEAST	ENT4_YEAST
WDR59_HUMAN	YD128_YEAST	YBK4_YEAST
CP2B6_HUMAN	CP2BB_CANFA	CP2CL_CANFA
IF4A3_HUMAN	FAL1_YEAST	IF4A_YEAST
O5AP2_HUMAN	O1020_MOUSE	O1086_MOUSE
DDX6_HUMAN	DHH1_YEAST	DBP6_YEAST
MCM4_HUMAN	CDC54_YEAST	CDC47_YEAST
OR5J2_HUMAN	O1052_MOUSE	O1094_MOUSE

Table 3.10: A random sample of 28 uncontested IPs (Inparanoid pairs)
These are uncontested in FOSTA (i.e., FOSTA does not assign any FEP from that species).

Human	**Inparanoid**
TIM_HUMAN	TOF1_YEAST
CC45L_HUMAN	CDC45_YEAST
SURF1_HUMAN	SHY1_YEAST
TEX10_HUMAN	IPI1_YEAST
FA2H_HUMAN	SCS7_YEAST
IPO9_HUMAN	IMB5_YEAST
ISK5_HUMAN	IOV7_CHICK
LETM1_HUMAN	A60DA_DROME
FRK_HUMAN	SRC42_DROME
NVL_HUMAN	RIX7_YEAST
MMS19_HUMAN	MET18_YEAST
DYR1A_HUMAN	MNB_DROME
ATBP3_HUMAN	NCS6_YEAST
DCR1A_HUMAN	PSO2_YEAST
PDXK_HUMAN	BUD16_YEAST
PLAP_HUMAN	DOA1_YEAST
JAZF1_HUMAN	SFP1_YEAST
EXTL3_HUMAN	EXT3_DROME
ZUBR1_HUMAN	POE_DROME
IPO11_HUMAN	KA120_YEAST
RBBP6_HUMAN	MPE1_YEAST
TRIPC_HUMAN	UFD4_YEAST
PAP1L_HUMAN	EPAB_XENTR
PINX1_HUMAN	YG5W_YEAST
CFDP1_HUMAN	SWC5_YEAST
FGF17_HUMAN	FG17B_DANRE
XPOT_HUMAN	LOS1_YEAST
TM11A_HUMAN	DESC4_RAT

Table 3.11: Insensitivities in the FOSTA functional match methodology
By benchmarking FOSTA against the popular (but fundamentally different) method Inparanoid, some insensitivies in the FOSTA description field matching (described in Section 3.2.2.2) have become apparent. Two of these are shown below.

Mapping to/from acronyms and long forms	
CC45L_HUMAN	CDC45-related protein; PORC-PI-1; Cdc45
CDC45_YEAST	Cell division control protein 45
Allowing for slight variations in names and numbers	
FGF17_HUMAN	Fibroblast growth factor 17 precursor; FGF-17
FG17B_DANRE	Fibroblast growth factor 17b precursor; FGF-17b

elucidate.

Only two examples highlight where the FOSTA functional match methodology may lack sensitivity; these are shown in Table 3.11. The first example—CC45L_HUMAN/CDC45_YEAST—suggests that mapping from acronyms to long forms and vice versa may be valuable in a future version of FOSTA. In the first example shown in Table 3.11, CDC would be extended to 'Cell division control'. In the second FGF17_HUMAN/FG17B_DANRE example, some flexibility in names and numbers used by the matching machinery would lead to these two proteins being identified as FEPs. However, introducing such additional flexibility without careful consideration would increase the likelihood of false positives being introduced into the FOSTA dataset. The priority in FOSTA has been to minimize the number of false positives in order to have a reliable dataset.

3.4 Conclusions

FOSTA is a novel method that extracts functionally equivalent proteins (FEPs) from the UniProtKB/Swiss-Prot database by 'reading' the UniProtKB/Swiss-Prot annotations. As such, it is a grouping of UniProtKB/Swiss-Prot proteins that are annotated similarly. FOSTA takes advantage of the fact that UniProtKB/Swiss-Prot annotations are the result of many hours of manual annotation and should encapsulate all knowledge available to the annotator at the time.

Since FOSTA simply assimilates existing annotations, it is difficult to separate the

performance of the FOSTA method from the quality and consistency of annotation in UniProtKB/Swiss-Prot. Manual analysis of eight FOSTA families (those rooted around PROC_HUMAN, P5CR1_HUMAN, AMPN_HUMAN, COX10_HUMAN, TRY1_HUMAN, DDX51_HUMAN, G6PI_HUMAN and HXB7_HUMAN) and two benchmarking evaluations were carried out which indicate that FOSTA performs well and that UniProtKB/Swiss-Prot annotations are generally of high quality. In a comparison with the popular but conceptually quite different Inparanoid method, the results were largely complementary. In addition to providing researchers with genuine FEP families for tasks such as studying sequence conservation, FOSTA could be used to provide datasets to evaluate function prediction methods.

Given the methodology, FOSTA has a few limitations. Firstly, FOSTA is clearly dependent on UniProtKB/Swiss-Prot annotations. Any method based on database annotations is potentially problematic as it relies on possibly mistaken, incomplete, inconsistent, ambiguous or outdated information. However, the UniProtKB/Swiss-Prot database is considered to be the gold standard for protein annotation (the benchmarking results reflect that the annotations are indeed very reliable), and annotations are constantly revised (for example, 210 454 annotation revisions were made between release UniProtKB/Swiss-Prot v52.0 and UniProtKB/Swiss-Prot v53.0[13]). The continuous revision of UniProtKB/Swiss-Prot with the regular update of FOSTA ensures that FOSTA FEP assignments can only improve in parallel with UniProtKB/Swiss-Prot.

Secondly, only proteins described in UniProtKB/Swiss-Prot can be assigned to FOSTA families. Given that UniProtKB/Swiss-Prot is growing at an exponential rate[14] and that it is the aim to include all proteins in UniProtKB/Swiss-Prot, this limitation is not considered significant. A related problem is that for many species, UniProtKB/Swiss-Prot does not describe the entire proteome. In a few cases, a gene duplication may have resulted in two or more similar sequences of which only one appears in UniProtKB/Swiss-Prot with an annotation which should more appropriately be applied to one of the other sequences. Thus the true FEP may be a protein not yet present in UniProtKB/Swiss-Prot. However, it should be noted that FOSTA is simply trying to assimilate the current, curated knowledge of protein function to identify evolutionarily related proteins that have been described similarly; manual examination of UniProtKB/Swiss-Prot entries would make the same errors.

If FOSTA cannot discriminate between two candidate FEPs on the basis of function, it will

[13] http://www.expasy.ch/txt/old-rel/relnotes.53.htm
[14] http://expasy.org/sprot/relnotes/relstat.html

choose the candidate with the higher sequence identity to the root; only 6 047 of FEP assignments (5.00%) are made on this basis. Any sequence matching is undesirable, as high sequence similarity does not necessarily imply precise functional equivalence (see Section 3.1). It may be avoided if more sensitive information extraction methods could be implemented to improve functional discrimination. UniProtKB/Swiss-Prot keywords and GO terms may have some value, but these tend to be at a higher level of annotation and are unlikely to improve discrimination of very detailed functional information. More sophisticated natural language processing methods (Rice *et al.*, 2005) would not be expected to improve performance, as the text being examined is simply a list of nouns. Alternatively, a more sensitive sequence matching protocol could be implemented where annotated functional residues, or a consensus profile of FEPs already assigned with high confidence could be used, rather than the whole sequence (which may be misleading). Furthermore, a vocabularly mapping acronyms to their long forms and vice versa, and/or mapping between known synonyms may improve the functional comparison step.

FOSTA's insistence on one-to-one FEP relationships may also be viewed as a limitation, but is considered to be justified. Consider the protein X in species A that has two homologues Y_1 and Y_2 in species B. If Y_1 and Y_2 are both homologous to X, one must have been derived via a gene duplication event. Gene duplication is a mechanism for functional divergence and one can therefore argue that either Y_1 or Y_2, most likely (though not necessarily) the one with the poorer sequence identity to X, has acquired novel, or lost existing, functionality (or is in the process of doing so), and should not be selected as a FEP.

Currently, FOSTA roots families around human proteins because the priority is to identify FEPs to human proteins, with a view to examining human disease. 58.36% (169 523 of 290 484) of UniProtKB/Swiss-Prot proteins are not assigned to a FOSTA family in UniProtKB/Swiss-Prot v53.0. Using the median size of a FOSTA family (87), one can estimate that another 1949 families will be formed if FOSTA were to cluster around non-human proteins. It is proposed that a future version of FOSTA will root FOSTA families around decreasingly well defined (in terms of proteome coverage and functional annotation in UniProtKB/Swiss-Prot) species, until all proteins are assigned to a FOSTA family. While it is hoped that this will be addressed in future versions, it must be noted that human proteins are the most thoroughly annotated, and it is unclear whether proteins from other organisms will be annotated well enough to identify functional equivalencies across species.

Where there are annotation problems and inconsistencies across species, these are often not strictly attributable to UniProtKB/Swiss-Prot, as the description fields are generated from annotations provided by the species-specific annotation communities, who may differ in their annotation practices. However, as a widely used and trusted resource, UniProtKB/Swiss-Prot is in a unique position to rectify such problems, and could implement a second layer of description above that of the separate annotation communities, which would aim to provide a standardised nomenclature across all species. It is hoped that the FOSTA results may contribute to any such effort. It has already done so with the recent correction of 'PROC' prefixes as described in Section 3.3.4.

More generally, a controlled vocabulary for UniProtKB/Swiss-Prot description fields which would allow description of all proteins across all species, would facilitate text mining and result in more reliable hypotheses. This might be implemented as a second, computer-friendly description field, keeping the existing descriptions for human inspection. In addition, it would be desirable to move some information from the description field into separate tags in the UniProtKB/Swiss-Prot flatfile format; for example, flags for fragmented or hypothetical sequences. Given the size of UniProtKB/Swiss-Prot (UniProtKB/Swiss-Prot v53.0 contains 290 484 proteins), the resource must expect to be interrogated computationally, more so with every new release. Any effort from UniProtKB/Swiss-Prot to make its contents more computationally accessible would be valuable. Note that recent UniProtKB/Swiss-Prot releases have split the description field into an easier-to-parse, structured form, although this was done in such a way that old parsers are not longer able to extract information from UniProtKB/Swiss-Prot.

As stated above, a guarantee of unique UniProtKB/Swiss-Prot protein ID prefixes for equivalent proteins in different species would preclude the need for hypotheses to be drawn by software such as FOSTA. It is clear that the UniProtKB/Swiss-Prot team are making efforts to standardise such annotations across species[15]; however it is also clear that some efforts are not yet propagated fully across all relevant proteins and species. As stated above, the protein C/pyrroline-5-carboxylate reductase case described in Section 3.3.4 has since been rectified by the UniProtKB/Swiss-Prot annotators.

It is clear that not only is the automatic extraction of FEPs a surprisingly difficult problem, but that it is also very difficult to evaluate these methods. The evaluation that was performed

[15] http://www.expasy.ch/txt/old-rel/relnotes.53.htm

not only demonstrated that FOSTA performs well, but also that the vast majority of UniProtKB/Swiss-Prot annotations used by FOSTA are of high quality. This provides further justification of an annotation-based methods such as FOSTA and indicates that any concern about FOSTA's dependence on annotations need not be over-emphasized. In addition, it is expected that FOSTA will improve with every revision of UniProtKB/Swiss-Prot.

Chapter 4

Generating an Improved Protein Alignment Conservation Threshold

Chapter 3 described FOSTA, a method for extracting functionally equivalent proteins (FEPs) from UniProtKB/Swiss-Prot. In this chapter, a conservation scoring method is described, which analyses alignments of FEPs (or any other MSA) to generate an **Im**proved **P**rotein **A**lignment **C**onservation **T**hreshold, or **ImPACT** score, for any given protein alignment.

4.1 Introduction

When structural analyses fail to 'explain' a disease-causing mutation, it may be possible to infer functional relevance from sequence conservation: if a residue is conserved across many different branches of evolution, it is likely that that residue is functionally significant.

The extent of conservation is often described using some system of scoring. Conservation scores are a function of genuine functional equivalence across species. However, they are also a function of the species set represented and a function of properties of the proteins they contain. As the species set represented by a multiple sequence alignment (MSA) widens, it becomes less likely that residues will be conserved by chance, because the evolutionary distance between the species represented in the MSA is greater. As such, lower conservation scores will become

more significant as markers of functional relevance.

In addition, some proteins are highly conserved throughout the MSA, due to a high number of functional residues or a highly conserved network of stabilising interactions, for example. An alignment of proteins highly conserved for functional or structural reasons might therefore appear to represent a more closely related set of sequences than they actually do. On the other hand, other proteins will be generally *not* well conserved, having diverged between species, and will appear to represent a set of more *distant* species. Once again, lower conservation scores will become significant as markers of functional relevance when considering residue conservation in the context of a globally poorly conserved protein.

To elucidate genuine trends of conservation, it is necessary to (a) take into account the influence of the species set represented by the MSA and (b) consider how well the protein is conserved at the 'global' level. In the context of SAAPdb, any method developed must be automatic. As such, it is necessary to factor out the influence of the species set and global conservation by taking a *statistical* approach to defining high conservation.

4.1.1 What is conservation?

Conservation describes whether a residue is seen at the equivalent position, in an equivalent protein, in different species. Alignment methods identify which residues are equivalent. If a residue is maintained across species, it has been subject to evolutionary pressure and therefore is likely to be critical to the protein, in terms of function, stability or fold. Mutations affecting such residues could therefore disrupt protein function, potentially causing disease. Where a mutation cannot be explained using *structural* analyses, the functional information implicit in alignments of FEPs may offer an explanation.

4.1.2 Scoring conservation

Conservation could simply be calculated as the fraction of sequences in the MSA that have the residue of interest at an equivalent position. However, protein function can be maintained if a residue with similar characteristics replaces the original. To take this into account, conservation

scoring methods use amino acid substitution matrices to calculate conservation (see Section 2.3.4). A typical calculation for residue x is shown below in Equation 4.1:

$$C(x) = \lambda \sum_{i}^{N} \sum_{j>i}^{N} M(s_i(x), s_j(x)) \tag{4.1}$$

where the MSA contains N sequences, $s_a(x)$ is the residue at position x in sequence a, $M(p, q)$ is the amino acid substitution value between residues p and q, and λ scales $C(x)$ between 0 and 1, that is, $\lambda = \frac{1}{N}$. In this chapter, M is always the PET91 matrix (Jones *et al.*, 1992), an update of the Dayhoff matrix, which has been normalised such that all scores on the diagonal are equal (see Section 2.3.4.3).

An alternative approach uses the concept of 'entropy' and is borrowed from information theory. The most commonly used entropy-based method is Shannon's entropy (Shannon, 1948), summarised in Equation 4.2:

$$H(X) = -\sum_{i=1}^{K} p_i \log_2 p_i \tag{4.2}$$

where $H(X)$ is the Shannon's entropy of a set of residues X; i denotes an amino acid; K is the number of amino acids (therefore, $K = 20$), and p_i is the fractional frequency of residue i (that is, $n_i/|X|$ where n_i is the number of i residues in X and $|X|$ is the length of X). A column containing one of each of the twenty amino acids (X_{20}) would score $H(X_{20}) = 4.32$, while a column containing twenty of the same amino acid (X_1) would score $H(X_1) = 0$, reflecting that there is less *information* contained in the X_1 residue set than in the X_{20}. This can be used to measure conservation, where low entropy indicates high conservation and high entropy indicates low conservation. Such methods, however, are unable to account for the similarity between amino acids and are therefore inappropriate for use here.

In 2002, Valdar published a comprehensive review of conservation scoring methods, which describes many methods including weighted scores and Shannon's entropy. One scoring system that Valdar describes is that of Sander and Schneider (1991). This system is assessed later in Section 4.2.1.

4.1.3 Identifying highly conserved residues

Although various methods exist to score conservation, few exist that identify 'highly conserved' residues. Two well known methods that are used to calculate conservation from MSAs, and aim to identify highly conserved residues (and therefore *functional* residues) are Rate4Site and SIFT. Rate4Site estimates the maximum likelihood rate at each position in the alignment, based on a hypothesized phylogeny (Pupko *et al.*, 2002). SIFT (which '**S**orts **I**ntolerant **F**rom **T**olerant mutations') is a popular sequence-based method for identifying deleterious mutations (Ng and Henikoff, 2001). It calculates a probability that an amino acid would be tolerated in the alignment, based on the observed variability and the estimated variability in a theoretical, complete alignment. Neither of these methods explicitly considers the alignment with respect to species coverage or with respect to global patterns of conservation, and further, neither method generates a threshold with which highly conserved residues can be identified.

4.1.4 Generating an improved protein alignment conservation threshold

As yet, no method exists that accounts for (i) species coverage and (ii) background conservation levels in the alignment, that is also amenable to automated, distributed processing as required for SAAPdb. In this chapter, a method is described for generating an **Im**proved **P**rotein **A**lignment **C**onservation **T**hreshold (ImPACT), that explicitly 'normalises' the effects of species coverage and models the distribution of conservation scores to allow a threshold for high conservation to be generated.

4.2 Methods

4.2.1 Accommodating the species set bias

Where species are very similar, some proportion of the conservation will be due to the small evolutionary distance between the species. To 'normalise' for the species set, it is necessary to reduce the influence of pairwise sequence comparisons between similar species. This can be achieved by adding a weighted component to the conservation score.

In their paper, Sander and Schneider (1991) (hereafter referred to as SS) present a method that deals with high levels of sequence similarity in protein alignments. When the conservation score is calculated, pairwise scores are moderated by the similarity (or, strictly, dissimilarity) of the two sequences. The calculation for column x of the MSA is shown in Equation 4.3.

$$C_{SS}(x) = \lambda \sum_{i=1}^{N} \sum_{j=i+1}^{N} d(s_i, s_j) m(s_i(x), s_j(x)) \tag{4.3}$$

$$\lambda = \left(\sum_{i=1}^{N} \sum_{j=i+1}^{N} d(s_i, s_j) \right)^{-1} \tag{4.4}$$

$$d(s_i, s_j) = 100 - \frac{1}{L} \sum_{k=1}^{L} \delta(R_{ki}, R_{kj}) \tag{4.5}$$

where $m(s_i(x), s_j(x))$ is the score from the Dayhoff substitution matrix (Dayhoff *et al.*, 1978) for the residues in column x in sequences s_i and s_j; $\delta(R_{ki}, R_{kj})$ is the identity {1,0} of R_{ki} and R_{kj} (the kth residue in the alignment of sequences i and j respectively); N is the number of sequences in the MSA; and L is the length of the alignment.

This goes some way to accommodating the species bias inherent in MSAs. However, it uses the similarity of sequences in the MSA to *approximate* species similarity. This will be misleading if sequences are highly conserved for functional reasons. Should two similar sequences between two very different species be compared as described by Equations 4.3-4.5, their contribution to the overall alignment will be downweighted rather than correctly upweighted. As such, a more rational and direct weighting system, which weights pairwise comparisons by the *species* similarity, has been developed here.

Thus, species similarity is calculated directly: the FEPs identified by FOSTA (see Chapter 3) are used to calculate how similar two species are, on average. The system of equations becomes:

$$C_{specsim}(x) = \lambda' \sum_{i=1}^{N} \sum_{j=i+1}^{N} d'(s_i, s_j) m'(s_i(x), s_j(x)) \tag{4.6}$$

$$\lambda' = \left(\sum_{i=1}^{N} \sum_{j=i+1}^{N} d'(Q(s_i), Q(s_j)) \right)^{-1} \tag{4.7}$$

$$d'(A, B) = 100 - \frac{1}{F} \sum_{i}^{F} nw(A_i, B_i) \tag{4.8}$$

where $m'(s_i(x), s_j(x))$ is the score from the normalised PET91 matrix (Jones *et al.*, 1992); A and B are two species for the residues in column x in sequences s_i and s_j; $Q(a)$ is the species from which sequence a is derived; F is the number of FEPs that are identified between A and B; A_i and B_i are the ith FEPs; and $nw(X, Y)$ is the NW aligment score calculated across the aligned regions of the protein sequences of X and Y (using a gap initialisation penalty of 11 and a gap extension penalty of 2, see Section 2.3.3 for a description of NW). $d'(A, B)$ is therefore a dissimilarity coefficient for A and B, scaled between 0 and 100. Using Equations 4.6-4.8, conservation scores are normalised with respect to species diversity. Hereafter, the resulting species similarity matrix will be referred to as `specsim` (**spec**ies **sim**ilarity).

To generate the matrix d', all observed species pairings from FOSTA (i.e., all species pairs which share membership of a FOSTA family) are extracted from the database. In the current implementation of FOSTA (November 2008), there are 6 801 254 protein pairs from 1 147 685 possible species pairings, requiring that 6 801 254 pairwise alignments are run; this is not feasible on a single machine. Further, each species pairing can be processed in parallel. As such, the code is ideal for distribution across processors and has been developed for execution across the local 116-core grid. The pseudocode for each species comparison (i.e., each distributed job) is shown in Figure 4.1.

4.2.2 Accommodating protein-specific patterns of conservation

Using the `specsim` scoring method described in Section 4.2.1 (Equations 4.6-4.8), conservation scores are normalised with respect to the species set represented. The next step is to isolate conservation patterns that are independent of the properties of the proteins.

To define an alignment-specific 'high conservation' threshold, it is necessary to characterise the distribution of conservation scores appropriately. As the distribution of conservation scores

(a) specsim inputs

- `species1` : the first species
- `species2` : the second species

(b) specsim processing

(I) identify all *good* pairs of FEPs for `species1` and `species2`

(II) for each protein pair:

  (a) extract the sequences from the database

  (b) align the sequences using NW, calculate and record the identity score

(III) calculate and record the mean of the identity scores

Figure 4.1: Calculating a similarity score for a pair of species
NW: Needleman & Wunsch (1970) using a gap initialisation penalty of 11 and gap extension penalty of 2. `species1` and `species2` are UniProtKB species suffices. *Good* FEPs are those that are (a) not fragments (b) not unreliable and (c) not assigned using the 'last-resort' sequence matching (see Section 3.2.2.3).

cannot be expected to conform to a single Gaussian distribution, standard parametric, statistical approaches are not suitable. Non-parametric ranking methods (e.g., identifying the top 5% of conservation scores) are also inappropriate as MSAs will vary with respect to the proportion of columns that are highly conserved.

Mixture models allow distributions to be described using *multiple* Gaussian components (Aitkin and Wilson, 1980). For the purposes of the current analysis, the distribution is characterised as having three components, G_0, G_1 and G_2: G_0 will characterise the unconserved residues, G_1 will capture the distribution of moderately conserved residues, and G_2 will describe the distribution of the highly conserved residues. It is expected that these three classes have some functional relevance: G_0 describes freely mutating residues, G_1 describes residues with some minor structural role, while G_2 describes residues that are critical to structure and/or function. Thus, residues defined by G_2 are the ones that should be identified as 'highly conserved'.

Figure 4.2 shows the mixture modelling of some example data. These data were randomly generated using the `rnorm()` function in `R`: 3000 points were drawn evenly from three Gaussian distributions at $\mu = \{-5, 0, 5\}, \sigma = \{2, 5, 4\}$. Although the Gaussians overlap considerably, the data are appropriately captured by the three Gaussians: the means of the fitted Gaussians (shown by the dashed blue lines) correspond closely to the true values $\{-5, 0, 5\}$; compare the raw data, plotted in black, with the cumulative model in red to evaluate how well the model fits the data.

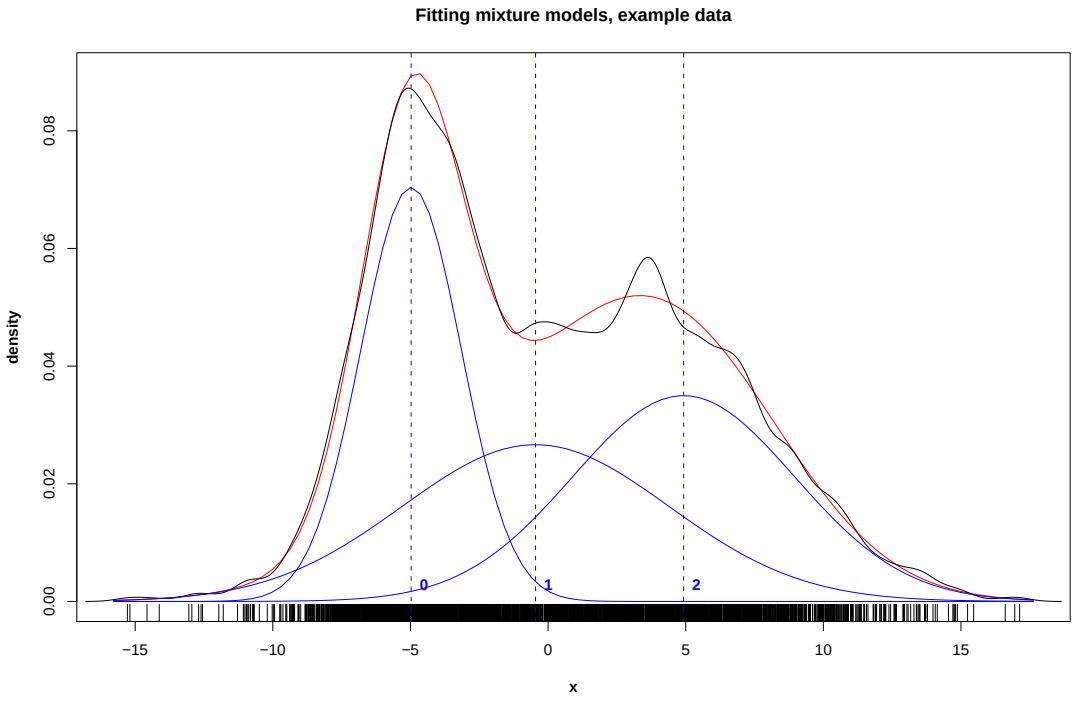
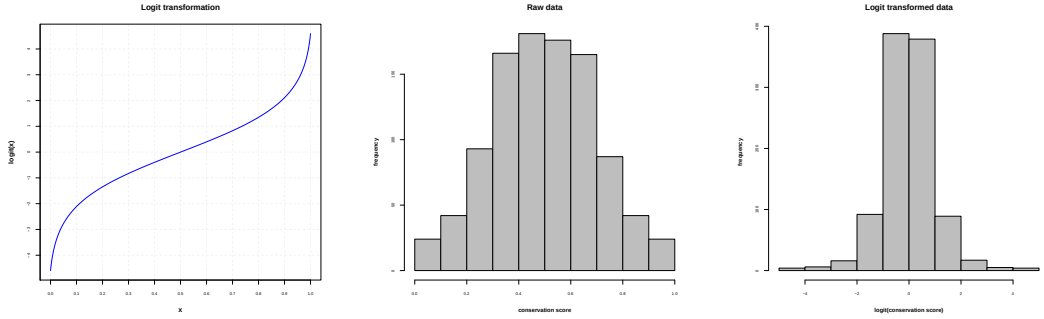


Figure 4.2: Using multiple Gaussians to model data

The data modelled above has been generated randomly using `rnorm()` in R. 3000 numbers were drawn evenly from three Gaussian distributions: distribution A ($\mu = -5, \sigma = 2$), distribution B ($\mu = 0, \sigma = 5$), distribution C ($\mu = 5, \sigma = 4$). The black line represents the raw data. A 3-component mixture model is fitted to these data and is shown in blue. Fitted components are numbered from 0 to 2, from left to right; the means of these components are indicated with dashed blue lines. The cumulative modelled distribution is shown in red. For details on the optimisation method (BFGS), see Section 4.2.2.



(a) The logit transformation (b) Raw data (mean=0.5, SD=0.2) (c) Logit transformed data

Figure 4.3: The logit function and logit transformation

To ease discrimination at the extremeties of the distribution, the data are logit transformed (see Figure 4.3). The formula is given in Equation 4.9 below.

$$x_{logit} = \log\left(\frac{x}{1-x}\right) \tag{4.9}$$

The three Gaussian components G_0, G_1 and G_2 are fitted to the logit transformed data and two constraints are applied to the resulting model to define high conservation in the MSA. The mixture model is fitted using the `optim()` command in R, using the BFGS method (Broyden-Fletcher-Goldfarb-Shanno) (Fletcher, 1970). ImPACT has been implemented to run fifty rounds of optimisation and to draw priors from the uniform Dirichlet conjugate prior ($\alpha = 1$). The Dirichlet distribution is used to represent any prior belief with regards to the values of the parameters of the mixture model. Using a uniform prior where $\alpha = 1$ in the context of the ImPACT mixture model means that no prior belief as to the densities of the three component Gaussians exists in the modelling system.

4.2.2.1 Constraint one: applying a basic concept of conservation

The method must first ascertain whether the MSA contains any significant conservation at all. In the context of the modelled distribution described above, this translates to considering whether μ_{G_2} is high enough. If G_2 exists at, for example, $\simeq 0.60$, it may be high relative to

the other residues in the protein, but it is not high enough to infer functional significance. To define a basic concept of conservation, the lower bound of 'high conservation' must be defined. This value will be described as constraint one, or $C1$.

To establish a suitable value for $C1$, the conservation scores for combinations of eight residue duples or triples were considered. That is, sets of two or three residues were combined to represent alignment columns, with residues being chosen based on an existing expectation of what would constitute conservation and what would not. By considering the conservation for each combination and considering which sets of columns should be classified as conserved, the value for $C1$ can be defined. For example, the combination of residues CCCCCCWWWW would not be considered conserved, as cysteine and tryptophan are very different, but the combination of residues FFFFFFYYYY would be, as phenylalanine and tyrosine are much more similar physicochemically.

Of the set of eight residue pairs, two residue tuples were included as examples of very different residues—{EW} and {CW}—with the remaining six—{ILV}, {ST}, {DE}, {RK}, {FY} and {NQ}—representative of sets of more similar residues. The intention here is to set the value for $C1$ such that appropriate mixtures of the similar residues are included, while excluding appropriate mixtures of the dissimilar residues.

In this section, the term 'X:Z ratio' will be used to describe the composition of the combinations of residues (X and Z are used so as to avoid confusion with amino acids). A high X:Z ratio indicates that the combination predominantly comprises one residue (e.g., 'FFFFFFFFFFY' has an X:Z ratio of 11:1 where X represents F and Z represents Y), whereas an X:Z ratio close to 1:1 indicates that there are equal numbers of each/all residues in the combination (e.g., 'SSSSSSTTTTTT' has an X:Z ratio of 1:1 where X represents S and Z represents T). For each residue combination, all possible combinations of 12 residues were generated. 12 was used because (a) it allows a 1:1 residue composition in a residue duple and a 1:1:1 residue composition in a residue triple (b) it allows a reasonable range of X:Z ratios, from 11:1 to 1:1 for duples and 10:1:1 to 1:1:1 for triples, and (c) it provides enough residues from which a reliable conservation score can be calculated.

Conservation scores were generated as described in Equation 4.1 using amino acid substitution scores from the normalised PET91 matrix (Jones *et al.*, 1992). The conservation scores for {ILV}, {ST}, {DE}, {RK}, {NQ}, {FY}, {EW} and {CW} are shown in Tables 4.1-4.8.

Table 4.1: The conservation scores for all combinations of twelve {ILV} residues
Column residues: the combination of residues; **Conservation**: the conservation score for the **column residues**. Thresholds of 0.75, 0.80, 0.85, 0.90 and 0.95 are marked on the right hand side. Columns are ordered by decreasing conservation score. {ILV} are a set of similar residues.

Column residues	Conservation	Threshold
LLLLLLLLLLLL	1.000000	
IIIIIIIIIIII	1.000000	
VVVVVVVVVVVV	1.000000	**0.95**
IVVVVVVVVVVV	0.933333	
IIIIIIIIIIIV	0.933333	
LLLLLLLLLLLV	0.911111	
ILLLLLLLLLLL	0.911111	
IIIIIIIIIIIL	0.911111	
LVVVVVVVVVVV	0.911111	**0.90**
IIVVVVVVVVVV	0.878788	
IIIIIIIIIIVV	0.878788	
IIIIIIIIIILV	0.850505	
ILVVVVVVVVVV	0.850505	**0.85**
IILLLLLLLLLL	0.838384	
LLLLLLLLLLVV	0.838384	
LLVVVVVVVVVV	0.838384	
IIIIIIIIIILL	0.838384	
IIIIIIIIIVVV	0.836364	
IIIVVVVVVVVV	0.836364	
ILLLLLLLLLLV	0.832323	
IIIIVVVVVVVV	0.806061	
IIIIIIIIVVVV	0.806061	
IILVVVVVVVVV	0.802020	
IIIIIIIIILVV	0.802020	**0.80**
IIIIIIIVVVVV	0.787879	
IIIIIVVVVVVV	0.787879	
IIIIIIIIILLV	0.783838	
ILLVVVVVVVVV	0.783838	
IIILLLLLLLLL	0.781818	
LLLVVVVVVVVV	0.781818	
LLLLLLLLLVVV	0.781818	
IIIIIIIIILLL	0.781818	
IIIIIIVVVVVV	0.781818	
IILLLLLLLLLV	0.769697	
ILLLLLLLLLVV	0.769697	
IIILVVVVVVVV	0.765657	
IIIIIIIILVVV	0.765657	**0.75**
IIIIIIIILLVV	0.741414	
IIIIIIILVVVV	0.741414	
IIIILLLLLLLL	0.741414	
LLLLLLLLVVVV	0.741414	
IIIIIIIILLLL	0.741414	
IILLVVVVVVVV	0.741414	
IIIILVVVVVVV	0.741414	
LLLLVVVVVVVV	0.741414	

continues next column...

...continues here

ILLLVVVVVVVV	0.733333
IIIIIIIILLLV	0.733333
IIIIILVVVVVV	0.729293
IIIIIILVVVVV	0.729293
ILLLLLLLLVVV	0.723232
IIILLLLLLLLV	0.723232
IIIIIIILLLLL	0.717172
LLLLLLLVVVVV	0.717172
IILLLLLLLLVV	0.717172
IIIIILLLLLLL	0.717172
LLLLLVVVVVVV	0.717172
IIIIIIILLVVV	0.711111
IIILLVVVVVVV	0.711111
LLLLLLVVVVVV	0.709091
IIIIIILLLLLL	0.709091
ILLLLVVVVVVV	0.698990
IIIIIIILLLLV	0.698990
IIIIIIILLLVV	0.696970
IILLLVVVVVVV	0.696970
ILLLLLLLVVVV	0.692929
IIIIIILLVVVV	0.692929
IIIILLLLLLLV	0.692929
IIIILLVVVVVV	0.692929
IIIIILLVVVVV	0.686869
IIIIIILLLLLV	0.680808
IIILLLLLLLVV	0.680808
ILLLLLVVVVVV	0.680808
IILLLLLLLVVV	0.680808
IIIIIILLLLLV	0.678788
ILLLLLLVVVVV	0.678788
IIILLLVVVVVV	0.672727
IIIIIILLLVVV	0.672727
IILLLLVVVVVV	0.668687
IIIIIILLLLVV	0.668687
IILLLLLLVVVV	0.660606
IIIIILLLVVVV	0.660606
IIIILLLVVVVV	0.660606
IIIILLLLLLVV	0.660606
IILLLLLVVVVV	0.656566
IIIIILLLLLVV	0.656566
IIILLLLLLVVV	0.654545
IIIIILLLLVVV	0.650505
IIILLLLVVVVV	0.650505
IIILLLLLVVVV	0.644444
IIIILLLLVVVV	0.644444
IIIILLLLLVVV	0.644444

Table 4.2: The conservation scores for combinations of twelve {ST} residues
Column residues: the combination of residues; **Conservation**: the conservation score for the **column residues**. Thresholds of 0.75, 0.80, 0.85, 0.90 and 0.95 are marked on the right hand side. Columns are ordered by decreasing conservation score. {ST} are a set of similar residues.

Column residues	Conservation	Threshold
SSSSSSSSSSS	1.000000	
TTTTTTTTTTT	1.000000	**0.95**
STTTTTTTTTT	0.900000	
SSSSSSSSSST	0.900000	**0.85/0.90**
SSSSSSSSSTT	0.818182	
SSTTTTTTTTT	0.818182	**0.80**
SSSSSSSSTTT	0.754545	
SSSTTTTTTTT	0.754545	**0.75**
SSSSSSSTTTT	0.709091	
SSSSTTTTTTT	0.709091	
SSSSSSTTTTT	0.681818	
SSSSSTTTTTT	0.681818	
SSSSSSTTTTT	0.672727	

Table 4.3: The conservation scores for combinations of twelve {DE} residues
Column residues: the combination of residues; **Conservation**: the conservation score for the **column residues**. Thresholds of 0.75, 0.80, 0.85, 0.90 and 0.95 are marked on the right hand side. Volumns are ordered by decreasing conservation score. {DE} are a set of similar residues.

Column residues	Conservation	Threshold
DDDDDDDDDDD	1.000000	
EEEEEEEEEEE	1.000000	**0.95**
DEEEEEEEEEE	0.933333	
DDDDDDDDDDE	0.933333	**0.90**
DDEEEEEEEEE	0.878788	
DDDDDDDDDEE	0.878788	**0.85**
DDDDDDDDEEE	0.836364	
DDDEEEEEEEE	0.836364	
DDDDEEEEEEE	0.806061	
DDDDDDDEEEE	0.806061	**0.80**
DDDDEEEEEEE	0.787879	
DDDDDDEEEEE	0.787879	
DDDDDEEEEEE	0.781818	**0.75**

Table 4.4: The conservation scores for combinations of twelve {RK} residues
Column residues: the combination of residues; **Conservation**: the conservation score for the **column residues**. Thresholds of 0.75, 0.80, 0.85, 0.90 and 0.95 are marked on the right hand side. Columns are ordered by decreasing conservation score. {RK} are a set of similar residues.

Column residues	Conservation	Threshold
RRRRRRRRRRRR	1.000000	
KKKKKKKKKKKK	1.000000	**0.95**
KRRRRRRRRRRR	0.933333	
KKKKKKKKKKKR	0.933333	**0.90**
KKRRRRRRRRRR	0.878788	
KKKKKKKKKKRR	0.878788	**0.85**
KKKKKKKKKRRR	0.836364	
KKRRRRRRRRRR	0.836364	
KKKKKKKKRRRR	0.806061	
KKKRRRRRRRRR	0.806061	**0.80**
KKKKKKKRRRRR	0.787879	
KKKKKRRRRRRR	0.787879	
KKKKKKRRRRRR	0.781818	**0.75**

Table 4.5: The conservation scores for combinations of twelve {NQ} residues
Column residues: the combination of residues; **Conservation**: the conservation score for the **column residues**. Thresholds of 0.75, 0.80, 0.85, 0.90 and 0.95 are marked on the right hand side. Columns are ordered by decreasing conservation score. {NQ} are a set of similar residues.

Column residues	Conservation	Threshold
QQQQQQQQQQQQ	1.000000	
NNNNNNNNNNNN	1.000000	**0.90/0.95**
NNNNNNNNNNNQ	0.888889	
NQQQQQQQQQQQ	0.888889	**0.80/0.85**
NNQQQQQQQQQQ	0.797980	
NNNNNNNNNNQQ	0.797980	**0.75**
NNNQQQQQQQQQ	0.727273	
NNNNNNNNNQQQ	0.727273	
NNNNQQQQQQQQ	0.676768	
NNNNNNNNQQQQ	0.676768	
NNNNNNNQQQQQ	0.646465	
NNNNNQQQQQQQ	0.646465	
NNNNNNQQQQQQ	0.636364	

Table 4.6: The conservation scores for combinations of twelve {FY} residues
Column residues: the combination of residues; **Conservation**: the conservation score for the **column residues**. Thresholds of 0.75, 0.80, 0.85, 0.90 and 0.95 are marked on the right hand side. Columns are ordered by decreasing conservation score. {FY} are a set of similar residues.

Column residues	Conservation	Threshold
YYYYYYYYYYYY	1.000000	
FFFFFFFFFFFF	1.000000	**0.95**
FFFFFFFFFFFY	0.944444	
FYYYYYYYYYYY	0.944444	**0.90**
FFFFFFFFFFYY	0.898990	
FFYYYYYYYYYY	0.898990	
FFFYYYYYYYYY	0.863636	
FFFFFFFFFYYY	0.863636	**0.85**
FFFFYYYYYYYY	0.838384	
FFFFFFFFYYYY	0.838384	
FFFFFYYYYYYY	0.823232	
FFFFFFFYYYYY	0.823232	
FFFFFFYYYYYY	0.818182	**0.75/0.80**

Table 4.7: The conservation scores for combinations of twelve {EW} residues
Column residues: the combination of residues; **Conservation**: the conservation score for the **column residues**. Thresholds of 0.75, 0.80, 0.85, 0.90 and 0.95 are marked on the right hand side. Columns are ordered by decreasing conservation score. {EW} are a set of very different residues.

Column residues	Conservation	Threshold
EEEEEEEEEEEE	1.000000	
WWWWWWWWWWWW	1.000000	**0.85/0.90/0.95**
EEEEEEEEEEEW	0.833333	
EWWWWWWWWWWW	0.833333	**0.75/0.80**
EEWWWWWWWWWW	0.696970	
EEEEEEEEEEWW	0.696970	
EEEWWWWWWWWW	0.590909	
EEEEEEEEEWWW	0.590909	
EEEEEEEEWWWW	0.515152	
EEEEWWWWWWWW	0.515152	
EEEEEEEWWWWW	0.469697	
EEEEEWWWWWWW	0.469697	
EEEEEEWWWWWW	0.454545	

Table 4.8: The conservation scores for combinations of twelve {CW} residues
Column residues: the combination of residues; **Conservation**: the conservation score for the **column residues**. Thresholds of 0.75, 0.80, 0.85, 0.90 and 0.95 are marked on the right hand side. Columns are ordered by decreasing conservation score. {CW} are a set of very different residues.

Column residues	Conservation	Threshold
CCCCCCCCCCCC	1.000000	
WWWWWWWWWWWW	1.000000	**0.95**
CCCCCCCCCCCW	0.900000	
CWWWWWWWWWWW	0.900000	**0.85/0.90**
CCCCCCCCCCWW	0.818182	
CCWWWWWWWWWW	0.818182	**0.80**
CCCWWWWWWWWW	0.754545	
CCCCCCCCCWWW	0.754545	**0.75**
CCCCWWWWWWWW	0.709091	
CCCCCCCCWWWW	0.709091	
CCCCCCCWWWWW	0.681818	
CCCCCWWWWWWW	0.681818	
CCCCCCWWWWWW	0.672727	

A threshold of 0.80 identifies *any* combination of twelve tyrosine and phenylalanine residues as being conserved (Table 4.6); increasing the threshold to 0.85 only identifies columns with an X:Z ratio of 3:1 or higher. Unless the hydroxyl group of tyrosine is critical, tyrosine and phenylalanine can generally replace each other without compromising protein function, suggesting that a threshold of 0.80 is appropriate for this pair of amino acids. The threshold of 0.80 also seems appropriate for the {DE} and {RK} combinations: both pairs of residues generate the same conservation scores, and applying a threshold of 0.80 allows columns with an X:Z ratio of 2:1 or better to be identified as highly conserved.

The remaining sets of similar residues—{ILV} (Table 4.1), {ST} (Table 4.2) and {NQ} (Table 4.5)—may suggest that a lower threshold of 0.75 is more appropriate, allowing for columns such as 'IILLLLLLLLLV', 'SSSSSSSSSTTT' and 'NNNNNNNNNNQQ' to be identified as highly conserved. However, dropping the threshold to 0.75 would compromise performance where very different residues are being compared. For example, a threshold of 0.75 would identify an alignment column comprising of 'CCCCCCCCCWWW' or 'CCCWWWWWWWWW' as highly conserved.

With a view to applying a conservative threshold, an initial criteria that $\mu_{G_2} \geq logit(0.80)$ is defined.

4.2.2.2 Constraint two: generating the threshold

If $C1$ is not violated (i.e., the predefined, basic model of conservation exists in the data), the method then assesses whether the density at the high end of the distribution is sufficiently discrete from that of the middle of the distribution. To do this, it must assess how *separate* G_1 and G_2 are (e.g., in Figure 4.2, the distance between the second and third vertical dashed lines would be used to assess whether the highest distribution (distribution 2) is distant from the middle distribution (distribution 1)). Using the parameters of G_1, which models the distribution of moderately conserved residues, it is possible to identify those residues that exist at the upper extreme of what constitutes 'moderately conserved': by moving two standard deviations in the positive direction, the top ∼2.5% of the moderately conserved data is identified. If μ_{G_2} (the Gaussian which represents highly conserved residues) exists at this point or higher, the two Gaussians (G_1 and G_2) are considered to be separate; that is, if $\mu_{G_2} >= \mu_{G_1} + C2 * \sigma_{G_1}; C2 = 2$, there is adequate distance between G_1 and G_2, and $\mu_{G_1} + C2 * \sigma_{G_1}; C2 = 2$ becomes the threshold for high conservation. Otherwise, the basic concept of conservation is applied (i.e., $C1$ becomes the threshold).

Constraint two ($C2$) defines how far from μ_{G_2} (in standard deviations) the threshold should be set. The ImPACT threshold (I_T) is therefore calculated as follows:

$$I_T = \begin{cases} \mu_{G_1} + C2 * \sigma_{G_1} & \text{if } \mu_{G_2} >= \mu_{G_1} + C2 * \sigma_{G_1}, \\ C1 & otherwise. \end{cases} \tag{4.10}$$

By default, $C1 = 0.80$ and $C2 = 2$. Note that the threshold for conservation can be < 0.80. $C1$ assesses whether the mean of G_2 (the Gaussian representing the *conserved* residues) is greater than 0.80. If this constraint is met, the threshold is calculated from the mean and standard deviation of G_1 (the Gaussian representing the *moderately conserved* residues). As such, the threshold generated by the calculation $\mu_{G_1} + 2\sigma_{G_1}$ could be less than 0.80.

4.3 Results and Discussion

It is difficult to assess how successful conservation scoring methods are. In his comprehensive summary of protein conservation scoring methods, Valdar (2002) states that "*there is no rigourous mathematical test for judging a conservation measure...rather than accuracy then, a conservation score may be judged on its verisimilitude: its ability to depict realism and its concordance with biochemical notation*". However, further to scoring conservation, ImPACT generates a *threshold*, the application of which aims to classify residues in an MSA as highly conserved or not highly conserved. As such, should an appropriate dataset of conserved residues exist against which ImPACT can be benchmarked, it is possible to use standard binary classification performance statistics (see Section 2.3.5) to evaluate the performance of ImPACT.

To perform a multi-faceted evaluation of ImPACT, it has been assessed using three datasets. The first is a dataset of four representative human proteins: glucose-6-phosphate 1-dehydrogenase (G6PD) [UniProtKB:P11413/G6PD_HUMAN]; ornithine carbamoyltransferase (OTC) [UniProtKB:P00480/OTC_HUMAN]; cellular tumor antigen P53 [UniProtKB:P04637/P53_HUMAN] and haemoglobin subunit beta (HBB) [UniProtKB:P68871/HBB_HUMAN]. Secondly, the sequence motif database PROSITE (Hulo *et al.*, 2006) is parsed to extract residues that should be conserved in an alignment of functionally equivalent proteins. Finally, artificial conservation data, for which the global conservation patterns can be controlled, are used to evaluate the scoring method.

First, however, the `specsim` weighted scoring system is evaluated.

4.3.1 Normalising conservation using the `specsim` matrix

Figure 4.4 shows an unrooted phylogenetic tree (constructed using the `fitch` method from the PHYLIP[1] package of phylogenetic software) generated from a subset of fifty species' pairwise dissimilarity scores calculated from the `specsim` matrix. It is clear that the dissimilarity scores are representative of species diversity: the two main branches of the tree represent the eukaryotes and prokaryotes. There is appropriate subdivision of eukaryotic species into mammal, fly, yeast, fungi and plant groups, and appropriate prokaryotic subdivisions including cynobateria

[1] `http://evolution.genetics.washington.edu/phylip.html`

(including the Nostocales subfamily), mycobacteria, E-coli subspecies, helicobacteria and so on.

Figure 4.5 compares the unweighted, species-similarity-weighted and sequence-similarity-weighted conservation scores for the four representative proteins introduced above. HBB and P53 are the two most heavily adjusted proteins in terms of weighted conservation scores, while G6PD and OTC are less affected. This indicates that the species similarity, as measured by `specsim`, and the sequence similarity are both high within the P53 and HBB alignments. The results for G6PD and OTC are clustered more closely around the identity line, demonstrating that the proteins are more different from each other, with respect to species and sequence.

For all four proteins, the sequence and species weighting do correspond closely to each other, indicating that sequence similarity is a reasonable approximation of the species similarity. In general, however, it does appear that weighting by species similarity has a greater effect on the score than sequence similarity, with blue points closer to the x axis than red points. The exception is OTC, the conservation scores for which appear to be more heavily corrected when weighting by sequence similarity than when weighting by species similarity.

4.3.2 Using four representative proteins to assess ImPACT

Four representative human proteins have been chosen to evaluate ImPACT (see Table 4.9). **G6PD** (515 residues long, 49 proteins in alignment) is representative of a protein that is generally not highly conserved. It has a low mean conservation score of 0.58 and a relatively large standard deviation of 0.22. Only 5.05% of the residues are 100% conserved in an alignment with 515 columns. **OTC** (354 residues long, with 178 proteins in the alignment) has a very similar conservation profile to that of G6PD, with a mean conservation score of 0.59, a standard deviation of 0.20 and 3.39% of the 354 residues 100% conserved. The mean conservation score of the tumour suppressor protein **P53** is 0.68; P53 displays the most amount of variation around the mean, with a standard deviation of 0.24 and 21.63% of residues 100% conserved. **HBB** has the highest mean score (0.74) and with less variation ($\sigma = 0.18$) about the mean and 6.12% of residues 100% conserved. P53 and HBB are examples of highly conserved proteins. HBB has higher mean conservation value (0.74 compared to 0.69), but far fewer residues that are 100% conserved (6.12% compared to 21.63%). As such, it is possible to describe HBB as the *globally* more conserved protein although P53 has more 100% conserved residues. The aim of ImPACT is to identify high or low global conservation and generate appropriately higher or

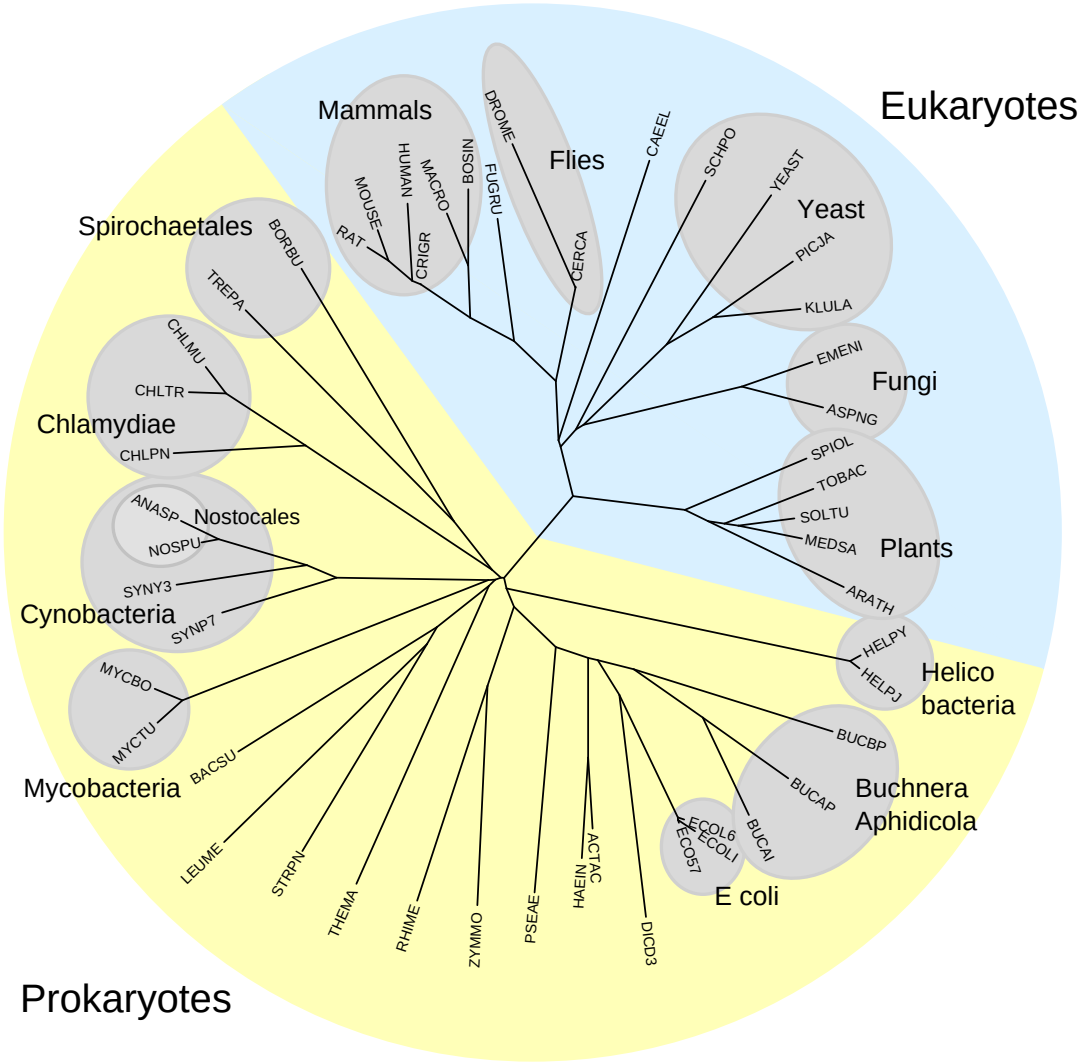
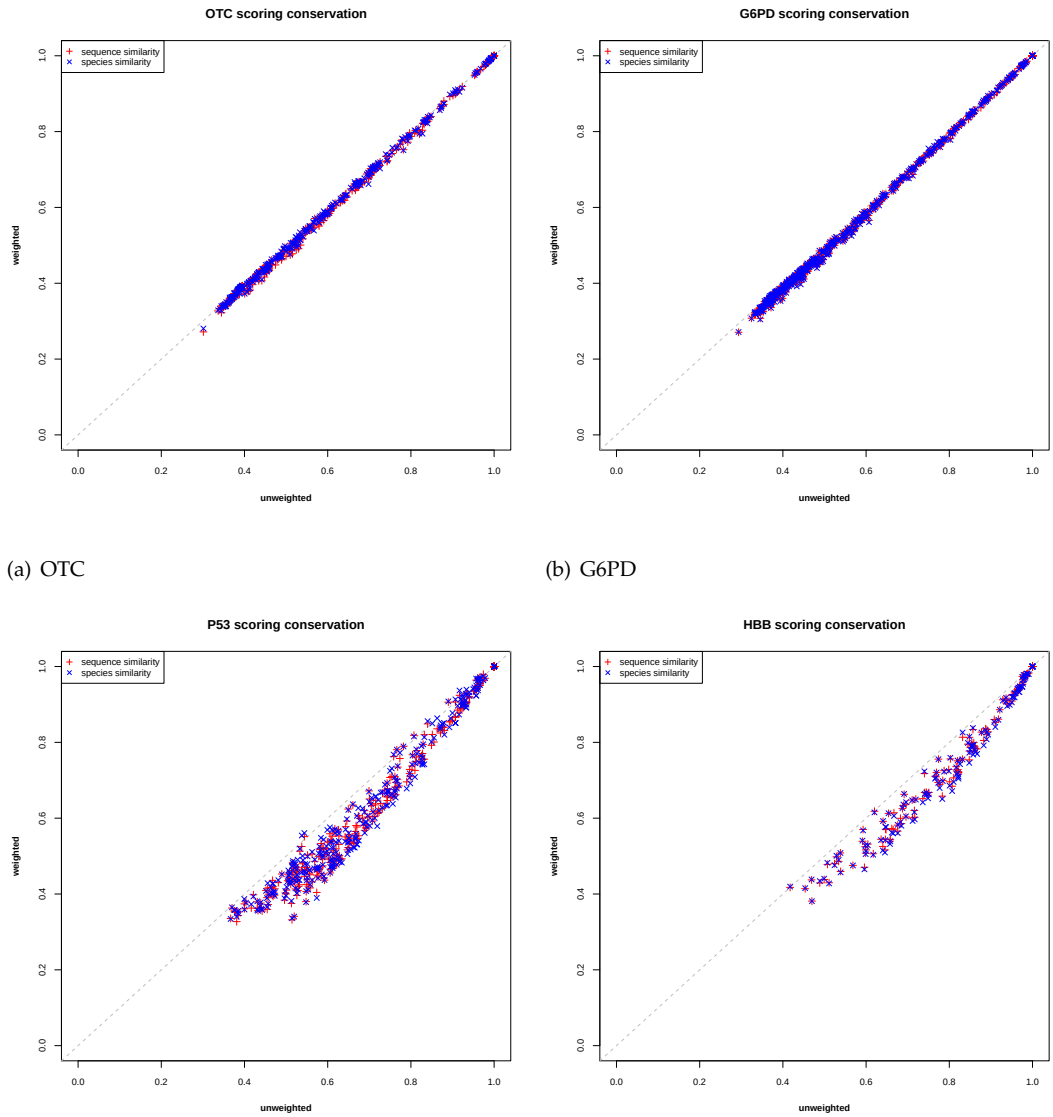


Figure 4.4: Evaluating the `specsim` matrix using phylogeny

Dissimilarity scores between 50 species were taken from the `specsim` matrix (see Section 4.2.1) and used as distance metrics with which to construct a phylogenetic tree (using `fitch` from the `Phylip` package). Species represented using their UniProtKB/Swiss-Prot species suffix. The first branch of the tree divides the species into eukaryotes (highlighted in blue) and prokaryotes (highlighted in yellow). Further subdivisions are highlighted in grey.



(a) OTC

(b) G6PD

(c) P53

(d) HBB

Figure 4.5: Comparing `specsim`, SS and unweighted conservation scoring methods

The four representative proteins (OTC, G6PD, P53 and HBB) are used to evaluate the sequence similarity weighted (SS, see Equations 4.3-4.5), species similarity weighted (`specsim`, see Equations 4.6-4.8) and unweighted conservation scoring methods. The unweighted score is along the x axis and the weighted scores are plotted along the y axis, with scores weighted by species plotted as blue $+$s and scores weighted by sequence plotted as red $\times$s. The identity line (where the weighted and unweighted scores are equal) is marked by a dashed grey line.

Table 4.9: Conservation patterns vary across proteins
n_{res}: the number of residues/columns in the alignment; n_{seq}: the number of sequences/rows in the alignment; max: the maximum conservation score; min: the minimum conservation score; μ_c: the average conservation score; σ_c: the standard deviation of the conservation scores; $perc_x$: the percentage of columns that have a conservation score of x. All numbers rounded to 2dp.

Protein	n_{res}	n_{seq}	max	min	μ_c	σ_c	$perc_{1.00}$	$perc_{0.95}$	$perc_{0.90}$
OTC	354	178	1.00	0.28	0.59	0.20	3.39	10.17	12.43
G6PD	515	49	1.00	0.27	0.58	0.22	5.05	9.32	13.59
P53	393	31	1.00	0.33	0.69	0.23	21.63	24.68	30.79
HBB	147	239	1.00	0.38	0.74	0.17	6.12	14.29	27.21

Table 4.10: ImPACT results for the four representative proteins
n_{res}: the number of residues/columns in the alignment; n_{seq}: the number of sequences/rows in the alignment; I_T: the threshold generated using ImPACT; $perc_{I_T}$: the percentage of residues identified as highly conserved, according to I_T. I_T rounded to 4dp, all other figures rounded to 2dp.

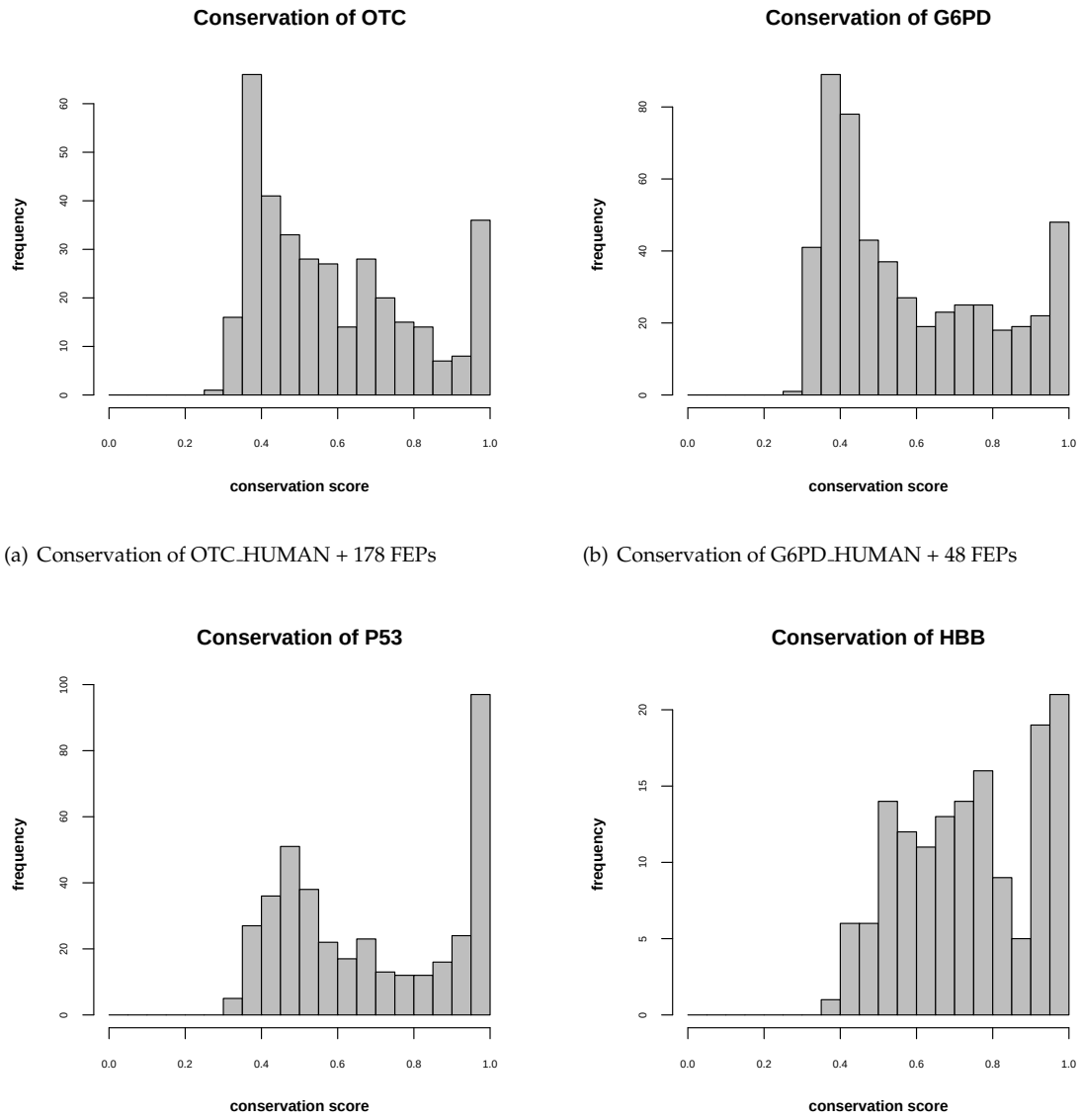
Protein	n_{res}	n_{seq}	I_T	$perc_{I_T}$
OTC	354	178	0.8672	13.27
G6PD	515	49	0.9612	8.54
P53	393	31	0.9636	23.66
HBB	147	239	0.9763	10.20

lower scores. All proteins have at least one residue that is 100% conserved.

The graphs in Figure 4.6 plot the distribution of `specsim` -weighted (Section 4.2.1) conservation scores for each protein and its FEPs (as defined by FOSTA, see Chapter 3) as a proportion of the total number of residues (i.e., the data are normalised for the sequence length, n_{seq} in Table 4.9). It is clear that the four proteins vary with respect to conservation score distribution and as such are appropriate datasets with which ImPACT can be assessed.

Figure 4.7 shows the logit transformed distributions as modelled by ImPACT and the results of the ImPACT analysis. Visual inspection of the fitted mixture models shows that the distribution of raw, `specsim` -weighted conservation scores (shown in Figure 4.7 in black) is successfully captured using the three Gaussians in all four proteins (compare the black and magenta traces).

The corresponding alignment representation on the right of each graph in Figure 4.7 depicts the global conservation patters for each protein, where conservation increases from green (where



(a) Conservation of OTC_HUMAN + 178 FEPs

(b) Conservation of G6PD_HUMAN + 48 FEPs

(c) Conservation of P53_HUMAN + 31 FEPs

(d) Conservation of HBB_HUMAN + 239 FEPs

Figure 4.6: Varying conservation patterns in the four representative proteins

The `specsim` -weighted conservation scores for the four representative proteins (OTC, G6PD, P53 and HBB). See Table 4.9 for more more details.

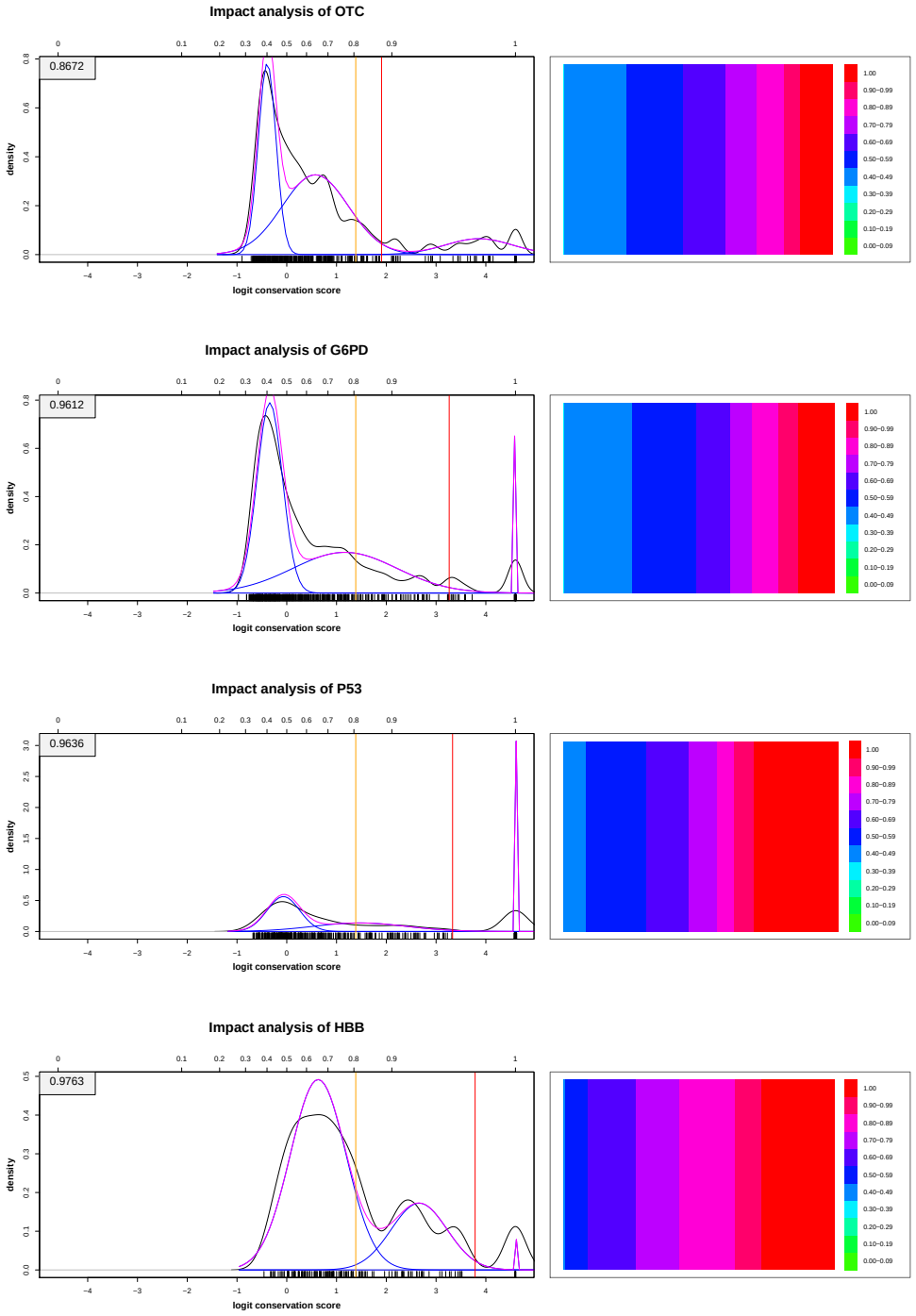


Figure 4.7: Assessing ImPACT using four representative proteins

ImPACT analysis for (from top to bottom) OTC, G6PD, P53, and HBB, the four representative proteins. Distribution of logit transformed raw data is shown in black, the three fitted Gaussians are shown in blue and the cumulative model is shown in magenta. The values for the first and second constraints (see Section 4.2.2) are depicted as vertical orange and red lines respectively. The resulting threshold is given in the grey box in the top-left corner of the graph. In addition to the fitted Gaussian components, to aid comparison of each protein's global conservation trends, a depiction of conservation trends for each protein's MSA is shown to the right of the graph, where high conservation scores are shown in red, more moderate scores are shown in blue and low scores are shown in green.

the conservation score is 0) to red (where the conservation score is 1) via blue and purple. The width of the colour band indicates how many columns in the alignment are scored in that range. Therefore, an alignment representation that is predominantly red is more highly conserved than an alignment representation containing large blue bands. Using these alignment representations to sort the four proteins according to increasing global conservation—taking into account *both* the prevalence of high conservation *and* the lack of very low conservation—would order the proteins as follows: OTC, G6PD, P53, HBB. Appropriately, ImPACT assigns these proteins increasing ImPACT scores: 0.8672, 0.9612, 0.9636 and 0.9763 for OTC, G6PD, P53 and HBB respectively. As discussed at the beginning of this section, although P53 has a higher proportion of 100% conserved residues, HBB has a higher mean conservation score and as such is the more *globally* conserved protein. HBB should therefore have a higher conservation threshold. ImPACT successfully captures these trends and generates a higher threshold for HBB than for P53.

4.3.3 A large scale analysis of ImPACT: PROSITE

PROSITE is a databank of biologically relevant protein motifs (Hulo *et al.*, 2006). Motifs are identified using structural alignments of proteins and several levels of profile extraction, refinement and scaling methods (for full details, see Gribskov *et al.* (1990), Lüthy *et al.* (1994), Thompson *et al.* (1994b)). Motifs in PROSITE can be both functional (e.g., an N-linked glycosolation site) and indicative of homologous protein families (e.g., an apple domain).

4.3.3.1 Defining the data

PROSITE (version 20.36, 02/09/08) data were obtained from Expasy[2]. This version of PROSITE contains 1315 PROSITE motifs. The motifs are described using a PROSITE-specific format: where each element is separated by a '-'; standard single-letter symbols are used to represent amino acids; 'x' indicates any residue; '[]'s indicate an inclusive choice (i.e., any residue in the '[]' brackets); '{}'s indicate an exclusive choice (i.e., any residue *except* those in the '{}' brackets); < and > indicate the start and end of the sequence respectively; and numbers or ranges in '()'s are quantifiers. For example, the AP endonuclease family 1 signature 1 (PROSITE family PS00726) describes the sequence motif `[APF]-D-[LIVMF](2)-{T}-[LIVM]-Q-E-{G}-K`,

[2] `ftp://ftp.expasy.org/databases/prosite/prosite.dat`

which translates as "ala or pro or phe / asp / leu or ile or val or met or phe / leu or ile or val or met or phe / any residue except thr / leu or ile or val or met / gln / glu / anything but gly / lys".

In this chapter, PROSITE is being used to define 'conserved' residues in human proteins; this requires that 'conservation' must be defined within PROSITE patterns. This is a question of leniency: how many different residues can exist in equivalent positions before the position becomes unconserved? For the results described in this chapter, the leniency has been set at two; that is, those elements in the PROSITE sequence motifs that describe a set of at most two amino acids are considered to be conserved. Although this may be viewed as a rather strict definition of conservation, ImPACT has been designed to be conservative in its approach and it is therefore undesirable to extend the leniency any further. These are the data against which ImPACT has been benchmarked.

Figure 4.8 describes how PROSITE is parsed to identify conserved residues in human proteins. Each PROSITE pattern (PA) record is extracted and the PROSITE formatted motif is translated into a Perl regular expression (REGEX). An example is shown in Figure 4.9(a). As described above, a leniency of two has been set to identify conserved residues, which results in the fourth, sixth, eighth, ninth and fourteenth elements in the example being described as conserved as well as the first, second, fifth and seventh single residue elements. To illustrate the use of leniency, the conserved elements using leniencies {1,2,3} are indicated with an asterisk beneath the aligned PA and REGEX in Figure 4.9(a).

Then, each TP (true positive) human protein[3] named in the PROSITE record is searched using the Perl regular expression to identify where the motif occurs (the sequence is searched for exhaustively, returning all occurrences of the motif). Figure 4.9(b) shows the motif described in Figure 4.9(a) being identified in all the PROSITE-TP human proteins. Again, the conserved residues, as defined by the sequence motif and the leniency, are marked with an asterisk.

Once the PROSITE records are parsed to identify the conserved residues, the conserved residues for each human protein are aggregated across PROSITE families into one set of residues. For example, ACES_HUMAN contains conserved residues at positions 221, 222, 223, 232, 234 and 236 according to the Carboxylesterase B1 motif (PROSITE family PS00122) and contains further conserved residues at positions 126, 127 and 128 according to the

[3]These are proteins that contain the motif as expected by PROSITE (Hulo *et al.*, 2006), not to be confused with the later use of 'TP' to indicate a true positive in the ImPACT/PROSITE benchmarking

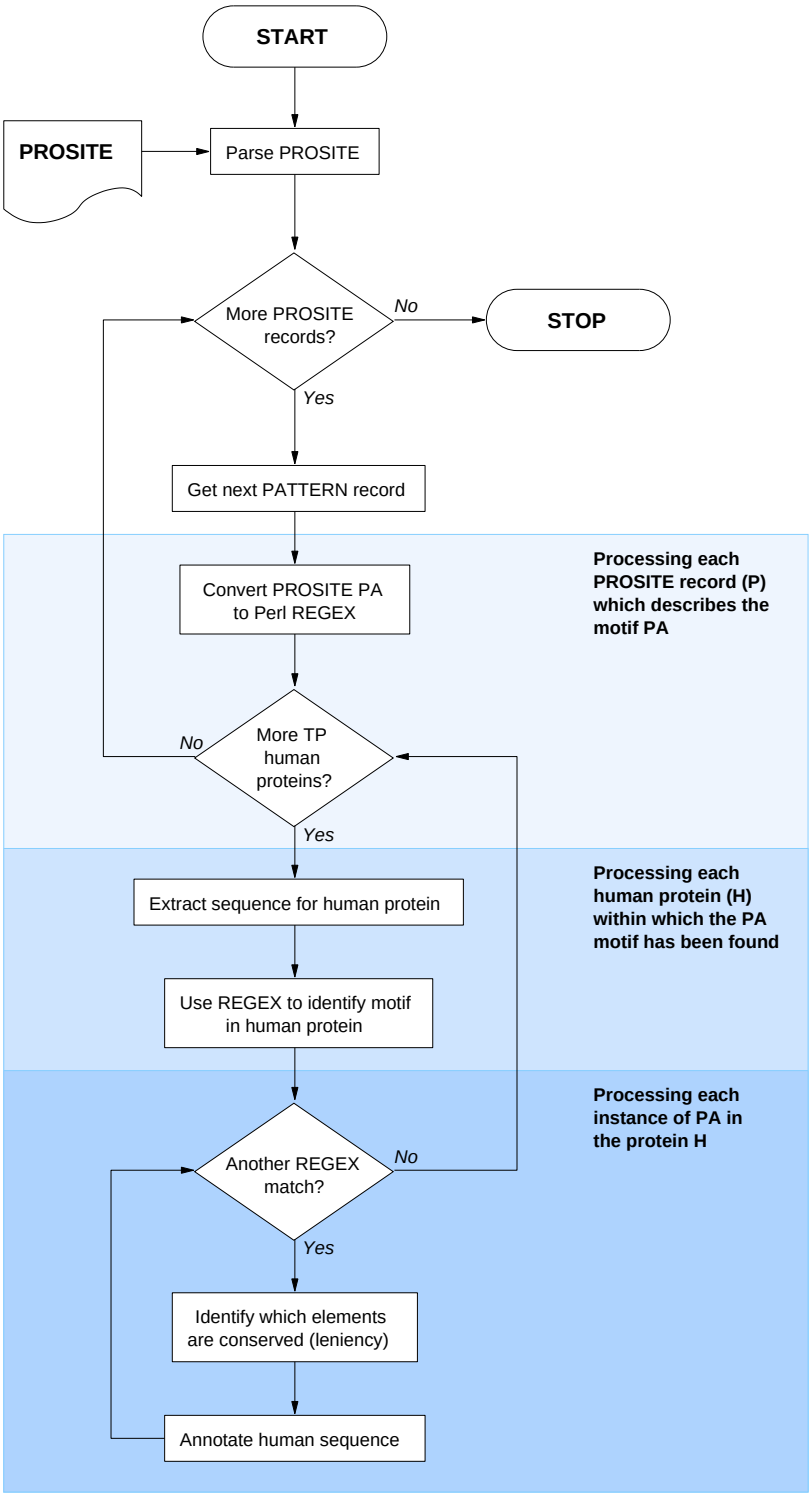


Figure 4.8: Extracting conserved residues from PROSITE

For each PROSITE record, a Perl regular expression is generated from the PROSITE motif string (see Figure 4.9(a) for an example). All instances of the motif are identified in each true positive (TP) human protein (see text) using the regular expression. Using a predefined leniency value (throughout this chapter, a value of 2 is used), the conserved elements of the motif are identified. The human sequence is annotated with the conserved PROSITE residues and these data are recorded.

PROSITE PA	S -	Q -	[STK] -	[TA] -	I -	[SC] -	R -	[FH] -	[ET] -	x -	[LSQ] -	x(0,1) -	[LIR] -	[ST]	.
Perl REGEX	/ S	Q	[STK]	[TA]	I	[SC]	R	[FH]	[ET]	.	[LSQ]	.{0,1}	[LIR]	[ST]	/g
Leniency ≤ 1	*	*			*		*								
Leniency ≤ 2	*	*		*	*	*	*	*	*					*	
Leniency ≤ 3	*	*	*	*	*	*	*	*	*		*		*	*	

(a) The POU-specific (POUs) domain signature 2 (PROSITE family PS00465)
Converting the POU domain signature 2 from the PROSITE PA format into a Perl regular expression (REGEX). The consequence of applying different conservation leniencies (1, 2 and 3) are shown beneath the PA/REGEX using $*$s. Throughout this chapter, a leniency of 2 is used to identify conserved elements of PROSITE motifs (shown in red).

```
P02F1_HUMAN / P14859 ...321 SQTTISRFEALNLS 334...
P02F2_HUMAN / P09086 ...236 SQTTISRFEALNLS 251...
P02F3_HUMAN / Q9UKI9 ...224 SQTTISRFEALNLS 239...
P03F1_HUMAN / Q03052 ...285 SQTTICRFEALQLS 300...
P03F2_HUMAN / P20265 ...303 SQTTICRFEALQLS 318...
P03F3_HUMAN / P20264 ...355 SQTTICRFEALQLS 370...
P03F4_HUMAN / P49335 ...227 SQTTICRFEGLQLS 242...
P04F1_HUMAN / Q01851 ...308 SQSTICRFESLTLS 323...
P04F2_HUMAN / Q12837 ...295 SQSTICRFESLTLS 310...
P04F3_HUMAN / Q15319 ...223 SQSTICRFESLTLS 238...
P05F1_HUMAN / Q01860 ...179 SQTTICRFEALQLS 194...
P05F2_HUMAN / Q8N7G0 ...159 SQTTICRFEAQQLS 174...
P05L1_HUMAN / Q06416 ...179 SQKTICRFEALQLS 194...
P06F1_HUMAN / Q14863 ...180 SQSAICRFEKLDIT 195...
P06F2_HUMAN / P78424 ...509 SQSAICRHTIL RS 523...
PIT1_HUMAN  / P28069 ...165 SQTTICRFENLQLS 180...
                            ** ****** *
```

(b) Identifying the POU-2 motif in the human proteins
Each TP human protein in the POU-2 PROSITE family (PS00465) is searched for the occurence of the motif, using the regular expression given in Figure 4.9(a); The motif is highlighted here in pale yellow and the conserved elements are marked with an $*$; positions for the preceeding and following residue are shown in grey.

Figure 4.9: Extracting data from the POU-specific (POUs) domain signature 2 PROSITE family (PS00465)

Carboxylesterase B2 motif (PROSITE family PS00941). ACES_HUMAN is therefore annotated as being conserved at positions 126, 127, 128, 221, 222, 223, 232, 234 and 236.

MUSCLE alignments (see Section 2.3.2 for a description of the MUSCLE method) of functionally equivalent proteins or FEPs (as identified by FOSTA, see Chapter 3), are extracted from the FOSTA database. Only the most reliable FEPs are used: those non-fragmented proteins that have matched on protein prefix or share a synonym with the root human protein (see Section 3.2.2.2). In this chapter, these FEPs will be described as 'strict'. The alignments are further constrained in that they must contain twenty or more proteins, to ensure that the alignment is adequately informative and will therefore generate reliable data for the modelling process. This results in 231 proteins containing PROSITE conserved residues against which ImPACT can be benchmarked.

Figure 4.10(a) shows a section of the alignment of RS27_HUMAN with the 28 strict FEPs that are extracted from the FOSTA database. RS27_HUMAN contains the ribosomal S27E motif (PROSITE family PS01168). This sequence motif is highlighted along the bottom of the alignment in yellow or red. Using a leniency of two, seven of these elements are considered 'conserved'; these are marked in red in Figure 4.10(a). These will be the positive examples in the PROSITE benchmarking of ImPACT, all non-annotated residues will be used as negative examples. Any positive example not identified by ImPACT is a 'false negative' (FN) and any negative example identified as conserved by ImPACT is a 'false positive' (FP); correct assignments of positive and negative examples are true positives (TPs) and true negatives (TNs) respectively (see Section 2.3.5 for an introduction to TPs, TNs, FPs and FNs and binary classification performance statistics).

There are two characteristics of this dataset that constrain how far the results can be interpreted. Firstly, the number of negative examples will, in the vast majority of sequences, far outnumber the number of positive examples. As such, it is inappropriate to use performance statistics such as accuracy ($(TP + TN)/(TP + TN + FP + FN)$), as a high accuracy can be achieved simply by classifying everything as not conserved.

More importantly, there will be residues that are conserved in the alignment, but will not be annotated by PROSITE. PROSITE is a database of sequence motifs. Residues involved at catalytic sites, ligand binding and those important for structural conservation need not be described by PROSITE. Indeed, a phenylalanine at position 11 in Figure 4.10(a) is 100% conserved but not

included in the PROSITE motif. In terms of the present dataset, this translates to uncertainty in the set of TNs; in terms of performance statistics, this suggests that statistics that use the number of FPs (including the postive predictive value or PPV; the false discovery rate, or FDR; and the false positive rate or FPR) will be misleading, and will underestimate the performance of ImPACT.

In the remainder of this chapter, the benchmarking of ImPACT against PROSITE will be described. The ImPACT threshold generated will be compared with thresholds varying from 0 to 1, at increments of 0.01; these thresholds will be referred to as the 'standard' thresholds. Residues defined as conserved by PROSITE will be described as **PC** residues; all other PROSITE residues will be described as **!PC** residues and residues that are defined as conserved by ImPACT will be described as **IC** residues.

Bearing in mind the caveats described above, the performance statistics have been limited to Receiver Operating Characteristic (ROC) plots and Matthews Correlation Coefficient (MCC) (again, for a description of these terms, see Section 2.3.5). When considering the ROC plots in the context of the caveats described above, it is more imporant that ImPACT maximises the TPR, however, small FPRs are also desirable. A good ImPACT result will be maximally distant from the TPR/FPR identity line (the line that describes performance no better than random).

4.3.3.2 Examples of good results

There are many proteins for which ImPACT is successful in identifying the PROSITE conserved (PC) residues. Figure 4.11 shows the ROC and MCC results for four such proteins: 40S ribosomal protein S27 [UniProtKB:P42677/RS27_HUMAN], triosephosphate isomerase [UniProtKB:P60174/TPIS_HUMAN], transthyretin [UniProtKB:P02766/TTHY_HUMAN] and calcium-dependent phospholipase A2 [UniProtKB:P39877/PA2G5_HUMAN].

ImPACT generates a threshold of 0.9404 for high conservation in the RS27_HUMAN alignment, which gives an MCC of 0.93. The alignment for these data is shown in Figure 4.10(a). It is clear that this is the optimal value with respect to MCC (Figure 4.11(b)). In addition, the ImPACT ROC result is maximally distant from the TPR/FPR identity line as compared to the other thresholds (Figure 4.11(a)). The reason that ImPACT doesn't attain a perfect MCC score of 1.00 is that one !PC residue is identified as an IC residue: the phenylalanine at position 11

is identified by ImPACT. In other words, ImPACT identifies an additional highly conserved residue which is not included in the PROSITE pattern.

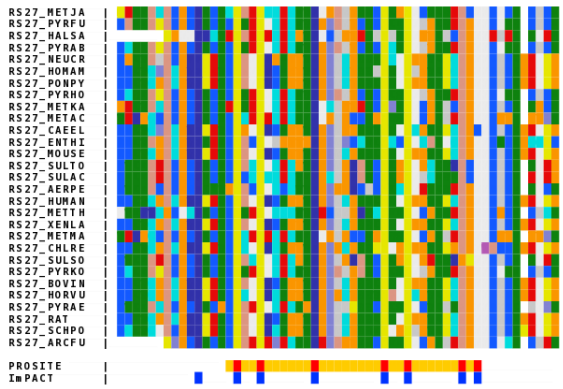
The patterns of conservation in the TPIS_HUMAN alignment (containing 318 proteins including TPIS_HUMAN, see Figure 4.12) result in an ImPACT threshold of 0.9786 which has an MCC of 0.5047. This is clearly better than any of the standard thresholds (Figure 4.13(b)). Again, the ImPACT result is maximally distant from the TPR/FPR identity line (Figure 4.13(a)). In addition to demonstrating that ImPACT performs well, this result shows that very precise thresholds (in the case of ImPACT, precision to the fourth decimal place) can enhance detection of highly conserved residues compared with thresholds of lower precision (in the case of the standard thresholds, precision to the second decimal place): the ImPACT result does not fall on the line drawn by the standard thresholds.

Although the MCC result for the PROSITE benchmarking of TTHY_HUMAN and its 20 FEPs (see Figure 4.10(b)) is not as high as the previous two examples—the ImPACT threshold of 0.9544 has an MCC of 0.3649, see Figure 4.14(b)—it is the best result, surpassing the small peak of the standard thresholds (concentrated at ≈ 0.95) and achieving maximal distance from TPR/FPR identity. Again, this demonstrates the increased discriminative power of higher precision thresholds; without a data modelling process such as that described in Section 4.2.2, it is difficult to generate such high precision thresholds. The relatively poorer performance in this dataset is owing to high numbers of FPs: compare the number of IC residues and the number of PC residues in Figure 4.10(b).

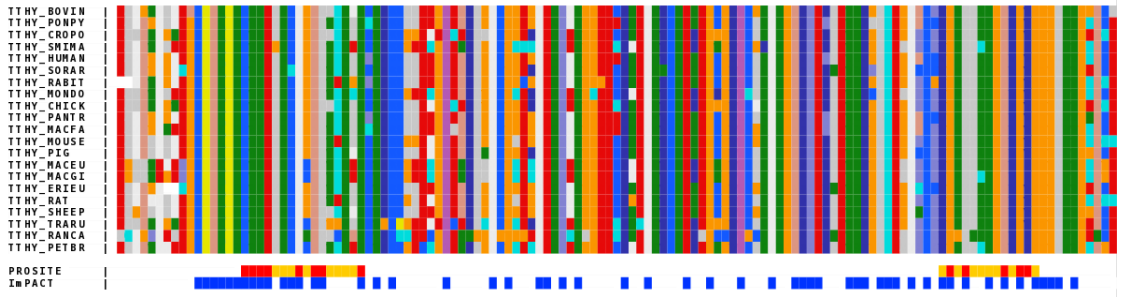
The final example is for the protein PA2G5_HUMAN (see Figures 4.10(c), 4.15(a) and 4.15(b)). ImPACT generates a high conservation threshold of 0.9623 for the MUSCLE alignment of PA2G5_HUMAN and its 21 FEPs. Unlike in the previous examples, the best result is achieved by several of the standard thresholds, clear by the performance 'plateau' at an MCC of 0.6850 from the standard threshold 0.97 onwards. The ImPACT threshold of 0.9623 achieves the same MCC value as the best standard thresholds.

4.3.3.3 Examples of average results

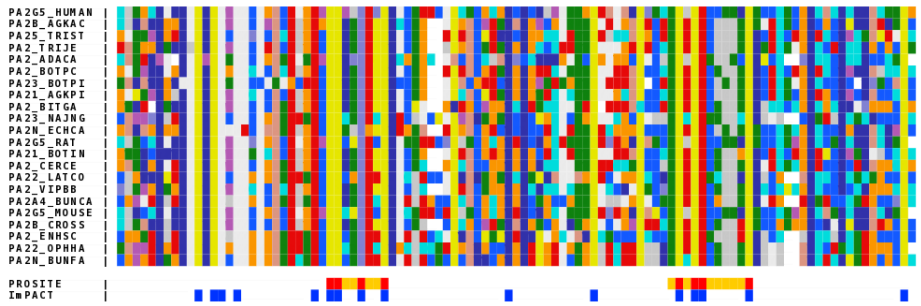
In the previous section, four examples of very successful results were described. Given the caveats regarding the dataset (see Section 4.3.3.1), it is expected that most results will underes-



(a) RS27_HUMAN alignment, ImPACT threshold is 0.9408



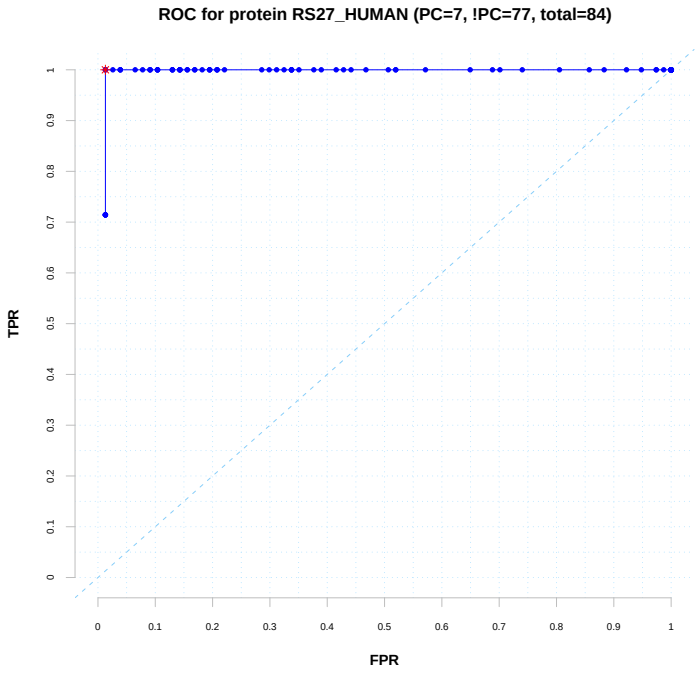
(b) TTHY_HUMAN alignment, ImPACT threshold is 0.9544



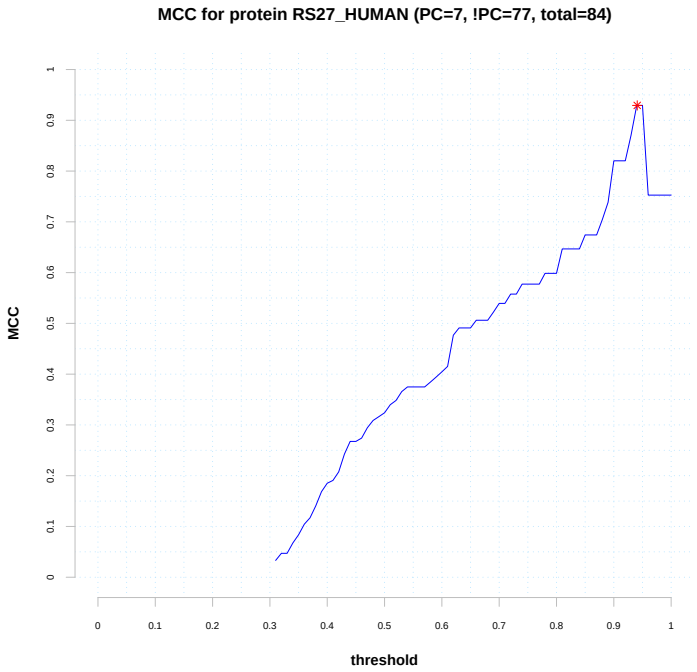
(c) PA2G5_HUMAN alignment, ImPACT threshold is 0.9623

Figure 4.10: Example annotated alignments analysed by ImPACT: RS27_HUMAN, TTHY_HUMAN, PA2G5_HUMAN

PROSITE and ImPACT annotations are shown below the MSA: PROSITE motifs are indicated using orange, PC residues (conserved PROSITE residues, as defined using a leniency of 2) are indicated using red, residues classified as IC (highly conserved after application of the ImPACT threshold) are indicated using blue. Amino acid colours as shown in Appendix [B.i].



(a) RS27_HUMAN+28 FEPs, ROC

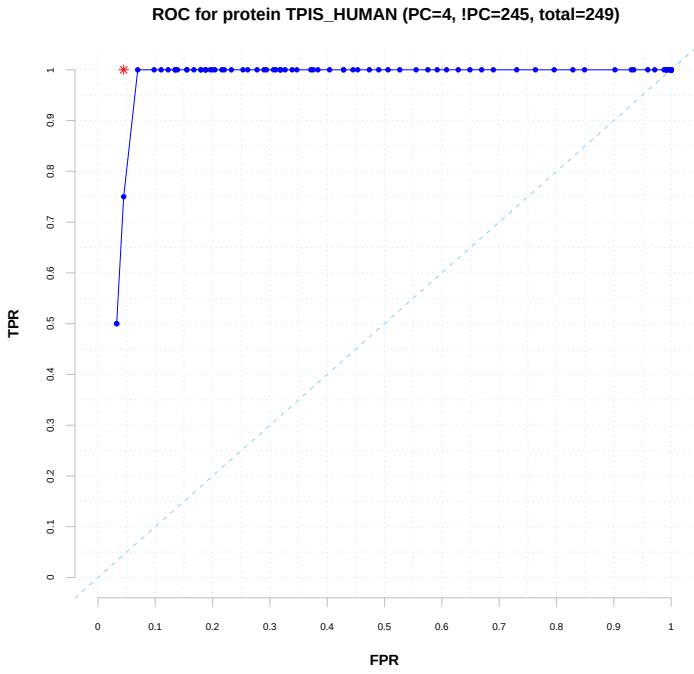


(b) RS27_HUMAN+28 FEPs, MCC

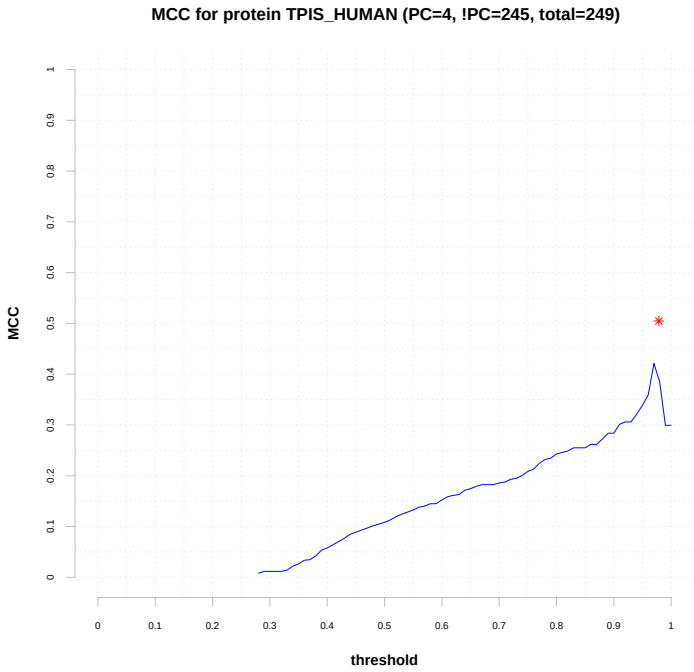
Figure 4.11: Benchmarking ImPACT against PROSITE: RS27_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).

Figure 4.12: Example annotated alignments analysed by ImPACT TPIS_HUMAN
PROSITE and ImPACT annotations are shown below the MSA: PROSITE motifs are indicated using orange, PC residues (conserved PROSITE residues, as defined using a leniency of 2) are indicated using red, residues classified as IC (highly conserved after application of the ImPACT threshold) are indicated using blue. Amino acid colours as shown in Appendix [B.i].



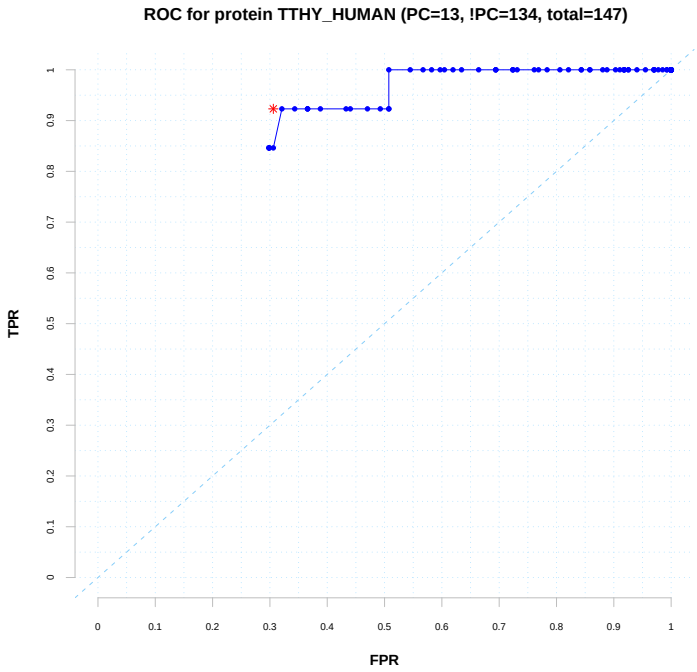
(a) TPIS_HUMAN+318 FEPs, ROC



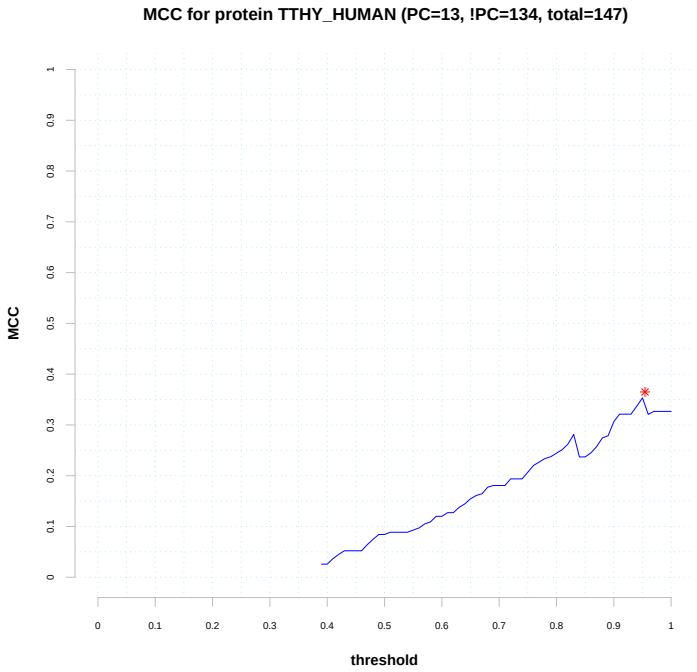
(b) TPIS_HUMAN+318 FEPs, MCC

Figure 4.13: Benchmarking ImPACT against PROSITE: TPIS_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).



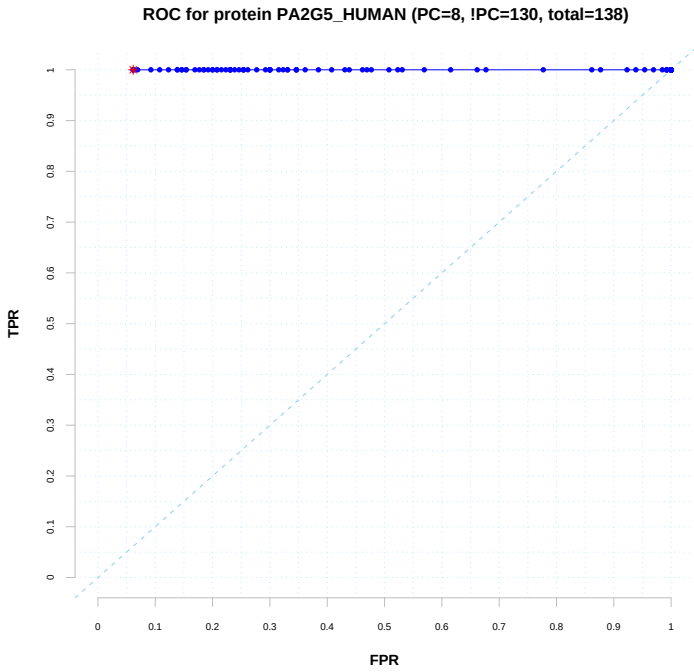
(a) TTHY_HUMAN+20 FEPs, ROC



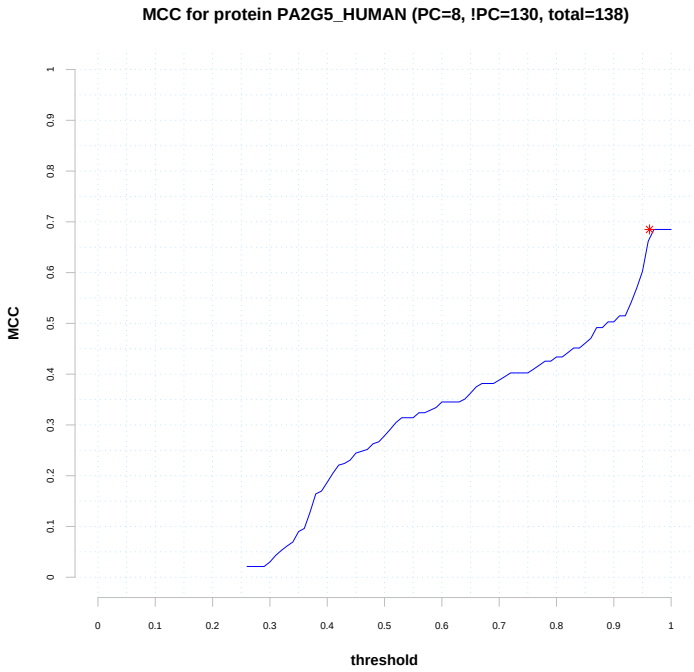
(b) TTHY_HUMAN+20 FEPs, MCC

Figure 4.14: Benchmarking ImPACT against PROSITE: TTHY_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).



(a) PA2G5_HUMAN+21 FEPs, ROC



(b) PA2G5_HUMAN+21 FEPs, MCC

Figure 4.15: Benchmarking ImPACT against PROSITE: PA2G5_HUMAN

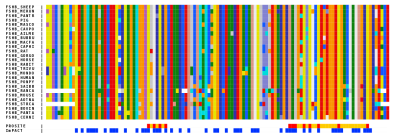
The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).

timate the performance of ImPACT; this is indeed the case. In this section, three 'typical' results are described.

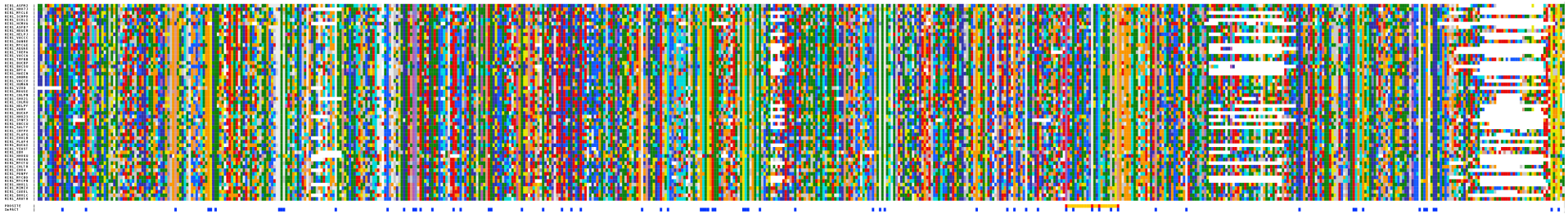
Approximately 25% of PROSITE benchmarking results are similar to that of FSHB_HUMAN. That is, all standard thresholds and the ImPACT threshold result in a 100% TPR (Figure 4.17(a)) (i.e., all of the residues that should be identified as conserved are identified as conserved, or all PC residues are identified as IC residues), and the ImPACT threshold yields a near-optimal MCC score (Figure 4.17(b)). Given that *all* thresholds return a 100% TPR, all of the PC residues must be 100% conserved. Further, the FPR is always 30% or greater regardless of the threshold chosen. This indicates that for all thresholds, at least 30% of !PCs are 100% conserved and, according to the PROSITE annotations, wrongly identified by ImPACT as highly conserved. It is clear from Figure 4.16(a) that there *are* many residues in FSHB_HUMAN that are 100% conserved and not included as part of a motif in the PROSITE dataset.

The MCC and ROC results for RIR1_HUMAN are conflicting: the ImPACT threshold is near-optimal with respect to the ROC plot (see Figure 4.18(a)), but the MCC results are rather poor (Figure 4.18(b)). The ImPACT threshold has been set to 0.7639, which yields a much lower MCC than many of the standard thresholds (performance peaks at $\approx$0.94). The alignment for RIR1_HUMAN and its strict FEPs is shown in Figure 4.16(b). It is clear that many !PC residues are identified as IC. It is also clear that there are regions of significant insertions, despite all FEPs sharing the 'RIR1' protein prefix. This suggests that there is considerable diversity between the species and perhaps the lower threshold of 0.7639 generated by ImPACT is appropriate.

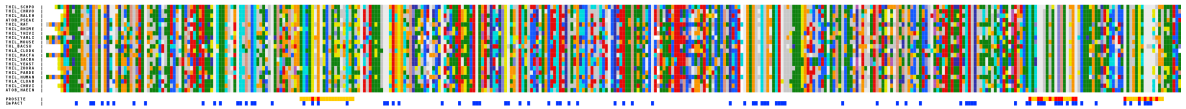
The final typical example is of the analysis of THIL_HUMAN. Again, it is evident that approximately 10% of !PC residues in this dataset are 100% conserved, given that the FPR does not fall below 0.10 (Figure 4.19(a)). Unlike the FSHB_HUMAN example discussed above, ImPACT does not achieve 100% TPR, however it is near-optimally distant from the TPR/FPR line, as compared with the standard thresholds. Again, however, the MCC is less than optimal (Figure 4.19(b)). Figure 4.16(c) shows that there are many highly conserved residues outwith the motif regions in the alignment of THIL_HUMAN and its 19 FEPs.



(a) FSHB_HUMAN alignment, ImPACT threshold is 0.9601



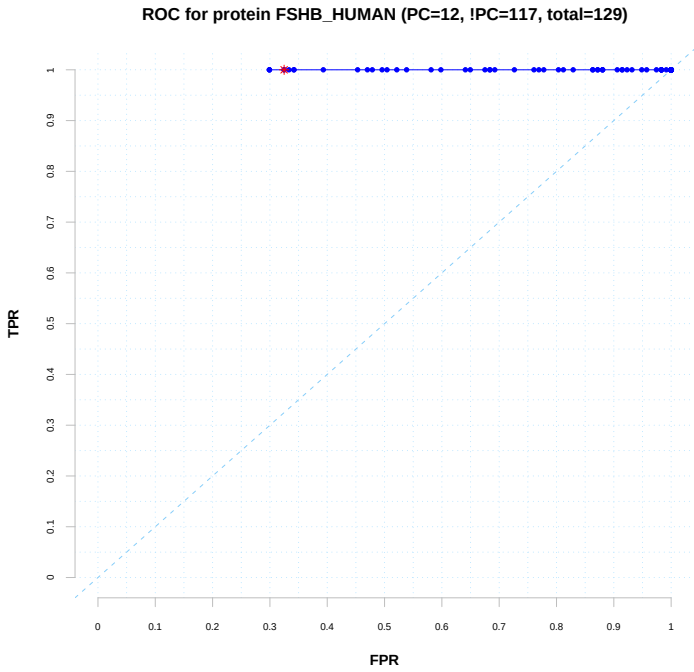
(b) RIR1_HUMAN alignment, ImPACT threshold is 0.7639



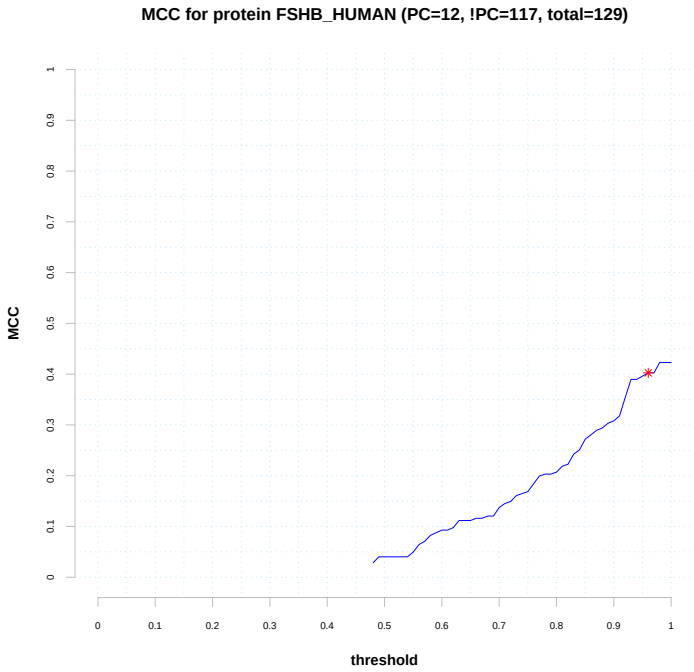
(c) THIL_HUMAN alignment, ImPACT threshold is 0.9204

Figure 4.16: Example annotated alignments analysed by ImPACT: FSHB_HUMAN, RIR1_HUMAN and THIL_HUMAN

PROSITE and ImPACT annotations are shown below the MSA: PROSITE motifs are indicated using orange, PC residues (conserved PROSITE residues, as defined using a leniency of 2) are indicated using red, residues classified as IC (highly conserved after application of the ImPACT threshold) are indicated using blue. Amino acid colours as shown in Appendix [B.i].



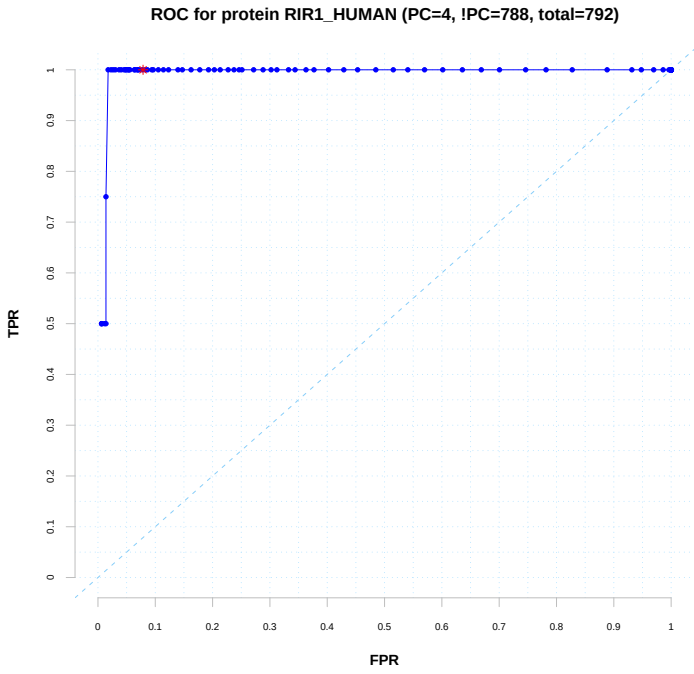
(a) FSHB_HUMAN+25 FEPs, ROC



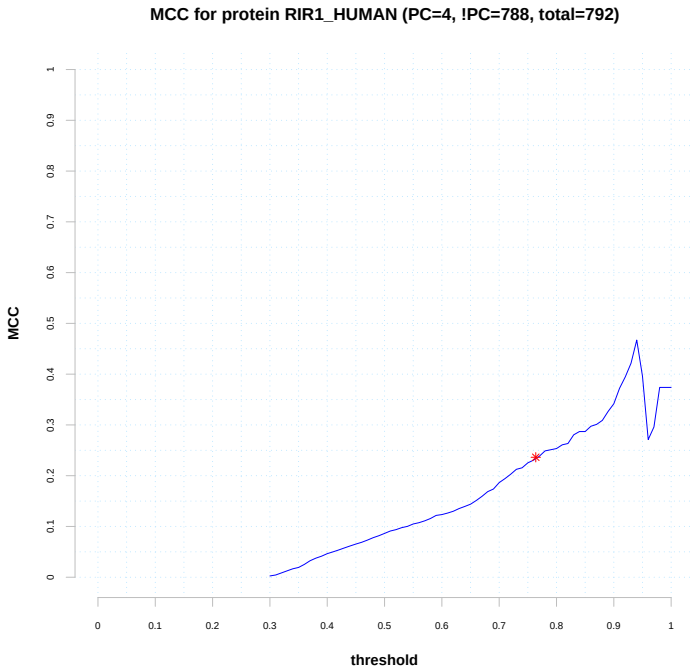
(b) FSHB_HUMAN+25 FEPs, MCC

Figure 4.17: Benchmarking ImPACT against PROSITE: FSHB_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).



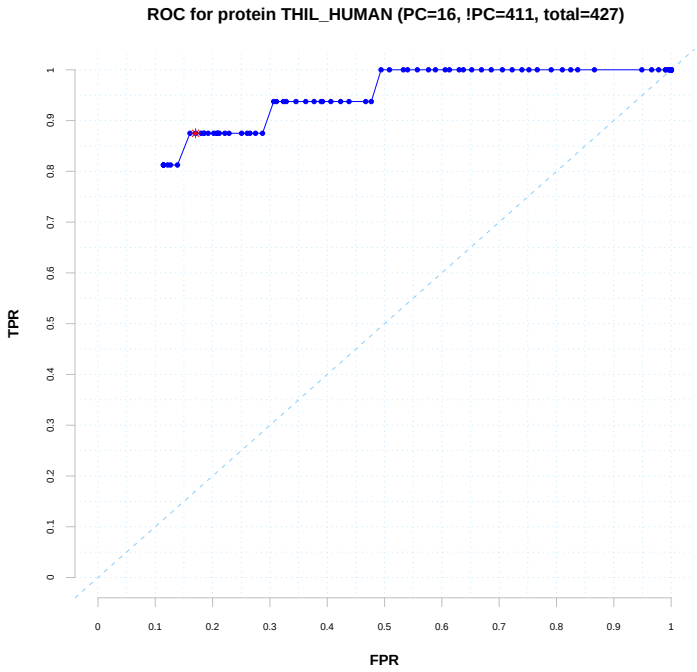
(a) RIR1_HUMAN+54 FEPs, ROC



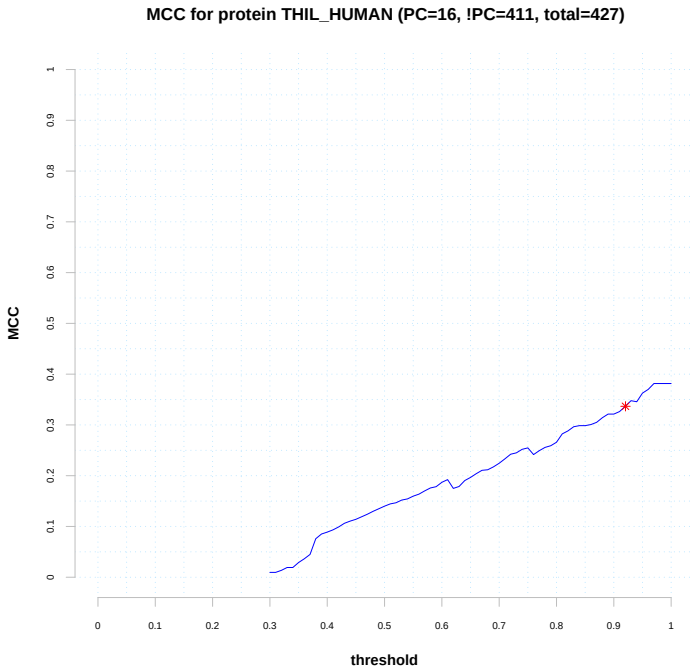
(b) RIR1_HUMAN+54 FEPs, MCC

Figure 4.18: Benchmarking ImPACT against PROSITE: RIR1_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).



(a) THIL_HUMAN+19 FEPs, ROC



(b) THIL_HUMAN+19 FEPs, MCC

Figure 4.19: Benchmarking ImPACT against PROSITE: THIL_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).

4.3.3.4 Examples where sequence conservation fails to identify PROSITE motif residues

For five proteins (2.16%) in the PROSITE dataset, more than half of the standard thresholds are found on the 'wrong-side' of the TPR/FPR identity line. That is, the FPR is greater than the TPR. For one protein in the dataset, the general level of conservation is so low that the first criteria (see Section 4.2.2.1) is not met. In this section, these results are discussed.

Human glycyl-tRNA synthetase [UniProtKB:P41250/SYG_HUMAN] is assigned 61 strict FEPs; the MUSCLE alignment of this protein and its FEPs is shown in Figure 4.20(a). The WHEP-TRS domain signature motif (PS00762) is identified at positions 74-102 in SYG_HUMAN. As the alignment clearly shows these residues (highlighted in yellow and red) occur in a very sparse section of the alignment, where only five (including SYG_HUMAN) of the proteins are represented.

Not only do all these proteins share their protein prefix 'SYG' with the human protein, but they all (but one) share the same four synonyms:

- Glycyl-tRNA synthetase
- Glycine–tRNA ligase
- GlyRS
- EC 6.1.1.14

The one exception is SYG_YEAST, which is annotated with the synonyms:

- Glycyl-tRNA synthetase 1
- Glycine–tRNA ligase 1
- GlyRS 1
- GlyRS1
- EC 6.1.1.14

Given these annotations, this set of proteins appears to be functionally coherent; the annotations do not suggest that the alignment is inappropriate. Rather, the motif is only found in a specific subset of proteins and this *sub*functionality is not reflected in the UniProtKB/Swiss-Prot annotations.

Some species have gained (or lost) the WHEP-TRS domain without affecting the overall function of the protein, suggesting that the presence (or absence) of the motif has little impact on functionality. On closer inspection, it appears that the WHEP-TRS domain is a eukaryotic embellishment: only the eukaryotic species—*Homo sapiens*, *Mus musculus*, *Bombyx mori*, *Pongo pygmaeus* and *Caenorhabditis elegans*—contain the domain, with *Saccharomyces cerevisiae* being the only eukaryotic exception. Indeed, the WHEP-TRS domain has been shown to exist in several higher eukaryotic aminoacyl-transfer RNA synthetases and that the same functionality in Prokaryotes is encoded by distinct genes (Cerini *et al.*, 1991), explaining the absence of the domain in the non-eukaryotic species.

The fitted mixture model for RNC_HUMAN (ribonuclease III) and its 219 strict FEPs violates the first constraint of the ImPACT analysis, suggesting that the minimal model of conservation as defined in Section 4.2.2.1 does not exist in the alignment. Figure 4.20(b) shows the entire alignment for RNC_HUMAN and its 219 strict FEPs. It is immediately apparent that this alignment is very sparse: a small number of ribonuclease III proteins (including RNC_HUMAN) have acquired extensive insertions. The paucity of amino acid representation in these regions will result in many very low scoring columns that will dominate the distribution of conservation scores, forcing the mixture modelling to be biased towards accounting for the low conservation scores.

Figure 4.22 shows a column-wise subsection of the full RNC_HUMAN alignment. Two ribonuclease III family signatures (PROSITE family PS00517) are identified in RNC_HUMAN, at positions 966-974 and 1144-1152; of these positions, 969, 971, 972 and 973 in the first occurence of the motif are PC, and 1147, 1149, 1150 and 1151 in the second occurence of the motif are PC. The first set of PC residues is very well-conserved across all FEPs, with `specsim` corrected conservation scores of 1.00, 0.96, 1.00 and 1.00 (to 2dp) respectively. However the second occurence of the motif is present only in the human protein, and as such the conservation scores for these values are very low (0.40 to 2dp).

All 219 FEPs share the following synonyms:

- Ribonuclease 3
- Ribonuclease III
- RNase III
- EC 3.1.26.3

Further, all but one share the "RNC" protein prefix, the only exception being RNT1_YEAST. As in the SYG_HUMAN results, the proteins aligned here are clearly all ribonuclease III proteins. Ribonuclease III is expressed in most eukaryotic and prokaryotic cells (Wu *et al.*, 2000; Conrad and Rauhut, 2002) and therefore, although the *function* of the protein has been maintained throughout evolution, it is likely that significant changes have occurred between species. Figure 4.20(b) shows that this is indeed the case, with large inserts in several proteins between smaller, reasonably well conserved regions.

Unlike SYG_HUMAN however, the vast majority of the proteins in the alignment *do* contain the PROSITE motif. Therefore, it is *not* the case that the UniProtKB/Swiss-Prot annotations fail to represent some subfunctionality within the ribonuclease III proteins, at least not at the level of this particular PROSITE motif. More disruptive to the successful performance of ImPACT in this case are the extensive insertion regions evident in the alignment and, with respect to the PROSITE benchmarking, the acquisition of a *second* ribonuclease III family signature only in the human protein.

The strict FEPs used to create the alignment are all prokaryotic or archaeal, excepting *Homo Sapiens*, two worm species (*Caenorhabditis briggsae* and *Caenorhabditis elegans*) and two yeast species (*Schizosaccharomyces pombe* and *Saccharomyces cerevisiae*). Figure 4.21 (taken and modified from Conrad and Rauhut (2002)) shows the domain structure of ribonuclease III proteins across different species and demonstrates that there have indeed been several extensive embellishments in eukaryotic species. Even within the eukaryotic domain, significant additions have been aquired (compare the structure of the yeast species, *S. pombe* and *S. cerevisiae* in Figure 4.21, with that of the human species, *H. sapiens* in Figure 4.21). If other eukaryotic species closer to *Homo sapiens* were to be included in the alignment, the second occurrence of the PS00517 PROSITE motif might be represented.

In addition, as seen previously for SYG_HUMAN, the extensive insertion regions will result in a distribution of conservation scores saturated with very low values, which will dominate the

mixture model optimisation, and limit the extent to which ImPACT can accurately model the distribution.

4.3.3.5 The representative proteins in PROSITE

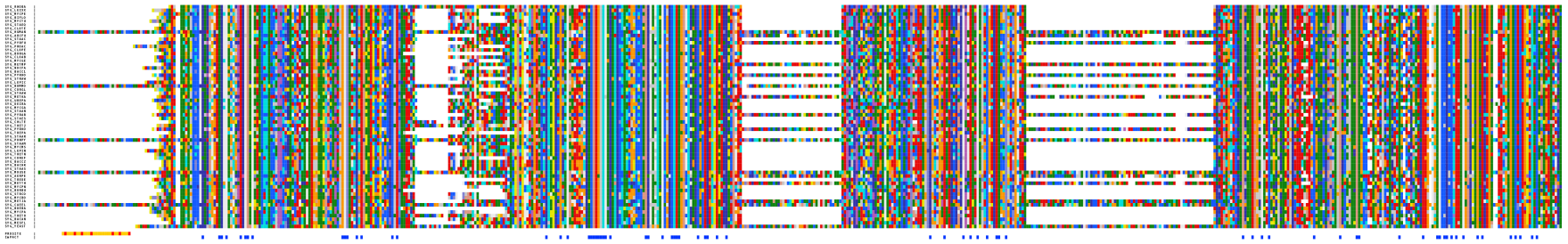
Three of the four proteins discussed in Section 4.3.2 contain PROSITE motifs: G6PD_HUMAN, P53_HUMAN and OTC_HUMAN. Their alignments are shown in Figures 4.23(a), 4.23(b) and 4.23(c) respectively. In contrast to the alignments shown in Figures 4.16 and 4.20, a higher proportion of the motif residues (those highlighted in yellow or red in Figure 4.23) are classified as PC (those coloured in red in Figure 4.23). That is, the motifs are more strict than seen previously.

The results for G6PD_HUMAN (Figures 4.23(a), 4.24(a), 4.24(b)) and P53_HUMAN (Figures 4.23(b), 4.25(a), 4.25(b)) are similar to those of FSHB_HUMAN described in Section 4.3.3.3. That is, 100% TPR rate is achieved by all thresholds, indicating that all PC residues are 100% conserved, and the FPR is never 0%, indicating that there are !PC residues that are 100% conserved. The MCC values for G6PD_HUMAN and P53_HUMAN approach optimal performance.

The variation in TPR in Figure 4.26(a) demonstrates that there are some PC residues that are not 100% conserved in OTC_HUMAN. However, ImPACT still achieves 100% TPR with a comparatively low threshold of 0.8672 and is close to optimal performance with respect to the FPR. Like the previous results for G6PD_HUMAN and P53_HUMAN, ImPACT approaches the maximal MCC. Here however, it is significant that ImPACT does not apply too rigourous a threshold for the patterns of conservation in the OTC_HUMAN alignment. If the ImPACT threshold generated for the P53_HUMAN alignment (0.9636) were applied to the OTC_HUMAN data, the TPR would drop significantly from 100% to 66.67%.

4.3.4 Using artificial alignments to assess ImPACT

As discussed in Section 4.3, no gold standard dataset exists against which ImPACT can be benchmarked. Thus far, ImPACT has been evaluated by considering four representative proteins (OTC, G6PD, P53 and haemoglobin (HBB)) and by using PROSITE data. To evaluate ImPACT further, a battery of artificial alignment data have been generated. Three Gaussian components—being assumed to underlie the real data—have been used to generate artificial



(a) SYG_HUMAN alignment, ImPACT threshold is 0.9247

(b) RNC_HUMAN alignment, constraint 1 is violated

Figure 4.20: Example annotated alignments analysed by ImPACT: SYG_HUMAN and RNC_HUMAN

PROSITE and ImPACT annotations are shown below the MSA: PROSITE motifs are indicated using orange, PC residues (conserved PROSITE residues, as defined using a leniency of 2) are indicated using red, residues classified as IC (highly conserved after application of the ImPACT threshold) are indicated using blue. Amino acid colours as shown in Appendix [B.i].

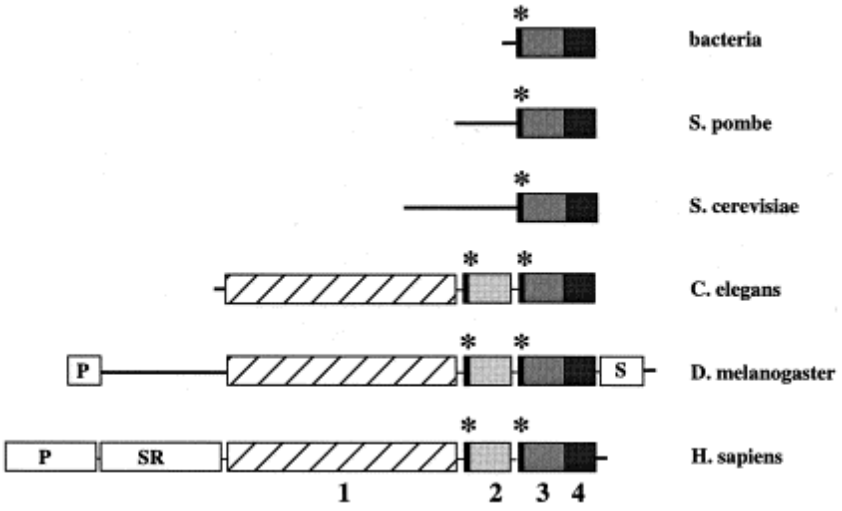


Figure 4.21: Domain structures of eukaryotic and prokaryotic ribonuclease IIIs
Asterisks indicate sequence signatures. (1) N-terminal extension in eukaryotic ribonuclease IIIs (except yeasts); (2/3) duplicated bacterial N-terminus; (4) double strand RNA-binding domain. P, S and SR indicate proline, serine and serine-arginine rich regions respectively. Figure taken and adapted from Conrad and Rauhut (2002), figure 1.

distributions of conservation scores that might be derived from protein MSAs. By varying the parameters of the data-generating Gaussians (called D_0, D_1 and D_2 to correspond with the G_0, G_1 and G_2 components used to model the data, see Section 4.2.2), the distribution of conservation scores can be finely controlled.

When generating the artificial data, it is possible to vary the mean, standard deviation and relative densities of the three Gaussian components (the density of a component describes how much of the data the component accounts for, or models). To ensure that the randomly generated data are representative of real conservation data, the fitted mixture models for the four representative proteins described in Section 4.3.2 were considered. The densities for the three fitted Gaussian components for OTC, G6PD, P53 and HBB are shown in Table 4.11. Using the average of these values as a starting point, and taking into account the known conservation patterns of the proteins, the relative densities of G_0, G_1 and G_2 for the artificial alignments were chosen as 45%, 45% and 10% (as shown in the final line of Table 4.11). Experience of many MSA-fitted mixture models was used to determine the placement and spread of these components as $\{\mu = 0.40, \sigma = 0.10; \mu = 0.65, \sigma = 0.10; \mu = 0.95, \sigma = 0.025\}$ for G_0, G_1 and G_2 respectively.

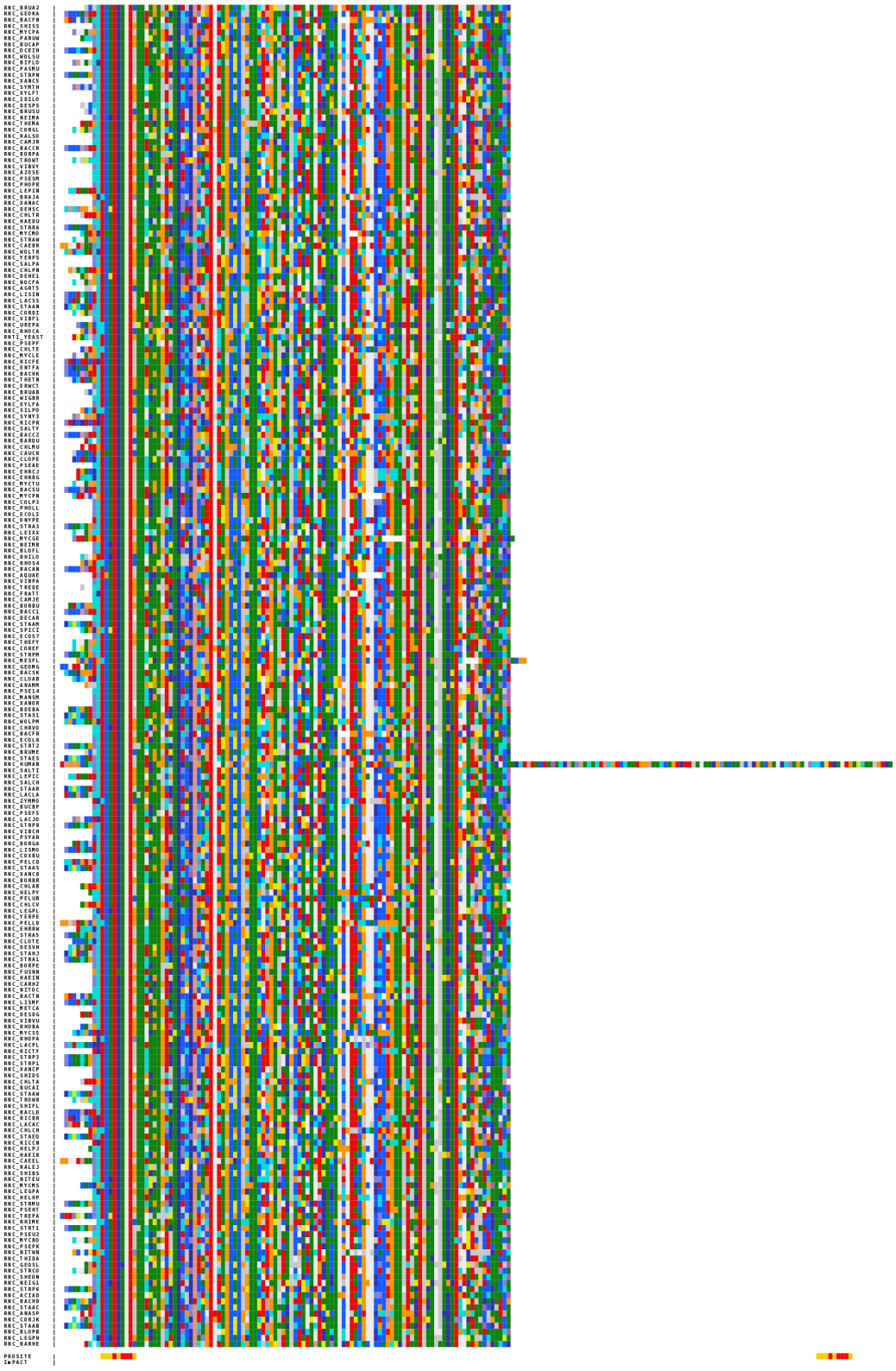
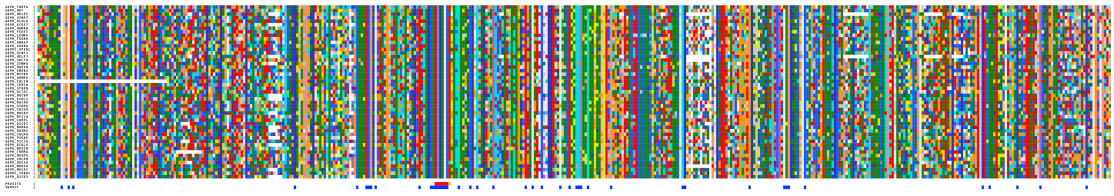
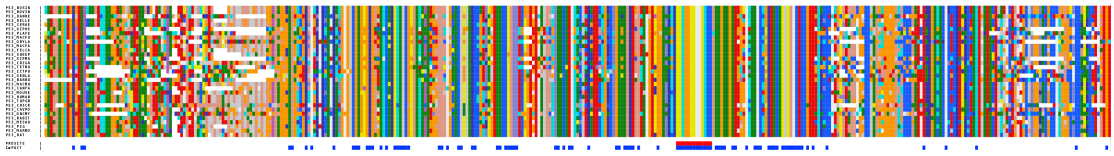


Figure 4.22: A subsection of the RNC_HUMAN alignment

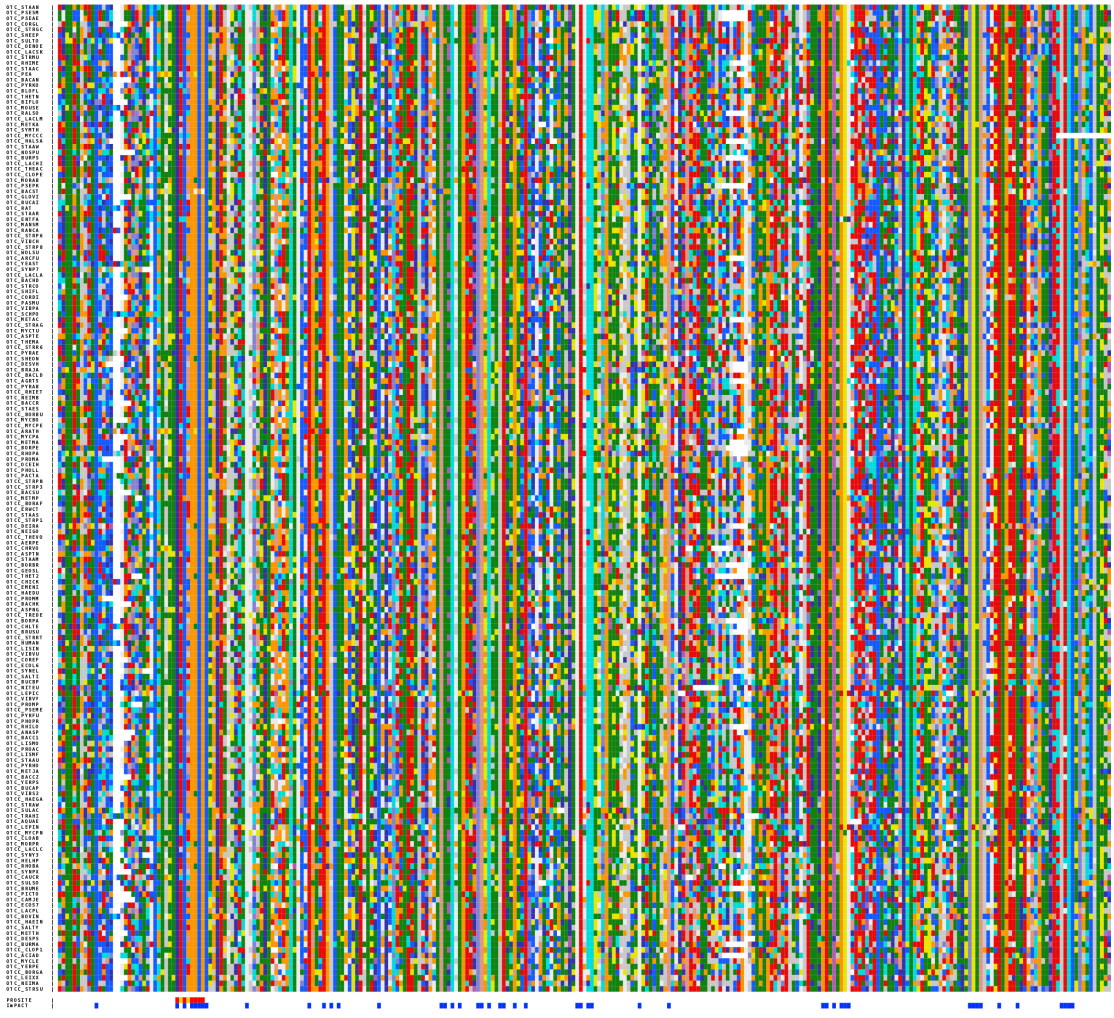
PROSITE and ImPACT annotations are shown below the MSA: PROSITE motifs are indicated using orange, PC residues (conserved PROSITE residues, as defined using a leniency of 2) are indicated using red, residues classified as IC (highly conserved after application of the ImPACT threshold) are indicated using blue. Amino acid colours as shown in Appendix [B.i].



(a) G6PD_HUMAN alignment, ImPACT threshold is 0.9612



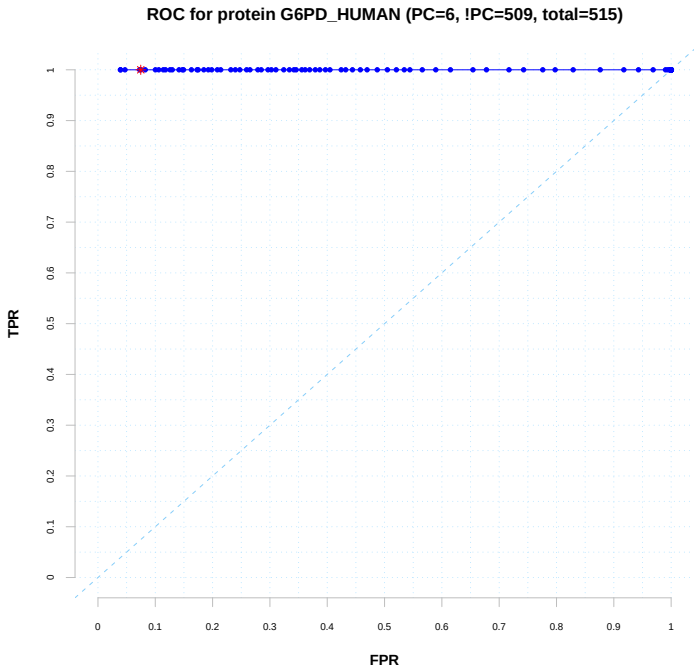
(b) P53_HUMAN alignment, ImPACT threshold is 0.9636



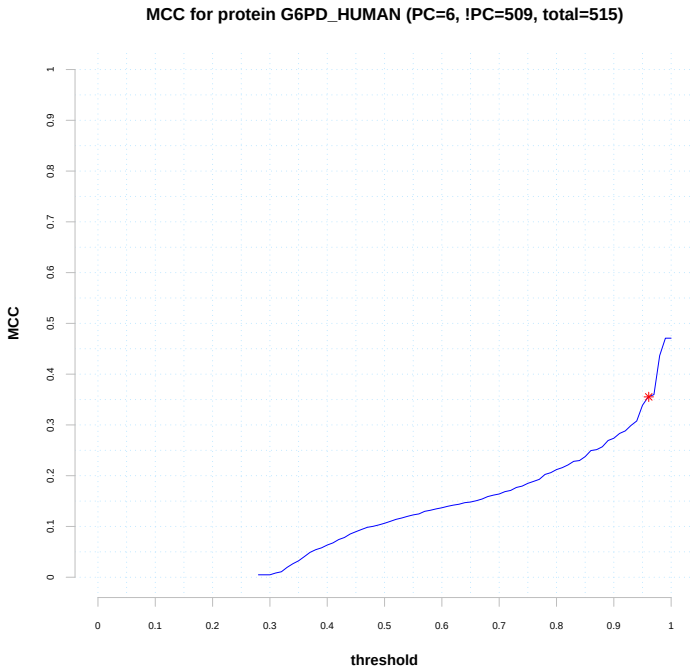
(c) OTC_HUMAN alignment, ImPACT threshold is 0.8672

Figure 4.23: Example annotated alignments analysed by ImPACT, three representative proteins: G6PD_HUMAN, P53_HUMAN and OTC_HUMAN

PROSITE and ImPACT annotations are shown below the MSA: PROSITE motifs are indicated using orange, PC residues (conserved PROSITE residues, as defined using a leniency of 2) are indicated using red, residues classified as IC (highly conserved after application of the ImPACT threshold) are indicated using blue. Amino acid colours as shown in Appendix [B.i].



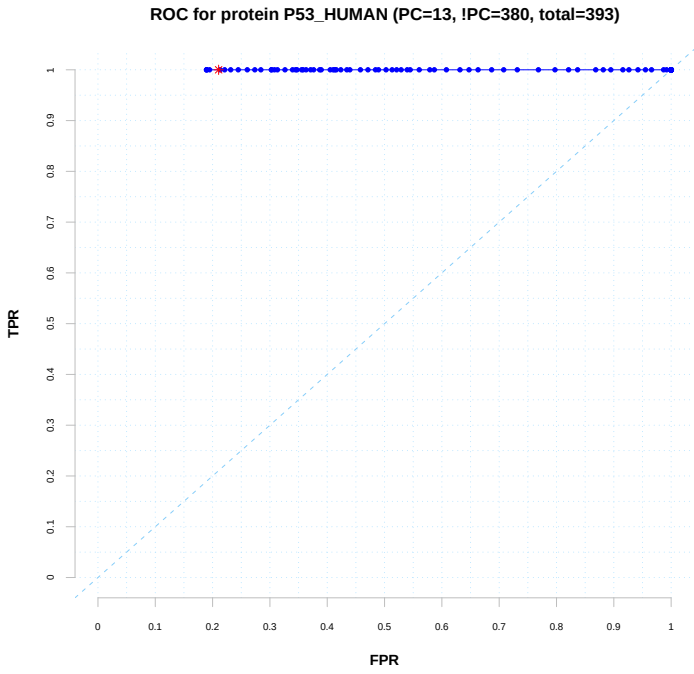
(a) G6PD_HUMAN+48 FEPs, ROC



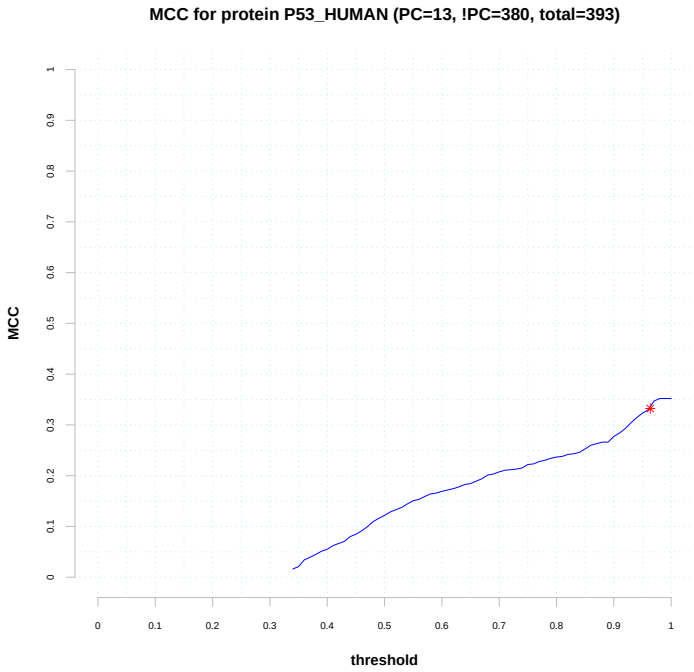
(b) G6PD_HUMAN+48 FEPs, MCC

Figure 4.24: Benchmarking ImPACT against PROSITE: G6PD_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).



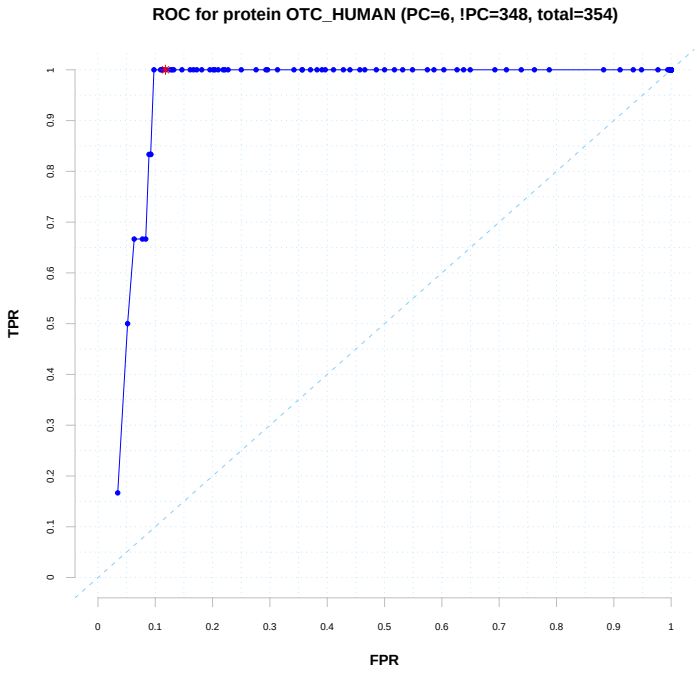
(a) P53_HUMAN+30 FEPs, ROC



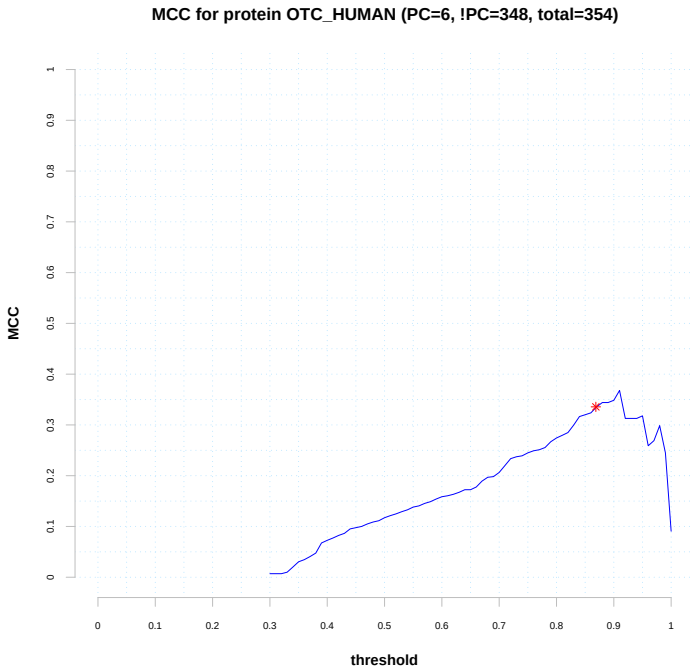
(b) P53_HUMAN+30 FEPs, MCC

Figure 4.25: Benchmarking ImPACT against PROSITE: P53_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).



(a) OTC_HUMAN+176 FEPs, ROC



(b) OTC_HUMAN+176 FEPs, MCC

Figure 4.26: Benchmarking ImPACT against PROSITE: OTC_HUMAN

The 'standard' thresholds (values from 0 to 1, at increments of 0.01) are shown in blue. The ImPACT threshold is shown as a red asterisk. To give some idea of the composition of the datasets, the numbers of PROSITE-conserved (PC) and non PROSITE-conserved (!PC) residues are given in the graph titles, along with the total number of residues considered (i.e., PC+!PC).

Table 4.11: The densities of three Gaussian components for OTC, G6PD, P53 and HBB
Protein: the protein to which the mixture model has been fitted ('Chosen values' indicates which values were chosen to generate the artificial alignments); G_0, G_1, G_2: the names of the Gaussian components (see Section 4.2.2). The 'density' of a component describes how much of the data the component accounts for, or models.

Protein	G_0	G_1	G_2
OTC	27.24	51.33	21.43
G6PD	52.58	45.41	2.10
P53	50.65	30.92	18.43
HBB	44.31	23.46	32.23
Chosen values	45.00	45.00	10.00

Table 4.12: The test sets of artificially generated conservation data
Set: the name of the test set; D_0, D_1, D_2: the names of the Gaussians used to generate the data (the unconserved, moderately conserved and highly conserved respectively). Values that are varied within an example set are *italicised*.

Set	μ_{D_0}	μ_{D_1}	μ_{D_2}
1.1	*0.35*	0.65	*0.95*
1.2	*0.40*	0.65	*0.90*
1.3	*0.45*	0.65	*0.85*
1.4	*0.50*	0.65	*0.80*
2.1	0.40	0.65	*0.70*
2.2	0.40	0.65	*0.80*
2.3	0.40	0.65	*0.90*
2.4	0.40	0.65	*0.95*
3.1	0.40	*0.90*	0.95
3.2	0.40	*0.85*	0.95
3.3	0.40	*0.80*	0.95
3.4	0.40	*0.70*	0.95

4.3.4.1 Fitting increasingly homogeneous Gaussian components

The first set of artificial alignments has been devised simply to demonstrate the mixture modelling process; no ImPACT analysis has been carried out. Here, the three underlying Gaussian components that generate the data become increasingly homogenous from example (1.1) to example (1.4) (see Figure 4.27). However, it is clear throughout the examples that the mixture model fitting process accurately models the data: the peaks of the fitted Gaussians correspond closely with the means of the underlying Gaussians (compare with the non-logit transformed axis across the top of the plots in Figure 4.27).

4.3.4.2 Generating thresholds as D_2 increases

This second set of artificial conservation score data keeps the parameters D_0 and D_1 constant, while increasing the mean of D_2 from 0.7 to 0.95. This will specificially test the second constraint as the distance between the D_1 and D_2 will increase. It is expected that as $\mu_{D_2} - \mu_{D_1}$ increases—that is, as the distance between $\mu_{D_2} - \mu_{D_1}$ increases—the threshold will increase. This is appropriate because, as $\mu_{D_2} - \mu_{D_1}$ increases, the highly conserved residues will become distinct from that of the moderately conserved residues and the *global* patterns of conservation will rise.

In examples (2.1) and (2.2), where D_2 is placed at 0.7 and 0.8 respectively, the first constraint is violated; i.e., the mean of D_2 does does not exceed 0.80. However, as the mean of D_2 increases to 0.9 and 0.95 in examples (2.3) and (2.4), constraint one is met and ImPACT thresholds of 0.7951 and 0.8405 are calculated. It appears that ImPACT is generating appropriate thresholds.

4.3.4.3 Generating thresholds as D_1 decreases

This set of artificial data is analogous to that of the second set in that the distance between D_1 and D_2 is being varied. Here, however, the placement of D_1 is being varied rather than the placement of D_2. Therefore, unlike in the previous set of examples, the first constraint should always be met; indeed, as Figure 4.29 shows, this is the case. As the distance between the two underlying Gaussians increases, it is expected that the ImPACT threshold generated

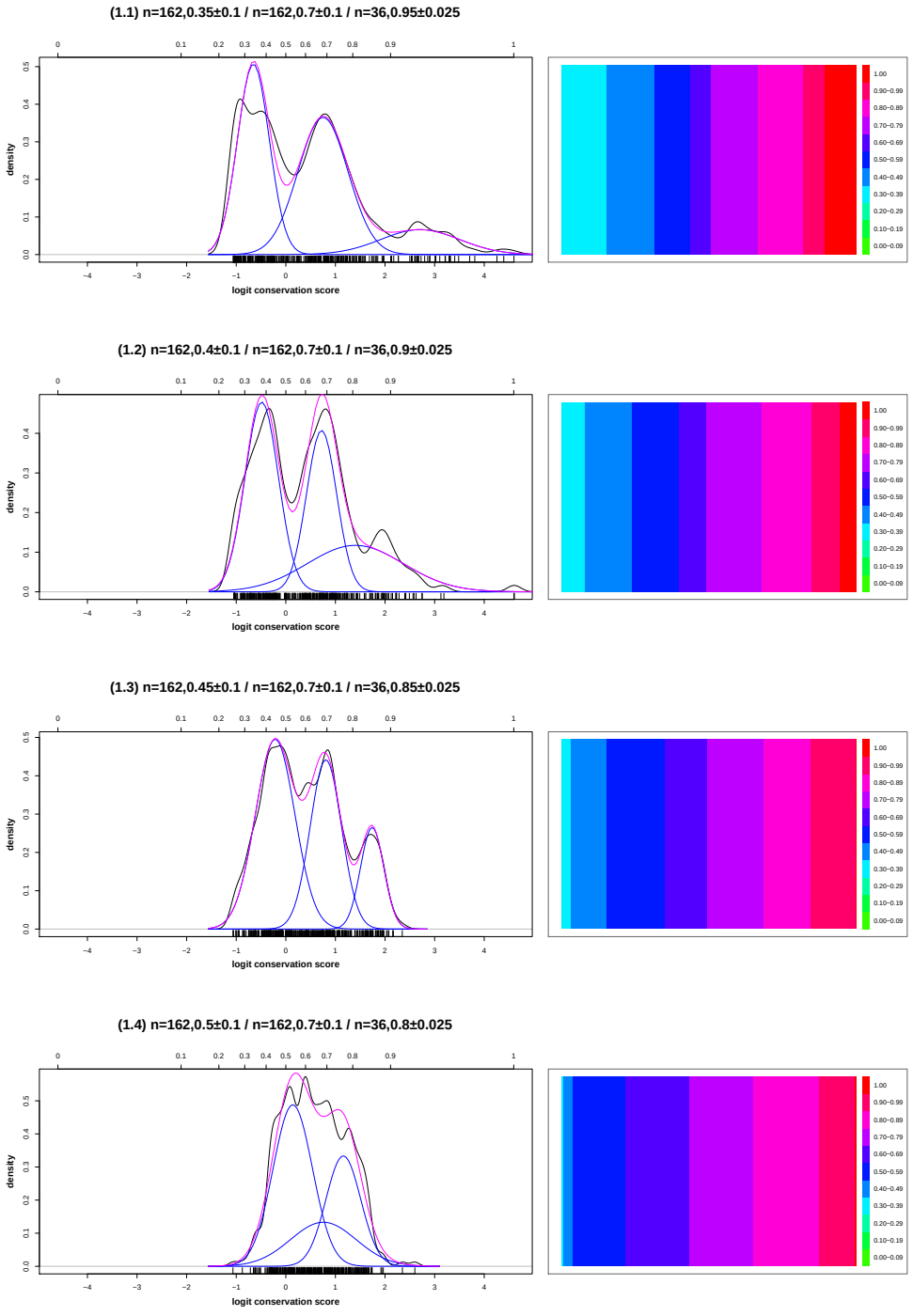


Figure 4.27: Assessing ImPACT using artificially generated data: fitting a mixture model

Parameters for the three underlying Gaussian components of test tests (1.1)-(1.4) are shown in Table 4.12. Distribution of logit transformed raw data is shown in black, the three fitted Gaussians are shown in blue and the cumulative model is shown in magenta. In this set of examples, no threshold has been generated. In addition to the fitted Gaussian components, to aid comparison of each protein's global conservation trends, a depiction of conservation trends for each protein's MSA is shown to the right of the graph, where high conservation scores are shown in red, more moderate scores are shown in blue and low scores are shown in green.

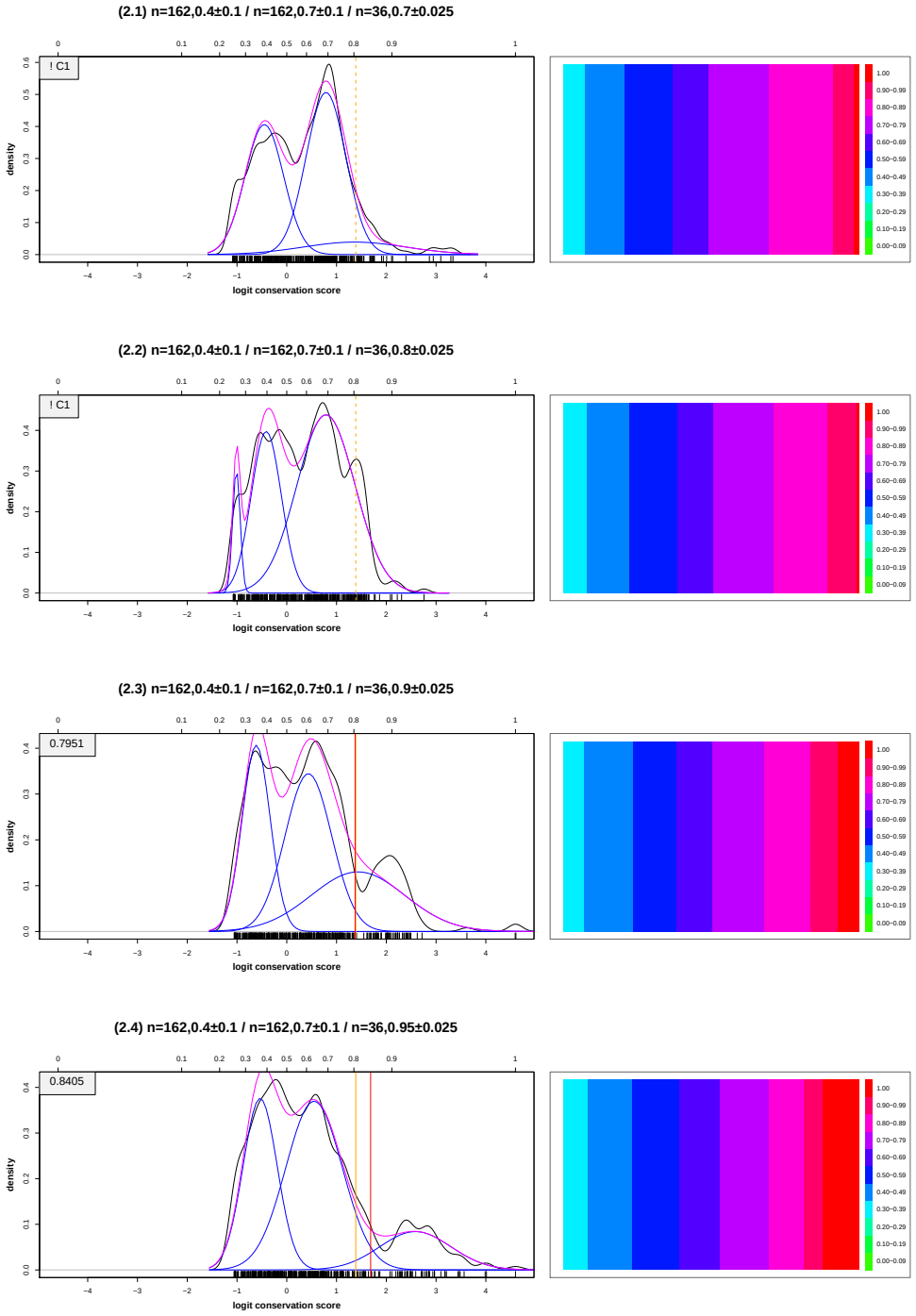


Figure 4.28: Assessing ImPACT using artificially generated data: increasing D_2

Parameters for the three underlying Gaussian components of test tests (2.1)-(2.4) are shown in Table 4.12. Distribution of logit transformed raw data is shown in black, the three fitted Gaussians are shown in blue and the cumulative model is shown in magenta. The values for the first and second constraints (see Section 4.2.2) are depicted as vertical orange and red lines respectively. The resulting threshold is given in the grey box in the top-left corner of the graph. In addition to the fitted Gaussian components, to aid comparison of each protein's global conservation trends, a depiction of conservation trends for each protein's MSA is shown to the right of the graph, where high conservation scores are shown in red, more moderate scores are shown in blue and low scores are shown in green.

will decrease to reflect the ease with which D_1 and D_2 can be distinguished and to reflect the reduction of global conservation patterns.

As shown in Figure 4.29, as μ_{G_1} decreases from 0.9 to 0.7 and $\mu_{D_2} - \mu_{D_1}$ increases, the ImPACT threshold generated decreases appropriately.

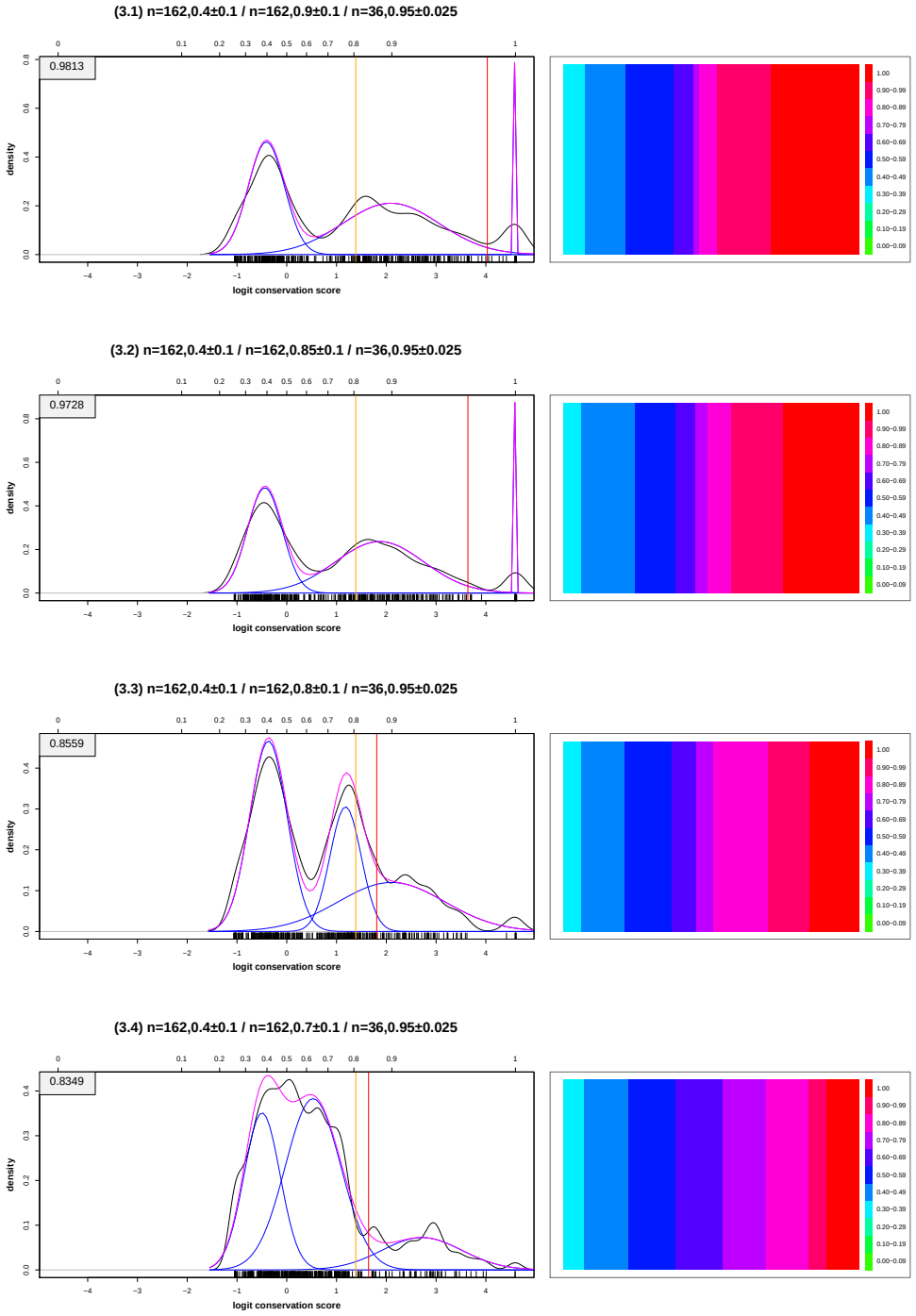


Figure 4.29: Assessing ImPACT using artificially generated data: decreasing D_1

Parameters for the three underlying Gaussian components of test tests (3.1)-(3.4) are shown in Table 4.12. Distribution of logit transformed raw data is shown in black, the three fitted Gaussians are shown in blue and the cumulative model is shown in magenta. The values for the first and second constraints (see Section 4.2.2) are depicted as vertical orange and red lines respectively. The resulting threshold is given in the grey box in the top-left corner of the graph. In addition to the fitted Gaussian components, to aid comparison of each protein's global conservation trends, a depiction of conservation trends for each protein's MSA is shown to the right of the graph, where high conservation scores are shown in red, more moderate scores are shown in blue and low scores are shown in green.

4.4 Conclusions

SAAPdb is primarily concerned with *structural* analysis of single amino acid polymorphisms, with a view to explaining deleterious phenotypes. However, where structural information is not available or not informative, functional relevance can be inferred from sequence conservation in multiple sequence alignments (MSAs). MSAs are products of the protein characteristics, the species set represented and genuine functional traces. In this chapter, a method (ImPACT) for isolating the genuine functional data to define high conservation within an MSA and its multi-faceted evaluation have been described.

Given difficulties in evaluation of conservation scoring methods, two *qualitative* evaluations of ImPACT were carried out. The first considered four 'representative' proteins (P53, G6PD, OTC and haemoglobin) and the second considered artificial conservation data. Both qualitative analyses demonstrated that ImPACT is successful in selecting appropriate thresholds for high conservation. A third more quantitative evaluation of ImPACT used PROSITE motifs to define which residues, in an alignment of functionally equivalent proteins (FEPs, see Chapter 3), should be classified as highly conserved by the ImPACT threshold.

A concern over the definition of a negative example (i.e., those residues that should not be considered as highly conserved) limits the extent to which the PROSITE benchmarking may be considered a fair evaluation of ImPACT. PROSITE records sequence motifs, but other residues may also be conserved for structural (e.g., hydrogen-bonding) or functional (e.g., ligand binding) reasons. The observation that the analysis of many (approximately 25%) PROSITE proteins generated a positive, non-zero FPR for *all* thresholds indicates that it is indeed common for a residue not identified in a PROSITE motif to be 100% conserved.

Given the concerns regarding negative examples, the most accurate measure with which ImPACT can be evaluated using these data is the TPR (see Section 2.3.5). A TPR of 100% is achieved in 99/231 (42.86%) of the proteins in the PROSITE dataset using the ImPACT threshold. As maintained throughout this chapter (and, indeed, throughout this thesis), the aim is to be conservative in the predictions that are made. For example, the first constraint is set reasonably high (0.80) and a strict leniency of 2 is set for the extraction of conserved residues in PROSITE motifs, which, as a result of limiting over-prediction (i.e., limiting the number of FPs), will increase under-predicting (i.e., increasing the number of FNs). This conservative approach

will, at least in part, explain why a 100% TPR is not found more often in the PROSITE datset.

Despite concerns with regards to the definition of negative examples, good performance with respect to MCC and ROC plots was observed in many cases, with ImPACT often approaching optimal performance. Most informative, however, is the close consideration of cases where sequence conservation data fails to identify PROSITE motif residues. For example, the alignment of RNC_HUMAN and its corresponding (strict) FEPs have large insertions (Figure 4.20(b)), but the proteins are clearly functionally equivalent given their UniProtKB/Swiss-Prot annotations. It is apparent that where genuinely functionally equivalent, but evolutionarily distant proteins, are aligned, extensive inserts could dominate the distribution of conservation scores, in that many columns will have very low conservation scores. A valuable addition to ImPACT would be to consider only the conservation scores of columns that are adequately represented in all proteins in the alignment (for example, SIFT (Ng and Henikoff, 2001) limits predictions to those columns with at least 50% MSA coverage).

ImPACT performance may improve if the set of PROSITE motifs were refined. PROSITE sequence motifs can be both functional (e.g., an N-linked glycosolation site) and indicative of homologous protein families (e.g., an apple domain). It may be more appropriate to benchmark ImPACT against only those *family* motifs described by PROSITE, as the functional motifs may only apply to the single protein. However, it is not straightforward to identify which PROSITE motifs are functional and which describe protein families. A more straightfoward approach would be to constrain the dataset to those residues within PROSITE motifs, with the negative examples being those that do not make the leniency threshold, for example, those residues in Figures 4.9(a)-4.9(b) that are *not* marked with an asterisk.

In summary, three-component Gaussian mixture modelling is sensitive to small but significant changes in the distribution of conservation scores and is able to model the changes accordingly. The ImPACT threshold generation method provides appropriate thresholds as patterns of global conservation vary, as observed in the artificial alignment dataset. Where some belief is held as to what the threshold for high conservation should be, as in the case of the four representative proteins, the ImPACT threshold is consistent with expectations. By analysing the performance of ImPACT using sequence motif data, while being aware of caveats regarding the dataset, it is evident that ImPACT often approaches near optimal performance, when compared with static thresholds.

Chapter 5

SAAPdb: The analysis pipeline

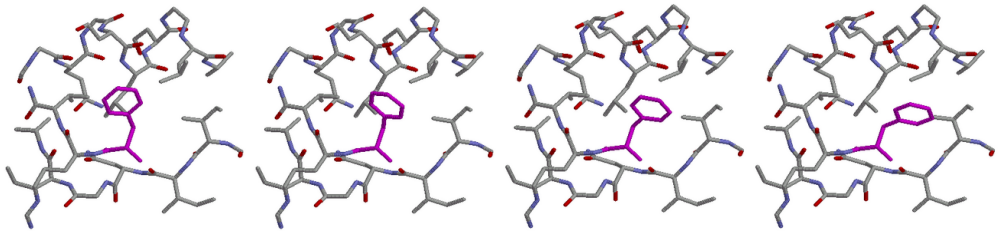
SAAPdb is a database of disease-causing and neutral mutations, which have been analysed to assess what effect, if any, they may have on protein structure and therefore function. The hypothesis is that disease mutations will more often affect protein structure, thus introducing a deleterious phenotype. SAAPdb attempts to identify the structural effect and therefore *explain* the mutation[1]. The development of a conservative, comprehensive structural analysis pipeline with which to analyse SAAPs is one of the main aims of the SAAPdb project. In this chapter, the suite of analyses with which SAAPdb assesses each mutation is described.

5.1 Introduction

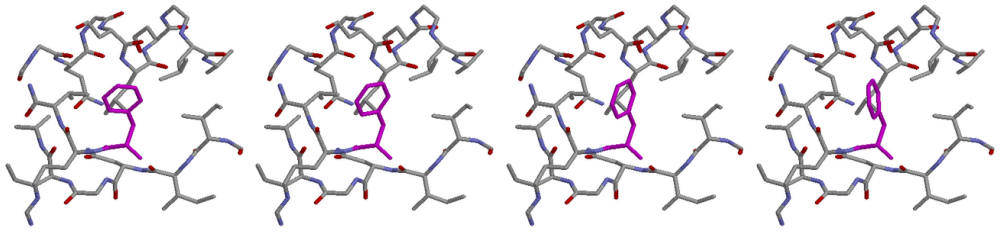
The SAAPdb structural analysis pipeline aims to identify whether a mutation will affect the native protein structure. Therefore, motivating the choice of structural analyses to include in the pipeline are the known fundamentals of protein structure: hydrogen bonding, interactions with ligands, characteristics of the protein core and so on (see Section 1.5 for an introduction to protein structure).

Currently, the pipeline consists of fifteen structural analysies, of which eight have been developed as part of this thesis. A ninth novel sequence-based analysis (ImPACT) was described in

[1] As stated in Chapter 1, the term 'explain' or 'explanation' is used to refer to predicted structural effects even for neutral mutations where there is no phenotypic effect to be explained



(a) Rotating the mutant residue (shown in magenta) about χ_1 in 30° increments



(b) Rotating the mutant residue (shown in magenta) about χ_2 in 30° increments

Figure 5.1: Using mutmodel to model a mutant residue into an existing structure: rotation about the χ angles.

Chapter 4. The pipeline is implemented in python as a series of 'wrappers', allowing information to be passed to each analysis by the main driver program. The analyses themselves are implemented in various languages, including C, Perl and SQL queries and functions.

This chapter will first describe the method by which mutant structures are generated where necessary (Section 5.2) and then describe the seven analyses that were developed previously for SAAPdb in Andrew Martin's group (Section 5.3). It will then describe the implementation and incorporation of the nine new analyses that have been developed (Sections 5.4-5.12).

Some of the work presented in this chapter has been published elsewhere (Martin *et al.*, 2002; Kwok *et al.*, 2002; Cuff and Martin, 2004; Cuff *et al.*, 2006; Hurst *et al.*, 2009).

5.2 Generating mutant structures

For two of the analyses described in this chapter—the void and clash analyses—it is necessary to generate a mutant structure. A minimum pertubation protocol (MPP) (Shih *et al.*, 1985; Snow

and Amzel, 1986) has been used to model the mutant residue into the native structure.

The method is as follows:

(I) Use maximum overlap protocol (MOP) (Snow and Amzel, 1986) to replace the sidechain, inheriting torsion angles from the native residue where possible

(II) Identify neighbouring residues within 8Å of the residue

(III) Rotate the sidechain around χ_1 (Figure 5.1(a)) and χ_2 (Figure 5.1(b)) and record whether a bad contact is made or not (a bad contact is defined as two atom centres within 2.50Å of each other)

(IV) If the MOP conformation makes ≤ 1 bad contacts the conformation is accepted; otherwise a choice is made from the set of conformations generated in step III

(V) If no rotamer exists that makes ≤ 1 bad contacts, the one with the least number of bad contacts is chosen

With a view to being conservative in 'explaining' mutations, sidechain replacements that clash with two or fewer other residues are considered acceptable. As described above, a clash is defined as two atom centres that are within 2.50Å of each other.

5.3 Existing analyses

The analyses described in this section have been previously published in Martin *et al.* (2002), Cuff and Martin (2004) and Cuff *et al.* (2006). They have been used elsewhere to explain disease mutations in disease-specific example datasets, including P53 (Martin *et al.*, 2002) and G6PD (Kwok *et al.*, 2002). They will be described briefly in this section, together with information about how they are integrated into the analysis pipeline.

5.3.1 Disrupting native hydrogen bonding

Hydrogen bonding is critical to maintaining the native protein fold. Using a grid-based approach, Cuff *et al.* (2006) analysed the occurrence and geometry of hydrogen bonds in the PDB for each hydrogen bonding donor and acceptor residue pair. Hypothetical mutant structures can then be compared with the observed hydrogen bonding residue profiles to assess whether a hydrogen bond is possible or not using the program `checkhbond`, which is available for use over the web at `http://www.bioinf.org.uk/`.

Each mutation must be analysed by `checkhbond`, but the algorithm is designed to be fast and each mutant structure does not need to be modelled: only the native structure is required by `checkhbond`. The 'pseudo-energy' score generated by `checkhbond` is extracted and stored in the SAAPdb database. The pseudo-energy score uses data on the likelihood that a hydrogen bond exists between two given residues for a given geometry and approximates the energy for the interaction, where a score of 0 implies that it is very unlikely that a hydrogen bond is formed. At present this processing is done sequentially on one machine although it is suitable for distributed processing.

Mutations that break hydrogen bonds (i.e., those with a pseudo-energy score of 0) are identified between backbone/sidechain and sidechain/sidechain donor and acceptor atoms.

5.3.2 Mutations at the interface

Residues at the interface between PDB chains, or between chains and ligands, will be critical in forming the biologically relevant multimer. Mutating such residues may disrupt native structure and may be deleterious. Interface residues are identified by a $> 10\%$ ΔASA (accessible surface area) in the monomer state as compared with the multimer state. ASA is calculated using a local implementation of the Lee and Richards algorithm (Lee and Richards, 1971). These data are obtained from XMAS files, an existing local resource of XML/ASN.1-like formatted PDB files (see Section 2.2.3).

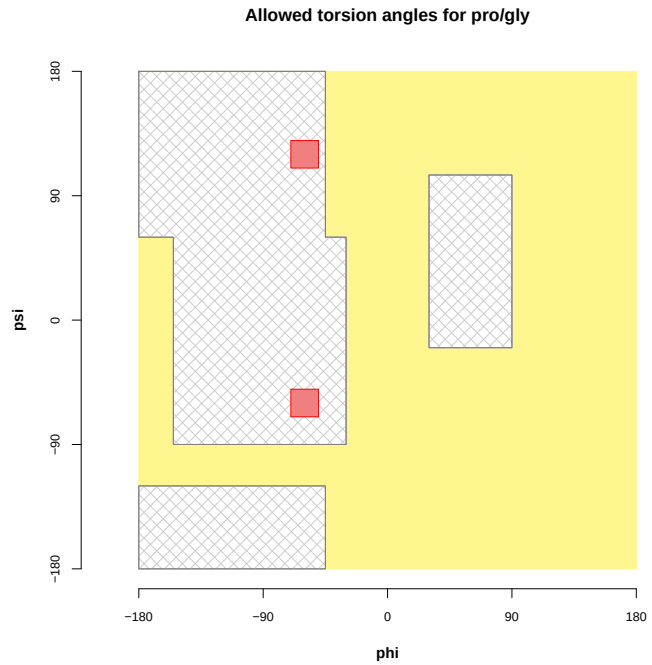


Figure 5.2: Allowed regions for proline and glycine
The pink areas mark the restricted comformation for proline residues, the hatched grey area marks the regions for non-proline, non-glycine residues, and the pale yellow colour marks the rest of the conformational space, primarily occupied by glycine residues.

5.3.3 Mutations to binding residues

Many PDB structures describe proteins in complex with ligands and other proteins; mutations to protein-ligand or protein-protein binding residues will hinder native protein function. Mutations to residues that form hydrogen bonds, as described by Baker and Hubbard (1984), and 'non-bonds' are identified by parsing the XMAS formatted PDB files (see Section 2.2.3). Non-bonds are formed between non-consecutive, inter-residue atoms that do not meet the criteria of Baker and Hubbard (1984) and whose centres are within 2.7-3.35Å of each other; this will include Van der Waals contacts and electrostatic interactions. Residues meeting these criteria will be a subset of the interface residues described in Section 5.3.2.

5.3.4 Mutations to proline

The cyclic nature of the proline sidechain limits the conformations which the residue can adopt. It is therefore likely that introducing a proline where the torsion angles are unfavourable will distort the protein structure or inhibit folding entirely. X→P mutations are identified outwith the region: $-70° \leq \phi \leq -50°$ and ($-70° \leq \psi \leq -50°$ or $110° \leq \psi \leq 130°$). In Figure 5.2, this area is marked in pink.

5.3.5 Mutations from glycine

Glycine has no sidechain and so can adopt backbone conformations that other amino acids cannot. Replacing a glycine with another amino acid, where the torsion angles are unfavourable, will affect protein structure. G→X mutations that occur outwith the region ($-180° \leq \phi \leq -30°$, $60° \leq \psi \leq -180°$) or ($-155° \leq \phi \leq -15°$, $-90° \leq \psi \leq 60°$) or ($-180° \leq \phi \leq -45°$, $-180° \leq \psi \leq -120°$) or ($30° \leq \phi \leq 90°$, $-20° \leq \psi \leq 105°$) are identified. In Figure 5.2, this area is coloured yellow.

5.3.6 Mutations that cause steric clashes

It may not be possible to accommodate a larger mutant residue in the native structure without disrupting the fold, and therefore the function. MutModel calculates the number of steric clashes caused by introducing a mutant residue in a protein structure (Section 5.2). Mutations that can be modelled into the native structure without clashing with three or more other atoms are identified. As discussed in Section 5.2, two residues clash if any atom centres are within 2.50Å of each other.

5.3.7 Introducing a void in the core

Where the previous section considered small to large residue mutations, here, large to small residue mutations are considered. Replacing a large amino acid with a smaller one could affect protein stability by introducing an internal void or surface crevice. A void is defined as a cavity

or crevice with a protein structure that is not accessible to bulk solvent. The software AVP is used to identify and measure the size of internal voids in protein structures (Cuff and Martin, 2004). AVP allows independent probe sizes for definition of solvent and voids with probe radii of 1.4Å and 0.5Å respectively being used.

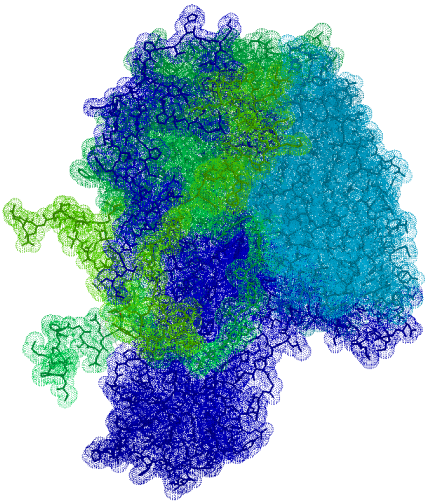
Obtaining these data requires significant preprocessing: all mutant structures must be generated using MutModel (see Section 5.2) before AVP is run on each individual structure. The compute time for each structure is dependent on the size of the protein chain being analysed, and can vary from a few seconds to several minutes.

5.4 Improving the analysis of disruption of quaternary structure

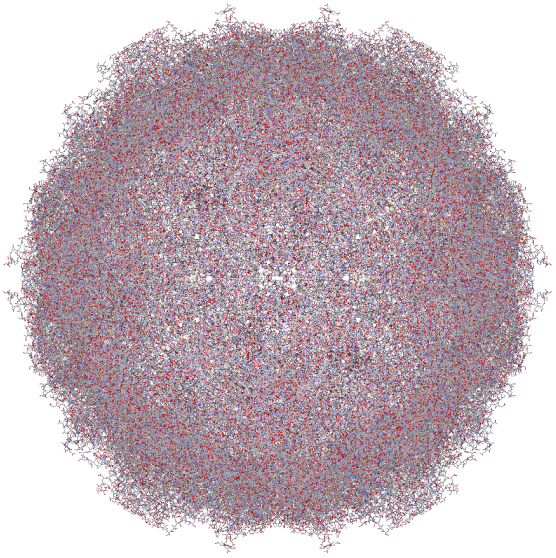
5.4.1 Background

The assembly of multiple tertiary protein structures into biologically relevant multimers is described as the quaternary structure (see Section 1.5). Residues at the quaternary interface will be critical to the native protein fold. The 'interface' and 'binding' analyses (Sections 5.3.2 and 5.3.3 respectively) attempt to identify mutations at the quaternary interface. However, this analysis is based on crystallographic unit cells from PDB files. These can have artificial crystal contacts or missing biologically relevant contacts (Janin, 1997).

The Protein Quaternary Structure (PQS) database describes hypothetical quaternary structures for PDB structures (Henrick and Thornton, 1998). All interatomic contacts $\leq$3.7Å for all space-group symmetry operations of the unit cell are calculated. Potential quaternary structures are assembled by the addition of monomeric chains; chains are selected based on the number of interchain contacts and the number of residues in the chain. Figure 5.3(b) shows the complete hypothetical quaternary structure of the Human poliovirus capsid protein 2plv with the original PDB structure shown in Figure 5.3(a). Although *some* of the binding contacts will be recognised by the binding and interface analyses, many will be lost (compare Figures 5.3(a) and 5.3(b) with respect to the number of interface surfaces).



(a) The PDB structure of Human poliovirus capsid protein



(b) The hypothetical PQS assembly of Human poliovirus capsid proteins

Figure 5.3: Quaternary structure information from PQS

Figure 5.3(a) shows the PDB representation (2plv) of the Human poliovirus capsid protein which has four chains. The biologically relevant structure, as assembled by PQS is shown in Figure 5.3(b) (*2plv.mmol*).

Further to identifying residues at the quaternary interface, the PQS database removes non-biological interactions that arise from the crystallisation process. The structure of the homo-dimer quinone reductase (PDB record 1qrd) is shown in Figure 5.4(a); the interface between the two chains is minimal. A more plausible homo-dimer is calculated by PQS and is shown in Figure 5.4(b) (in fact, PQS provides two similar alternative assemblies: *1qrd_1.mmol* (shown below) and *1qrd_2.mmol*). Several factors (ΔASA, interface size, interchain crosslinks and change in solvation energy) contribute to the discrimination between true macromolecular contacts and crystal packing artefacts. The analysis of PQS structures has been used elsewhere to identify biologically relevant contacts (Salama *et al.*, 2001).

Interchain contacts derived from the analysis of PQS structures are more likely to be genuine, biologically relevant interactions than those derived from an analysis of PDB structures.

5.4.2 Incorporating PQS information into the pipeline

The PQS database is mirrored locally. For each mutation identified in a structure, all corresponding PQS files are identified (for example, for PDB record 1qki, there are two corresponding PQS files: *1qki_1.mmol* and *1qki_2.mmol*). Each PQS file containing the relevant chain is retained for analysis.

PQS files are generated from PDB structures and maintain the PDB numbering. It is therefore straightforward to map PDB residues to PQS residues. To identify interface residues in the PQS structure, the following method was used:

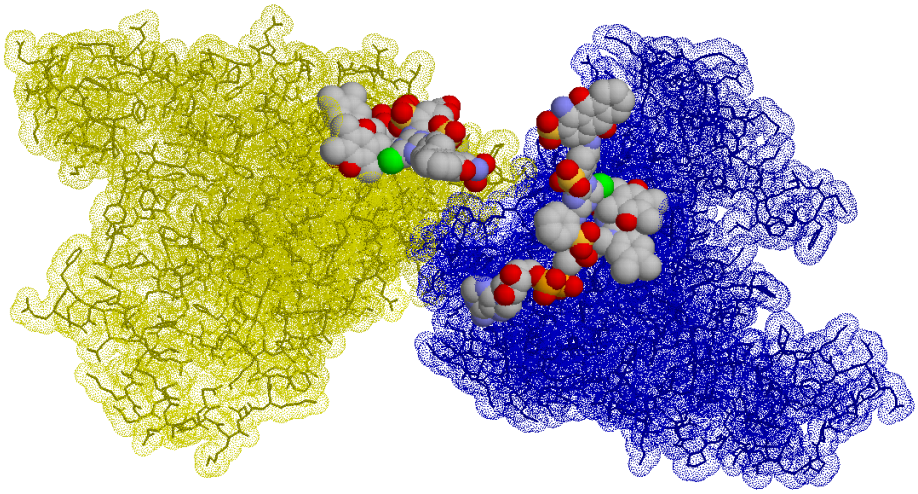
(I) For each PQS record:

  (a) Strip the waters from the structure

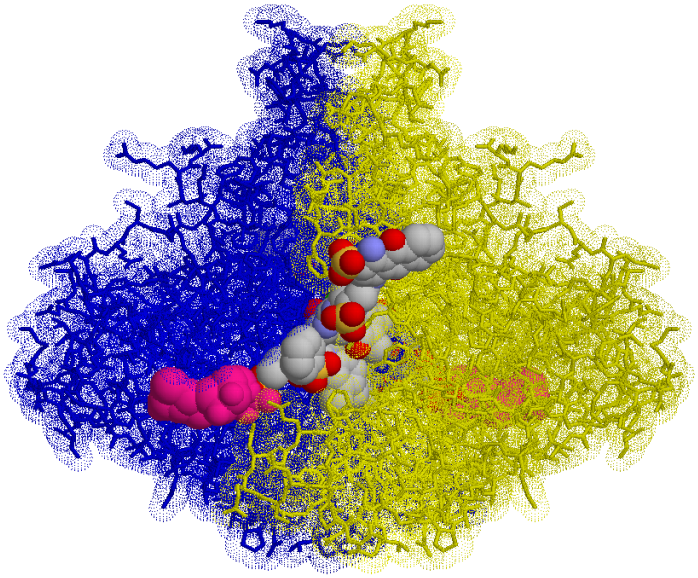
  (b) Convert the PDB-formatted PQS data into XMAS format (see Section 2.2.3)

  (c) Parse the generated XMAS file to identify residues with $> 10\%$ increase in relative accessibility as a monomer as compared with the multimer structure

This process is identical to that of the interface analysis described in Section 5.3.2, but PQS structures are used rather than PDB structures.



(a) PDB structure (1qrd)



(b) Suggested PSQ structure: `1qrd_1.mmol`

Figure 5.4: Artificial X-ray contacts in PDB unit cells

PDB describes quinone reductase as a homo-dimer in the PDB structure 1qrd (Figure 5.4(a)). A more plausible homo-dimer is calculated by PQS (*1qrd_1.mmol*) and is shown above in Figure 5.4(b).

As each PQS structure can be processed independently, these analyses were distributed across the local 20-core grid. A Perl script was written to analyse the PQS data and to distribute it cleanly across the grid using the Sun GridEngine.

388 452 PQS interface residues were identified in 7 487 PDB chains.

5.5 Mutations to binding residues (MMDBBIND)

5.5.1 Background

MMDBBIND (Salama *et al.*, 2001) is an assimilation of the three-dimensional structure information described by Entrez's MMDB database (Wang *et al.*, 2007b) and the mmCIF PDB chemical component dictionary[2] (Feng *et al.*, 2003), and is part of the larger BIND database (Bader *et al.*, 2003; Bader *et al.*, 2001). MMDB itself is a refined and extended representation of the PDB: the data are represented in ASN.1 format; multiple conformations for atoms are removed; all non-standard or modified residues are explicitly annotated, and binding and secondary structure data are explicitly recorded[3].

To identify binding residues in MMDB structures, all inter-molecule residue pairs (a) within a 10Å radius, and (b) with a van der Waals interatomic distance of ≤ 0.5Å are identified. Redundant interaction records are removed and the PQS database (see Section 5.4) is used to remove nonbiological interaction artefacts that arise from the crystallography process.

This analysis identifies all small-to-medium range interaction types between proteins, DNA (excluding complementary DNA interactions, i.e., DNA base pairing), RNA and small molecules in the PDB. Mutations to residues that form intermolecular contacts are likely to disrupt native protein function and therefore cause disease. In effect, MMDBBIND provides a refined version of the 'binding' analysis described in Section 5.3.3.

[2] formerly the HET group dictionary
[3] `http://www.ncbi.nlm.nih.gov/Structure/MMDB/mmdb.shtml`

```
>65310|1BOM|383|B|A|809234|809233
xqpqavhTYCGrhLarTLadLCweagvd
gIVdeCClrpcsVdvLLsYC
```

Figure 5.5: An example of an MMDBBIND record
In the header line, `65310` is the BIND identifier; `1BOM` is the PDB ID; `383` is the MMDB ID; `B` names the first sequence — given on the second line — as chain B of `1BOM`; `A` names the second sequence — given on the third line — as chain A of `1BOM`; `809234` is the GI (Genbank ID) of molecule A and `809233` is the GI of molecule B. GIs are taken from the MMDB database. Binding residues are highlighted in the sequences, using capital letters.

5.5.2 Incorporating MMDBBIND data into the pipeline

To include the MMDBBIND data in SAAPdb, the content of the MMDBBIND flat file[4] is parsed for intermolecular contacts involving at least one protein sequence. The annotation in this file does not define binding partners or describe the nature of the bond, it simply annotates a residue as binding or non-binding.

Figure 5.5 gives an example of an MMDBBIND record. The record consists of a header line (preceeded by the $>$ symbol) and two annotated sequences. The header describes which chains in which PDB structure are being annotated and the sequences provide the annotation, where binding residues are identified using capital letters. The sequences provided are derived from the SEQRES records of the PDB files, which do not necessarily correspond directly with the sequence derived from the amino acids in the structure. To include MMDBBIND annotations in the pipeline, the numbering must be resolved with respect to the residues described by the structure.

First, the method ensures that the sequences described in the MMDBBIND record are identical to the `SEQRES` records in the named PDB structure. If this is the case, the numbering is resolved as follows:

(I) For each record containing at least one protein sequence:

(a) Record seq_{seqres} : the sequence as described by the `SEQRES` record

(b) Record seq_{str} : the sequence as described by the `ATOM` records

(c) Record $first_{seqres}$: the number describing the first residue in seq_{seqres}

[4] `http://bond.unleashedinformatics.com/downloads/data/BIND/data/MMDBBIND/mmdbbind.txt`

(d) Record $last_{seqres}$: the number describing the last residue in seq_{seqres}

(e) Identify flanking regions ($flank_{pre}$ and $flank_{post}$) described in seq_{seqres}, but not in seq_{str}

(f) Calculate the offset as:
$offset = (num_{first} - flank_{pre}) - 1$

(g) To ensure correct numbering, check whether the num_{last} is as expected using the offset value. The value should be:
$num_{first} + length(seq_{seqres}) - (flank_{pre} + flank_{post}) - 1$

If the value of num_{last} is not as expected, then the sequence of residues given by the `ATOM` records is different to that of the `SEQRES` records (most commonly, residues in flexible loop structures are missing). Only those MMDBBIND data that satisfy both criteria are recorded, for structures to which a SAAP has been mapped in SAAPdb. Any interactions described in the MMDBBIND record are numbered using the offset value as calculated above.

This method does result in more MMDB data being rejected than is necessary and there is definite scope for improvement using alignments of the `SEQRES` and `ATOM` records. One approach for improving the verifation of this numbering is included in the discussion (Section 5.13).

A set of Perl methods was developed to handle the MMDBBIND data. First, the MMDBBIND flatfile[4] is parsed, and all relevant, verified data are recorded in a second tab-delimited file. A second set of methods generate SQL statements for these data and record them in the database.

94 277 interacting residues are identified in 3 731 PDB chains.

5.6 Disrupting disulphide bonding

5.6.1 Background

Disulphide bonds (see Section 1.5.2) are crosslinks that form between cysteine residues in polypeptides and stabilise protein structure (Figure 5.6). Mutations to disulphide bonding cysteines may compromise protein stability and therefore compromise native protein function.

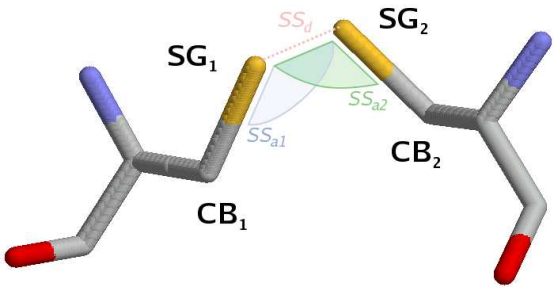


Figure 5.6: A disulphide bond between CYS6 and CYS127 of lysozyme (7lyz), showing SS_d, SS_{a1} and SS_{a2}

5.6.2 Incorporating disulphide data into the pipeline

A Perl script was written to identify potential disulphide bonding cysteine residues in PDB files. First, all cysteine residues are identified. Secondly, each pair of cysteine residues is assessed as to whether it forms a disulphide bond. The residues must satisfy the following criteria (Hazes and Dijkstra, 1988):

- Sγ_1–Sγ_2 bond length should be $\leq$2.50Å
- Cβ_1–Sγ_1–Sγ_2 and Cβ_2–Sγ_2–Sγ_1 bond angles should be $104° \pm 10\%$

Standard trigonometry calculations and methods from the Perl `Math::Trig` module were used to calculate distances and angles from PDB coordinates.

Each protein structure described in SAAPdb is analysed to identify potential disulphide bonding cysteine residues. Isolated PDB chains are used, as interchain disulphide bonding will be identified by the PQS analysis. All candidate sulphur atoms from cysteine residues are extracted from the PDB file. All possible pairs of the cysteine-sulphur atoms are considered and those that meet the criteria described by Hazes and Djikstra (1988) are recorded as disulphide bond partners.

The computational pre-processing for this analysis is comparatively light: PDB files are parsed and simple calculations are carried out to calculate potential bond angles and distances. The pre-processing therefore need not be distributed across the grid and each structure (PDB file)

is processed sequentially. The script analyses the PDB structures, extracts potential disulphide bonding partners and generates the corresponding SQL to record disulphide bonding cysteine residues in SAAPdb. Multiple occupancy cysteines are processed as any other cysteine; that is, the atoms for each alternative conformation are grouped together and each alternative conformation is considered as a disulphide bonding cysteine.

15 963 potential disulphide bonds are identified in 9 223 PDB structures.

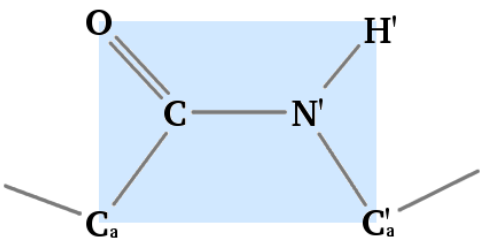
5.7 Mutations to cisprolines

5.7.1 Background

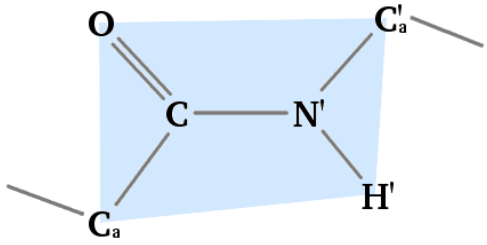
The peptide bond forms a partial double-bond between the carboxylate C and amide N atoms of amino acids. Energetically, this favours two conformations where the C_α, O, C, N', H', C_α' atoms form a planar unit (i.e., those atoms exist in the same plane): the *trans* conformation where $\omega \simeq 180°$ (Figure 5.7(a)) and the *cis* conformation where $\omega \simeq 0°$ (Figure 5.7(b)). The vast majority of peptide bonds are found in the *trans* conformation: the proximity of C_α and C_α' makes the *cis* conformation less stable.

However, peptide bonds between any residue and proline (Xaa-Pro) more readily adopt the *cis* conformation than other peptide bonds (Xaa-nonPro). The *cis* conformation is more than 1000 times less stable than the *trans* conformation in Xaa-nonPro peptide bonds, while the *cis* conformation is only four times less stable than the alternative *trans* conformation in Xaa-Pro peptide bonds (Branden and Tooze, 1999). It has been shown that approximately 5-6.5% of Xaa-Pro bonds are *cis*, and 0.03-0.05% of Xaa-nonPro are *cis* (Jabs *et al.*, 1999; Stewart *et al.*, 1990). Figure 5.7(c) shows a histogram of cispeptide frequency for each amino acid in a representative set of CATH v3.0.0 HReps (Pearl *et al.*, 2003). Although the Xaa-nonPro peptide bond can adopt the *cis* conformation, it is clear that *cis* peptide bonds are predominantly between a non-proline and a proline residue.

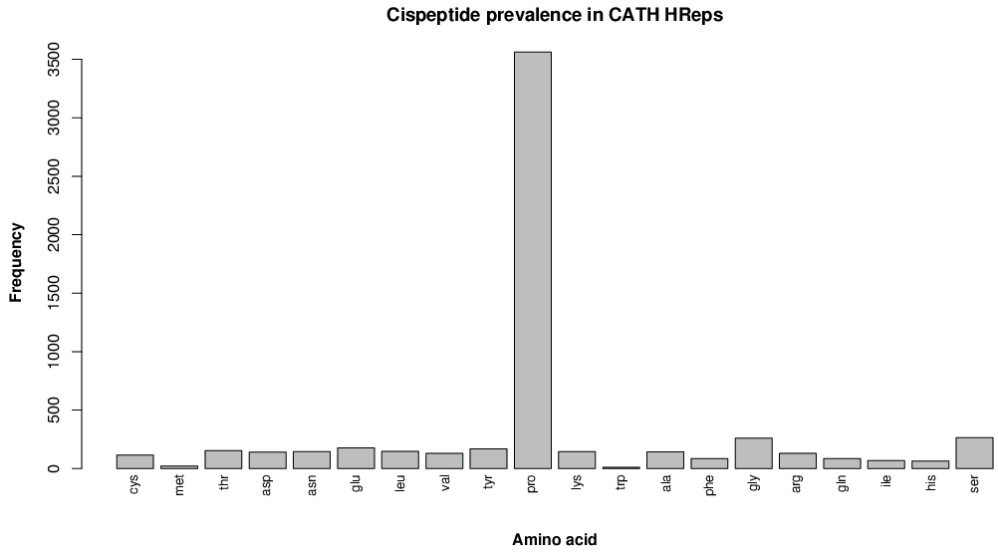
Thus, a mutation from a cis-proline to another amino acid, forcing an Xaa-nonPro peptide bond to adopt a *cis* conformation, is likely to destabilise the protein structure.



(a) The *cis* conformation of the peptide bond



(b) The *trans* conformation of the peptide bond



(c) *cis* peptide prevalence in CATH v3.0.0 HReps
The raw frequencies of *cis*-peptide bonds in CATH v3.0.0 HReps was calculated. The **Amino acid** defines the second residue in the peptide-bond. Most *cis*-peptide bonds are between Xaa-Pro residues.

Figure 5.7: The *cis*-peptide bond

5.7.2 Incorporating these data into the pipeline

The ω torsion angle measurements are calculated using the `torsions` program (Martin, unpublished) and are calculated when populating the `structural_analysis` table (see Section 6.2.7). The analysis can therefore be implemented as a simple SQL query which identifies mutations from proline to non-proline where $-90 < \omega < 90$.

4.40% (6 584/14 9642) of Xaa-Pro and 0.15% (4 342/2 946 025) of Xaa-nonPro peptide bonds described by SAAPdb adopt the *cis*-conformation. This is in agreement with existing data (MacArthur and Thornton, 1991; Weiss *et al.*, 1998).

5.8 Introducing a charge shift in the core

subsectionBackground

Charged residues are often functional in protein structures (Torshin and Harrison, 2001). Arginine and lysine, and histidine to a lesser extent, are positively charged and often form salt bridges with negatively charged groups, while aspartic acid and glutamic acid are negatively charged and able to form salt bridges with positively charged groups. These amino acids almost invariably occur as satisfied pairs of oppositely charged residues in the protein core (Torshin and Harrison, 2001). Removing or introducing a charged residue from or into the protein core may disrupt the fold and cause a deleterious phenotype. Surface charged residues are solvated and therefore do not need to occur as charge pairs.

It is of course possible for a charged residue on the surface to interact with other molecules and therefore be critical to protein function. However, these residues should be identified by the PQS, binding and/or MMDBBIND analysis which are described elsewhere (Sections 5.4, 5.3.3 and 5.5 respectively). Here, the focus is specifically on the effects of removing charge in the protein core.

Table 5.1: Charge shift values for mutations between charged and neutral residues
Mutations between residues that are identically charged do not generate a charge shift, mutations between oppositely charged residues generate a charge shift of ± 2, mutations between charged and neutral residues generate a charge shift of ± 1. Negative scores indicate a movement towards a more negative charge, positive scores indicate a movement towards a more positive charge.

Native charge	Mutant charge	Charge shift
positive	negative	-2
positive	neutral	-1
positive	positive	0
neutral	neutral	0
negative	negative	0
negative	neutral	1
negative	positive	2

5.8.1 Incorporating these data into the pipeline

This analysis does not require any additional processing as all the required data are parsed from the XMAS files (see Section 2.2.3). A PostgreSQL function was written to calculate the 'charge shift' of a mutation (see Appendix [E.i] for the definition of this function). Table 5.1 shows the charge shift values for mutations between all possible pairs of charged and neutral amino acids. With this PostgreSQL function, it is possible to implement this analysis as a single SQL query, where mutations with a non-zero charge shift occurring in the core (where the relative, monomer accessibility statistic $\leq 5\%$) are easily identified as introducing a buried, unsatisfied charge.

5.9 Introducing hydrophobic residues on the protein surface

5.9.1 Background

Hydrophobic residues are concentrated in the protein core (Branden and Tooze, 1999). Replacing a hydrophilic residue with a hydrophobic residue on the surface of a protein could result in protein aggregation or misfolding and therefore a deleterious phenotype (for example, the E6V mutation that causes sickle-cell anaemia, see Section 1.6).

5.9.2 Incorporation into the pipeline

Phenylalanine, isoleucine, leucine, methionine, valine and tryptophan are classified as hydrophobic and aspartate, glutamate, histidine, lysine, asparagine, glutamine, arginine, serine, threonine and tyrosine are classified as hydrophilic.

All data required to identify the hydrophobic mutations on the surface—i.e., native/mutant amino acids and accessibility statistics—are recorded when the XMAS format of each mapped PDB structure is parsed to populate the `structural_analysis` database table (see Section 6.2.7). The analysis can therefore be performed by a single SQL query. Mutations from a hydrophilic residue to a hydrophobic residue where the relative surface accessibility in the monomer state is $> 5\%$ are identified.

5.10 Introducing hydrophilic residues in the protein core

5.10.1 Background

Replacing a hydrophobic residue with a hydrophilic residue could destabilise the native protein fold based on the observation that the vast majority of hydrogen bonding capable sidechains participate in hydrogen bonding (McDonald and Thornton, 1994). Without potentially stabilising native hydrogen bonding in the protein core, native protein folding may be compromised.

5.10.2 Incorporation into the pipeline

Again, phenylalanine, isoleucine, leucine, methionine, valine and tryptophan are classified as hydrophobic and aspartate, glutamate, histidine, lysine, asparagine, glutamine, arginine, serine, threonine and tyrosine are classified as hydrophilic.

As described in the Section 5.9, the information required to identify the introduction of a hydrophilic residue in the protein core already exists in SAAPdb and no additional processing is required. As such, the analysis can be implemented as a single SQL query identifying muta-

tions from any hydrophobic residue to any hydrophilic residue where the relative accessibility of the residue in the monomer is $\leq 5\%$.

5.11 UniProtKB/Swiss-Prot features

5.11.1 Background

UniProtKB/Swiss-Prot uses a controlled vocabulary and the FT tag to annotate regions of interest in protein sequences[5]. A small number of these annotations are manual, however many more are transferred 'by similarity' from another annotated protein.

Many of these regions will be critical to protein function (for example, post-translational modifications and binding sites) and others will be critical to protein stability (for example, disulphide bonds and other crosslinks). Mutations to such residues could disrupt protein function.

5.11.2 Incorporating these data into the pipeline

The UniProtKB/Swiss-Prot DAT flatfile is parsed and residues annotated with FT tags are identified. As the aim is to explain the effects of mutations, a subset of features which have the potential to affect protein stability or function are relevant. These are described in Table 5.2.

In UniProtKB/Swiss-Prot, the FT tag annotations can describe the start and end of contiguous regions of annotation, or they can describe two non-adjacent residues (see third 'Numbering scheme' column of Table 5.2). If the start and end number are the same, it describes a single residue. When parsing the UniProtKB/Swiss-Prot data, the two numbering schemes are handled appropriately, annotating all residues between the start and end of contiguous feature regions with the corresponding feature. FT tag numbering that includes the non-digit characters ?, $<$ or $>$ is unreliable and these data are not extracted. After extracting the annotations from the UniProtKB/Swiss-Prot flatfile, all feature residues that have been extracted are stored in the database; 1 488 092 residues are annotated in 135 883 UniProtKB/Swiss-Prot records. The PDBSWS mapping (Martin, 2005) that is imported to SAAPdb allows these annotations to be

[5] http://www.expasy.org/sprot/userman.html#FT_line

Table 5.2: UniProtKB/Swiss-Prot FT annotations used to identify functional residues in SAAPdb

Feature tag: the UniProtKB/Swiss-Prot FT tag; **Description**: a description of the feature; **Numbering scheme**: what the UniProtKB/Swiss-Prot FT numberings describe - a contiguous region or a pair of non-adjacent residues.

Feature tag	Description	Numbering scheme
ACT_SITE	Residues involved in enzymatic activity	contiguous
BINDING	A ligand or substrate binding site	contiguous
CA_BIND	Residues involved in calcium binding	contiguous
DNA_BIND	A DNA binding site	contiguous
NP_BIND	A nucleotide phosphate-binding region	contiguous
METAL	A metal binding site	contiguous
LIPID	Residues binding to a lipid substrate	contiguous
CARBOHYD	A glycosylation site	contiguous
MOD_RES	A site of PTM	contiguous
MOTIF	A short sequence motif of biological interest	contiguous
DISULFID	Location of a disulphide bond	non-adjacent
CROSSLNK	Crosslinks formed after PTMs	non-adjacent

mapped to protein structure (see Section 2.1.5 for a description of PDBSWS and Section 6.2.2 for a description of how PDBSWS is imported into SAAPdb).

The mapping process used to populate SAAPdb requires that all mutations are mapped initially to a residue in a UniProtKB/Swiss-Prot record. With the relevant data extracted from the UniProtKB/Swiss-Prot DAT flatfile and stored in the database, this analysis can be implemented by a simple PostgreSQL query.

Upon closer inspection, it appears that these data can be unreliable. Figure 5.8 shows the structure of human P53 (PDB record 1tsr) in complex with DNA (highlighted in red). Residues near to the DNA (within 10Å) are shown in yellow. The corresponding protein record ([UniProtKB:P04637/P53_HUMAN]) describes residues 102-292 as DNA_BINDing. These residues are shown in blue in Figure 5.8, having been mapped onto the protein structure using PDBSWS (see Section 2.1.5), and comprise most of the protein chain. It is clear from this example that the UniProtKB/Swiss-Prot functional annotation is too coarse-grained, with many residues remote from the DNA (i.e., distant by > 10Å) being annotated as DNA_BINDing.

Given this observation, the UniProtKB/Swiss-Prot FT data have not been included in the later analysis stage (see Chapter 7). However, the data are retained in the hope that annotations will improve.

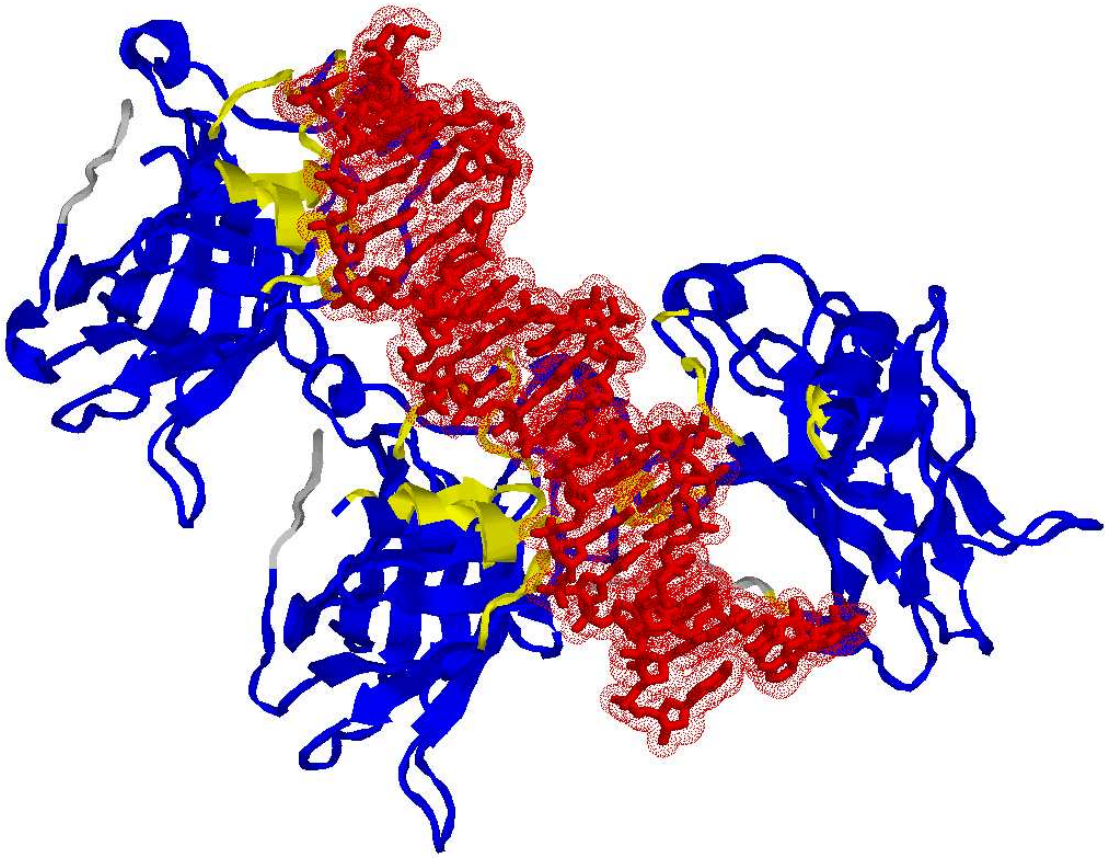


Figure 5.8: An example of coarse-grained UniProtKB/Swiss-Prot FT annotation

The figure shows the structure of human P53, PDB record 1tsr. The DNA to which P53 binds is shown in red with the protein chains shown as cartoon ribbons. The yellow residues indicate those within 10Å of the DNA, the blue residues are those annotated as DNA_BINDing by UniProtKB/Swiss-Prot, all other residues are coloured grey. Even using the very generous distance threshold of 10Å, the UniProtKB/Swiss-Prot DNA_BINDing annotation appears to be very coarse-grained.

5.12 Mutating conserved residues

5.12.1 Background

Where it is not possible to identify the structural effect of a disease mutation, sequence information can be used to infer functionality. Comparing the same protein in different species will highlight which residues are conserved and therefore likely to be critical to protein function and/or stability.

This led to the development of a novel method for identifying highly conserved residues which accounts for species diversity and protein-global conservation patterns. This method, called ImPACT, is described in Chapter 4. Here, the method by which these data are incorporated into the pipeline is described.

5.12.2 Incorporating ImPACT scores into the pipeline

In SAAPdb, each mutation must be mapped to a UniProtKB accession number to order to exploit the sequence-to-structure mapping in PDBSWS (see Section 2.1.5). Using these accession numbers, all functionally equivalent proteins (FEPs) for each protein (should it exist in UniProtKB/Swiss-Prot) can be identified by querying the FOSTA database. FOSTA is a method for identifying FEPs in UniProtKB/Swiss-Prot and is described in Chapter 3. The sequence data used to populate FOSTA are cloned for populating the ImPACT database so that all sequences can be retrieved, including records that have been replaced, merged or deleted since the last FOSTA run. A multiple sequence alignment (MSA) is generated by aligning the FEPs using MUSCLE (Edgar, 2004a) (see Section 2.3.2 for a description of this method). To identify a protein-specific threshold for high conservation, the distribution is modelled and analysed as described in Chapter 4.

As each protein can be processed independently, the ImPACT analyses are distributed across the local 20-core grid. For each MSA, the ImPACT threshold, target protein and size (i.e., number of sequences) are recorded, and for each residue in each MSA, the position (with respect to the target protein), the species similarity conservation score (see Section 4.2.1) and whether or not this exceeds the ImPACT threshold for the MSA are recorded. With these data recorded,

the sequence conservation analysis can be implemented as a single SQL query.

5.13 Discussion

The pipeline has been extended significantly as part of this thesis and analysis shows that the recent augmentation has been valuable with respect to explaining mutations (see Chapter 7). However, there remains considerable scope for improvement of the current analyses and incorporation of new ones.

Currently, the void analysis is rather crude: a static threshold of 275Å^3 is used to identify deleterious void creating mutations. This threshold was selected based on an analysis of PDB structures that showed that the largest void in 80% of protein structures is $\leq$275Å^3 (Cuff and Martin, 2004). However, it is likely that the threshold for deleterious void creation is dependent on the protein structure, its size and stability, its environment and its resistance to destabilising voids. Similar to what has been done for the sequence analysis—where MSA-specific thresholds for high conservation are calculated—it would be valuable to consider each protein structure individually, and, based on its properties, estimate the maximum void size that may be tolerated.

The existing method for resolving residue numbering in MMDBBIND should also be improved by dealing with structures with missing residues. Currently, the method corrects the numbering for leading and trailing flanking regions, but rejects any MMDBBIND record which describes a PDB structure with absent residues in the structure, i.e., where the `SEQRES` and `ATOM` records differ. This method could be improved by using the `SEQRES/ATOM` alignment provided in the XMAS files (see Section 2.2.3) to correctly map *all* MMDBBIND annotations to the PDB.

Further, the current definition of a clash—identifying atoms that are within 2.50Å of each other, as measured from the centre of the atom—could be improved. Using a static threshold does not differentiate between two residues that are slightly overlapping and two residues that are largely occupying the same space. Using a more informative Van der Waals energy calculation would refine the clash analysis and may improve results.

More generally, the set of structures within which the SAAPs are analysed could be refined. It may be beneficial to eliminate structures of lower resolution from the dataset, or to identify

the best structure within which to analyse the mutations. In addition, replacing the binding analysis with the corresponding PQS analysis would further refine the structural dataset, as PDB structures can contain apparent interaction artefacts that arise from the crystallography process as well as missing biologically relevant contacts (see Section 5.4).

With a view to extending the pipeline, it would be possible to incorporate external datasources such as the Catalytic Site Atlas (Porter *et al.*, 2004), PROCOGNATE (Bashton *et al.*, 2008) or dbPTM (Lee *et al.*, 2006) to identify functionally relevant sites in protein structures. In addition, it may be beneficial to consider the protein in a wider context, for example, its role in known pathways (Kanehisa *et al.*, 2008).

Chapter 6

The SAAPdb machinery and mechanics

In Chapters 4 and 5, the fifteen structural analyses that comprise the current suite of analyses in SAAPdb were described. This chapter will describe how SAAPdb is populated and how the pipeline is implemented.

Much of the work described in this chapter was developed by Jacob Hurst. The work completed as part of this thesis included porting and updating software to work on a new system with new versions of PostgreSQL and python; updating the pipeline to include the new analyses (see Sections 5.4-5.12); cache-ing of data; and retrieving, parsing and using the dbSNP XML data.

Some of the methods described in this chapter are described in Hurst *et al.* (2009).

6.1 Introduction

The raw data upon which SAAPdb is based describe two kinds of genomic variation, the first of which—the single nucleotide polymorphism or SNP—is assumed to have a negligible effect on protein structure and therefore function, and the second of which—the pathogenic deviation or

Table 6.1: Data overlap in SAAPdb

Numbers describe how many mutations are common to the two corresponding datasets. The emboldened identity numbers (i.e., where a dataset is compared with itself) show how many mutations are described by that dataset. The dbSNP and OMIM datasets are separated from the other LSMDB datasets using double ruled lines. Dataset names are self-explanatory (apart from 'P53-G' which represents the Germline IARC p53 Database and 'P53-S' which represents the 'Somatic IARC p53 Database'). All datasets are further described in Section 2.1.2. Empty cells indicate that there is no overlap between two datasets.

OMIM	**7119**									
ADABase	19	**38**								
G6PD	44		**103**							
HAMSTeRS	135			**526**						
P53-G	23				**94**					
P53-S	27				89	**1396**				
OTC	12						**148**			
SOD1db	27							**96**		
ZAP70	1								**5**	
dbSNP	3				3					**34081**
	OMIM	**ADABase**	**G6PD**	**HAMSTeRS**	**P53-G**	**P53-S**	**OTC**	**SOD1db**	**ZAP70**	**dbSNP**

PD—has been associated with disease and, therefore, are thought to have a deleterious effect(s) on protein structure and function. Here, the resources from which these data are taken are reviewed (they were introduced in Chapter 2).

6.1.1 SNP data

Two SNP resources are described in Section 2.1.1: dbSNP (Sherry *et al.*, 1999; Smigielski *et al.*, 2000) and HGVBase (Brookes *et al.*, 2000). However, as of October 2008, only dbSNP data are analysed by SAAPdb. The decision to eliminate HGVBase data was taken for two reasons, one of which is the data themselves and the second of which is the processing of these data.

Firstly, HGVBase has not be consistently maintained, with only sporadic updates since 2003. Secondly, HGVBase, unlike recent builds of dbSNP, does not provide reliable mappings to protein sequences. This requires that the genomic data be mapped to protein sequences via coding sequence assembly from genomic records, translation and ORF identification, and finally alignment and mapping with the referenced protein sequence or sequences. This is a computationally expensive process.

However, the analysis described in Chapter 7 was carried out before the HGVBase data were removed from the system and when the in-house mapping system was used to map *all* SNPs to protein sequence. As such, the method for importing these data is described below in Section 6.2.4.

Note that HGVBase has recently become HGVBaseG2P and it is expected that many of these problems will soon be resolved. However, at the time of writing, no downloadable set of mutations was available.

6.1.2 PD data

OMIM and LSMDBs (Locus Specific Mutation Databases) were discussed in Section 2.1.2. The resources are obtained in varying formats. The major challenges in processing LSMDB data are (i) to standardise the format of these data to allow them to be processed identically and (ii) to verify the sequence numbering that is provided by each LSMDB community.

6.1.3 SNP/PD overlap

Table 6.1 shows the size of the datasets currently used in SAAPdb and the overlap between them. The central PD resource OMIM has, as would be expected, at least some overlap with all of the other PD datasets. It is also by far the largest resource, being five times larger than the next largest, the somatic P53 dataset. However, larger still is the dataset of neutral non-synonymous SNPs, which is approximately five times larger than the OMIM resource. Within the LSMDBs, the only overlap that exists is between the germline and somatic P53 datasets.

Encouragingly, very few mutations are described as disease-associated *and* neutral. Only six mutations are described simultaneously as a PD and a SNP: three are common to the dbSNP and OMIM datasets and three are common to the dbSNP and P53 somatic datasets. When analysing the data in Chapter 7, these mutations are removed from the SNP dataset but retained in the disease dataset, working on the assumption that the large-scale genomic scanning technology by which the SNPs are identified happens to have sequenced the genome of an individual carrying a disease mutation.

It is worth noting that the unique complexity of cancer (where many mutations are acquired over a short period of time) means that there is less certainty as to the pathogenicity of those mutations found in both the somatic P53 dataset and the dbSNP dataset. Apparently carcinogenic mutations may simply be 'passenger' mutations that have little or no pathogenic effect, having 'hitchhiked' into the cancer cell by virtue of being coincident with a deleterious mutation (Greenman *et al.*, 2006). However, none of these three SNPs are mapped onto protein structures and are therefore not analysed in Chapter 7.

6.1.4 Additional resources

Several additional resources are required to processes these data: UniProtKB (Section 2.1.3) is required to map gene names to proteins and identify annotated functional residues; EMBL and Genbank (Section 2.1.6) are required to map genomic data to protein sequences where mappings are unreliable or absent; and PDBSWS (Section 2.1.5) is used to map protein sequences to protein structures accurately.

6.2 Materials and Methods

There are two main stages of data processing: (1) importing the SAAP data (PDs or SNPs) and (2) pushing it through the structural analysis pipeline. This naturally leads to a three-part data 'architecture' division, both with respect to data storage and data processing: (a) SNP data; (b) PD data and (c) pipeline data. These three sections are described below in Sections 6.2.3-6.2.4, 6.2.6 and 6.2.7 respectively, following a brief description of the database (Section 6.2.1) and information regarding the import of additional 'reference' data (Section 6.2.2).

Several other people have contributed to the design, development and maintainence of SAAPdb, including Jacob Hurst, James Allen, Craig Porter and Antonio Cavallo. Where appropriate, the contribution of each individual has been indicated in italics and marked with a ▷ symbol under the section heading.

6.2.1 The database

▷ *The database was designed and previously maintained by Jacob Hurst, James Allen, Craig Porter and Antonio Cavallo. It has been extended to include additional analysis data by Lisa McMillan.*

Figure 6.1 describes structure of the SAAPdb database. The database is divided into three sections: (a) handling the SNPs, (b) handling the PDs and (c) pushing the SAAP data through the pipeline. Within the SNP and PD sections, tables separate the storage of sequence and structural data (using `lsdb` and `lsdb_saap` for PD sequence and structural data respectively, and `snp2annotated` and `saap` for SNP sequence and structural data respectively). In the pipeline section, the storage of purely structural data (i.e., data pertaining to native structures, prior to mutation analysis) is kept separate (in the `structural_analysis` table) from the storage of mutation analysis data. The `mutanalysis` table contains the results of the structural analyses. In the pipeline section, summary tables are created to allow the fast retrieval of data via a webserver.

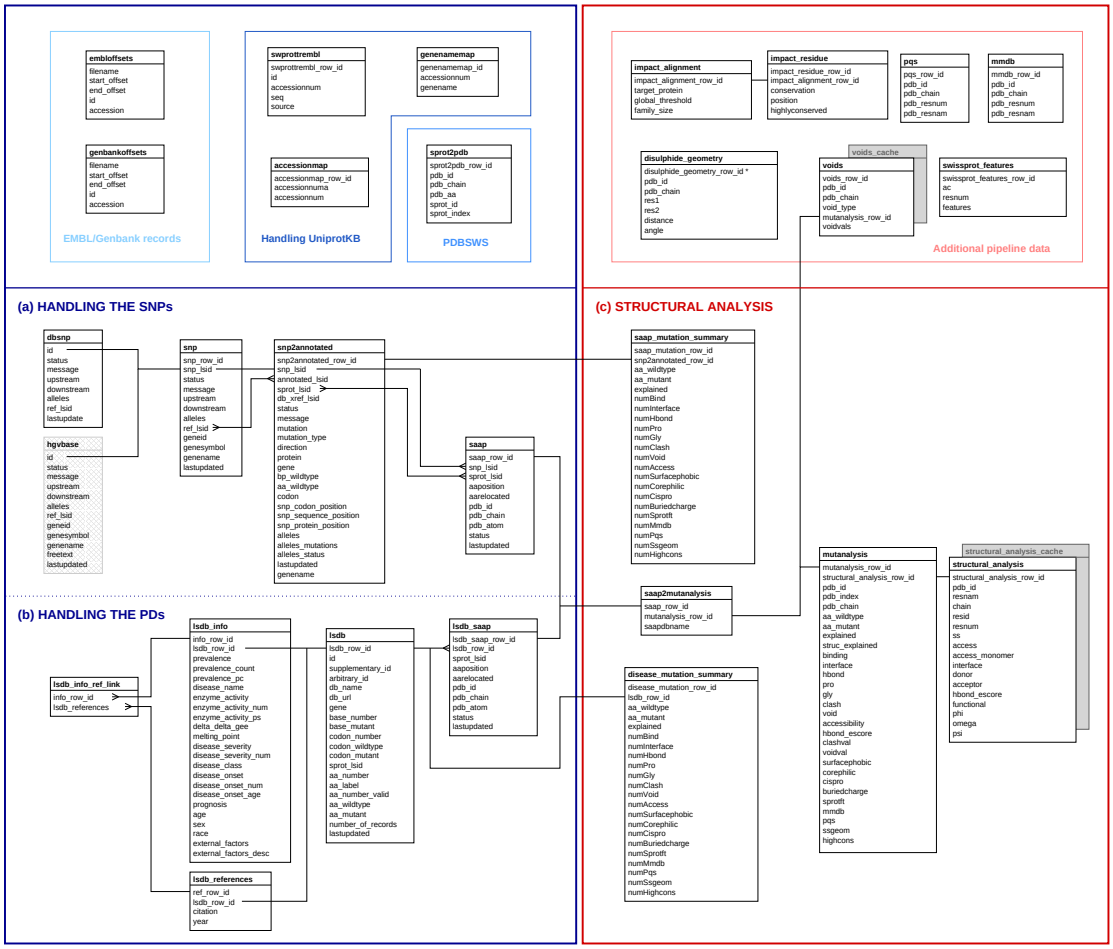


Figure 6.1: The structure of the SAAPdb database

Tables pertaining to data importation are enclosed by a blue box while tables containing pipeline data are enclosed by a red box. Within the data importation section, the PD and SNP data are separated with a dotted blue line. Additional data required are shown at the top of each section, delimited from the rest of the data by a solid blue or red line. Tables containing cached data are shown in grey. Foreign keys are linked with solid black lines. Excepting the additional data within each processing stage, data flow is largely from left to right; that is, the tables on the left hand side of the figure are populated first.

6.2.2 Populating reference tables

▷ *These methods were developed by Jacob Hurst and Craig Porter.*

Three tables in SAAPdb are populated by parsing UniProtKB: `swprottrembl`, `genenamemap` and `accessionmap`. These tables contain sequence data, mappings between gene names and proteins and a mapping between secondary and primary accession numbers respectively. An existing local mirroring of UniProtKB is cached locally before processing begins to ensure that the most recent version of UniProtKB is used in all relevant SAAPdb processes.

SAAPdb uses PDBSWS (Martin, 2005) to map those mutations identified in UniProtKB sequences to structures described by the PDB. The mappings are obtained from `http://www.bioinf.org.uk/pdbsws/pdbsws_res.txt`. This file is parsed to populate the `sprot2pdb` table.

6.2.3 Importing the dbSNP data: new method

▷ *These methods were developed by Lisa McMillan.*

This section describes how the SNP data are imported and mapped to protein sequences and structures, represented in section (a) of Figure 6.1.

The Entrez Programming Utilities[1] (or eUtils) are used to obtain the most recent dbSNP data from the NCBI. XML records of valid, non-synonymous, human SNPs are retrieved. 'Valid' SNPs are defined as those annotated with validation strings "by frequency", "by 2hit 2allele" or "by hapmap". All records retrieved are combined into one XML file and parsed to populate the `snp` and `snp2annotated` tables with dbSNP data.

It is expected that the new HGVBaseG2P[2] (Thorisson *et al.*, 2009) release will be available in the near future which is likely to include protein sequence mappings. This will allow HGVBase to be handled in much the same way as dbSNP currently is in SAAPdb and may render the in-house mapping process redundant.

[1] `http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html`
[2] `http://www.hgvbaseg2p.org`

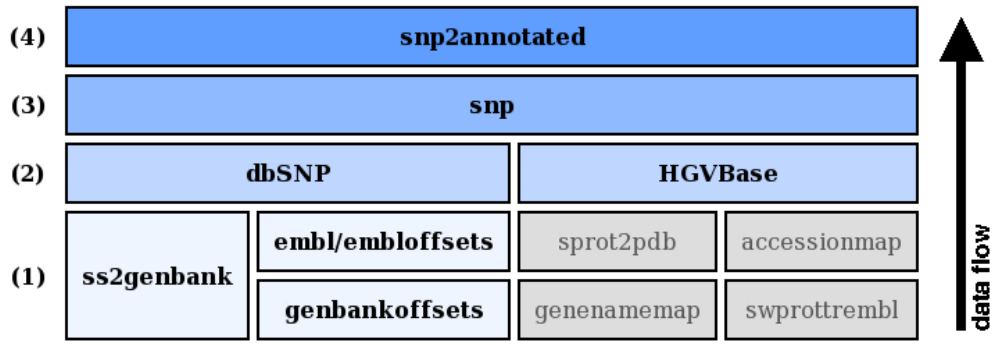


Figure 6.2: SNP data processing in SAAPdb
The layers indicate dependencies in the data, i.e., data in layer (1) must be present before data in layer (2) can be generated (the arrow indicates data flow). **Layer (1)**: data required to map SNPs to genes/proteins; **Layer (2)**: the raw dbSNP and HGVBase data (stored separately); **Layer (3)**: the combined dbSNP and HGVBase data; **Layer (4)**: the mapped SNP data. Grey boxes indicate data required but described elsewhere (Section 6.2.2).

6.2.4 Importing the dbSNP/HGVBase data: old method

▷ *These methods were developed by Jacob Hurst.*

As described above (section 6.2.3), the method for obtaining and retrieving dbSNP data has been changed to improve data integrity and accelerate processing: SAAPdb now uses protein sequence mappings provided by dbSNP. However, all SNP data (both dbSNP and HGVBase) that are analysed in Chapter 7 were mapped using the in-house mapping methodology. Therefore, this section will summarise the in-house system by which the SNPs were mapped to protein sequence. This method is an extension to that described in Cavallo and Martin (2005) and was deceloped by Jacob Hurst.

The mapping system consists of four layers of processing, each of which requires the previous layer to be complete before it itself can be initiated. Processing within a layer may be completed in any order. These layers of processing and their dependencies are shown in Figure 6.2, where layer (n) must be complete before processing in layer (n+1) can commence. Each layer of data processing is described below.

6.2.4.1 First layer: fundamental genomic and proteomic data

Both dbSNP and HGVBase provide the upstream and downstream flanking regions, and an EMBL or Genbank record ID describing the SNP. As the intention is to map to protein structures using PDBSWS, it is necessary to map the SNPs to UniProtKB records. The EMBL/Genbank entries provide database cross-references to protein sequence databases including UniProtKB. To map to UniProtKB/Swiss-Prot sequences, it is necessary to identify the SNP in the EMBL/Genbank record and map forward onto the protein sequence.

The most recent version of EMBL and Genbank are obtained via ftp from the EBI and NCBI respectively. To allow retrieval of individual records, all EMBL and Genbank files are parsed, with each record 'indexed' in the tables `embloffsets` and `genbankoffsets` respectively.

The `embl` and `embloffsets` tables store data about EMBL entries and are populated by parsing the EMBL `.dat` files. The `genbankoffsets` table stores the same information for Genbank files and is populated by parsing the Genbank `.seq` files. These data are required to map the SNPs to sequence and populate the `snp2annotated` table.

Also required at this stage are the tables `swprottrembl`, `genenamemap`, `accessionmap` and `sprot2pdb`, previously described in Section 6.2.2.

The `ss2genbank` table links dbSNP SS IDs to their Genbank accession IDs; these data are extracted from the 'Sub*' files included in the dbSNP release. These data are used later to populate the `dbsnp` table.

6.2.4.2 The second layer: importing the raw data from dbSNP and HGVBase

Raw dbSNP data in XML format are obtained via FTP from the NCBI. These files are then parsed to generate the appropriate SQL which is executed in SAAPdb.

The most recent version of HGVBase is mirrored. An HGVBase release takes the form of several g-zipped files. Each of these is unzipped and parsed to identify the necessary data. These data are then piped directly into SAAPdb.

6.2.4.3 The third layer: combining the dbSNP and HGVBase data in the snp table

The snp table comprises the third layer of SNP data in SAAPdb. This layer simply combines the SNP data from dbSNP and HGVBase into a single table, ensuring that there is no redundancy within each dataset (note that there may be redundancy between the datasets), so that both sets of SNPs can be processed identically henceforth. The raw genomic data are now ready to be mapped to protein sequence.

6.2.4.4 The fourth layer: mapping to UniProtKB sequence records

There are several significant challenges in mapping these genomic data to protein sequence records. Firstly, the EMBL/Genbank data contain introns and exons; to determine whether the SNP occurs in a coding region, the coding sequence must be reconstructed from CDS records in the EMBL/Genbank record. Secondly, it is not known (a) whether the flanking regions and alleles are derived from the forward or reverse complement sequence, and (b) whether the EMBL/Genbank record itself is given in the forward or reverse direction. Thirdly, no information is provided describing which reading frame should be used to translate the coding sequence into the gene product. In addition to the challenge of defining the mapping, the size of the SNP repositories requires that distributed computing be used to process the data within a reasonable time frame.

The process of batching the SNPs and submitting them to the local grid is described in Figure 6.3. The first step is to extract the relevant data from SAAPdb and save it to a file. Next, the SNPs are grouped according to the EMBL/Genbank file in which they occur. This is necessary as these files will be cached locally on the individual processing nodes; by consolidating all SNPs from one particular EMBL/Genbank file into one file, caching transactions will be minimised. Finally, these record-specific files are split into 2000 SNP batches which are processed individually across the grid. Figure 6.4 describes the processing carried out by each job on the compute nodes.

EMBL and Genbank are stored centrally on a single compute node (acrm3). To overcome sporadic NFS file system errors when using these files from nodes on the grid, the relevant EMBL/Genbank file must be cached locally. Once the relevant file has been cached locally, all SNPs in the batch are analysed with findsnp5, the software central to this mapping process.

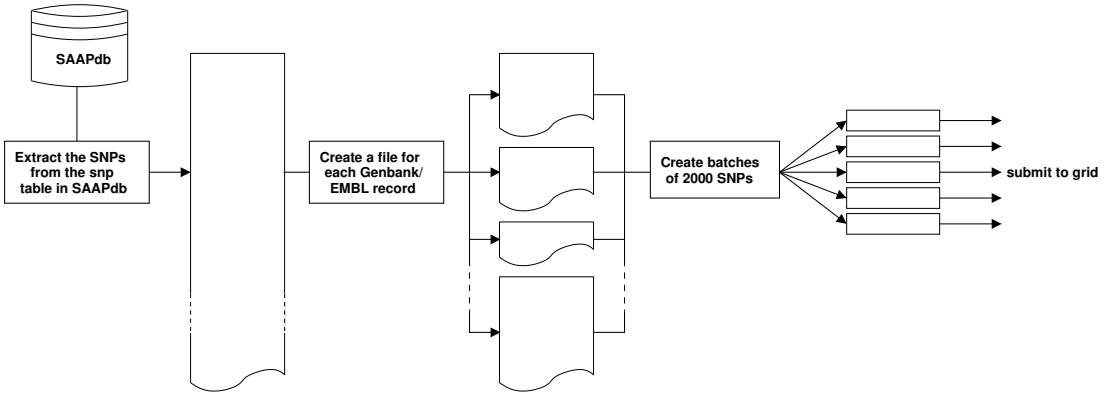


Figure 6.3: Submitting `findsnp5` to the grid
First, all SNPs are extracted and grouped according to the EMBL/Genbank record to which they are mapped. These EMBL/Genbank record-specific lists of SNPs are then divided into batches of 2000 to be distributed across the grid.

`findsnp5` (developed by Jacob Hurst) is depicted in Figure 6.5 and described in pseudocode in Figure 6.6.

`findsnp5` initiates the search by constructing an appropriate search term, consisting of the concatenation of the downstream sequence, the native allele and the upstream sequence. `findsnp5` determines the direction of the reading frame by searching for this assembled search term in the genomic sequence as described by the record and in the reverse complement sequence.

As `findsnp5` relies on annotated coding regions, if the EMBL/Genbank record does not contain any annotated coding regions no mappings can be made and `findsnp5` reports a failed mapping. Otherwise, `findsnp5` assembles the full coding sequence from the exons described by the EMBL/Genbank CDS records. The assembly can accommodate external CDS references and reverse complement CDS records (reverse complement CDS records are explicitly annotated in EMBL/Genbank). The position of the SNP is then identified in the new assembled coding sequence (ACS) by scanning for the original search term; if this is not found, the SNP exists in a non-coding region of the genome, no protein mapping is possible and processing is terminated.

Next, the longest reading frame in the ACS is identified and the ACS is translated. At this stage the codon; the position of the corresponding amino acid in the ACS; and the position of the SNP

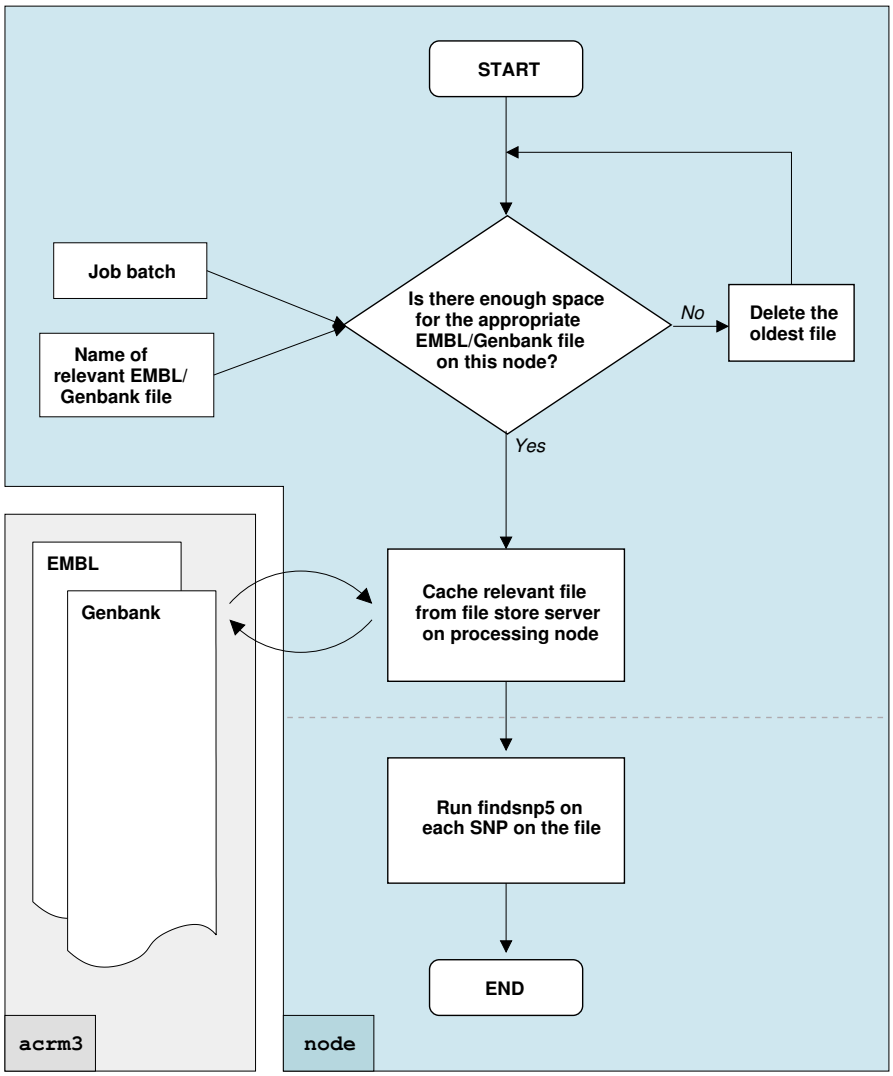


Figure 6.4: The processing done by a batch of `findsnp5` jobs

Each batch of jobs is submitted to a node on the local 20-core grid (shown in blue above). First, the oldest file in the data directory is recursively deleted to accommodate the new EMBL/Genbank file (if space is not available already) which is then retrieved from `acrm3` (shown in grey). After the EMBL/Genback file has been cached successfully, each SNP in the batch is analysed with `findsnp5`.

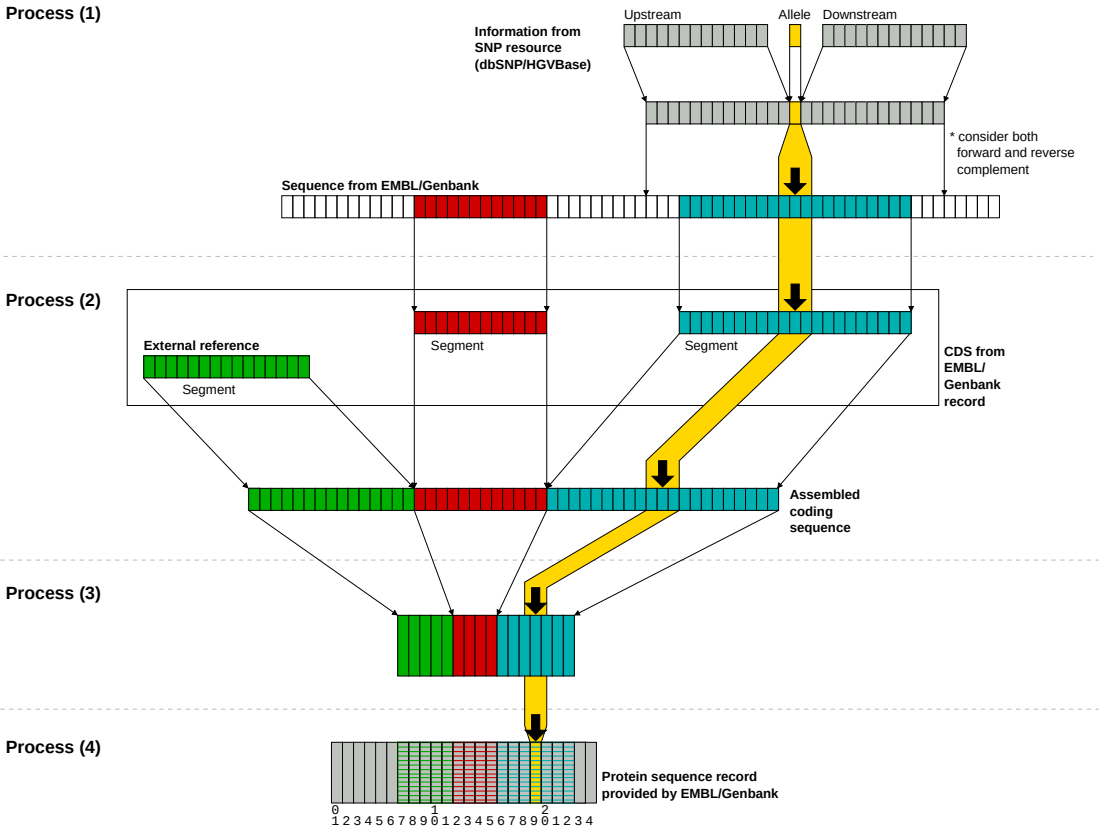


Figure 6.5: The `findsnp5` mapping process

Process (1): look for the allele plus the flanking regions and determine which direction the ORF will be; **Process (2)**: reconstruct coding sequence from CDS records (using both internal and external references; **Process (3)**: find the longest RF and translate to obtain the protein sequence; **Process (4)**: compare the correct translated ORF to the protein sequence provided and identify the relevant residue. The mapping progress of the original allele is marked in yellow; other colours are used to match regions of the genome or the proteome between mapping stages. See Figure 6.6 for pseudocode describing `findsnp5`.

(a) `findsnp5` inputs

- `snp_id` : the ID of the SNP in the source database
- `record_id` : the record containing `snp_id`
- `searchterm` : upstream + 'X' + downstream
- `fseekinfo` : position of `record_id` in file (format is file:start:end)
- `alleles` : the alleles (separated by ',')

(b) `findsnp5` process

(I) Determine record type, construct appropriate parser
(II) Determine which direction to use (forward/reverse complement)
(III) if (`record_id` contains CDS records):
  (a) Assemble CDS into `ACS`
  (b) Find SNP in ACS using `searchterm` (`newsnp_pos`)
  (c) Find longest RF in ACS (`bestrf`)
  (d) Translate ACS using `bestrf` (`bestRF_seq`)
  (e) Identify codon affected (`codon_seq`)
  (f) Identify calculated position in `bestRF_seq` (`codon_pos`)
  (g) Identify which allele of `codon_seq` is affected (`allele_pos`)
  (h) Extract "translation" field from `record_id` (`record_seq`)
  (i) if (`bestRF_seq` not in `record_seq`):
    i. Mapping failed, exit
  (j) Calculate offset value (`offset`) for `bestRF_seq` in `record_seq`
  (k) Calculate `offset_codon_pos` as `codon_pos` + `offset`
  (l) Identify wildtype AA (`record_seq[offset_codon_pos]`)
  (m) Determine mutation type (nonsense/missense/silent)
  (n) Obtain `dbxref` and `genename` pairs from `record_id`
  (o) Write record for each `dbxref`
(IV) else:
(V) Mapping failed, exit

Figure 6.6: The `findsnp5` mapping process in pseudocode
ACS = assembled coding sequence; **RF** = reading frame. Process at line (#1) identifies whether `record_id` is an EMBL or Genbank record (all `record_id` names are preceeded by 'embl:' or 'genbank:' to indicate which record type they are). Process at line (#2) searches for the `searchterm` in the "SQ" feature of the EMBL record. Process at line (#3a) allows for reverse complement CDS records. Process at line (#3i) will check for the first 5 residues if the entire sequence fails to map. Process at line (#3m) compares the native allele with the mutant allele, as constructed by replacing the allele at `allele_pos` with the mutant base.

in the codon are determined, and the protein sequence as provided by the EMBL/Genbank record is extracted.

The protein sequence as derived from the ACS and the protein sequence as provided by the EMBL/Genbank record can then be compared in order to map the SNP onto the protein sequence correctly. If `findsnp5` cannot find the entire translated ACS in the protein sequence record, it tries to find the first five residues of the translated ACS in the protein sequence record. If this also fails, no mappings can be made and `findsnp5` reports a failed mapping. Otherwise, `findsnp5` can proceed in identifying the relevant residue in the protein sequence record and determine whether the base change(s) described by the SNP will result in a silent, nonsense or missense mutation by comparing the translated mutant codon with the translated native codon. After further information is extracted from the EMBL/Genbank record (namely the 'db_xref' and 'gene' name records), the successful mapping is reported.

The `snp2annotated` table is populated with the mappings as identified by `findsnp5`.

6.2.5 Mapping the SNPs to protein structure

▷ *These methods were developed by Jacob Hurst.*

The `sprot2pdb` table is used to map all UniProtKB records in the `snp2annotated` table to protein structures. The resulting mappings are described in the `saap` table.

6.2.6 Importing the PDs

▷ *These methods were developed by James Allen.*

6.2.6.1 The PD data

In many ways, the task of mapping and processing the PD data is more straightforward: there is less data, allowing processing to be sequential, and protein sequence mappings are usually provided, avoiding computationally expensive mapping procedures. Instead, the challenges are in accommodating the different file formats of the source databases. In this section, the method by which the PD data (described in Section 2.1.2) are imported is described.

6.2.6.2 The dataset-specific wrapper

As the PD data are coalesced from different sources, the primary data vary in format. To separate the parsing, verification and import phases, and to permit easy integration of all the data into SAAPdb, it is necessary to represent all the data in the same format. An XML format has been developed within the Martin group to represent mutation data and therefore process each dataset identically. An extract from an example record is shown in Figure 6.7.

This approach requires that each dataset be accommodated by a dataset-specific 'wrapper' which converts the original data into the XML format. Along with the retrieval of the raw datafiles themselves, these are the only manual steps required to import the PD data.

6.2.6.3 Verifying the protein sequence numbering

OMIM is a centrally maintained, curated resource for disease mutation data. However, given that the described mutations are derived from multiple sources and from the literature, it is not surprising that there are inconsistencies in the numbering of amino acids. It is important to verify that the numbering provided by the primary datasets is correct.

A version of OMIM with corrected numbering is currently automatically maintained within the Martin group. Figure 6.8 shows how the verified OMIM mapping is derived for each disease dataset. First, a partial sequence is constructed from the native residues described in OMIM (Figure 6.8(a)). This partial sequence is then compared with the protein sequence named by OMIM, by sliding it along in increments of one residue and storing the number of residue matches for each comparison (Figure 6.8(b)). The alignment that is the best match to the named protein sequence is used to calculate an offset value describing how the OMIM numbering should be corrected; in the example, the offset is -3 (Figure 6.8(c)). The offset is then applied to these 'matching' residues to correct their numbering. If any mutations remain unmatched that would match the sequence with an offset of 0 (e.g., the A20L mutation in the example, highlighted in blue in Figure 6.8(c)), it is assumed that these were submitted to OMIM in a separate batch where correct UniProtKB numbering was used and these data are added to the

```
<lsdb name='DatabaseABC' url='http://DatabaseABC.com'>

<mutation id='001' supplementary_id='456' arbitrary_id='1' number_of_records='6'>

<dna_data>
<gene>ABC</gene>
<dna_base wildtype='T' mutant='G'>1</dna_base>
<codon wildtype='ATT' mutant='AGT'>1</codon>
</dna_data>

<protein_data ac='P00123'>
<amino_acid aa_label='1' wildtype='I' mutant='S' valid='t'>1</amino_acid>
</protein_data>

<occurrence>
<prevalence_text>High</prevalence_text>
<prevalence_count>1000</prevalence_count>
<prevalence_percentage>10</prevalence_percentage>
</occurrence>

<patient_data>
<age>12</age>
<sex>M</sex>
<race>UK</race>
<phenotype mendelian='dominant'>
<disease_name>ABC Deficiency</disease_name>
<disease_class>4</disease_class>
<disease_severity numeric='2'>Moderate</disease_severity>
<disease_onset numeric='2' age='10'>Childhood</disease_onset>
<enzyme_activity numeric='3' percentage='6'>Severely-decreased </enzyme_activity>
<delta_delta_gee>-0.95</delta_delta_gee>
<melting_point>40</melting_point>
<prognosis>10 years</prognosis>
</phenotype>
<external_factors details='1'>Radiation exposure</external_factors>
</patient_data>

<references>
<citation year='2006'>Author, A. N. (2006)</citation>
</references>

</mutation>

...

</lsdb>
```

Figure 6.7: An example of the XML format

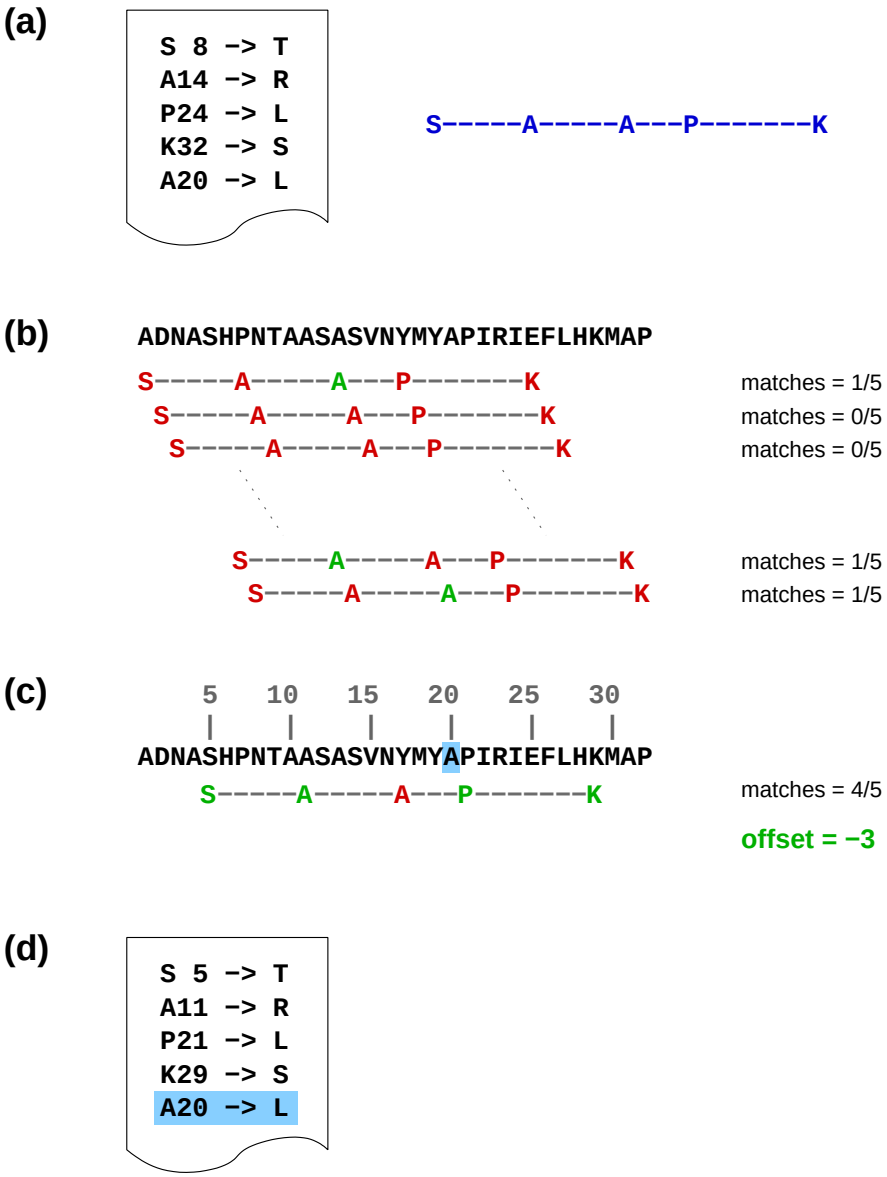


Figure 6.8: Verifying the OMIM mapping

(a): a partial sequence is reconstructed from the mutations described in the OMIM record; **(b)**: this partial sequence is slid along the UniProtKB/Swiss-Prot sequence to which it is mapped in OMIM and the number of matches for each position is recorded (matches are shown in green, mismatches are shown in red); **(c)**: the best matching position is used to calculate the offset (note that the A20 record (shown in blue) could be correct with an offset of 0 (i.e., the OMIM annotation is correct) as an alanine does exist at position 20); **(d)**: the offset is applied to the 'matched' original mutations (i.e., the residues found to match in (c)) to generate a corrected numbering and all 'probably correct' mutations (those matched using an offset of 0) are also included in the dataset (again, the 'probably correct' A20 example is highlighted in blue).

corrected dataset, flagged as 'probably correct'. Some mutations may remain unmapped after these stages. The completed corrected dataset is shown in Figure 6.8(d).

To provide some idea of the extent to which the OMIM data are corrected, 2318 of OMIM mutations (31.0% of all OMIM mutations) from 182 OMIM entries (14.6% of all crosslinked-to from UniProtKB/Swiss-Prot OMIM entries) available in August 2008 required an offset to be applied to correct the sequence numbering. These corrected OMIM data are publicly available at `http://www.bioinf.org.uk/omim`.

An identical scheme is applied to each of the LSMDB datasets in an attempt to maximise the amount of correct data extracted from the LSMDBs (see Section 6.2.6.4).

6.2.6.4 Pushing the data into the database

Figure 6.9 shows the complete workflow by which the PD data are entered into SAAPdb, including the manual 'Write wrapper function' step (highlighted in red). Clearly, this is only written once for each dataset, though it is not uncommon for updates to break the wrappers.

The first stage of processing is to convert the raw data into XML. Before a new dataset can be accommodated by SAAPdb, an appropriate wrapper function must be written which defines which data are where in the raw datafile. The pseudocode for the wrapper scripts is shown in Figure 6.10.

The system will attempt to identify the correct AC should the mutations not be mapped to a UniProtKB/Swiss-Prot sequence. It does this by constructing a partial native sequence by combining the wildtype residues from the data and representing all other residues with an 'X'. This partial sequence is then searched for in the most recent version of UniProtKB/Swiss-Prot using `ssearch34` (Pearson and Lipman, 1988). The raw data are updated accordingly so that the time consuming sequence search need not be repeated. This step is highlighted in green in Figure 6.9.

The corresponding wrapper function is run on each raw data file to generate XML representations of the PD data. Each XML file is then converted to SQL statements via an XSLT specifi-

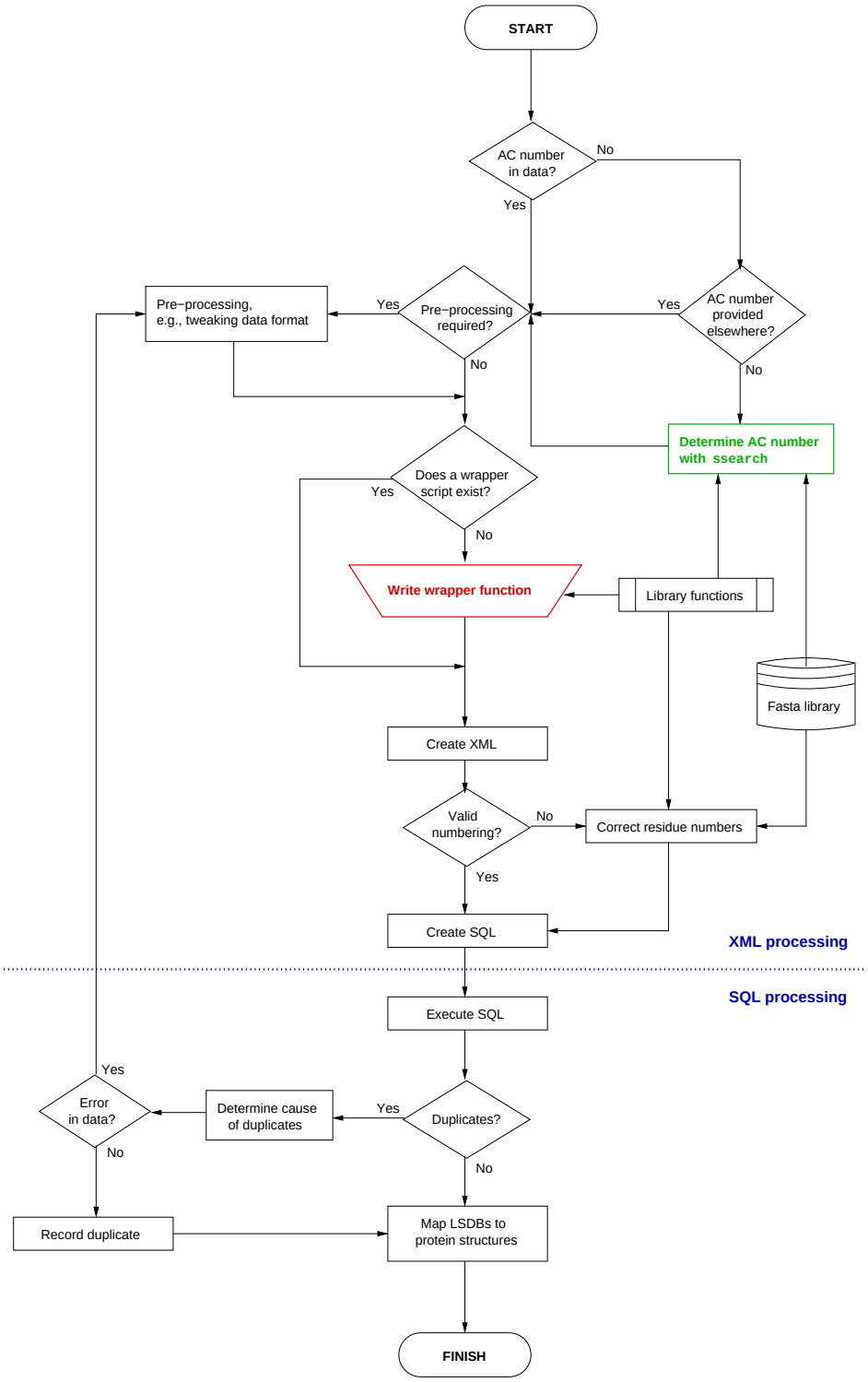


Figure 6.9: Importing an LSMDB dataset

A wrapper script converts the raw data into valid XML and this XML is translated to appropriate SQL using XSLT; the single manual intervention step, where the data wrapper is written, is highlighted in red. Should no AC be provided for the dataset, the AC number is determined using ssearch (highlighted in green, for details see text). This diagram describes the PD data flow for a single LSMDB dataset, from original data format to XML (via wrapper), to SQL (via XSL). In reality, all datasets are processed simultaneously; that is, all raw data-XML processing is done, then all XML-SQL processing. XML and SQL processing stages are separated by a dotted blue line.

(a) wrapper inputs

- `data_folder` : the folder containing the raw data
- `xml_folder` : the foler containing the xml

If these are not provided, the default values of '../data' and '../xml' respectively are used.

(b) wrapper process

(I) parse the `lsdb_info.txt` file to find the `dbname`, `dburl`, `sprotac`, `rawdatafile`

(II) open `rawdatafile` using `cvs.reader()` and the appropriate delimiter

(III) check whether a corresponding XML file already exists (if so, exit cleanly without doing anything)

(IV) identify the `sprotac` using `lsdb_utils.get_ac_number()` unless `sprotac` has been extracted from `lsdb_info.txt`

(V) for each entry in `rawdatafile`:

- (a) if no `mutation_id` exists:
  - i. increment an arbitrary mutation ID counter
- (b) define an appropriate `UI`
- (c) extract all the relevant information
- (d) increment the count for this particular mutation using the `UI`
- (e) record the basic mutation data using the `UI`
- (f) record the numbering (`res_num`, `aa_wildtype`) using the `UI`

(VI) verify the numbering using `lsdb_utils.validate_numbering()`:

- (a) retrieve the sequence of `sprotac` from the UniProtKB website
- (b) identify all possible offsets for each unverified `res_num/aa_wildtype` pair
- (c) identify the most commonly found offset (`most_common_offset`)
- (d) if all `res_num/aa_wildtype` pairs are offset by `most_common_offset`:
  - i. correct all values of `res_num` by `most_common_offset`
  - ii. mark all `res_num/aa_wildtype` pairs as fully validated ('t')
- (e) else:
  - i. if $\geq 50\%$ of the `res_num/aa_wildtype` pairs have an offset of 0:
    - A. Mark these `res_num/aa_wildtype` pairs as fully validated ('t')
  - ii. else if ≥ 2 of the `res_num/aa_wildtype` pairs have an offset of 0:
    - A. Mark these `res_num/aa_wildtype` pairs as probable ('?')
- (f) if there are more `res_num/aa_wildtype` pairs to validate:
  - i. repeatedly calculate offsets as described above until everything is probable or fully validated, or there are only a small number left

(VII) write the XML file using the validated data

Figure 6.10: The PD data wrapper: pseudocode

UI = unique identifier; the thresholds that define what is fully, probably or not validated (in processes #(6(e)i)-#(6(f)i)) can be changed; process at line #6a retrieves the sequence from `http://us.expasy.org/uniprot/`.

cation (see Section 2.2.2) and all SQL is executed in the database. This populates the database tables `lsdb`, `lsdb_references`, `lsdb_info` and `lsdb_info_ref_link` (see Figure 6.1) with the appropriate data.

The final step is to map the imported and verified PDs to protein structures and populate the appropriate database table (`lsdb_saap`) with the mappings (this step requires that the data described in Section 6.2.2 be present in the database). First, the UniProtKB/Swiss-Prot accession numbers to which the disease mutations are mapped are updated to their corresponding primary accession number which will be present in PDBSWS. Then, the `lsdb_saap` table is populated with the appropriate sequence and structural data. These steps are implemented as SQL statements.

Figure 6.9 describes the complete data flow for a single dataset. In reality, processing progresses through the data representations, rather than through each dataset. That is, *all* raw data to XML processing is executed, *all* SQL is generated by applying the XSLT schema to each XML file in turn and finally *all* SQL is executed. Once the sequence data are in the database, the SQL statements updating the AC numbers and the structural mappings are executed.

6.2.7 The pipeline

▷ *These methods were originally developed by Jacob Hurst and have been extended by Craig Porter and Lisa McMillan.*

Once the SNP and PD data are mapped to protein structures (i.e., once the `saap` and `lsdb_saap` tables have been populated), the SAAP data can be processed by the pipeline.

Eight of the analyses require additional data to be present in the database: the hydrogen bonding (Section 5.3.1), clash (Section 5.3.6), void (Section 5.3.7), MMDB (Section 5.5), UniProtKB/Swiss-Prot features (Section 5.11), sequence conservation (Section 5.12), PQS (Section 5.4) and disulphide geometry (Section 5.6) analyses. Detailed information regarding these analyses, what data are required and how they are derived is available in the Sections given above in parentheses.

Figure 6.11 shows how the pipeline is run and how the data are coordinated. There are four phases of processing, which are delineated in Figure 6.11 using dashed lines. In phase (A), the

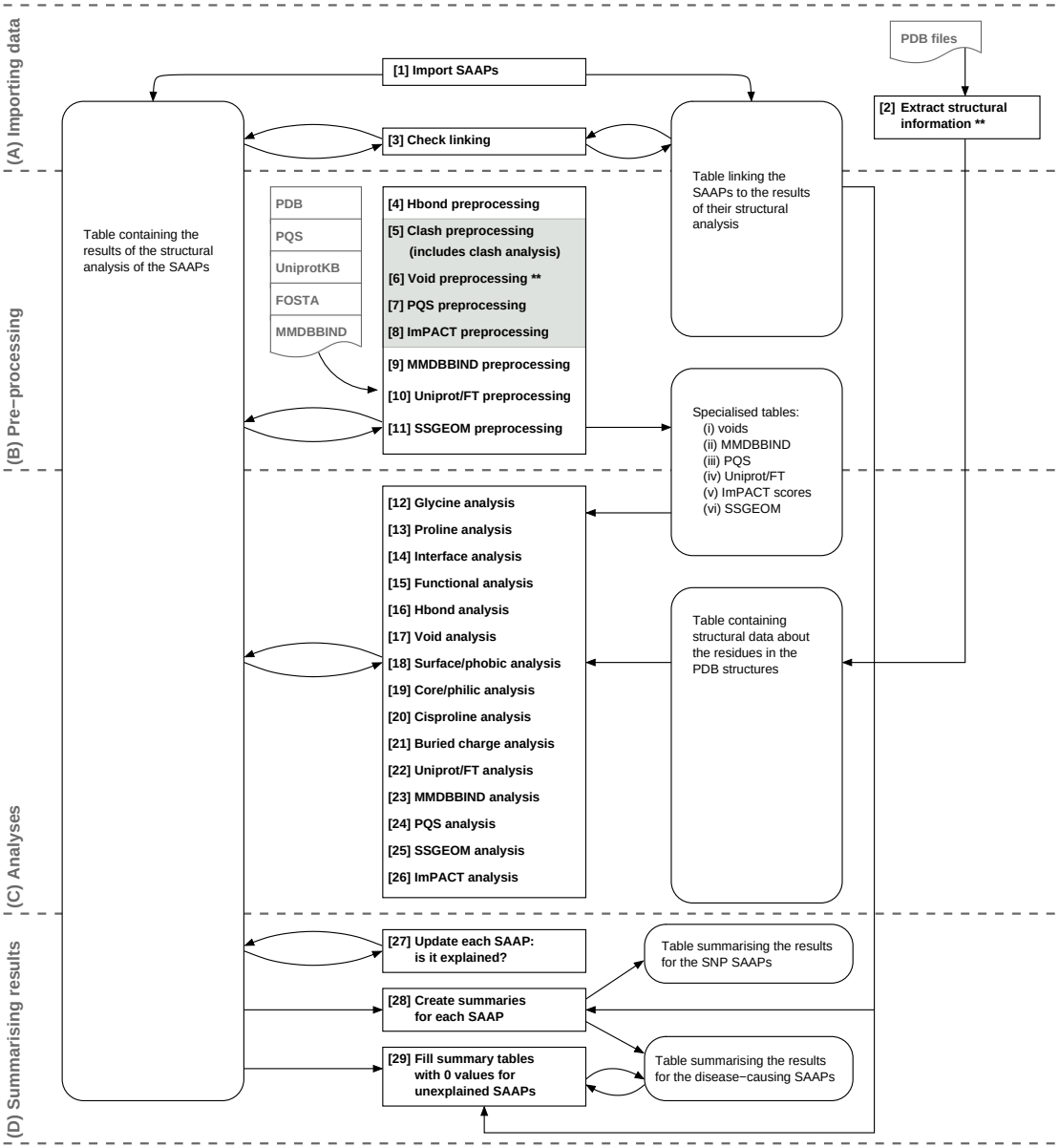


Figure 6.11: Pushing the SAAPs through the structural analysis pipeline

Square boxes indicate data processing, boxes with rounded corners represent database tables and arrows indicate information flow. In processing stage (A), steps [1-3] populate the database with all disease-associated SAAPs and structural information about all PDB structures. In processing stage (B), steps [4-11] generate mutant structures and carry out essential pre-processing for the hydrogen bonding, clash, void, MMDBBIND, Swiss-Prot/FT, PQS, ImPACT and SSGEOM analyses. In processing stage (C), steps [12-26] carry out the structural analyses. In the final processing stage (D), steps [27-29] generate summary information for each SAAP. Cached data are highlighted with $**$ and all distributed grid processing is highlighted with a grey background.

data from the saap and lsdb_saap tables are imported into the mutanalysis table (step [1]) and the structural_analysis table is populated with data extracted and calculated from the relevant PDB files (including torsion angle data; accessibility statistics; secondary structure, and interface and functional flags; step [2]). In step [3], the link between the mutanalysis and structural_analysis tables is created.

In phase (B), all the necessary preprocessing is carried out for the hydrogen bonding (step [4]), clash (step [5]), void (step [6]), PQS (step [7]), sequence conservation/ImPACT (step [8]), MMDB (step [9]), UniProtKB features (step [10]) and disulphide geometry analyses (step [11]). Four of the analyses—clash, void, PQS and ImPACT (steps [5-8])—require considerable preprocessing and as such are distributed across the local 20-core grid. Results are written to the specialist tables (impact_alignment/impact_residue, disulphide_geometry, voids, pqs, mmdb and swissprot_features, see Figure 6.1); the clash preprocessing also updates the mutanalysis table with the clash result and therefore carries out the clash analysis. In Figure 6.11, all distributed processing is highlighted in grey.

The two most time consuming processing steps are step [2] in phase (A)—extracting information from the PDB structures—and step [6] in phase (B)—calculating the void data. To avoid unnecessary and time-consuming repeated processing, these data are cached (in a 'cloned' table) before each run of SAAPdb. In the current implementation of SAAPdb, this creates the tables voids_cache and structural_analysis_cache (these are shown in grey in Figure 6.1). The original table is then dropped and recreated and the original data from the cached table are imported if requested. Processing can then proceed as normal.

With all of the additional data imported into SAAPdb, the remaining analyses can be implemented as SQL queries. These are carried out in phase (C) (steps [12-26]) and update the appropriate columns in the mutanalysis table.

The results are summarised in phase (D). First, each mutation described in the mutanalysis table is annotated as predicted to have a structural effect or not, based on the results of steps [5,12-26]. In step [28], the disease_mutation_summary and saap_mutation_summary tables are populated. These tables summarise the structural analysis results for each *sequence* mutation, as described in either saap or lsdb_saap, by summing over all mapped structures. Finally, any blank entries in the disease_mutation_summary and saap_mutation_summary tables are replaced by zeros (step [29]).

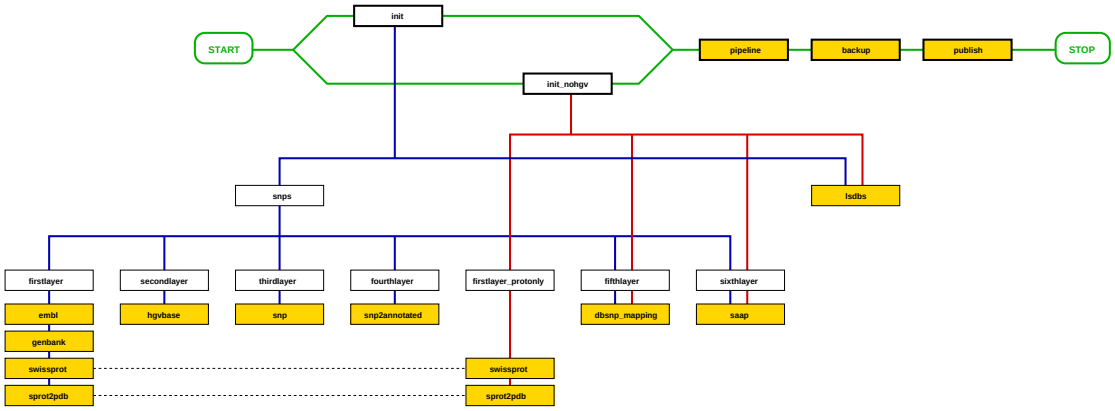


Figure 6.12: The SAAPdb Makefile

All targets are shown in square-edged boxes, with the five top-level targets (`init`, `init_nohgv`, `pipeline`, `publish` and `backup`) at the top of the diagram, highlighted in thick-lined boxes. The thick green line follows processing, from START to FINISH (indicated by round-edged green boxes); alternative processing steps are shown by a split in the green line. **`init`**: processes all SAAPs (dbSNP, HGVBase and LSMDBs); **`init_nohgv`**: processes only dbSNP and LSMDB SAAPs; **`pipeline`**: runs the pipeline on the SAAP data in SAAPdb; **`backup`**: creates a backup of SAAPdb, and **`publish`**: creates a web-live version of SAAPdb and creates an SQL download of the most recent backup of SAAPdb. Boxes highlighted in gold are 'executing' targets (i.e., they execute the scripts) whereas white boxes are 'abstract' targets, pointing to other targets. The blue lines trace the targets of the top-level `init` and the red lines trace the targets of `init_nohgv`. Dashed black lines link two uses of the same target.

6.2.8 Putting it all together: the Makefile

▷ *The Makefile was originally written by Craig Porter, with small additions (including the ability to backup and make the data web-live) by Lisa McMillan.*

The SAAPdb system is controlled via a Makefile. The targets for the Makefile are described in this section and illustrated in Figure 6.12.

The `init` target imports and processes dbSNP and HGVBase SNPs, and all LSMDB PDs. The SNP importing requires several layers of data processing as described in Section 6.2.3. The `init_nohgv` targets initiates processing of the dbSNP and LSMDB PD data *only* (i.e., HGVBase is not imported or processed). The `pipeline` target runs the pipeline on all SAAPdb data. The `backup` target runs a '`VACUUM FULL ANALYSE;`' on SAAPdb and creates a gzipped-tar backup of the data. `publish` can be used to make a web-live version of the database and to construct an SQL download of the relevant tables for remote reconstruction of SAAPdb.

Chapter 7

SAAPdb : data overview

In Chapters 4 and 5 the current suite of structural analyses in SAAPdb was described. In Chapter 6 the mechanics and machinery of SAAPdb were described, including the import of PD and SNP data; the mapping of SAAPs to protein structures; the integration of data necessary for the structural analyses and the implementation of the structural analyses themselves. In this chapter, the resulting data are analysed. The aim of this analysis is to characterise pathogenic deviations (PDs) and single nucleotide polymorphisms (SNPs), with a view to building models that predict whether a novel SAAP (single amino acid polymorphism) will cause disease.

Note that this analysis includes SNP data from HGVBase and that all SNPs have been mapped to sequence using the mapping procedure described in Section 6.2.4. Some of the work in this chapter has been published in Hurst *et al.* (2009).

7.1 Introduction

It is the intention that the data compiled in SAAPdb will eventually be used to train machine learning methods to predict whether a novel SAAP will disrupt the native protein structure, thus inducing a disease phenotype. In this context, it is important to characterise *both* datasets as fully as possible for several reasons.

Firstly, the analysis will suggest whether it is possible to train a prediction algorithm at all: the data analysis may conclude that there is no discriminatory power in the data and therefore no potential for successful prediction. Such a result would motivate the collection of more data, and, in particular, more varied data in the form of additional structural analyses.

Secondly, characterising the datasets would inform the choice of data representation and machine learning method. By drawing on existing literature and the results described in this chapter, it will be possible to maximise the discriminative power inherent in the feature vector with which the training data are represented. Further, should the analysis reveal characterstics of the dataset as a whole (e.g., whether the data are particularly sparse), this will inform the most appropriate choice of algorithm.

Finally, a maturing body of literature exists in this field (see Section 1.9). To date, SAAPdb is the most extensive collation of SAAPs, both deleterious and neutral, and their structural effects. A systematic analysis of the contents of SAAPdb will contribute significantly to the understanding of disease polymorphisms, and ultimately the treatment of the deleterious phenotype.

To recap the SAAPdb system, Figure 7.1 shows a simplified workflow for the population of SAAPdb. PD processing is highlighted in yellow and SNP processing is highlighted in grey. This colour scheme will be used throughout this chapter to denote each dataset. As described in detail in Chapter 6, SNPs are mapped to protein structure by assembling the genomic coding sequence from EMBL and Genbank records, and aligning the translated sequence with UniProtKB/Swiss-Prot sequences (step (1) in Figure 7.1, see Section 6.2.4). PDs, being derived from multiple sources (OMIM and several LSMDBs, see Section 2.1.2), are imported by first representing the data in the same XML format and then processing and verifying these data (step (2) in Figure 7.1, see Section 6.2.6 for details). At this point, both PDs and SNPs—together described as SAAPs—are mapped to protein sequences. The SAAPs are then mapped to protein structures using PDBSWS (step (4), see Section 6.2.2 for details) and the native structures are analysed to extract basic information, such as binding sites, relative accessibilities and secondary structure classifications (step (4)). Finally, each SAAP is processed by the structural analysis pipeline (step (5), see Chapter 5 for details) to ascertain whether it is expected to have a structural effect.

Table 7.1 summarizes the content of SAAPdb. After importing the raw data from the SNP repositories and the various LSMDBs (i.e., after step (2) in Figure 7.1), there are approximately ten

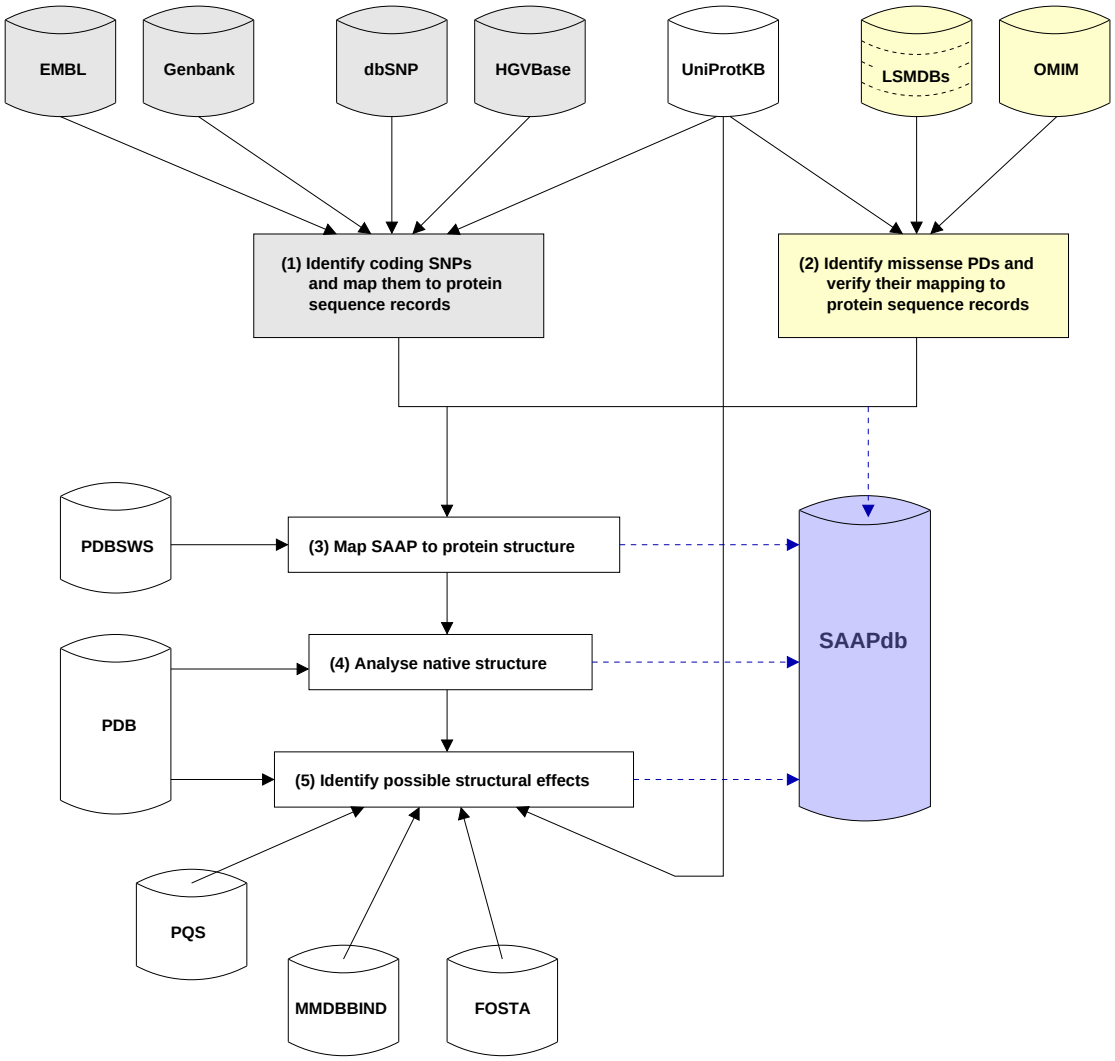


Figure 7.1: The SAAPdb workflow

STEP (1) SNPs are mapped to protein sequence by translating the genomic coding sequence into a protein sequence and aligning this with named UniProtKB/Swiss-Prot sequences (see Section 6.2.4); **STEP (2)** PDs are imported from OMIM and various LSMDBs and the protein sequence mappings provided are verified (see Section 6.2.6); **STEP (3)** From now on, both PDs and SNPs (together, described as SAAPs) are processed identically, in step (3), they are mapped to protein structures using PDBSWS (see Section 2.1.5); **STEP (4)** The native structure is analysed to extract basic data like accessibility and binding sites; **STEP (5)** Each SAAP is analysed by the structural analysis pipeline (see Chapters 5-6) which requires some additional data, external to SAAPdb, including FOSTA (Chapter 3), PQS (Section 5.4) and MMDBBIND (Section 5.5. Resources and processes specific to the PD dataset are highlighted in yellow, resources and processes specific to the SNP dataset are highlighted in grey, resources and processes common to both datasets are not coloured. SAAPdb is highlighted in blue. Solid black arrows indicate the direction of data flow; dashed blue lines indicate where data are stored in SAAPdb.

Table 7.1: A summary of the data in SAAPdb

	PDs	SNPs
Raw number described in database	9997	16227751
Raw number of SAAPs mapped to sequence	9617	24492
Unique sequence polymorphisms	8972	14015
Raw number mapped to at least one structure	4319	2022

thousand pathogenic deviations (PDs) and over 16 million neutral mutations described. 9 617 PDs (8 972 of which are unique) and 24 492 SNPs (of which 14 015 are unique) are successfully mapped to amino acid changes in a UniprotKB sequence. Using PDBSWS (Martin, 2005), the SAAPs are then mapped onto PDB structures (step (3) in Figure 7.1). Of the 9 617 mapped and coding PDs, 44.91% are mapped to at least one PDB structure, but only 8.26% of the neutral mutations are identified in a protein structure.

Despite having over one thousand times more 'raw' mutations in the SNP dataset as compared with the PD dataset, the two mapping stages (gene to protein sequence and protein sequence to protein structure) eliminate much of the SNP data to leave a more balanced dataset (many SNPs will have been lost when mapping to protein sequence as they will occur in non-coding areas of the genome).

It is possible that the SNP repositories (in particular dbSNP) describe polymorphisms as neutral that should be described as disease-causing. In SAAPdb, only six (see Section 6.1) polymorphisms are described in both datasets. For the analyses presented in this chapter, these polymorphisms are removed from the SNP dataset, but remain in the PD dataset. This is based on the assumption that the large-scale genomic scanning technology by which the SNPs are identified happens to have sequenced the genome of an individual carrying a disease mutation.

This chapter will report an analysis of the data obtained by the processes described in Figure 7.1.

7.2 Methods

7.2.1 Averaging across multiple structures

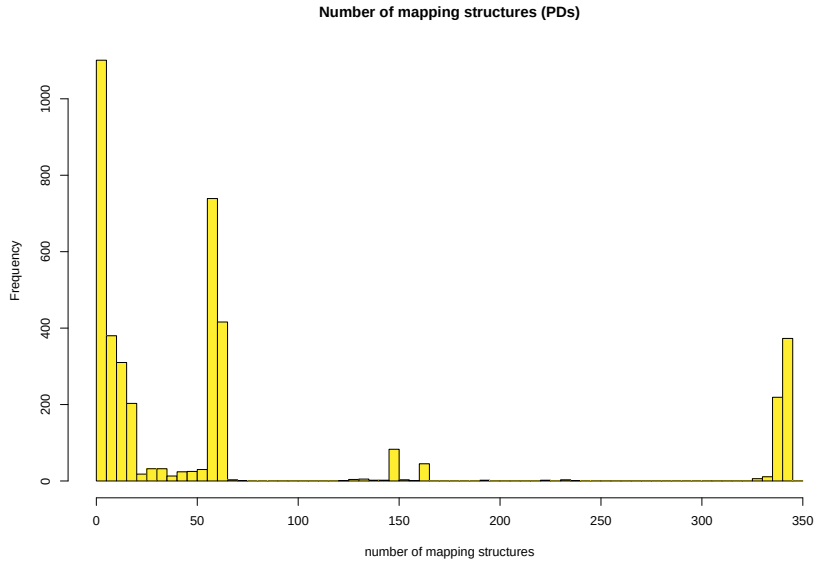
PDBSWS (Martin, 2005) is used to map mutations in UniProtKB/Swiss-Prot sequences to PDB structures (see Section 2.1.5). The redundancy in the PDB allows multiple SAAP/structure mappings to be identified by PDBSWS.

There is disparity in the number of structures to which SAAPs are mapped in the datasets: some are mapped to a single structure (e.g., mutations to the UniProtKB/Swiss-Prot record P02766, human transthyretin), while others are mapped to over three hundred (e.g., mutations to the UniProtKB/Swiss-Prot record P68871, human haemoglobin subunit β). This is primarily due to research bias: some proteins are more heavily researched than others. To illustrate this, the distributions of the number of structures to which PDs and SNPs map are shown in Figures 7.2(a) and 7.2(b) respectively. These graphs confirm the expectation that proteins implicated in disease are more heavily researched and more often structurally characterised than proteins not implicated in disease. Furthermore, there may be some structures that are of poorer quality and may give spurious ASA (accessible surface area), torsion angle measurements or hydrogen-bonding assignments.

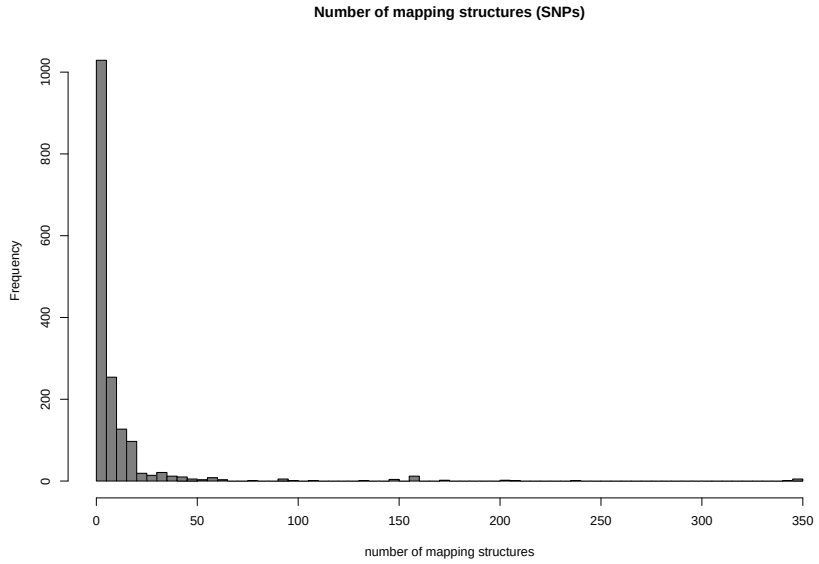
To accommodate multiple mappings fairly and to limit the effect of poor quality PDB structures, it is necessary to average over all structures to which a SAAP has been mapped when analysing the data. The median of measurements over all mapped structures is used to average numeric data. Where the measurement is nominal (for example, secondary structure DSSP code (Kabsch and Sander, 1983)), the mode of the data is used.

7.2.2 Statistics

The statistics used in this chapter were introduced in Section 2.3.6. In this section, any additional information specific to the analyses used in this chapter is given.



(a) The number of structures to which PDs are mapped



(b) The number of structures to which SNPs are mapped

Figure 7.2: Profiling SAAPs by the number of structures to which they are mapped
SAAPs are mapped to structures using PDBSWS (Martin, 2005). PDs are more often mapped to multiple structures due to research bias. PDs are shown in yellow; SNPs are shown in grey.

7.2.2.1 χ^2 tests

χ^2 tests are used to assess whether there is a significant difference between the PD and the SNP datasets with respect to the occurrence of a particular feature. All χ^2 tests with one degree of freedom (i.e., a 2x2 contingency table) are carried out using the Yates correction (which prevents underestimating the p-value, see Section 2.3.6.3). Note that, where results are reported with percentages, raw counts have been used in the χ^2 test.

Where possible, χ^2 statistics have been calculated using separately calculated expected values. In the case of individual native and mutant residues, standard amino acid frequencies (Robinson and Robinson, 1991) were used to estimate the numbers to be expected in the dataset. To analyse each polymorphism (i.e., each native/mutant residue pair), the PAM30 (Dayhoff *et al.*, 1978) matrix was normalised (i) to include only positive values, and (ii) to sum to 100. To reflect the SAAP data more accurately, only native/mutant residue pairs (a, b) that can be generated by a single base change were considered when normalising the matrix. The formula for this transformation is shown below:

$$P'_{sbc}(a,b) = 100 * \frac{P_{sbc}(a,b) - min(P_{sbc})}{\sum P_{sbc}} \tag{7.1}$$

where n is the number of different amino acids (therefore $n = 20$); P_{sbc} is the submatrix of the PAM30 matrix which describes only those mutations that can be derived from a single base change; P'_{sbc} is the normalised PAM30 single base change submatrix; $P_{sbc}(a, b)$ is the amino acid substitution matrix P_{sbc} score for replacing residue a with residue b; $min(P_{sbc})$ is the minimum value in the matrix P_{sbc}, and $\sum P_{sbc}$ is the sum of all the scores in the matrix P_{sbc}.

These normalised cell values were then used to approximate relative frequencies of mutations and therefore estimate the expected frequencies in the given dataset. The PAM30 substitution matrix was used because it is the most widely used amino acid substitution matrix that is derived from closely related sequences. This is appropriate because, in this dataset, human proteins are effectively being compared with themselves. Finally, expected values for secondary structure element occurrence were derived from a dataset of high resolution ($\leq$2.0Å) PDB structures.

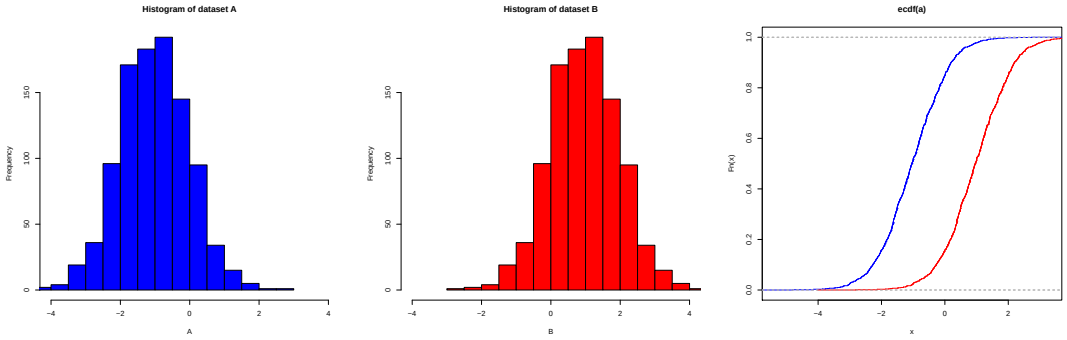


Figure 7.3: Interpreting a cumulative distribution function (CDF) plot
The two sets randomly generated data are shown above: the dataset A (shown in blue) has a mean of -1 and a standard deviation of 1, the dataset B (shown in red) has a mean of 1 and a standard deviation of 1. It becomes immediately apparent which dataset contains higher values in the CDF plot (far right): the CDF for dataset A is consistently higher than that of B, indicating that a higher proportion of datapoints in A are found at lower values.

χ^2 tests are unreliable if the data suffer low counts, specifically where the contingency table contains one or more cell values ≤ 5. Fisher-exact tests have been carried out on low-count datasets to verify χ^2 results. Low-count datasets are defined as datasets with at least one contingency table cell with a value ≤ 10.

7.2.2.2 Kolmogorov-Smirnov tests

The Kolomogorov-Smirnov (KS) test has been used to compare the distribution of numerical data in the SAAP datasets. All KS tests have been run in `R` using the `ks.boot` method (see Section 2.3.6.2).

Cumulative distribution function (CDF) plots are used extensively in this chapter. Often, when comparing large datasets, histograms are difficult to interpret; differences become more apparent by comparing CDFs. In Figure 7.3, two datasets are compared using their CDF functions. Although it is clear from the histograms that these datasets are different, the CDF plot makes it immediately apparent that the values in dataset A are lower than the values in dataset B.

7.2.2.3 Log ratios

Log ratios demonstrate clearly where features in the dataset are seen more or less often than expected when compared to some reference values. Where such reference values are available or can be reliably estimated, the corresponding log ratios have been calculated. As described in Section 7.2.2.1, expected values for the individual mutant and native residues were taken from Robinson and Robinson (1991), and expected values for each native/mutant residue pair were estimated by transforming the PAM30 matrix to (i) eliminate negative values and (ii) to sum to 100.

All log ratios are log_2.

7.2.3 Discriminative features

With a view to distinguishing between PDs and SNPs, and potentially training machine learning methods to predict whether a novel SAAP is deleterious or not, it is most important to identify those features that are 'discriminative'. In the context of the current dataset, a discriminative feature must meet two criteria. Firstly, it must be found at significantly different frequencies in the PD dataset and the SNP dataset, by χ^2 tests or otherwise. Secondly, where reliable expected frequencies are available, the feature must be over-represented in one dataset, while being under-represented in the other.

7.3 Results and Discussion

7.3.1 Illustrative examples

Before describing the analysis of the SAAPdb data, this section illustrates some of the results of the analysis pipeline.

Figure 7.4 shows the structure of human super-oxide dismutase [UniProtKB:P00441/SODC_HUMAN], as described by the PDB structure 2c9s. Mutations to

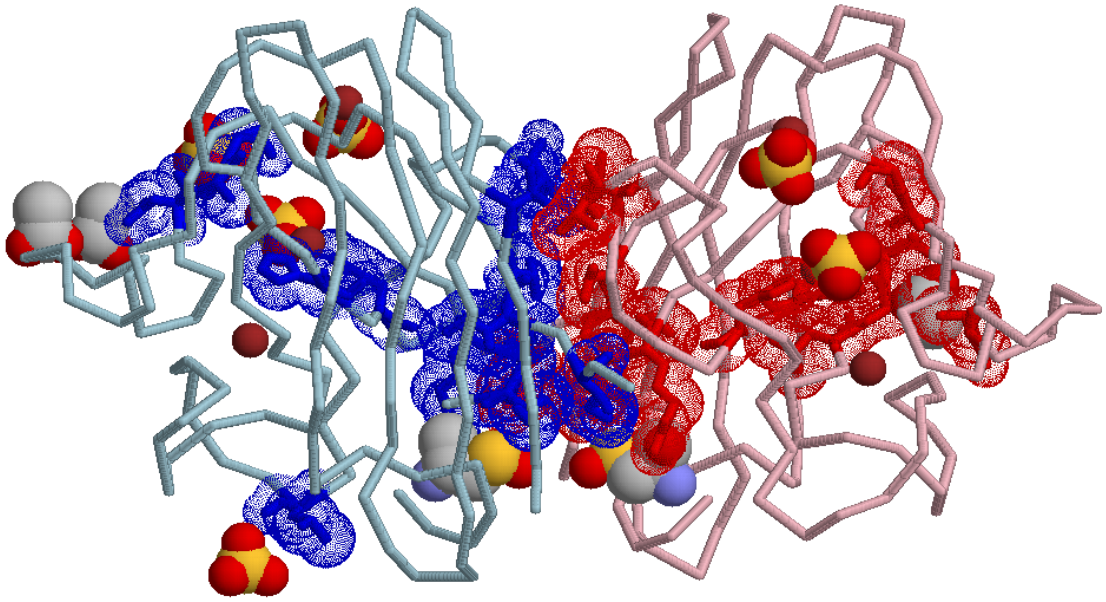
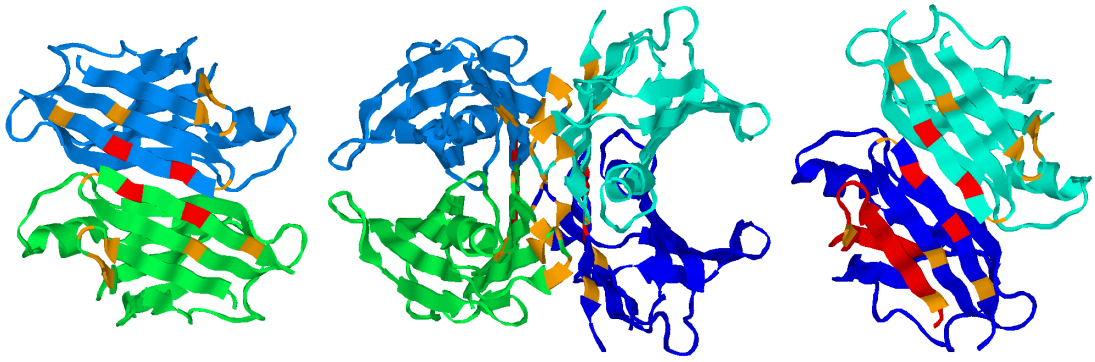


Figure 7.4: PDs identified at the interface
PDB structure 2c9s, chains A (in blue and light blue, on the left) and F (in red and pink, on the right). Ligands are shown in spacefill using the CPK colour scheme. Mutated residues identified by the interface (see Section 5.3.2) and PQS (see Section 5.4) analyses are shown in darker blue and red, with Van der Waals volumes indicated. Note that these analyses also identify residues near ligand binding sites, as well as residues at the chain interface.

super-oxide dismutase have been associated with amyotrophic lateral sclerosis or motor neurone disease (Aguirre *et al.*, 1999). Chain A is shown in blue, chain F is shown in red. Residues identified by the interface or PQS analyses are shown in darker blue and red respectively, with their Van der Waals suface indicated with dots. This illustrates that PQS/interface residues occur both at the inter-chain interface and at ligand binding sites.

Figure 7.5 shows three orientations of the human transthyretin protein [UniProtKB:P02766/TTHY_HUMAN] as described by the PQS record *1soq_1.mmol*. Mutations to transthyretin are associated with several amyloid diseases, including cardiomyopathy (Ranløv *et al.*, 1992) and polyneuropathology (Ferlini *et al.*, 2000). 48 transthyretin disease mutations (derived from OMIM, MIM:176300) are mapped to structure in SAAPdb.

The transthyretin PDs identified at the interchain protein interface are shown in orange or red in Figure 7.5 (the residues highlighted in red introduce a hydrophobic residue on the surface of the protein chain in addition to occurring at the interface). To demonstrate the value of using the 'corrected' PQS structures (see Section 5.4) rather than the basic PDB structures (see



(a) Native 1soq:B,D (b) Native 1soq:A,B,C,D (c) Native 1soq:A,C

Figure 7.5: PDs identified at the PQS interface
PQS structure `1soq_1.mmol`, chains A, B, C and D. Residues identified by the PQS analysis (see Section 5.4) but *not* the interface analysis (see Section 5.3.2) are highlighted in orange and red above (red residues are also identified by the surface-phobic analysis, see Section 5.9). Figure 7.5(b) shows all four chains (ABCD) together; Figure 7.5(a) shows chains B and D rotated +90°on the horizontal axis; Figure 7.5(c) shows chains A and C rotated -90°on the horizontal axis.

Section 5.3.2-5.3.3), only those PDs explained by the PQS analysis but *not* explained by the PDB interface analysis are highlighted. The BD and AC dimers are separated and rotated to display the interface in Figures 7.5(a) and 7.5(c) respectively. It is clear that the mutations cluster at the interface of the AC and BD dimers (no ligands are described by the 1soq structure).

The tumour suppressor protein P53 [UniProtKB:P04637/P53_HUMAN] is mutated in roughly half of human cancers (Greenblatt *et al.*, 1994; Sidransky and Hollstein, 1996; Lane and Fischer, 2004). Chain B of the solved P53 structure 1tsr is shown in complex with DNA in Figure 7.6; residues identified as 'functional' by the binding, PQS *and* UniProtKB/Swiss-Prot FT analyses (i.e., residues identified by *all* of these analyses) are highlighted in blue (these residues are also identified as highly conserved by ImPACT). These functional residues are clustered around the DNA-binding site.

Another P53 mutant is shown in Figure 7.7. Here, the native glycine residue at position 279 is mutated to tryptophan, the largest amino acid. When modelling the mutant residue into the native structure (using MutModel, see Section 5.2), the best orientation of the mutant sidechain clashes with 27 other native atoms. Figure 7.7(a) shows that the native glycine fits neatly inside the structure, while the tryptophan residue in Figure 7.7(b) protrudes out of the structure, clashing with other atoms, inhibiting formation of the native fold and inducing the disease

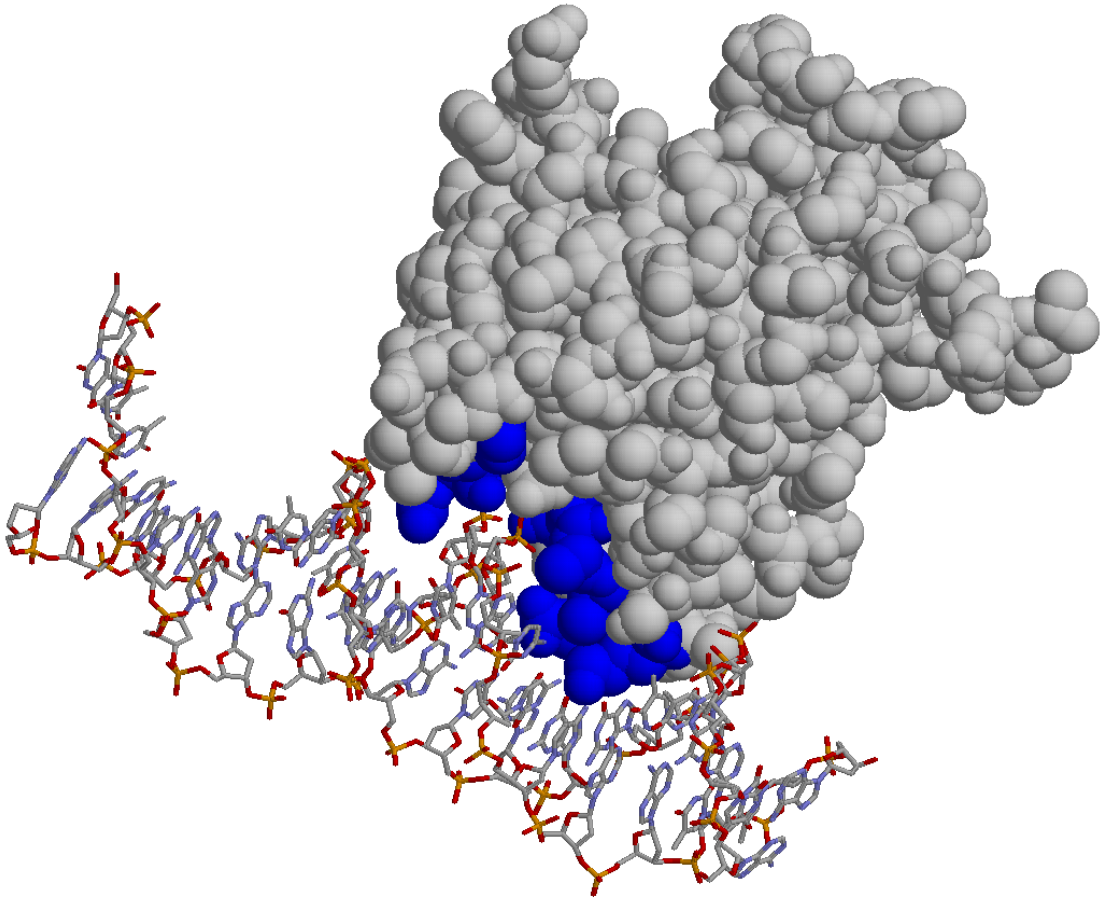
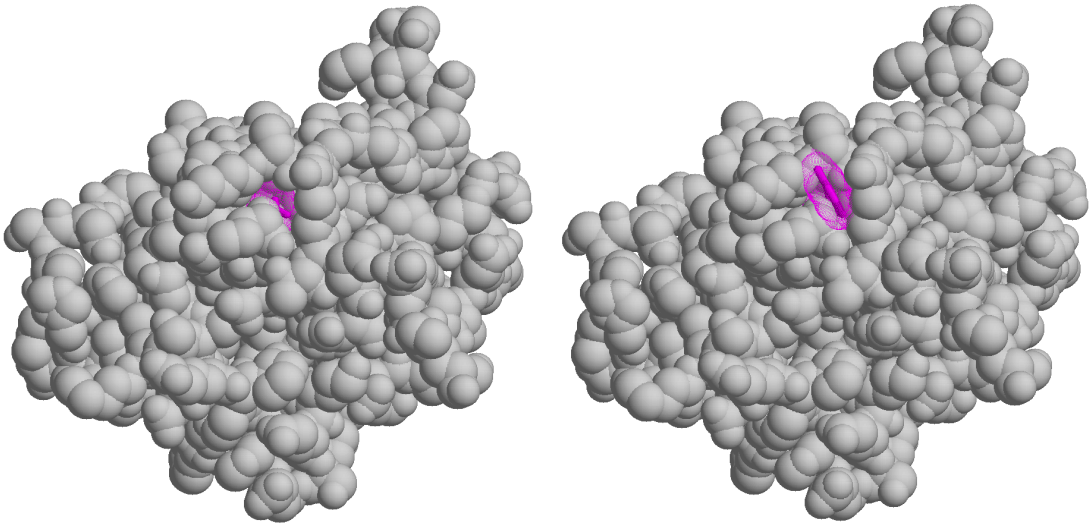


Figure 7.6: Binding PDs in P53
PDB structure 1tsr, chain B (in grey and blue) in complex with DNA. PDs that disrupt binding, as defined by the binding (see Section 5.3.3), PQS (see Section 5.4), ImPACT (high conservation, see Chapter 4) and UniProtKB/Swiss-Prot FT analyses (see Section 5.11) are highlighted in blue.

phenotype.

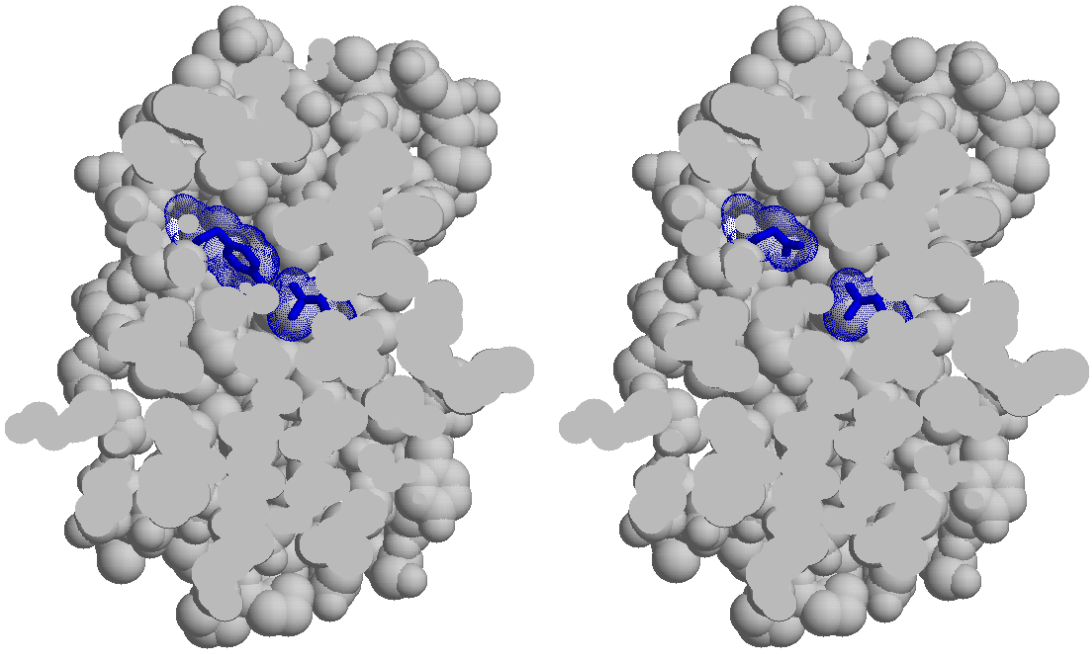
The native tyrosine residue at position 236 of human P53 forms a hydrogen bond with the threonine residue at position 253; these residues are highlighted in blue in Figure 7.8(a). This hydrogen bond is broken in the Y236D mutant structure in Figure 7.8(b), as the introduced aspartic acid sidechain is too distant to accept the hydrogen donor atom from T253. Note also that this hydrogen bond is buried, and therefore could be critical to the scaffold of interactions that stabilise the protein structure. In addition to breaking the hydrogen bond, this mutation will introduce an unpaired buried charge and is found to cause a de-stabilising internal void.

Many mutations to the structure of human haemoglobin have been reported: SAAPdb



(a) Native 2ata:A (b) Mutant G279W 2ata:A

Figure 7.7: PDs found to clash with other existing residues
PDB structure 2ata, chain A (shown in grey). The mutation G279W is described in the P53 somatic mutation dataset. The native and mutant structures are shown above, on the left and right respectively. The modelled tryptophan mutant residue clashes with 27 other atoms, and cannot be accommodated in the native structure.



(a) Native 2j1w:B (b) Mutant Y236D 2j1w:B

Figure 7.8: PDs that break hydrogen bonds
PDB structure 2j1w, chain B. The hydrogen bond that exists between the Y236 and T253 is not maintained in the mutant Y236D structure shown on the right (see Section 5.3.1). Residues 236 and 253 are highlighted in blue in both structures.

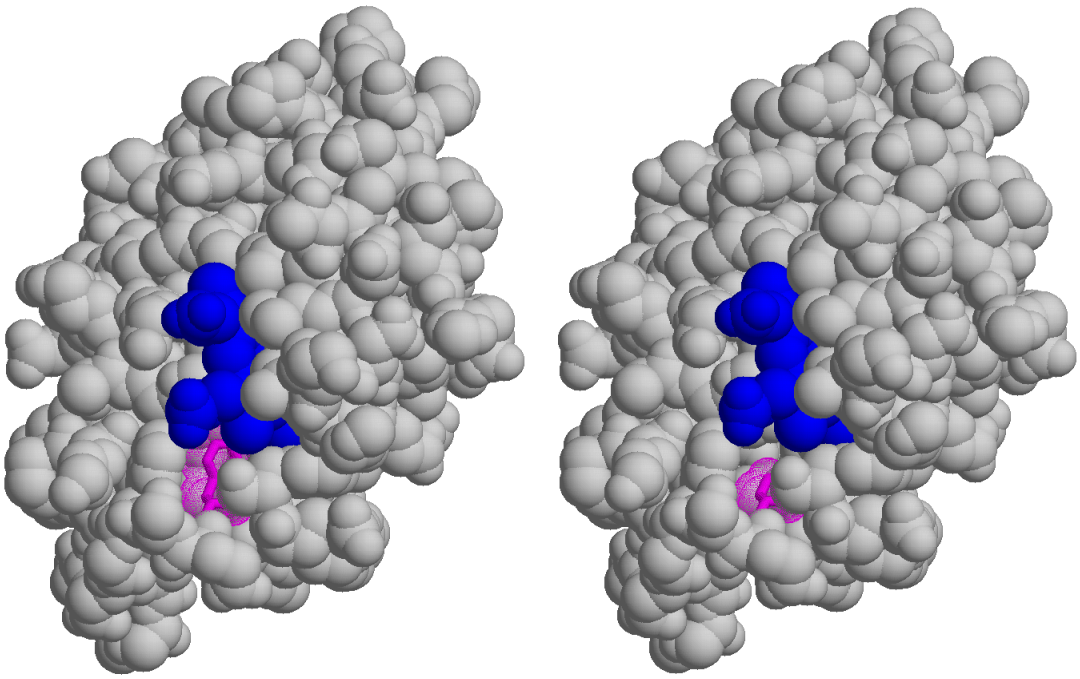
contains over six hundred. The mutation F42S in human haemoglobin [UniProtKB:P68871/HBB_HUMAN] is reported to be associated with cyanosis, moderate reticulocytosis and mild anaemia (Stabler *et al.*, 1994). SAAPdb 'explains' this mutation as introducing a void on the surface of the protein (compare the native structure in Figure 7.9(a) with the mutant form in Figure 7.9(b)). Note also that this void is close to the binding site of the haem ligand and therefore may affect the function of haemoglobin *directly* in addition to destabilising the structure. SAAPdb explains mutations at residue 42 with the interface, PQS and binding analyses.

Figure 7.10 shows the 1r1y crystal structure of human haemoglobin. Here, the mutation V54D introduces a buried, unsatisfied charge by replacing a neutral valine residue with the negatively charged aspartic acid (see Section 5.8). In addition, this mutation introduces an unfavourable hydrophilic residue in the protein core. An example of the complementary analysis—introducing a hydrophobic residue on the surface of the protein—is shown in Figure 7.11. The mutation seen here is the E6V mutation described in Chapter 1 that causes sickle cell anaemia, where the 'sticky' hydrophobic patch owing to the mutant valine residue causes aggregation and subsequent deformation of erythrocytes.

The example in Figure 7.12 shows a broken disulphide bond in super-oxide dismutase, identified both by the UniProtKB/Swiss-Prot FT analysis and the geometric disulphide analysis of the PDB files.

Figures 7.13 and 7.14 show the Ramachandran plots (Ramachandran *et al.*, 1963) obtained from a RAMPAGE (Lovell *et al.*, 2003) analysis of two mutant protein structures. RAMPAGE is a webserver that identifies unfavourable torsion angles for non-pro, non-gly, pro, gly and pre-pro residues in protein structures. Favoured conformations are plotted in black, less favoured conformations are plotted in orange and disallowed conformations are shown in red. Figure 7.13(a) shows the RAMPAGE analysis of human haemoglobin (PDB record 1ch4, chain A) and Figure 7.13(b) shows the same analysis for a multiple to-pro mutant (prolines are introduced at positions 2, 32, 38, 48, 76, 86, 88, 96, 97, 117, 138, 142, 143 and 146) of the same structure. Comparison of the two figures demonstrates that the torsion angle conformations in the mutant structure are generally less favoured, with some conformations disallowed.

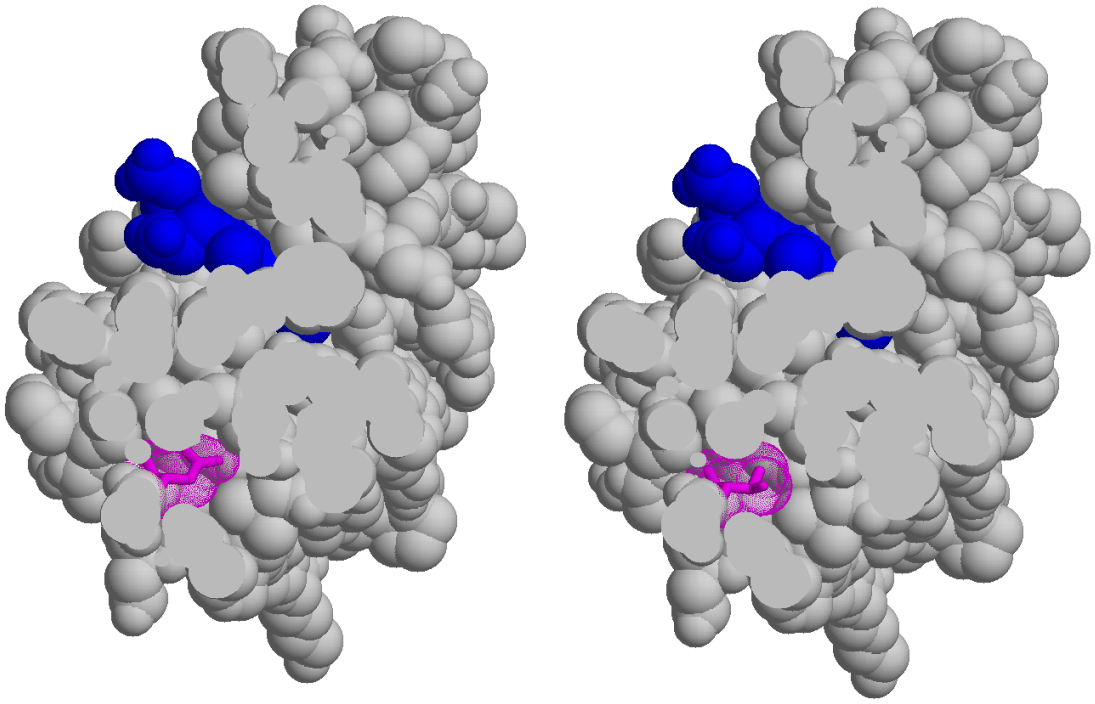
Figures 7.14(a) and 7.14(b) show the same comparison for the native and a multiple from-gly (specifically G→D mutations at positions 105, 154, 226, 245 and 262) mutant of chain A of the



(a) Native 1qsh:D (b) Mutant F42S 1qsh:D

Figure 7.9: PDs that create a void or crevice

PDB structure 1qsh, chain D. Replacing the native phenylalanine residue at position 42 with a serine residue (as shown on the right) creates a void or surface crevice which may destabilise the protein. Residue 42 is highlighted in magenta and the haem ligand is highlighted in blue. This mutation is also explained by affecting the PQS interface (i.e., affecting binding to the haem ligand).

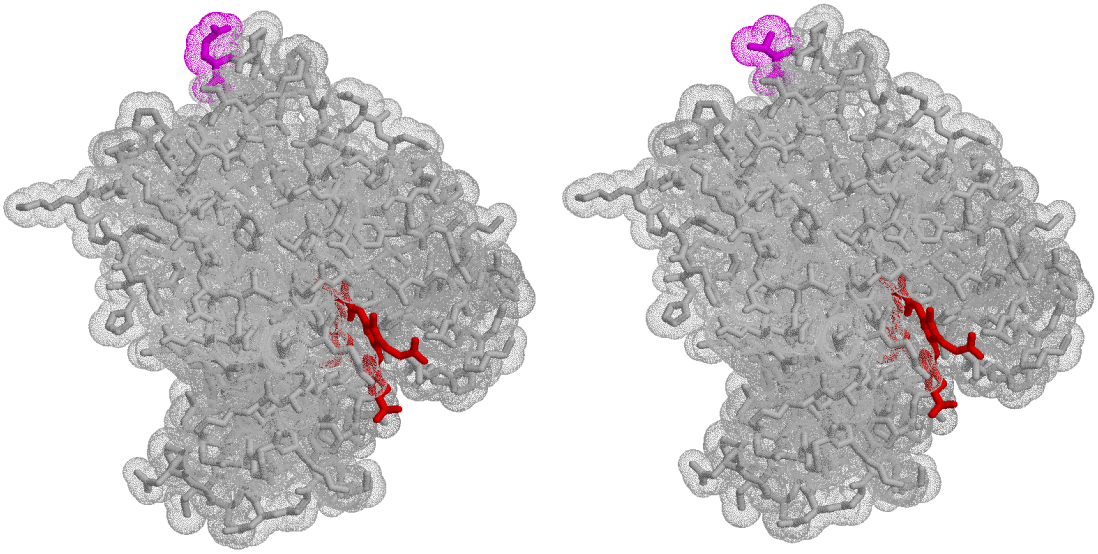


(a) Native 1r1y:D

(b) Mutant V54D 1r1y:D

Figure 7.10: PDs that introduce a buried, unsatisfied charge

PDB structure 1r1y, chain D. The buried charge analysis (see Section 5.8) identifies the V54D mutation in 1r1y as replacing a neutral valine residue (Figure 7.10(a)) with a negatively charged aspartic acid (Figure 7.10(b)), thus introducing a buried unsatisfied charge. Residue 54 is highlighted in magenta and the haem ligand is highlighted in blue. The mutation also introduces a hydrophilic residue in the core (see Section 5.10).

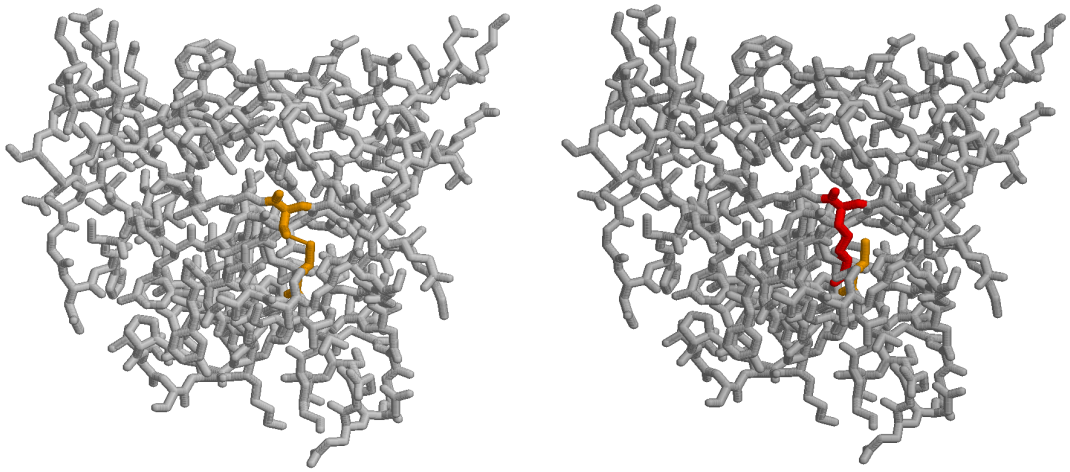


(a) Native 1hbb:D

(b) Mutant E6V 1hbb:D

Figure 7.11: PDs that introduce hydrophobic residues on the surface

PDB structure 1hbb, chain D. A mutation from glutamic acid to valine at residue 6 introduces a 'sticky' hydrophobic residue on the surface of 1hbb (see Section 5.9). Residue 6 is highlighted in magenta and the haem ligand is highlighted in red. This is the mutation that causes sickle cell anaemia.



(a) Native 1hl5:I

(b) Mutant C146R 1hl5:I

Figure 7.12: PDs that disrupt disulphide bonding

PDB structure 1hl5, chain I. A disulphide bond exists between C57 and C146 in chain I of 1hl5 (see native structure on the left). A mutation replacing C146 with an arginine (see mutant structure on the right, with the mutant arginine highlighted in red) is 'explained' both by the SSGEOM and UniProtKB/Swiss-Prot FT analyses (see Sections 5.6 and 5.11) as breaking a disulphide bond. The same mutation is also identified by ImPACT (see Chapter 4) and the clash analysis (see Section 5.3.6).

PDB structure 2bin (human P53). Again, there are a larger number of less well favoured and disallowed torsion angle conformations in the mutant structure.

Figure 7.15 shows the structure of glucosylceramidase [UniProtKB:P04062/GLCM_HUMAN] as described by the PDB record 2nt0. Many genomic aberrations have been identified in the corresponding GBA gene which are associated with Gaucher's disease, a disorder of lysosomal storage that results in a spectrum of neuropathologies (Jmoudiak and Futerman, 2005). 39 disease-associated glucosylceramidase mutations have been mapped to 2nt0, chain A and subsequently analysed by SAAPdb. In addition, a smaller number (10) of glucosylceramidase mutations described by dbSNP have been mapped to 2nt0, chain A and subsequently analysed by SAAPdb.

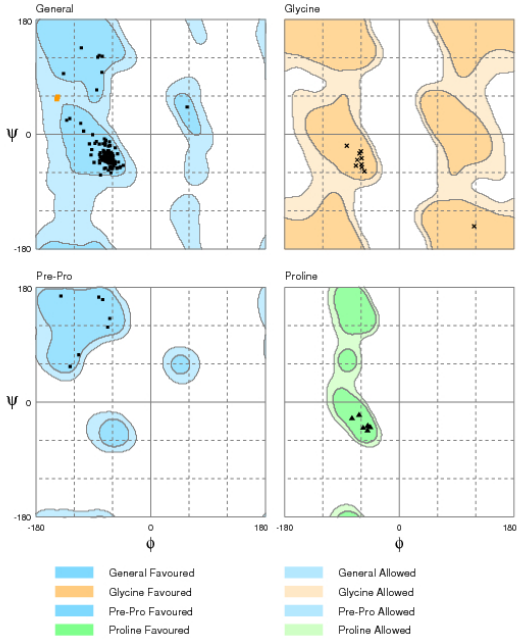
The 39 PDs (shown in yellow in Figures 7.15(a) and 7.15(b)) and 10 SNPs (shown in grey in Figures 7.15(c) and 7.15(d)) are mapped to chain A of 2nt0. In the left hand column (Figures 7.15(a) and 7.15(c)) all mutations mapped to the structure are shown, while only the *explained* mutations are shown in the figures in the right hand column (Figures 7.15(b) and 7.15(d)). Other protein chains in the structure are shown in grey with their Van der Waals volume suggested by a dotted surface.

It appears that the SNPs are more likely to be found near the surface of the protein chain, while the PDs are clustered more internally. Furthermore, SNPs appear to be more distant from ligands than PDs, although two SNPs are found very close to the interface with other protein chains.

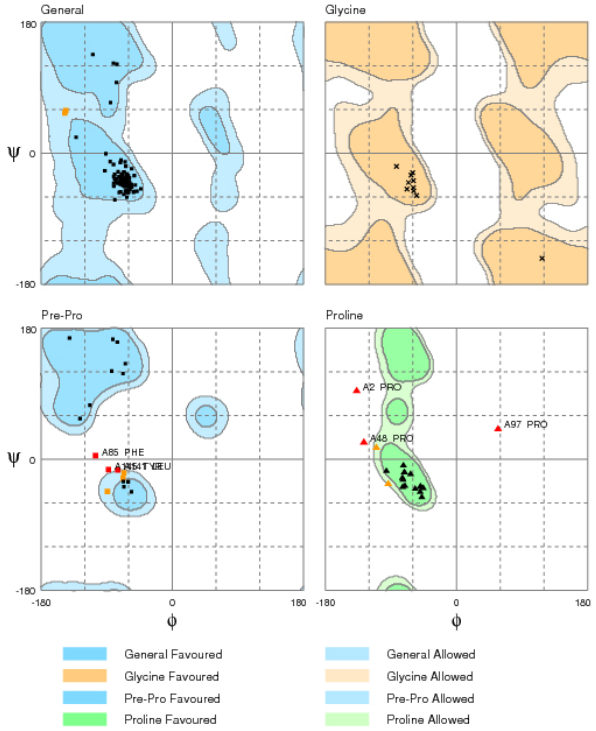
Figure 7.15 suggests that there are some differences between PDs and SNPs with respect to structure, at least in this example. The remainder of this chapter will discuss the large scale sequence, structural and structural effect analysis of disease-associated and neutral polymorphisms collated in SAAPdb, from which it is hoped statistically significant differences will be found.

7.3.2 PD residues are more often 'unique'

It is expected that neutral SNPs will more often be between 'replaceable' residues; that is, residues that can more easily replace each other without affecting protein structure and/or



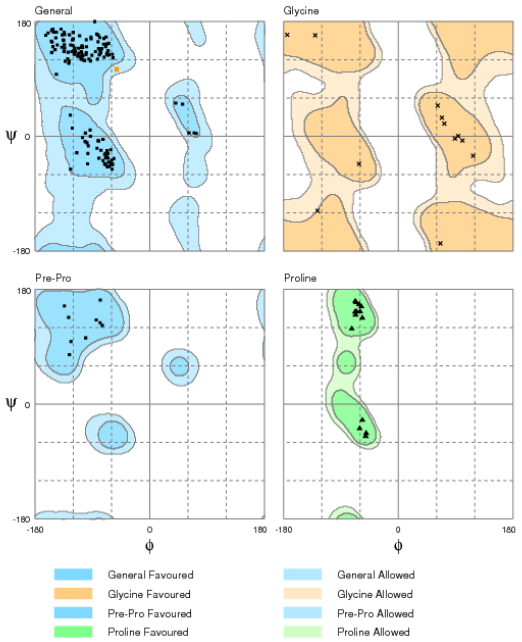
(a) Native 1ch4:A



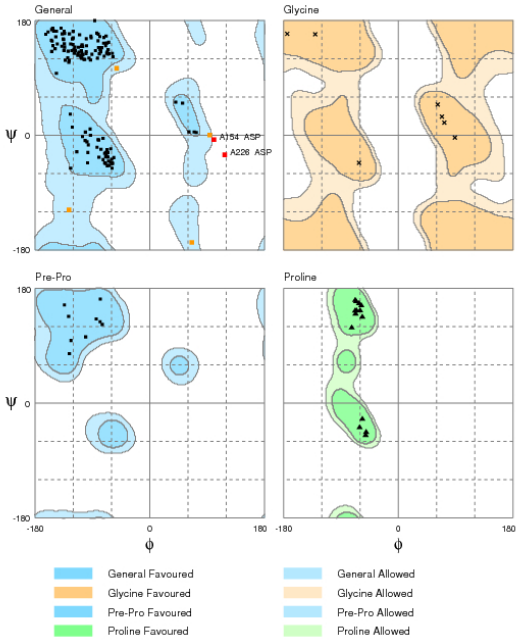
(b) Multiple mutant 1ch4:A

Figure 7.13: PDs that introduce proline where ϕ/ψ are not favourable

Fourteen conformationally unacceptable mutations to proline (at positions 2, 32, 38, 48, 76, 86, 88, 96, 97, 117, 138, 142, 143 and 146) are identified in PDB structure 1ch4, chain A [UniProtKB:P68871/HBB_HUMAN]. The RAMPAGE (Lovell *et al.*, 2003) analysis for the native and mutant structures are shown above in Figures 7.13(a) and 7.13(b) respectively. Each Figure shows the results for all residues (blue shading, top left), glycine residues (orange shading, top right), proline residues (green shading, bottom right) and pre-proline residues (blue shading, bottom left). Disallowed conformations are shown in red and less favourable conformations are plotted in orange. Note that there are many more unfavourable ϕ/ψ conformations in Figure 7.13(b).



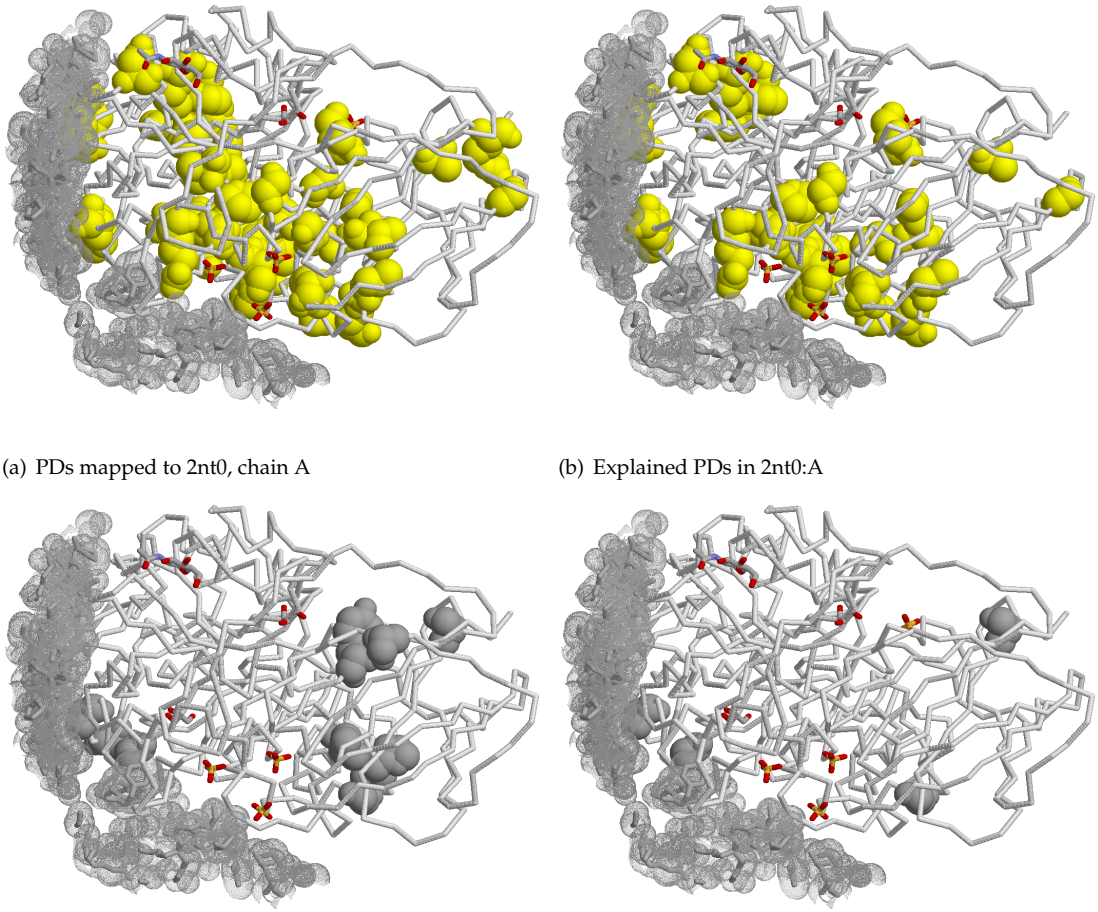
(a) Native 2bin:A



(b) Multiple mutant 2bin:A

Figure 7.14: PDs that replace glycine where ϕ/ψ are not favourable

Five conformationally unacceptable mutations from glycine to aspartic acid (at residues 105, 154, 226, 245 and 262) are identified in PDB structure 2bin, chain A [UniProtKB:P04637/P53_HUMAN]. The RAMPAGE (Lovell *et al.*, 2003) analysis for the native and mutant structures are shown above in Figures 7.14(a) and 7.14(b) respectively. Each Figure shows the results for all residues (blue shading, top left), glycine residues (orange shading, top right), proline residues (green shading, bottom right) and pre-proline residues (blue shading, bottom left). Disallowed conformations are shown in red and less favourable conformations are shown in orange. Note that there are many more unfavourable ϕ/ψ conformations in Figure 7.14(b).



(a) PDs mapped to 2nt0, chain A

(b) Explained PDs in 2nt0:A

(c) SNPs mapped to 2nt0, chain A

(d) Explained SNPs in 2nt0:A

Figure 7.15: The distribution of PDs and SNPs in glucosylceramidase

PDB structure 2nt0, chain A. Here, both the disease (shown in yellow in Figure 7.15(a)) and the neutral mutations (shown in grey in Figure 7.15(c)) are shown in spacefill in the diagrams in the left hand column. The *explained* PDs and SNPs are shown in yellow and grey spacefill in the diagrams in the right hand column (Figures 7.15(b) and 7.15(d) respectively). Residues from other chains are shown in grey, with their Van der Waals radii indicated by a dotted surface. Ligands are coloured using the CPK colour scheme.

Table 7.2: Characterising the twenty amino acids with respect to their 'replaceability'
Residue: the amino acid; δ: the average dissimilarity score (see text); Residues are sorted in ascending δ (i.e., from most 'unusual' residues to more 'replacable' residues). All data are rounded to 2dp.

Residue	δ
Tryptophan	-3.32
Cysteine	-3.21
Glycine	-3.16
Proline	-3.05
Aspartic acid	-2.68
Phenylalanine	-2.58
Isoleucine	-2.58
Leucine	-2.32
Valine	-2.16
Arginine	-2.05
Asparagine	-1.95
Tyrosine	-1.95
Histidine	-1.89
Glutamic acid	-1.79
Lysine	-1.68
Methionine	-1.53
Alanine	-1.42
Threonine	-1.37
Glutamine	-1.32
Serine	-1.16

function. To quantify how 'replaceable' or, conversely, how 'unusual' a residue is, an average dissimilarity value for each residue has been calculated by averaging the BLOSUM62 matrix values (see Section 2.3.4.2) of all mutations to/from the residue. The average dissimilarity value for a residue has been denoted with δ. The calculation is shown in Equation 7.2 below:

$$\delta_X = \frac{1}{n-1} \sum_{A \neq X}^{n} s(A, X) \tag{7.2}$$

where n is the number of amino acids (therefore, $n = 20$); X is the residue of interest; A denotes all other residues, and $s(A, X)$ denotes the BLOSUM62 amino acid substitution matrix score. The δ values for all twenty amino acids are shown in the second column of Table 7.2.

χ^2 tests show that there are significant differences between PDs and SNPs in terms of the native and mutant amino acids (Table 7.3). These results are discussed in detail in Sections 7.3.2.1-7.3.2.3.

Table 7.3: Mutant and native residues in the SAAP datasets

χ^2: the χ^2 statistic from a 2x2 Yates-corrected χ^2 test, testing for a difference in occurrence of the residue in the two datasets; **p**: the p-value ($\star\star$ denotes $p \leq 0.01$, $\star$ denotes $p \leq 0.05$); **set**: the SAAP set which has a higher occurrence of the corresponding amino acid. All numbers are rounded to 2dp.

Native residue		χ^2	**p**	**set**
Cysteine	$\star\star$	27.70	1.42×10^{-7}	PD
Glycine	$\star\star$	44.58	2.44×10^{-11}	PD
Arginine	$\star\star$	56.01	7.21×10^{-14}	PD
Tryptophan	$\star$	4.28	3.86×10^{-2}	PD
Tyrosine	$\star$	4.24	3.95×10^{-2}	PD
Alanine	$\star\star$	8.05	4.56×10^{-3}	SNP
Glutamic acid	$\star$	4.07	4.37×10^{-2}	SNP
Isoleucine	$\star\star$	32.62	1.12×10^{-8}	SNP
Lysine	$\star\star$	36.39	1.61×10^{-9}	SNP
Glutamine	$\star\star$	8.38	3.80×10^{-3}	SNP
Threonine	$\star\star$	15.32	9.06×10^{-5}	SNP
Valine	$\star\star$	19.20	1.18×10^{-5}	SNP

Mutant residue		χ^2	**p**	**set**
Cysteine	$\star\star$	29.66	5.16×10^{-8}	PD
Aspartic acid	$\star$	4.15	4.16×10^{-2}	PD
Proline	$\star\star$	46.38	9.74×10^{-12}	PD
Arginine	$\star\star$	22.62	1.98×10^{-6}	PD
Tryptophan	$\star\star$	8.76	3.07×10^{-3}	PD
Tyrosine	$\star\star$	8.58	3.39×10^{-3}	PD
Alanine	$\star\star$	9.74	1.80×10^{-3}	SNP
Phenylalanine	$\star\star$	19.43	1.04×10^{-5}	SNP
Isoleucine	$\star\star$	68.60	1.11×10^{-16}	SNP
Leucine	$\star\star$	9.23	2.38×10^{-3}	SNP
Asparagine	$\star\star$	7.11	7.66×10^{-3}	SNP
Valine	$\star\star$	17.59	2.74×10^{-5}	SNP

7.3.2.1 Native residues

Cysteine, arginine, glycine, tryptophan and tyrosine are more often native residues in the disease dataset; alanine, glutamic acid, isoleucine, lysine, glutamine, threonine and valine are mutated more often in the neutral dataset. The native residues associated with the SNP dataset are more 'replacable' than those associated with the PD dataset; that is, there is at least one other residue that behaves similarly and can often replace it without affecting function (for example, glutamic acid/aspartic acid, isoleucine/valine and lysine/arginine). The PD-associated native residues, which are more unusual in character, are less likely to be replaced without affecting function.

7.3.2.2 Replacement residues

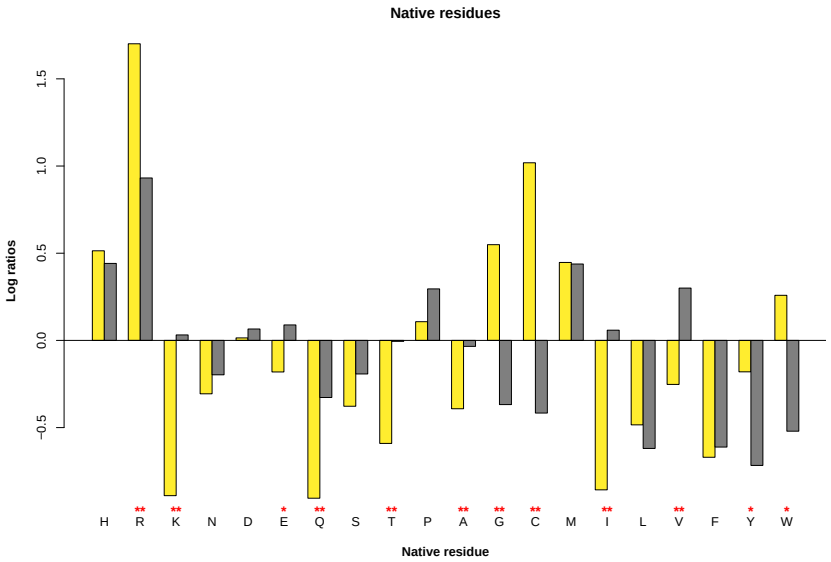
In terms of the residues introduced, there are significantly more deleterious cysteine, aspartic acid, proline, arginine, tryptophan and tyrosine mutant residues, and significantly more alanine, phenylalanine, isoleucine, leucine, asparagine and valine mutant residues in the neutral dataset. Once again, those residues common in the SNP dataset are more often replaceable, while the PD-associated residues are more unusual in character.

7.3.2.3 Discriminating residues

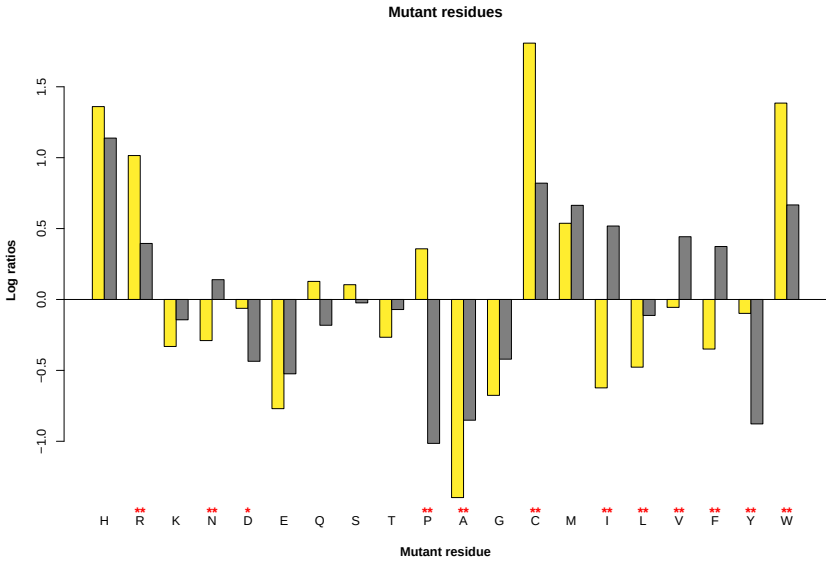
Figure 7.16(a) expresses each native amino acid as the log ratio of observed percentages over the expected percentages and Figure 7.16(b) shows the same data for the mutant residues (see Section 7.2.2.3 for a description of the log ratio calculation). The results described as significant in Table 7.3 are denoted with stars in Figure 7.16 (two where $p \leq 0.01$ and one where $p \leq 0.05$). Positive values in Figure 7.16 indicate that the amino acid is over-represented compared with the standard amino acid frequencies and negative values indicate under-representation.

With a view to discriminating between the two types of SAAP, the most interesting results are those that are significantly different from what is expected *and* over-represented in one dataset and under-represented in the other (see Section 7.2.3).

For native residues, cysteine and tryptophan were identified as 'discriminating' residues that



(a) Profiling SAAPs by native residue, normalising by standard amino acid frequencies



(b) Profiling SAAPs by mutant residue, normalising by standard amino acid frequencies

Figure 7.16: Profiling SAAPs by native and mutant residues

Data are normalised using standard amino acid frequencies (see Section 7.2.2.1). Statistically significant χ^2 results are denoted with red stars (two stars denote $p \leq 0.01$, one star denotes $p \leq 0.05$). Residues are ordered as suggested by Vitkup *et al.* (2003).PDs are shown in yellow; SNPs are shown in grey.

are enriched in the PD dataset, and glutamic acid, lysine, isoleucine and valine as 'discriminating' native residues that are enriched in the SNP dataset. For mutant residues, asparagine, isoleucine, phenylalanine and valine are favoured as mutant residues in the SNP dataset, while proline is the only mutant residue favoured in the PD dataset.

The discriminating residues associated with the PD dataset have unique roles in protein structure, while those associated with the SNP dataset have characteristics that are shared with other amino acids and so may more readily be replaced, without resulting in a disease phenotype. These results are supported by earlier work in which glycine, cysteine and tryptophan have been characterised as targets of deleterious polymorphisms (Vitkup *et al.*, 2003; Dobson *et al.*, 2006).

In Table 7.4, the δ dissimilarity score for the twenty amino acids is given; again, these data are ordered from the least to the most replaceable residue, as measured by δ. Those residues that have been described as discriminatory above (either as a native or mutant residue) are annotated with ‡ symbols; statistically significant results that were not found to be discriminatory are denoted with a †. The PD+^{ve} column is used for residues associated with or over-represented in the PD dataset, while the SNP+^{ve} column is used for residues associated with or over-represented in the neutral SNP dataset.

The top five residues, and therefore the five most unusual residues (as measured by δ) are more frequently found in the deleterious dataset. Of these five, three are discriminatory; that is, they are over-represented in the PD dataset while being under-represented in the SNP dataset. The SNP-associated residues have lower δ values, suggesting that they are more 'replaceable'.

7.3.3 PDs are more often between residues with different characteristics

In Section 7.3.2, the native or mutant residues were considered independent of their mutation partner. In this section, the native/mutant residue pairs are analysed.

Table 7.5 lists the discriminating mutations that (i) occur at significantly different rates compared with what is expected, and (ii) are found to be over-represented in one dataset and under-represented in the other (see Section 7.2.3). Ten of the eleven discriminating SAAPs that are associated with the deleterious dataset include at least one of glycine, cysteine or pro-

Table 7.4: Characterising the native/mutant amino acids observed in SAAPdb with respect to their 'replaceability'

Residue: the amino acid; δ: the average dissimilarity score (see text); **PD+**^{ve}: residues more often seen as mutant or native residues in the PD dataset are annotated with a † in the PD+^{ve} column, discriminating residues (see text) are annotated with a ‡. **SNP+**^{ve}: residues more often seen as mutant or native residues in the SNP dataset are annotated with a † in the SNP+^{ve} column, discriminating residues (see text) are annotated with a ‡. Residues are sorted in ascending δ (i.e., from most 'unusual' residues to more 'replacable' residues). All data are rounded to 2dp.

Residue	δ	**PD +**^{ve}	**SNP +**^{ve}
Tryptophan	-3.32	‡	
Cysteine	-3.21	‡	
Glycine	-3.16	†	
Proline	-3.05	‡	
Aspartic acid	-2.68	†	
Phenylalanine	-2.58		‡
Isoleucine	-2.58		‡
Leucine	-2.32		†
Valine	-2.16		‡
Arginine	-2.05	†	
Asparagine	-1.95		‡
Tyrosine	-1.95	†	
Histidine	-1.89		
Glutamic acid	-1.79		‡
Lysine	-1.68		‡
Methionine	-1.53		
Alanine	-1.42		†
Threonine	-1.37		†
Glutamine	-1.32		†
Serine	-1.16		

Table 7.5: Mutations found to be significantly over-represented in one dataset and under-represented in the other

χ^2: the χ^2 statistic from a 2x2 Yates-corrected χ^2 test; **p**: the p-value ($\star\star$ denotes where $p < 0.01$, $\star$ denotes where $p < 0.05$); **LR**: the log_2 ratio; **Blo62**: the BLOSUM62 score for this mutation. Results are ordered by the absolute difference between the two L_2R scores. All numbers are rounded to 2dp.

PD mutations		χ^2	**p**	**PD LR**	**SNP LR**	**Blo62**
Cys→Tyr	$\star\star$	14.72	1.25×10^{-4}	0.68	-3.53	-2
Tyr→Cys	$\star\star$	16.78	4.21×10^{-5}	1.09	-1.94	-2
Phe→Cys	$\star$	4.30	3.82×10^{-2}	0.86	-2.04	-2
Leu→Pro	$\star\star$	29.45	5.74×10^{-8}	2.32	-0.37	-3
Arg→Pro	$\star\star$	10.98	9.22×10^{-4}	0.80	-1.53	-2
Gly→Asp	$\star\star$	16.21	5.66×10^{-5}	1.40	-0.63	-1
Leu→Arg	$\star\star$	8.50	3.55×10^{-3}	1.21	-0.72	-2
Ser→Pro	$\star$	5.14	2.34×10^{-2}	0.19	-1.40	-1
Gly→Ser	$\star\star$	9.69	1.86×10^{-3}	1.18	-0.26	-1
Cys→Arg	$\star$	5.79	1.61×10^{-2}	1.19	-0.23	-3
Gly→Glu	$\star$	6.19	1.29×10^{-2}	0.95	-0.36	-2

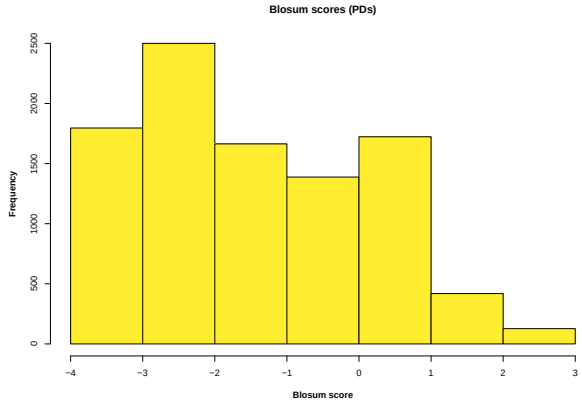
SNP mutations		χ^2	**p**	**PD LR**	**SNP LR**	**Blo62**
Asp→Glu	$\star\star$	36.05	1.92×10^{-9}	-1.21	0.77	2
Arg→Lys	$\star\star$	17.13	3.49×10^{-5}	-1.88	0.02	2
Ile→Phe	$\star\star$	16.13	5.91×10^{-5}	-1.65	0.19	0
Lys→Arg	$\star\star$	29.26	6.34×10^{-8}	-1.02	0.81	2
Val→Ile	$\star\star$	36.47	1.55×10^{-9}	-0.79	1.00	3
Ile→Val	$\star\star$	21.51	3.51×10^{-6}	-1.28	0.42	3
Leu→Val	$\star\star$	31.98	1.55×10^{-8}	-0.42	1.28	1
Gln→His	$\star\star$	21.51	3.51×10^{-6}	-1.21	0.49	0
Ala→Ser	$\star\star$	23.04	1.59×10^{-6}	-0.88	0.75	1
Glu→Asp	$\star\star$	31.86	1.65×10^{-8}	-0.45	1.13	2
Ile→Met	$\star\star$	10.58	1.14×10^{-3}	-1.35	0.10	1
Ser→Ile	$\star\star$	8.99	2.72×10^{-3}	-0.93	0.52	-2
Lys→Asn	$\star\star$	17.80	2.45×10^{-5}	-0.03	1.19	0
Glu→Gln	$\star\star$	8.34	3.88×10^{-3}	-1.06	0.12	2
His→Gln	$\star\star$	8.34	3.88×10^{-3}	-1.06	0.12	0
Met→Ile	$\star\star$	16.05	6.16×10^{-5}	-0.03	1.15	1
Val→Phe	$\star$	5.97	1.46×10^{-2}	-0.07	0.96	-1
Val→Ala	$\star\star$	9.24	2.37×10^{-3}	-0.15	0.87	0
Ser→Cys	$\star$	4.66	3.09×10^{-2}	-0.41	0.46	-1

line, the residues identified in Section 7.3.2 as a PD-favoured native or mutant discriminating residue. Interestingly, no from-tryptophan mutations are identified, despite Trp→X being identified as PD+^{ve} discriminating residue in Section 7.3.2. It is possible that the deleterious effect of from-tryptophan mutations is predominantly due to losing the characteristics of tryptophan and that the introduced residue is irrelevant; thus the counts for *each* Trp→X mutation are too low to be significant. It is also interesting to note that both Cys→Tyr and Tyr→Cys are the mutations that generate the two most disparate log ratio results in the two datasets (c.f. columns 4 and 5 in Table 7.5).

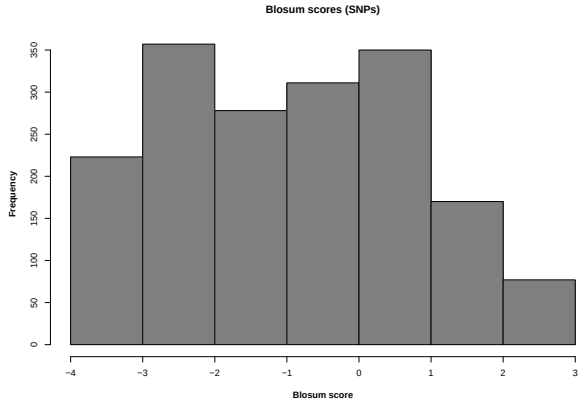
There are nineteen SNP-associated discriminating mutations, which include five pairs of 'commutative' mutations (i.e., X→Y and Y→X). The discriminating SNP-associated residues identified from Figure 7.16 commonly occur in this dataset (lysine, glutamic acid, isoleucine, valine, phenylalanine and asparagine). Once again, many of these mutations are between interchangable amino acids (for example, aspartic acid/glutamic acid, lysine/arginine, isoleucine/valine, leucine/valine and glutamine/glutamic acid).

The final column of Table 7.5 describes the discriminating mutations in terms of their BLOSUM62 score (Henikoff and Henikoff, 1992). It is striking that *all* eleven of the discriminating mutations enriched in the PD dataset have a negative score, while eleven of the nineteen discriminating mutations enriched in the neutral dataset have a positive score (five further mutations have a BLOSUM62 score of 0). This indicates that the mutations characteristic of the SNP dataset tend to be between similar residues, which more commonly replace one another in evolution, whereas PD mutations are more likely to be between amino acids which rarely replace one another in evolution.

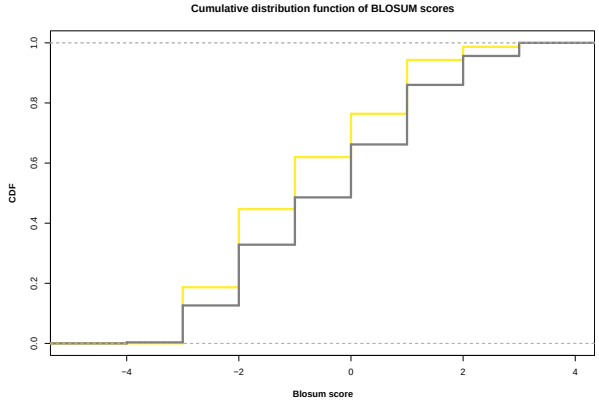
The distribution of BLOSUM62 scores for the deleterious and neutral mutations are shown in Figures 7.17(a)-7.17(c), and the distribution of their PAM30 scores are shown in Figures 7.18(a)-7.18(c). It is useful to compare PDs and SNPs using *both* amino acid substitution matrices because they are derived from different data: the BLOSUM62 matrix is derived from sequence alignments of more distantly related species than are used to derive the PAM30 matrix. Although the distributions are not continuous, the CDF plots in Figures 7.17(c) and 7.18(c) serve to show that PDs have lower BLOSUM62 and PAM30 scores than SNPs. A χ^2 test shows that these differences are statistically sigificant ($\chi^2_{df=15} = 533.55, p \simeq 0$ and $\chi^2_{df=33} = 315.34, p \simeq 0$ for BLOSUM62 and PAM30 scores respectively).



(a) Distribution of BLOSUM62 scores for PDs

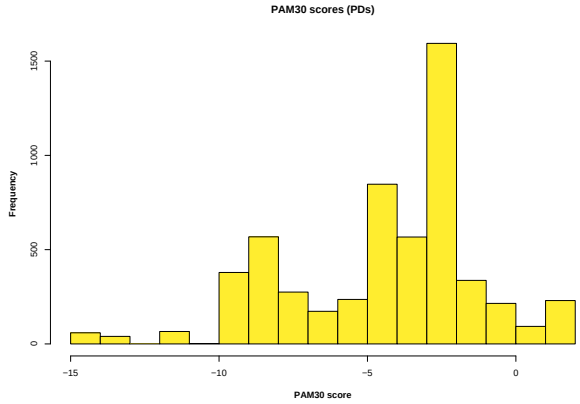


(b) Distribution of BLOSUM62 scores for SNPs

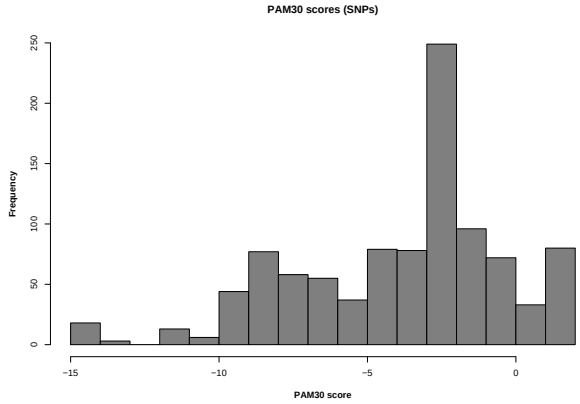


(c) CDF plot of PD/SNP BLOSUM62 scores

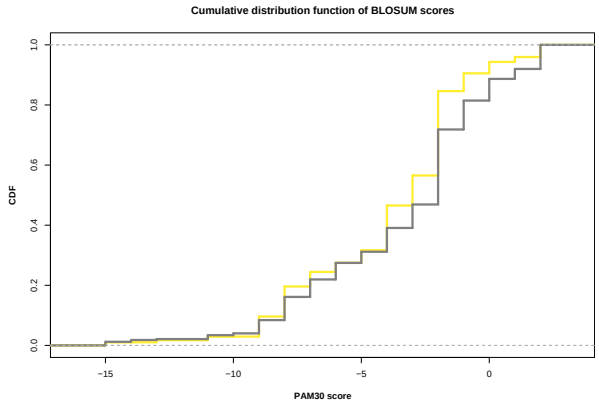
Figure 7.17: Profiling SAAPs by their BLOSUM62 AA substitution matrix scores
PDs are shown in yellow; SNPs are shown in grey.



(a) Distribution of PAM30 scores for PDs



(b) Distribution of PAM30 scores for SNPs



(c) CDF plot of PD/SNP PAM30 scores

Figure 7.18: Profiling SAAPs by their PAM30 AA substitution matrix scores
PDs are shown in yellow; SNPs are shown in grey.

To return to the concept of average dissimilarity as described in Section 7.3.2, Figure 7.19 shows a matrix, describing all possible mutations. The rows describe the native residues and the columns describe the mutant residues. Both the colums and rows are ordered by increasing δ. The matrix is partitioned into a pale yellow PD-associated area and a pale grey SNP-associated area. This is done by identifying the residue with the highest δ score that is identified as a PD+^{ve} discriminatory residue (proline, see Table 7.4) and using this residue as the inclusive PD+^{ve} threshold. Using this method, W, C, G and P cells are defined as the PD-associated area and all others are defined as the SNP-associated area.

PD+^{ve} discriminatory mutations (as described in Table 7.5) are then indicated by colouring the corresponding cell with a brighter yellow and SNP+^{ve} discriminatory mutations are indicated by colouring the corresponding cell with a darker grey. It is clear that the partitioning of the mutation matrix in this way is very successful in capturing the classification of the discriminatory mutations: only two discriminatory mutations are found in the 'wrong' region: the Leu→Arg mutation and the Ser→Cys mutation.

It is impossible at this stage to comment with any confidence as to whether the profile of residue substitutions will change if the site of the mutation is, for example, on the surface, in the core or at a functional site. However, there is a clear and significant tendency for PDs to be mutations to and from amino acids known to have a unique role in protein structure, and for SNPs to be mutations between physicochemically similar residues.

7.3.4 PDs affect sites of higher conservation

Residues that are highly conserved across diverse species have been consistently selected for across different branches of evolution. It is therefore likely that they are critical to protein function.

The histograms in Figures 7.20(a) and 7.20(b) describe the distribution of conservation scores for SNPs and PDs. These conservation scores are generated by the species-similarity weighted method (described in Section 4.2.1) used by ImPACT (see Chapter 4). Only data for those SAAPs that are mapped to alignments of 10 or more reliable, fully sequenced, functionally equivalent proteins as identified by FOSTA (see Chapter 3) are included here.

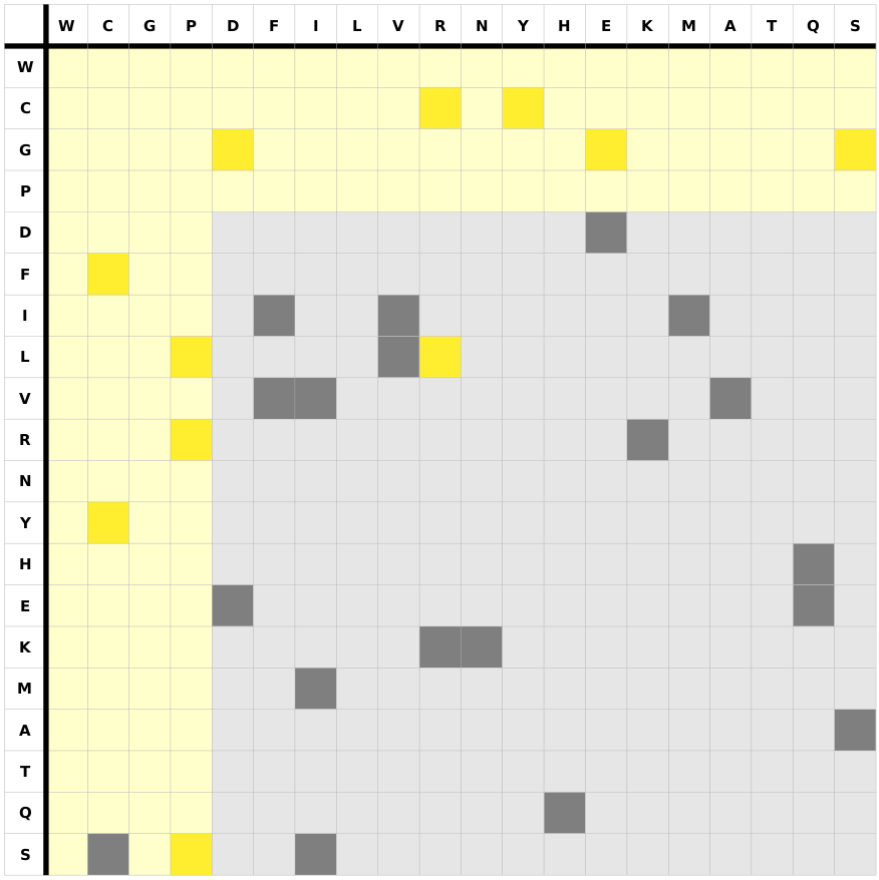


Figure 7.19: Characterising the SAAPs with respect to the 'replaceability' of the native and mutant residues

The rows describe the *native* amino acid, the columns describe the *mutant* amino acid in the polymorphism pair. Amino acids are ordered by increasing δ; that is, from most unique to more replaceable. The PD-associated mutation region (as defined by the dicriminatory mutations, see text) is coloured in pale yellow, while the SNP-associated mutation region is coloured in pale grey. Yellow and grey boxes denote discriminatory mutations (see text): yellow mutations are over-represented in the PD dataset and under-represented in the SNP dataset, grey mutations are over-represented in the SNP dataset and under-represented in the PD dataset. It is clear that PDs cluster in the top left hand half of the matrix, where mutations between the most unique residues are described.

It is clear from the CDF in Figure 7.20(c) that PDs more often occur at sites of high conservation. This trend is statistically significant ($D = 0.12, p \simeq 0$).

Despite the tendency for PDs to occur at sites of higher conservation, there is also a surprisingly large proportion of SNPs that are found at sites that are fully conserved. Such SNPs may be at more flexible or accommodating sites of protein structure (and the 100% conservation may therefore be a chance event) or there may be some tandem compensatory mutation nearby that stifles the effect of the polymorphism. Regardless of the mechanism by which the effects of SNPs at highly conserved sites are neutralised, it is clear that methods that rely exclusively on sequence data—in particular, sequence conservation—to discriminate between PDs and SNPs are limited in their potential.

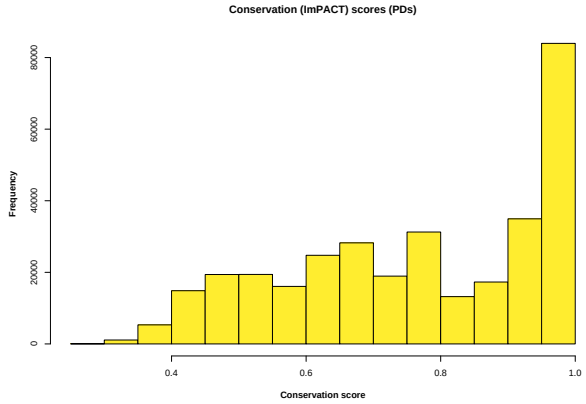
7.3.5 PDs and SNPs have the same torsion angle profiles

ϕ and ψ angles describe the dihedral angles involving the C-N'-C_α'-C' and N-C_α-C-N' respectively. Together they describe allowed conformational regions for amino acid backbones in the form of a Ramachandran plot (Ramachandran *et al.*, 1963). The ω torsion angle strictly adheres to a $\approx 0\ ^\circ$(*cis*), $\approx 180\ ^\circ$(*trans*) profile and defines the dihedral angle of the peptide bond (C_α-C-N'-C_α') which is delocalised. Beyond defining whether the peptide bond is *cis* or *trans*, the ω angle has little relevance to protein structure, at least in the context of the current analysis, and only ϕ and ψ angles are considered here.

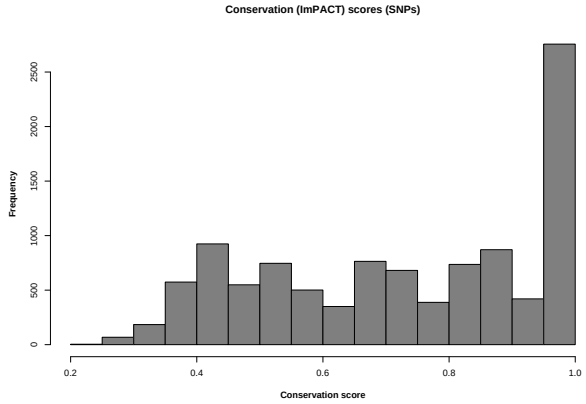
Figures 7.21(a)-7.21(c) and 7.22(a)-7.22(c) profile deleterious and neutral SAAPs in terms of the ϕ/ψ torsion angles at the site of mutation. There is no statistical difference between PDs and SNPs with respect to ϕ/ψ torsion angles ($D = 0.02, p = 0.54$ and $D = 0.02, p = 0.71$ respectively).

7.3.6 PDs and SNPs have the same secondary structure profiles

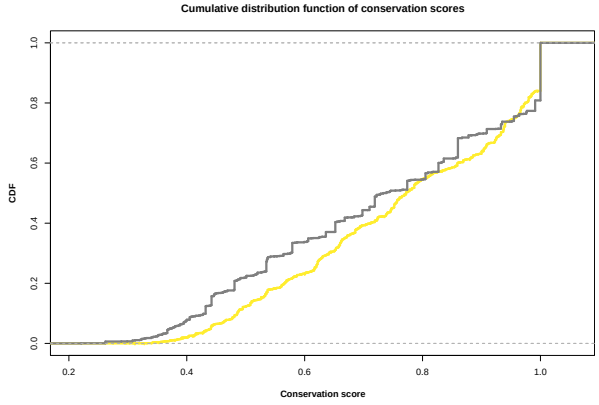
In Section 7.3.5, it was found that there is no difference between PDs and SNPs with respect to the ϕ/ψ torsion angles. In this section, a related analysis assesses whether there is a correlation between secondary structure and deleterious mutations.



(a) Distribution of conservation scores for PDs

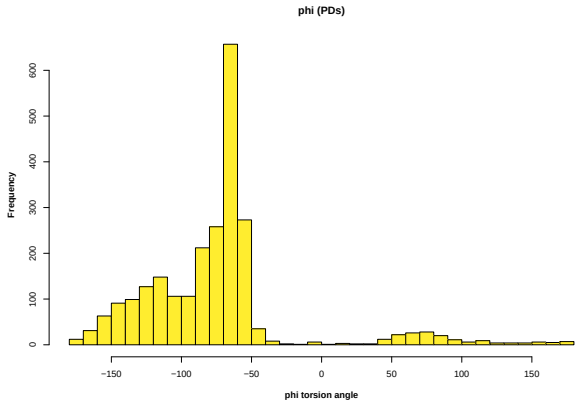


(b) Distribution of conservation scores for SNPs

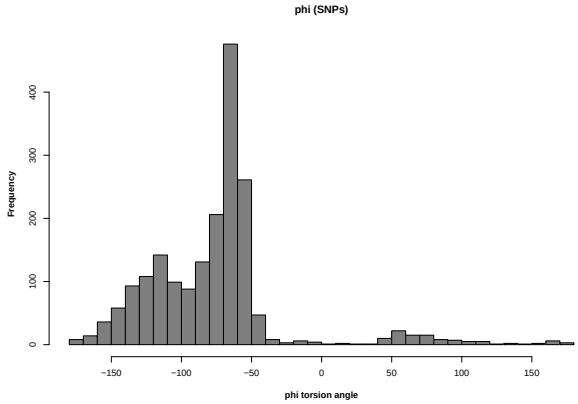


(c) CDF plot of PD/SNP conservation scores

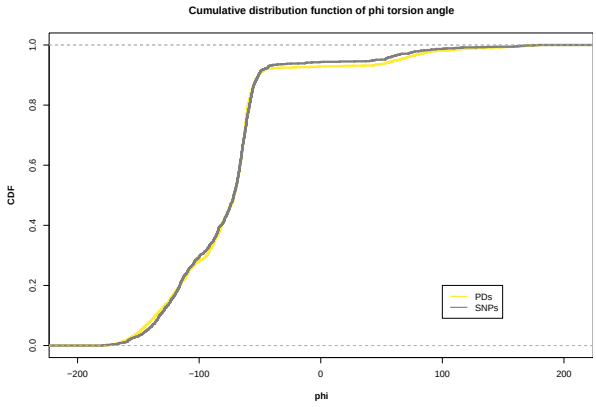
Figure 7.20: Profiling SAAPs by their `specsim` -weighted conservation scores
PDs are shown in yellow; SNPs are shown in grey.



(a) Distribution of ϕ angles for PDs

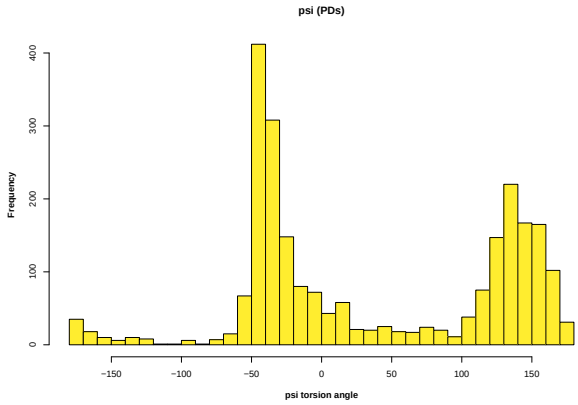


(b) Distribution of ϕ angles for SNPs

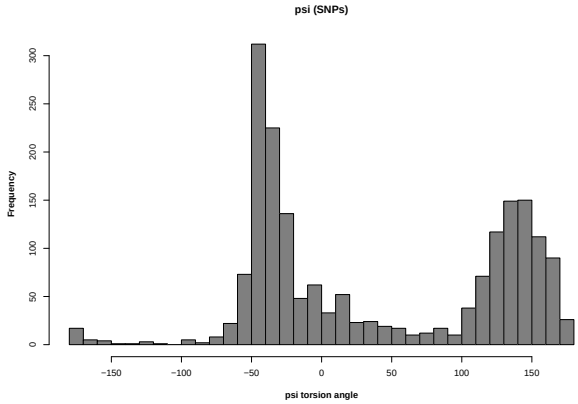


(c) CDF plot of PD/SNP ϕ angles

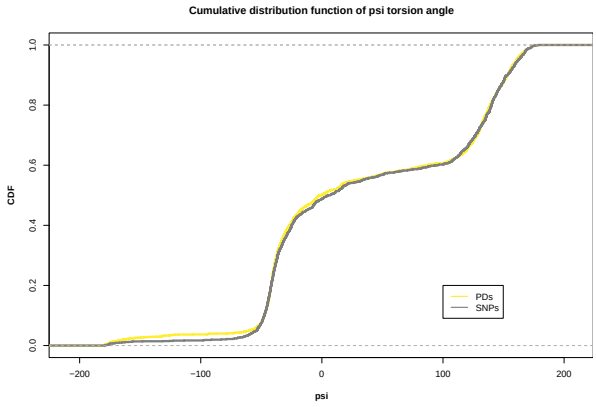
Figure 7.21: Profiling SAAPs by their ϕ torsion angles
PDs are shown in yellow; SNPs are shown in grey.



(a) Distribution of ψ angles for PDs



(b) Distribution of ψ angles for SNPs



(c) CDF plot of PD/SNP ψ angles

Figure 7.22: Profiling SAAPs by their ψ torsion angles
PDs are shown in yellow; SNPs are shown in grey.

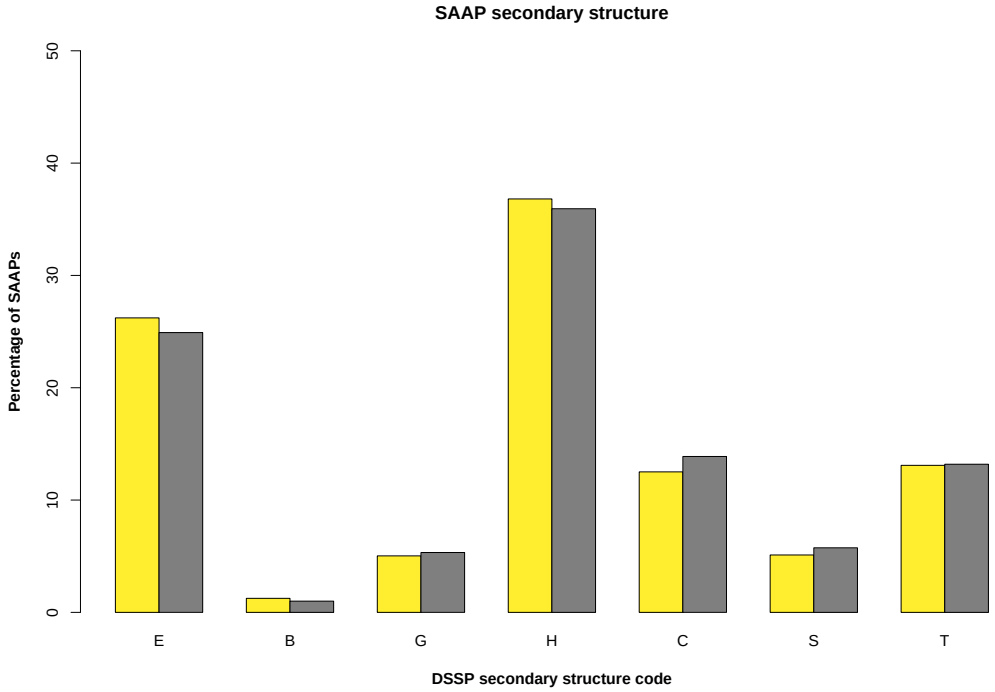


Figure 7.23: Profiling SAAPs by secondary structure
The proprotion of all represented secondary structures (represented by the corresponding DSSP code) in SAAPdb. PDs are shown in yellow, SNPs are shown in grey. **E** represents β-ladders; **B** represents β-sheets; **G** represents 3_{10} helices; **H** represents α-helices; **C** represents coils; **S** represents bends; **T** represents turns. E and B describe β structures; G and H describe helical structures; C, S and T describe loop structures. PDs are shown in yellow; SNPs are shown in grey.

DSSP (Kabsch and Sander, 1983) defines eight classes of secondary structure. The 3_{10}, α and π helical structures are denoted by G, H and I respectively and the β-strand and extended β-sheet are denoted by B and E. The third category describes all other 'loop' structures: C for coil, T for turn and S for bend.

Where a mutation is mapped to multiple structures, the mode secondary structure code has been recorded (see Section 7.2.1). No mutation is found to exist in the I (π-helix) conformation following this averaging process and no further data for this class are given.

Figure 7.23 shows how often PDs and SNPs are found in particular types of secondary structure. There does not appear to be any difference between PDs and SNPs with respect to secondary structure. This is confirmed by multiple χ^2 tests (which use pre-calculated expected values, see Section 7.2.2.1), the results of which are shown in Table 7.6.

Table 7.6: χ^2 tests comparing secondary structure in PDs and SAAPs
Comparison of PDs and SNPs with respect to their occurence in secondary structure elements. Results are for 2x2 (Yates corrected) χ^2 tests using real expected values, as calculated from CATH v3.0.0 HReps structures. **E** represents β-ladders; **B** represents β-sheets; **G** represents 3_{10} helices; **H** represents α-helices; **C** represents coils; **S** represents bends; **T** represents turns. E and B describe β structures; G and H describe helical structures; C, S and T describe loop structures.

DSSP code	χ^2	p	set
β **structures**	1.03	0.31	-
B	0.36	0.55	-
E	0.88	0.35	-
Helical structures	0.09	0.77	-
G	0.14	0.71	-
H	0.31	0.58	-
Loop structures	1.61	0.20	-
C	1.64	0.20	-
S	0.74	0.39	-
T	0.00	0.95	-

7.3.7 PDs are more commonly found in the protein core

Relative accessibility measures the accessibile surface area (ASA) in Å^2 as a proportion of the standard ASA observed for that amino acid in an extended Ala-X-Ala peptide. This is calculated using a local implementation of the Lee and Richards algorithm (Lee and Richards, 1971). Residues with a high relative accessibility will be found on the surface, while buried residues will have values of zero.

Figure 7.24(a) shows the distribution of monomer accessibility scores for PDs and Figure 7.24(b) shows the distribution of monomer accessibility scores for SNPs. Both sets of SAAPs appear to consist of two components: a concentration at $\leq$10% ASA (corresponding to buried residues) and another at 40-60% (corresponding to residues on the surface). The CDFs of both distributions (Figure 7.24(c)) show that there are proportionally more buried residues in the PD dataset than the SNP dataset. This difference is found to be signficant ($D = 0.076, p \simeq 0$).

A greater proportion of PDs are buried than SNPs. Residues in the core of the protein are generally critical to the stability of the structure and it follows that mutations in the core of the protein could critically affect protein stability and be deleterious. It is also likely that surface residues,

unless at critical functional sites or at tertiary or quaternary interfaces, can change more readily without disrupting protein structure and/or function. This trend has been identified elsewhere (Ferrer-Costa *et al.*, 2002; Chasman and Adams, 2001; Saunders and Baker, 2002; Krishnan and Westhead, 2003; Yue *et al.*, 2005).

7.3.8 PD residues are in contact with more other residues

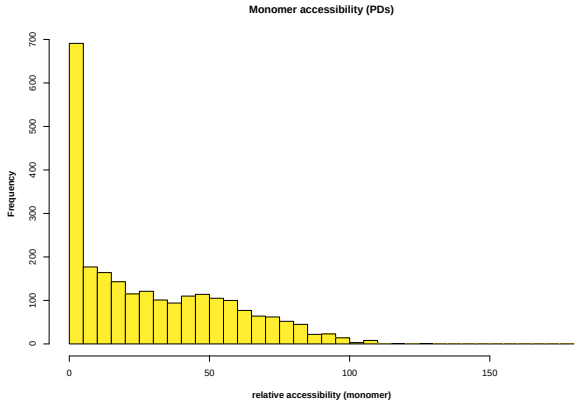
It is expected that mutating residues that are deeply embedded in the protein structure are more likely to cause disease. Indeed, in Section 7.3.7 it was shown that PDs are found to have lower relative ASA (i.e., are more buried) than SNPs. A complementary analysis considers the number of residue contacts. It is expected that mutations to residues that are in contact with many other residues will have a greater effect on structure than mutations to residues that have a low number of residue contacts.

An existing, locally-developed algorithm has been used to calculate the relative number of residue contacts (excepting primary-sequence-adjacent residues) for each residue in a protein structure. The raw number of residue contacts is normalised by the number of atoms in the target residue to generate the relative number of residue contacts, as described in Equation 7.3.

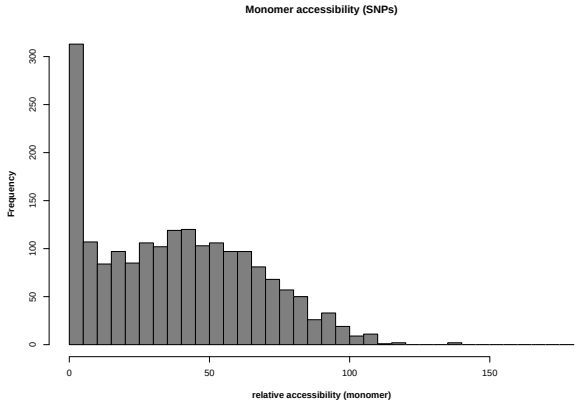
$$C_n^i = \frac{R_{d<3.5}^j}{N_i} \tag{7.3}$$

where C_n^i is the normalised contact number for a residue i; R is the number of residues, j, which make a contact at an atom centre distance of $< d$ and where $j \neq i$, and N_i is the number of atoms in residue i.

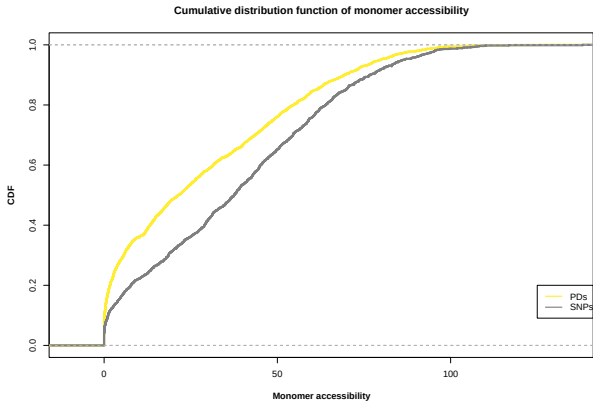
The graphs in Figure 7.25 describe the distribution of relative residue contacts in both datasets; the CDF in Figure 7.25(c) shows that PDs are in contact with a higher number of other residues than SNPs. This difference is found to be statistically significant ($D = 0.12, p = 3.05 \times 10^{-14}$).



(a) Distribution of ASA for PDs

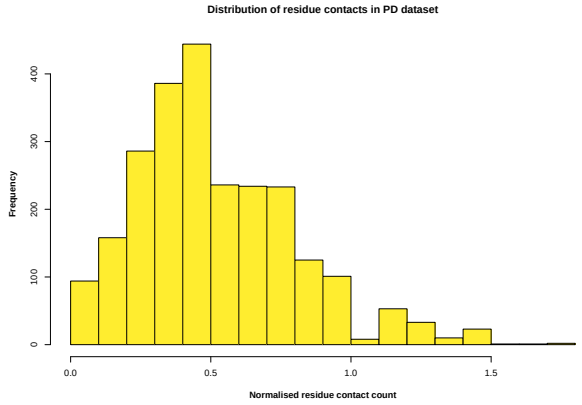


(b) Distribution of ASA for SNPs

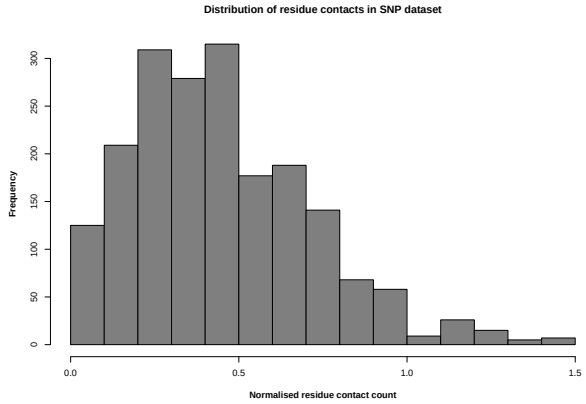


(c) CDF plot of PD/SNP ASA

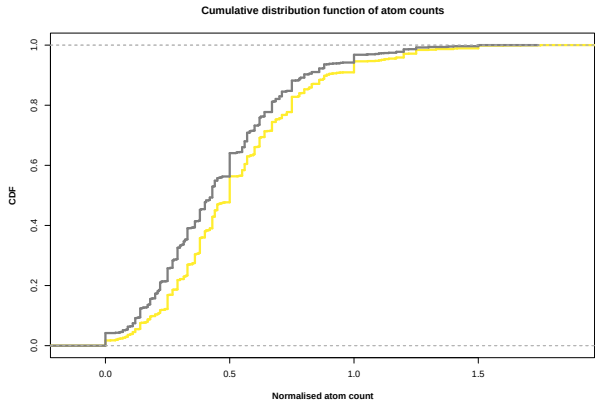
Figure 7.24: Profiling SAAPs by their relative accessible surface area (ASA)
PDs are shown in yellow; SNPs are shown in grey.



(a) Distribution of residue contacts for PDs



(b) Distribution of residue contacts for SNPs



(c) CDF plot of PD/SNP residue contacts

Figure 7.25: Profiling SAAPs by number of residue contacts
PDs are shown in yellow; SNPs are shown in grey.

7.3.9 PDs are more often explained

Sections 7.3.2-7.3.8 described PDs and SNPs with respect to basic sequence and structural characteristics. In this section, the SAAPs are profiled with respect to their structural 'explanations'.

One of the aims of SAAPdb is to 'explain' the effect of deleterious mutations by identifying their effect on local protein structure. It is hypothesized that mutations having an identifiable effect on protein structure will cause disease, whereas mutations that have little or no effect on protein structure will not cause disease. As such, it is expected that PDs will more often be explained by at least one of the analyses described in Chapter 5.

Figure 7.26 shows the results of the analyses, for each unique sequence mutation which has at least one mapped structure that generates a positive result for the corresponding explanation. The results of multiple χ^2 tests are compiled in Table 7.7. Disease mutations are more often explained by at least one of the analyses than neutral mutations (see bars marked 'EXPLAINED' in Figure 7.26): 87.17% of disease mutations are explained by at least one analysis compared with 58.68% of neutral mutations. This difference is highly statistically significant ($\chi^2_{df=1} = 552.99, p \simeq 0$).

7.3.10 PDs most often affect protein stability

All effects of mutations can be divided into three groups: (1) those which directly affect function (be it binding, catalysis, allosteries, etc.); (2) those which prevent correct folding; and (3) those which affect protein stability–i.e., they don't *prevent* correct folding, but destabilise the correct fold with respect to unfolded or misfolded states.

Much research has suggested that the deleterious effects of disease mutations are predominantly due to their effect on protein stability (Ferrer-Costa *et al.*, 2002; Ferrer-Costa *et al.*, 2004; Wang and Moult, 2001; Yue *et al.*, 2005). This is the most interesting category of mutations as these mutations have the potential to be 'rescued' by drugs which bind the correctly folded state (Boeckler *et al.*, 2008). In the current suite of structural analyses (see Chapter 5), there are six analyses that assess whether a mutation will make the protein structure unstable. Figure 7.26 shows that PDs are often explained by at least one of these analyses. In this section,

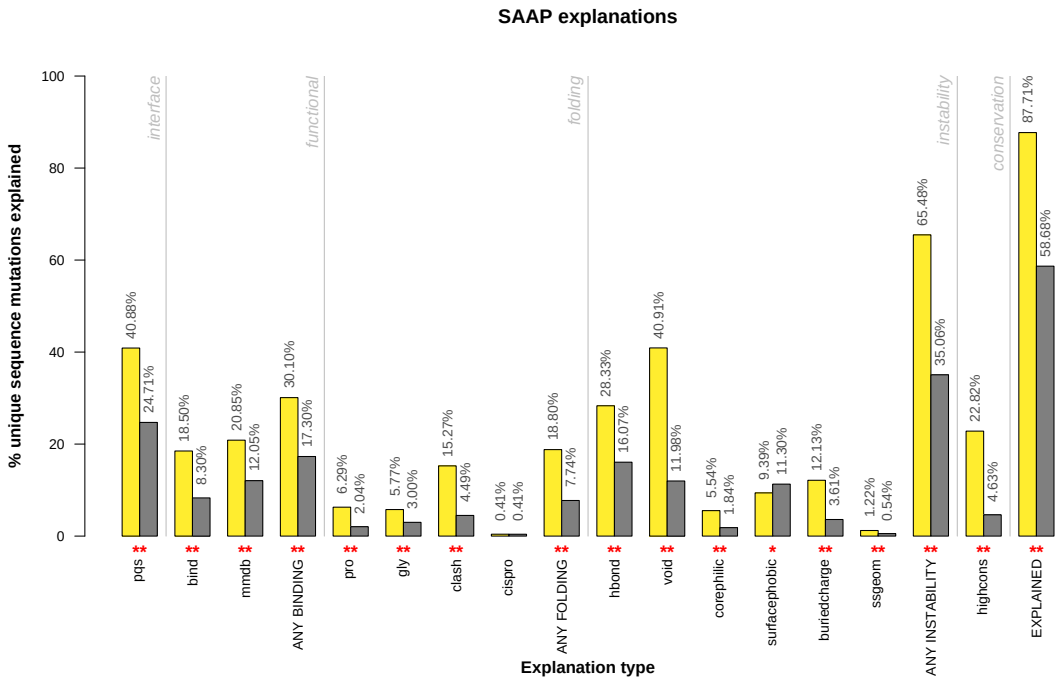


Figure 7.26: Profiling SAAPs with respect to explanations

pqs: the mutation disrupts an inter-chain interface as described by PQS; **bind**: the mutation disrupts a binding site as described by PDB; **mmdb**: the mutation disrupts a binding site as described by MMDBBIND; **ANY BINDING**: the mutation is positive for bind and/or mmdb; **pro**: the mutation introduces a proline where torsions are unfavourable; **gly**: the mutation replaces a glycine where torsions are unfavourable; **clash**: the mutation causes a steric clash in the hypothesised mutant structure; **cispro**: the mutation replaces a cisproline; **ANY FOLDING**: the mutation is positive for pro, gly, clash and/or cispro; **hbond**: the mutation breaks an existing hydrogen bond; **void**: the mutation creates a void in the protein core; **corephilic**: the mutation introduces a hydrophilic residue in the core; **surfacephobic**: the mutation introduces a hydrophobic residue on the surface; **buriedcharge**: the mutation introduces a buried unsatisfied charge; **ssgeom**: the mutation disrupts a disulphide bond as calculated from PDB coordinates; **ANY INSTABILITY**: the mutation is positive for hbond, void, corephilic, surfacephobic, buriedcharge and/or ssgeom; **highcons**: the mutation affects a highly conserved residue; **EXPLAINED**: the mutation is explained by at least one of the above analyses. Different 'classes' of explanation (i.e., interface, functional, folding, instability and conservation) are separated by pale grey vertical lines. Precise percentages are given above the corresponding bar. Statistically significant results are denoted with red stars (two where $p < 0.01$ and one where $p < 0.05$). For more information on these analyses, see Chapter 5. Yellow bars denote results for PDs, grey bars denote results for SNPs.

Table 7.7: Structural analysis of the SAAP datasets: individual explanations
Explanation the explanation (see Materials and Methods); χ^2: the χ^2 statistic from a 2x2 Yates-corrected χ^2 test; **p**: the p-value ($\star\star$ denotes $p < 0.01$, $\star$ denotes $p < 0.05$); **set**: the SAAP set which has a significantly higher occurrence of the corresponding explanation. ^{a} denotes a result with cell counts ≤ 10 but confirmed with Fisher-exact test. All numbers are rounded to 2dp.

Explanation		χ^2	**p**	**set**
PQS interface	$\star\star$	119.47	0	PD
PDB binding	$\star\star$	82.85	0	PD
MMDB binding	$\star\star$	54.31	1.71×10^{-13}	PD
Any binding	$\star\star$	173.85	0	PD
To-Proline	$\star\star$	38.63	5.13×10^{-10}	PD
From-glycine	$\star\star$	16.65	4.49×10^{-5}	PD
Clash causing	$\star\star$	113.52	0	PD
From-cisproline^a		0.04	8.33×10^{-1}	-
Any folding	$\star\star$	136.44	0	PD
Hydrogen bond break	$\star\star$	84.86	0	PD
Void creation	$\star\star$	401.84	0	PD
Hydrophilic in core	$\star\star$	32.91	9.67×10^{-9}	PD
Hydrophobic on surface	$\star$	4.13	4.22×10^{-2}	SNP
Buried unsatisfied charge	$\star\star$	86.69	0	PD
Disulphide (geometric)^a	$\star$	4.07	4.37×10^{-2}	PD
Any instability	$\star\star$	956.01	0	PD
High conservation	$\star\star$	239.39	0	PD
Explained	$\star\star$	552.99	0	PD

χ^2 tests are presented with percentages; note that where this is the case, the χ^2 tests have been carried out on the raw counts.

PDs more often break native hydrogen bonds (28.33% of PDs, 16.07% of SNPs/$\chi^2_{df=1} = 84.86, p \simeq 0$); more often create voids in the core of the protein (40.19% of PDs, 11.98% of SNPs/$\chi^2_{df=1} = 401.84, p \simeq 0$); more often introduce hydrophilic residues in the core of the protein (5.54% of PDs, 1.84% of SNPs/$\chi^2_{df=1} = 32.91, p = 9.67 \times 10^{-9}$); more often create a buried, unsatisfied charge (12.13% of PDs, 3.61% of SNPs/$\chi^2_{df=1} = 86.69, p \simeq 0$) and more often break disulphide bonds (1.25% of PDs, 0.54% of SNPs/$\chi^2_{df=1} = 4.07, p = 4.37 \times 10^{-2}$, $p = 0.033$ two-tailed Fisher exact test) than SNPs. Unexpectedly, it is found that SNPs more often introduce a hydrophobic residue on the surface of a protein (9.39% of PDs, 11.30% of SNPs/$\chi^2_{df=1} = 4.13, p = 4.22 \times 10^{-2}$); Saunders and Baker (2002) also found that this is a poor predictor of pathogenicity. This will in part be due to SNPs occurring more often on the protein surface than PDs (see Section 7.3.7).

Cumulatively, the instability analyses explain most of the PDs (65.48%), while only explaining

35.06% of SNPs, which is a significant result ($\chi^2_{df=1} = 956.01, p \simeq 0$). It appears that PDs are more often associated with destabilising changes in the core of the protein. This is consistent with the finding described in Section 7.3.7 that PDs are more often buried than SNPs.

7.3.11 Sequence conservation discriminates most successfully between PDs and SNPs

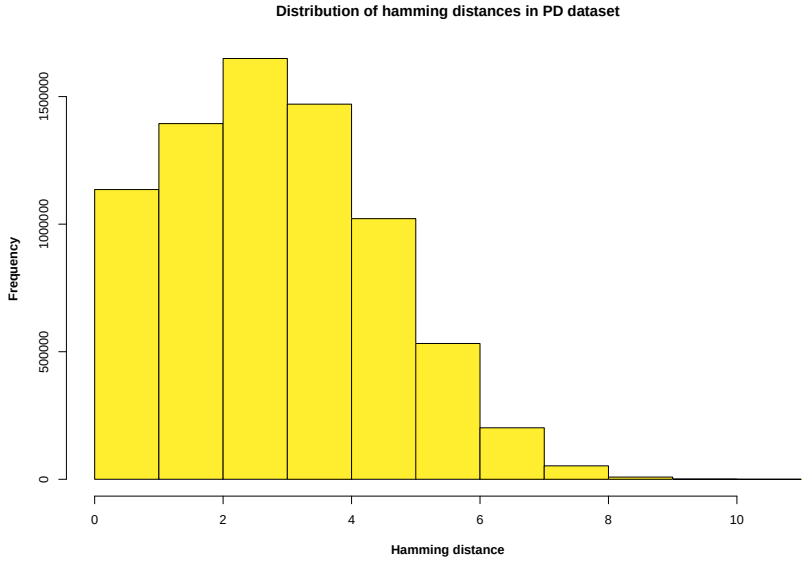
Figure 7.26 shows that 22.82% of disease mutations occur at a site of high conservation—as defined by ImPACT (Chapter 4)—whereas less than 5% of neutral polymorphisms affect highly conserved residues. This difference is highly statistically significant ($\chi^2_{df=1} = 239.39, p \simeq 0$) and is consistent with the hypothesis that mutating residues that are highly conserved is likely to be disease-causing.

7.3.12 PDs are more diverse in their structural explanations

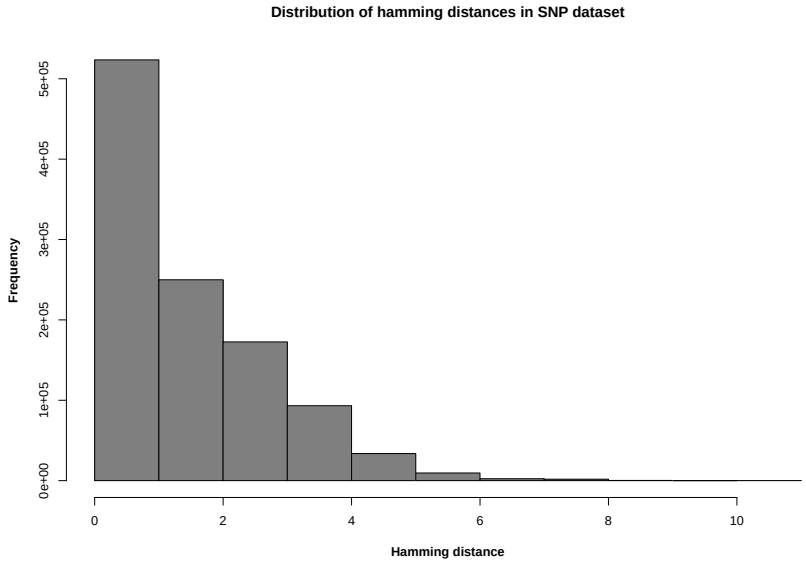
It is useful to characterise the PD and SNP datasets in terms of their homogeneity. It may be that SNPs are very similar to each other while PDs have a more diverse structural effect profile. Differences in explanation diversity as well as differences in the occurrence of particular explanations (as discussed in Section 7.3.10) are encouraging for future prediction work.

The explanation profiles can be represented as a binary vector, where 1 indicates a postive result for an analyses (i.e., an explanation), and 0 indicates a negative result (i.e., the absense of an explanation). A convenient metric for which to compare the explanation profiles, or calculate the *distance* between them, is the Hamming distance (here, denoted by D_H). The Hamming distance between two binary vectors is the number of corresponding elements in which they differ. For example, the Hamming distance between the binary vectors `01001` and `01100` is 2 (they differ at the third and fifth element). Explanation profiles that are very different will have a large Hamming distance (the maximum possible Hamming distance is the length of the vector, where all elements are different), while two identical profiles will have $D_H = 0$.

The Hamming distance between each pairwise comparison within each dataset was recorded. The graphs in Figure 7.27 characterise the PD and SNP dataset with respect to the diversity



(a) The distribution of hamming distances amongst PDs



(b) The distribution of hamming distances amongst SNPs

Figure 7.27: Profiling SAAPs by pairwise hamming distances within each dataset
PDs are shown in yellow; SNPs are shown in grey.

of the explanation profiles, as measured by the Hamming distance. Figure 7.27(b) shows that for the vast majority of SNP profile comparisons, $D_H \leq 1$. That is, most SNPs are identical in their explanation vectors, or differ by one explanation result. Note that the unexplained vector is included, and that the high frequency of $D_H = 0$ for the SNP data may be due to a high frequency of SNPs being unexplained and therefore identical. PDs, however, are more diverse: Figure 7.27(a) shows that the distribution of D_H is shifted to the right as compared to SNPs, indicating that for the majority of PD profile comparisons, $D_H \geq 2$.

However, it is possible that this effect is due to the PDs being explained more often by multiple analyses: if PDs are explained by multiple, simultaneous analyses more often than SNPs are (see Section 7.3.13), the Hamming distances will always be larger in an all-against-all pairwise comparison of PDs than in the same comparison within the SNP dataset. To account for potential bias in the number of *simultaneous* explanations, it is necessary to have some estimate of the 'background' or expected Hamming distance with which to normalised the data. This would require an, as yet unformed, comprehensive understanding of the co-occurrence of structural explanations in these datasets. Therefore, the Hamming distance distributions will not be compared and will only be used to characterise each dataset individually.

7.3.13 PDs are more often explained by multiple analyses

The analyses described in Section 7.3.9 showed that PDs are more often explained by at least one analysis. Here, the number of simultaneous explanations for PDs and SNPs are considered. In this section, the term 'simultaneous' will be used to describe two or more analyses that explain one SAAP. Given the hypothesis that PDs bring about significant structural disruption (potentially in more than one way) while SNPs do not, it is expected that PDs will be associated with multiple structural explanations while SNPs will be associated with very few.

Figures 7.28(a)-7.28(c) describe and compare the distribution of simultaneous explanations for PDs and SNPs and Table 7.8 describes the results of multiple χ^2 tests that compare PDs and SNPs with respect to the number of simultaneous explanations. SNPs are more likely than PDs to be explained by zero or one analyses ($\chi^2_{df=1} = 464.79, p \simeq 0$ and $\chi^2_{df=1} = 9.93, p = 1.63 \times 10^{-3}$ respectively). Cumulatively, these categories account for 71.46% of SNPs, but only 38.97% of PDs. PDs are more often explained by between two and seven explanations: 60.93% of PDs are explained by two to seven explanations, while only 28.40% of SNPs are explained by two to

Table 7.8: Structural analysis of the SAAP datasets: the simultaneous explanations
 # EXP: the number of simultaneous analyses that are positive, and so explain the SAAP; χ^2: the χ^2 statistic from a 2x2 Yates-corrected χ^2 test; **p**: the p-value ($\star\star$ denotes $p < 0.01$, $\star$ denotes $p < 0.05$); **set**: the SAAP set which has a significantly higher occurrence of the corresponding explanation; ^{a} denotes a result with cell counts ≤ 10 but confirmed with Fisher-exact test. All numbers are rounded to 2dp.

# EXP		%PDs	%SNPs	χ^2	p	set
0	$\star\star$	16.74	45.15	464.79	0	SNP
1	$\star\star$	22.23	26.31	9.93	1.63×10^{-3}	SNP
2	$\star\star$	21.84	15.59	25.90	3.59×10^{-7}	PD
3	$\star\star$	17.65	8.14	75.65	0	PD
4	$\star\star$	11.18	3.66	72.90	0	PD
5	$\star\star$	6.80	0.81	78.45	0	PD
6^a	$\star\star$	2.66	0.20	33.25	8.11×10^{-9}	PD
7^a	$\star\star$	0.80	0.00	11.90	5.61×10^{-4}	PD
8^a		0.10	0.14	0.10	0.75	-

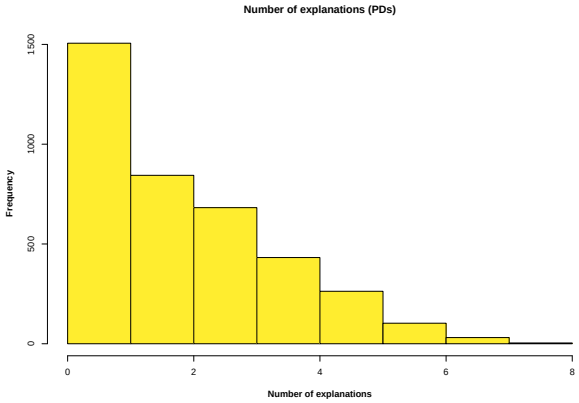
seven simultaneous explanations.

These results confirm that it is not possible to compare the distributions of Hamming distances (see Section 7.3.12) in the PD and the SNP datasets: as the PDs are more often explained by multiple analysis, the Hamming distances will be consistently greater than those derived from the SNP dataset. However, the Hamming distance analysis can contribute to the understanding of the structural effects of SAAPs as a measure of diversity *within* each dataset.

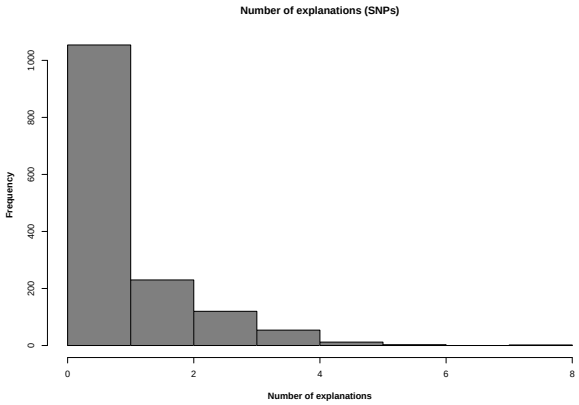
7.3.14 The most common explanation profiles are different for PDs and SNPs

It has been shown that PDs are more likely to be explained simultaneously by two or more analyses (see Section 7.3.13). Here, the most common explanation profiles for PDs and SNPs are identified.

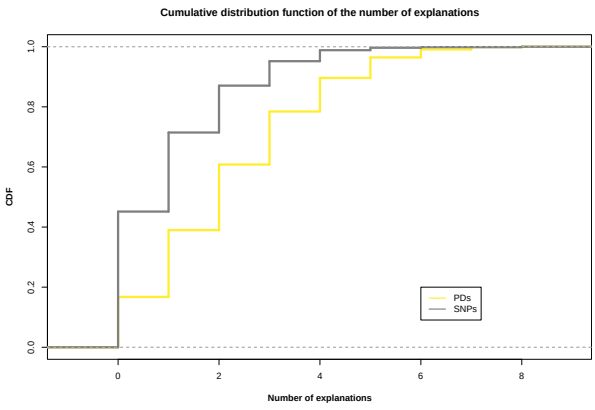
Not all the explanation profile combinations are shown here: to identify those explanation profiles that occur more often than expected, a rough estimate of the 'expected' frequency of each profile was calculated as N/E, where N is the size of the dataset (i.e., the number of SAAPs, or the number of observed explanation profiles) and E is the number of unique explanation profiles observed in that dataset. For example, consider a dataset of 100 SAAPs where five unique



(a) Distribution of the number of simultaneous explanations in PD dataset



(b) Distribution of the number of simultaneous explanations in SNP dataset



(c) CDF plot of PD/SNP simultaneous explanation data

Figure 7.28: Profiling SAAPs by the number of simultaneous explanations
PDs are shown in yellow; SNPs are shown in grey.

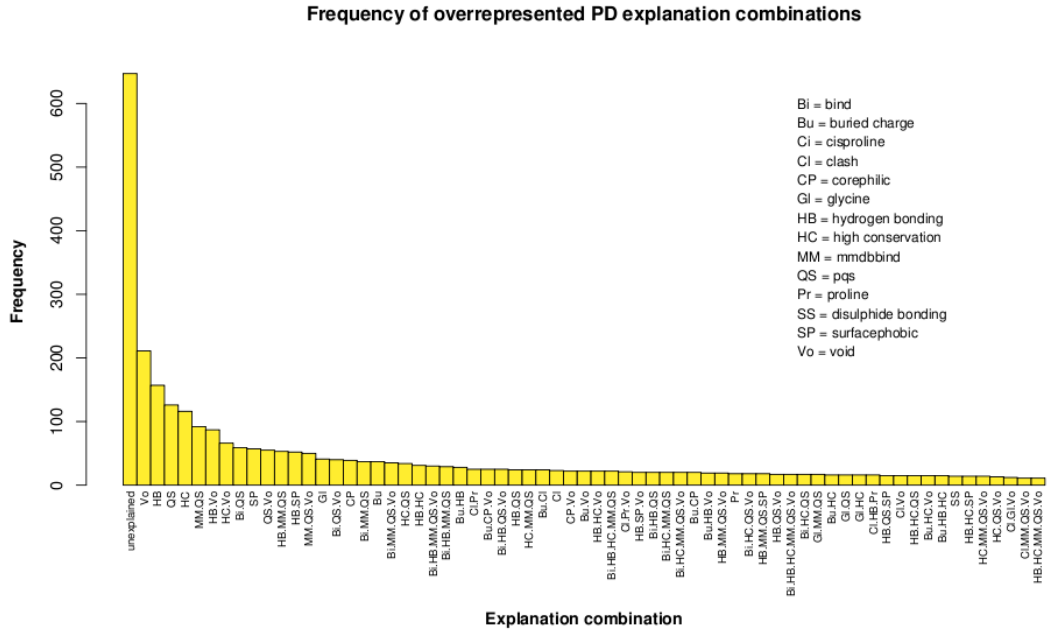
explanation profiles are observed; here, $N = 100$ and $E = 5$. The rough estimate of 'expected' frequency for each profile would be $100/5 = 20$. This estimation results in the same expected occurrence for each profile which is clearly not representative. Due to dependencies between analyses, the severity of the analyses themselves and aspects of protein structure, some explanations will occur together more often than others. For example, it may be expected that mutations identified as disrupting MMDBBIND sites (Section 5.5) will often also be identified as disrupting binding sites as extracted from the PDB (Section 5.3.3). However, as discussed in Section 7.3.12, it is difficult to estimate the expected frequencies of explanation profiles at present. Any profile that exceeds the rough estimate of the expected frequency will be described as 'enriched'.

Figures 7.29(a) and 7.29(b) show those explanation profiles that are enriched in the PD and SNP datasets respectively. The term 'positive profile' will be used to describe any explanation profile that contains at least one explanation, i.e., all observed explanation profiles excepting the 'unexplained' profile.

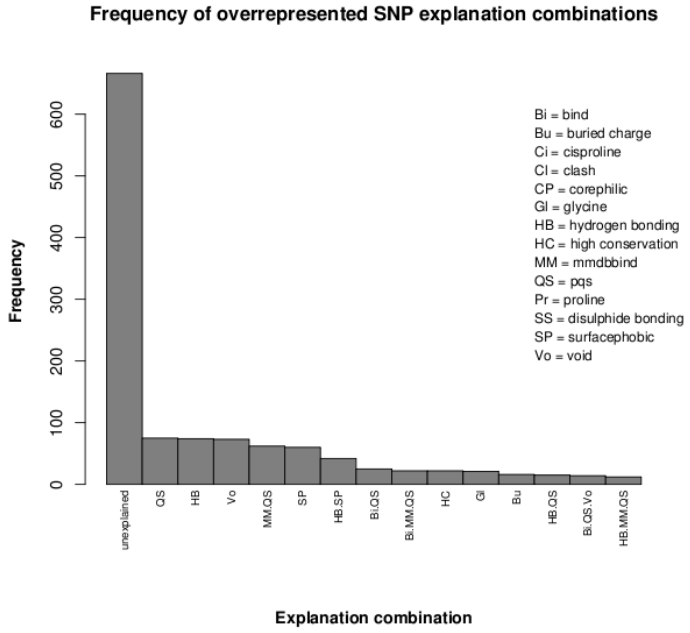
Firstly, far more positive explanation profiles are enriched in the PD dataset than in the SNP dataset. This suggests a stronger association between PDs and their positive explanation profiles than exists between SNPs and their positive explanation profiles. In Section 7.3.12 it was shown that SNPs are more similar to each other with respect to their explanation profiles than PDs, most often differing from each other in zero or one analyses. Taking this finding together with the trends shown in this Figure 7.29, SNPs can be characterised as primarily unexplained with the occasional explanation (which may be due to insensitivies in the analyses) while PDs can be characterised as primarily explained by at least two analyses.

Of the fourteen positive profiles that are enriched in the SNP dataset, half are single analysis explanations; similarly, a high proportion of the explanation profiles enriched in the PD dataset are multiple analysis explanation profiles. These observations support the findings of Section 7.3.13, where it was shown that SNPs are more likely to be associated with zero or one analyses and PDs are more likely to be associated with two to seven simultaneous explanations.

It is interesting to note that the three highest ranking positive profiles in both datasets are the void, hydrogen bonding and quaternary structure interface analyses. This indicates that—for these analyses more than any other single analyses—there is a differential in the extent to which the mutation affects protein structure. As discussed previously in Section 7.3.4 with respect to



(a) Frequency of explanation combinations for PDs



(b) Frequency of explanation combinations for SNPs

Figure 7.29: Profiling SAAPs by the explanation 'profile'

The explanation 'profile' of a SAAP is the set of analyses that 'explain' that SAAP. PDs are shown in yellow; SNPs are shown in grey.

the high number of SNPs found at sites of high conservation, it is possible that void-causing, hydrogen-bond-breaking and quaternary-structure-affecting mutations can more readily be accommodated by (i) adjustments in the existing structure or (ii) compensatory mutations. In an investigation into the phenomenon of compensatory mutations in SAAPdb, it was indeed found that PDs with identifiable compensatory mutations in other species are most commonly found to disrupt the quaternary interface, disrupt hydrogen bonding or introduce a void in the protein structure (Barešić *et al.*, 2009). Irrespective of the mechanism by which they are accommodated in SNPs, it is clear that these analyses in particular could be more sensitive and perhaps should *score* mutations rather than being binary classifiers.

Concentrating on those positive profiles common in the PD dataset, it is striking that neither the binding nor MMDBBIND analyses, both found to be significantly associated with PDs previously (see Figure 7.26 and Table 7.7), occur as single analyses or as a two-explanation pair. However, they are often found, both separately and together, in several ≥ 3 analysis explanation profiles. This suggests that for a mutation to occur at a binding site is often not enough to induce a deleterious phenotype. As suggested in Section 7.3.10, where it was found that PDs are more strongly associated with destabilising protein structure than affecting binding sites, perhaps the mechanics of protein-ligand binding are more flexible than might be expected.

7.4 Conclusions

This chapter described an analysis of the data in SAAPdb. PDs have most frequently been characterised as mutations between very different residues, where the introduced or replaced residue has a unique role in protein structure. They are found embedded in the protein structure, both in terms of accessibility and in terms of contacts with other residues, and they more often affect residues that have been conserved across different branches of evolution and therefore likely to have been subject to selection pressure. This characterisation is consistent with findings elsewhere (Wang and Moult, 2001; Ferrer-Costa *et al.*, 2002; Ferrer-Costa *et al.*, 2004; Yue *et al.*, 2005; Chasman and Adams, 2001; Saunders and Baker, 2002; Krishnan and Westhead, 2003; Dobson *et al.*, 2006; Vitkup *et al.*, 2003).

As described in Chapters 5 and 6, SAAPdb analyses all SAAPs with a view to identifying any local structural effect owing to the change in amino acid. PDs more often have a discernable

effect on protein structure than SNPs, and more often affect the structure in more than one way. Further, PDs have been found to affect protein structure in more diverse ways than SNPs.

It is perhaps surprising that PDs are not more often found at binding sites: disruption of ligand binding would directly affect protein function. More than double the number of PDs introduce instability in the protein structure than do affect binding sites (65.48% of PDs as opposed to 30.10%). There are several possible explanations for this trend: (i) protein-ligand binding is more flexible than is commonly believed; (ii) mutations at binding sites are fatal to the cell, therefore never observed in a patient, and therefore never reported as disease-associated, or (iii) given the mechanics of protein structure, a higher proportion of residues are involved in maintaining protein stability than are involved in binding sites. The similar ratio of instability explanations to binding explanations for SNPs (17.10% as compared to 35.05%) is roughly equivalent to the ratio of instability explanations to binding explanations for PDs ($17.10\% : 35.05\% \simeq 30.10\% : 65.48\% \simeq 2$), which lends some weight to the third theory. Further, Steward *et al.* (2003) demonstrated that the proportion of disease mutations identified as disrupting various types of interface (protein-ligand, protein-protein and so on) corresponds to the proportion of residues occuring at those interfaces. By estimating frequencies of each kind of 'explanation' in native protein structures, the results described in Section 7.3.9 could be corrected. However, without further investigation it is not possible to discount any of the proposed explanations.

In Sections 7.3.2 and 7.3.3, an average dissimilarity statistic δ was used to rank the amino acids from the most unique residue (tryptophan) to the most replaceable residue (serine). Although the calculation of this statistic is very simple (see Equation 7.2), it appears to be an effective statistic with which to describe amino acids when comparing disease-causing and neutral mutations and should be included in the feature vector when training machine learning algorithms. More work should be done to investigate such an average dissimilarity score.

Chapter 8

Conclusions

This thesis has described the SAAPdb database, a resource that collates information on single amino acid polymorphisms or SAAPs. SAAPdb attempts to identify the effects of disease mutations by providing hypotheses as to how they might disrupt structure and/or function.

In Chapter 3, a novel method of identifying functionally equivalent proteins (FEPs) was described, called FOSTA (Functional Orthologues from Swiss-Prot Text Analysis). In Chapter 4, a novel method of identifying high conservation within a multiple sequence alignment was described, called ImPACT (Improved Protein Alignment Conservation Threshold). Together, these chapters describe the one sequence analysis that has been incorporated into SAAPdb. Chapters 5 and 6 described the structural analysis pipeline with which SAAPdb analyses mutations and Chapter 7 described an analysis of the resulting SAAPdb data.

Here, the conclusions that can be drawn from the work presented in this thesis are collated.

8.1 Incorporating sequence data: FOSTA and ImPACT

8.1.1 FOSTA

The benchmarking of FOSTA demonstrated that not only are UniProtKB/Swiss-Prot annotations informative enough to capture the functionality of proteins, but that the simple string comparison methods employed by FOSTA are successful in identifying functional equivalence between proteins. Some inconsistencies identified by the FOSTA analysis of certain proteins (e.g., human protein C [UniProtKB:P04070/PROC_HUMAN]) have since been rectified by UniProtKB.

UniProtKB has recently changed the format of the description line from which FOSTA extracts the functional annotations[1]. Although the change is inconvenient, as existing parsers must be updated, the update has vastly improved the ease with which the desired data can be extracted. Furthermore, useful flags have been added to indicate whether the protein record describes, for example, a fragment. Figure 8.1 shows the change to the format of the horse protein C record, an example of a record of a protein fragment.

However, the comparison of FOSTA with Inparanoid did highlight some insensitivies in the methods with which FOSTA identifies functional equivalence. For example, mapping acronyms or short forms to long forms and vice versa would increase the functional match sensitivity of FOSTA. More generally, more 'fuzzy' matching should be employed to accommodate slight variations in names and numbers. See Table 3.11 for specific examples where these changes would lead to additional FEP assignments in FOSTA.

As pointed out in Chapter 3, annotation conventions in UniProtKB/Swiss-Prot are largely inherited from the source databases and organism-specific annotation communities. UniProtKB/Swiss-Prot is in a unique position to unify these annotations, thus avoiding any requirement for fuzzy matching, or for matching long forms of names with acronyms or abbreviations.

[1] http://www.expasy.ch/sprot/relnotes/spwrnew.html#rel56.0

```
ID   PROC_HORSE          Reviewed;        157 AA.
AC   Q28380;
DT   15-DEC-1998, integrated into UniProtKB/Swiss-Prot.
DT   01-NOV-1996, sequence version 1.
DT   12-JUN-2007, entry version 45.
DE   Vitamin K-dependent protein C (EC 3.4.21.69) (Autoprothrombin IIA)
DE   (Anticoagulant protein C) (Blood coagulation factor XIV) (Fragment).
 ⋮
```

```
ID   PROC_HORSE          Reviewed;        157 AA.
AC   Q28380;
DT   15-DEC-1998, integrated into UniProtKB/Swiss-Prot.
DT   01-NOV-1996, sequence version 1.
DT   02-SEP-2008, entry version 49.
DE   RecName: Full=Vitamin K-dependent protein C;
DE                EC=3.4.21.69;
DE   AltName: Full=Autoprothrombin IIA;
DE   AltName: Full=Anticoagulant protein C;
DE   AltName: Full=Blood coagulation factor XIV;
DE   Flags: Fragment;
 ⋮
```

Figure 8.1: Recent changes to the UniProtKB/Swiss-Prot flatfile format

Above is an extract from the record describing PROC_HORSE in UniProtKB/Swiss-Prot v53.0, below is the corresponding extract updated with the new formatting (as of UniProtKB/Swiss-Prot v56.0). The corresponding EC, synonym and fragment flag data are shown in red, blue and green respectively. The improvements facilitate fast extraction of the relevant information using regular expressions, for example.

8.1.2 ImPACT

ImPACT was benchmarked against four representative proteins and against a dataset of sequence motifs extracted from PROSITE. In addition, a battery of artificial conservation data was designed to test the response of ImPACT to controlled variations in the distribution of conservation scores.

The more qualitative evaluations (using four representative proteins and the artificial conservation data) demonstrated that ImPACT generates sensible thresholds for many different conservation score datasets. The more quantitative PROSITE analysis, despite concerns regarding the definition of negative examples, was possibly more informative in that it drew attention to the fact that ImPACT can perform poorly in sparse alignments. Extensive gaps will generate very low conservation scores which could dominate the distribution of conservation scores to be modelled. A valuable addition to the ImPACT method would be to disregard data from columns that are not adequately represented across the aligned proteins.

A further valuable addition to ImPACT would be some measure of confidence that would accompany each generated threshold. This could be a combination of (i) how well the mixture model has fitted the data; (ii) some measure of the deviation from the threshold that would be generated from random data; and (iii) some measure of the diversity and number of species included in the alignment. It is possible to calculate the former—all modelling methods provide a measurement of error—but it is more difficult to define how the latter confidence components might be calculated. An alternative to using a background distribution of random data (which is likely to deviate from characteristics of multiple sequence alignments) is to use permuted data, or to use alignments of known functionally deviant proteins to construct a background distribution of conservation scores. This might be possible by considering the functionally diverged homologues (FDHs) generated by FOSTA.

Included in the set of representative proteins with which ImPACT is assessed are P53 and haemoglobin (HBB), two proteins for which a large number of disease mutations (1 712 and 423 respectively) are analysed in SAAPdb. Although a higher proportion of P53 residues are 100% conserved, the HBB alignment has a higher mean conservation score and is the more *globally* conserved protein. To accommodate the higher global conservation, ImPACT correctly generates a higher ImPACT threshold of 0.9763 for HBB than it does for P53 (0.9636). However, mutations to HBB are rarely fatal and more often lead to mild anaemia, whereas mutations to

P53 are found in approximately 50% of human cancers (Greenblatt *et al.*, 1994; Sidransky and Hollstein, 1996). As such, a higher threshold for HBB appears to depart from what might be expected: because the phenotypic effect of HBB mutations is less severe than that of P53, it might be expected that the threshold for high conservation would be *lower* in HBB than P53. This apparent incongruity could be due to any number of differences between the two proteins, including their differential roles; their involvement in multiple pathways; the differential flexibility of the structures and so on. Regardless of the mechanisms by which HBB can withstand mutations, it is clear that sequence information alone cannot capture the differential phenotypic effects of these two proteins.

8.2 The analysis of disease mutations

Chapter 7 described a broad analysis of the data in SAAPdb. The existing characterisation of deleterious PDs (pathogenic deviations) and neutral SNPs (single nucleotide polymorphisms) was summarised in Table 1.9. This summary is updated in Table 8.2 to include the findings of Chapter 7 (which are highlighted in yellow) and to compare them to the characterisation that exists in the literature. Where the results described in this thesis overlap with previous work, they largely agree with the existing characterisation of disease and neutral SAAPs; where SAAPdb has used novel analyses (e.g., identification of unsatisfied buried charges, unfavourable voids and broken hydrogen bonds), the results complement the existing characterisation of disease mutations. Several significant differences—with respect to sequence, structure *and* structural effects—have been identified when comparing disease-associated PDs with phenotypically neutral SNPs. These findings have implications for future predictive methods and may facilitate the identification of drug targets.

8.2.1 Understanding the data

PDs are characterised as mutations involving the more 'unusual' amino acids, specifically tryptophan, cysteine and proline. The PDs have lower BLOSUM62 and PAM30 amino acid substitution matrix scores than SNPs, indicating that PDs tend to describe mutations between more different residues. PDs are found more often in the protein core, in contact with a larger number of other residues and most often disrupt the stability of the protein structure; they most often

Table 8.1: Comparing SAAPdb findings with the existing characterisations of PDs and SNPs
●: PDs were associated with this feature; ○: SNPs were associated with this feature. '-': no relationship was found. ⋆: the paper includes some prediction work. †: the paper considered only deleterious data. A blank cell denotes that the feature was not considered. **Datasets**: *A* = LacI repressor (Suckow *et al.*, 1996); *B* = T4 lysozyme (Rennell *et al.*, 1991); *C* = HIV protease (Loeb *et al.*, 1989); *D* = dbSNP (Smigielski *et al.*, 2000); *J* = uses natural residue variation across species to represent 'neutral' SAAPs; *M* = HMGD (Stenson *et al.*, 2003); *N* = HGVBase (Fredman *et al.*, 2002); *O* = OMIM (McKusick, 1998; Amberger *et al.*, 2009); *S* = UniProtKB/Swiss-Prot VARIANT (The UniProt Consortium, 2009); *X* = other LSMDB (various references). **Structural data used**: Y^* = where PDB structures were unavailable, models were used; Y^i = structural features were inferred from sequence. **AA**: amino acid. SAAPdb findings (see Chapter 7) are highlighted in yellow.

Reference	Data			Seq features			Str features						Extreme changes	Other
	Dataset(s)	Sequence data used?	Structural data used?	AA preference	AA matrix	Conservation	Buried	Destabilising	Secondary structure	Interface	Binding	Buried charge		
Bao & Cui (2005) ⋆	S	Y	Y			●	●		●				●	●
Cai *et al.* (2004) ⋆	AB	Y	Y			●							●	
Chasman & Adams (2001) ⋆	AB	Y	Y			●○	●					●	●	
Clifford *et al.* (2004) ⋆	DX	Y	Y^*											●
Dobson *et al.* (2006) ⋆	S	Y	Y	●							●			
Ferrer-Costa *et al.* (2002)	S	Y	Y		●		●	●	●	●			●	
Ferrer-Costa *et al.* (2004) ⋆	AS	Y	Y^i		●									
Khan & Vihinen (2007) †	MX	n	Y	●					-				●	
Krishnan & Westhead (2003) ⋆	AB	Y	Y^i			●	●					●	●	
Needham *et al.* (2006) ⋆	AB	Y	Y				●		-			●	●	
Ng & Henikoff *et al.* (2001) ⋆	ABC	Y	n			●								
Saunders & Baker *et al.* (2002) ⋆	AC	Y	Y		-	●	●						●	●
Stitziel *et al.* (2003)	DO	Y	Y			●	-	●						●
Steward *et al.* (2003)	O	Y	Y			●	●						●	
Sunyaev *et al.* (2001) ⋆[a]	DNS	Y	Y				●		●		●	●	●	●
Torkamani & Schork (2007)	X	Y	Y	●○		●								●
Verzilli *et al.* (2005) ⋆	AB	Y	Y			●	●	●				●		●
Vitkup *et al.* (2003)	S	Y	Y	●		●	●						●	
Wang & Moult (2001) [b]	DM	n	Y					●						
Yue *et al.* (2005) ⋆	JM	n	Y^*				●	●						●
SAAPdb [c]	DOX	Y	Y	●○	●	●	●	●	-	●	●	●	●○	●

[a] `http://genetics.bwh.harvard.edu/pph/`
[b] `http://www.snps3d.org/`
[c] `http://www.bioinf.org.uk/saap/db/`

introduce a void or crevice in the protein structure. Further, PDs are more likely than SNPs to disrupt sites of high conservation.

It appears that the simple average dissimilarity score used to rank amino acids in terms of their 'replaceability' (described by Equation 7.2) is a powerful measurement with which to represent the amino acids. More work is required to realise fully the potential of such a statistic in the context of disease mutations.

It is perhaps surprising that PDs are not more often found at binding sites: disrupting ligand binding would directly affect protein function. More than double the proportion of PDs introduce instability in the protein structure than do affect binding sites (65.48% of PDs as opposed to 30.10%, see Figure 7.26). This may indicate that interactions at binding sites are more flexible than previously thought. Alternatively, it may simply reflect the proportion of residues that are involved in instability compared with binding (as suggested in Steward *et al.* (2003)); or it may draw attention to the spectrum of disease mutations that *can* be observed: many mutations at the binding site may be fatal to the cell and therefore will never be observed in a living patient; should this be the case, the only observed mutations at binding sites will not be deleterious. More investigations are required before any confident statement can be made; a useful first step would be to quantify the number of binding-associated and instability-associated residues in protein structures in general: is it more likely that an instability-associated residue would be chosen at random than a binding-associated residue? With 'expected' frequencies of features known, the data represented in Figure 7.26 could be represented as log ratios (as in Figures 7.16(a) and 7.16(b)).

Indeed, there are many similar questions that have arisen from these investigations that, to be considered fully, require a more comprehensive understanding of the background or expected frequencies to be defined. For example, consider the frequency-of-explanation profile data described in Figures 7.29(a) and 7.29(b). It is not possible that the same mutation will be explained on the basis of introducing a hydrophobic residue on the surface *and* introducing a hydrophilic residue in the core; it is however likely that some mutations will be explained simultaneously by breaking a hydrogen bond *and* creating an internal void (see an example in Figure 7.8). Such inter-dependencies will not only allow the normalisation of data, but may also reveal important and interesting relationships that are fundamental to protein structure. For example, perhaps most hydrophobic residues that are introduced on the protein surface occur at the interface, which may improve inter-chain binding and therefore explain why the surfacephobic analysis

is the only structural analysis to be associated with SNPs (see Figure 7.26).

A recent collaboration concentrated on the analysis of mutations in the kinase domain (Izarzugaza *et al.*, 2009), comparing the explanation profiles of kinase PDs with all other PDs (non-kinase PDs) in SAAPdb, and comparing kinase PDs with kinase SNPs. This found that, unlike the vast majority of PDs analysed by SAAPdb previously (see Section 2.1.2), kinase PDs were significantly less likely to be attributable to some structural effect according to the SAAPdb analysis pipeline than non-kinase PDs, and that kinase PDs were no more likely to be attributable to some structural effect than kinase SNPs. It would seem that the current suite of structural analyses—the formation of which was motivated by a *general* understanding of protein structure—is failing to capture some important aspects of kinase structure, most importantly what differentiates kinase PDs from kinase SNPs. However, when using a set of kinase-specific features, and using a distance based analysis where *proximity* to features was measured rather than only considering the feature sites themselves, there was a clearer difference between kinase PDs and kinase SNPs.

It is possible that several of the kinase domain mutations interfere with the movement from the inactive to the active state and vice versa. Indeed, in a comprehensive study of oncogenic mutations in B-RAF by Wan *et al.* (Wan *et al.*, 2004), it was proposed that several oncogenic mutations destabilise the inactive state by disrupting the hydrophobic interactions between the P-loop and the DFG motif in the kinase domain, promoting the active conformation and thus mimicking the phosphorylated state.

8.2.2 Applying the findings to protein structure in general

Most (at the time of writing, November 2008, 85.50%) protein structures described by the PDB were derived via X-ray crystallography (see Section1.5); as such, most of the structures analysed by SAAPdb will also be X-ray structures. It is important to appreciate that protein crystals are merely 'snapshots' of protein structure. The manner in which PDB structures are derived—by enforced crystallisation—cannot capture the flexibility of protein structures. However, by considering the data and trends described in Chapter 7 as indirect measurements of the response to structural 'lesions', some comment on protein flexibility is possible: the extent to which protein structures can accommodate mutant residues, and what *kinds* (with respect to structural explanations) of mutant residues can be accommodated, will convey information regarding the

flexibility of the structures. For example, the analysis described in Chapter 7 demonstrated that PDs are more often found to affect protein stability than ligand binding sites. This may suggest that protein-ligand binding is more flexible, or can more easily accommodate mutations, than the network of scaffolding interactions that stabilise protein structure. Further, this trend may be owing to differences between obligate and transient interactions: perhaps transient binding is more flexible than obligate binding, which may be as constrained as the buried interior of a protein.

As discussed in Sections 7.4 and 8.2.1, such trends must be investigated further before attributing effects to variations in protein flexibility. Should the trends be verified, they could inform an alternative representation of protein structure to include hypothesised sites of flexibility, providing a more realistic model of protein structure with which to work.

8.2.3 Extending the pipeline

There are many potential structural effects of SAAPs that are currently not assessed by SAAPdb, as highlighted by the analysis of the kinase domain (see Section 8.2.1) where oncogenic mutations are known not only to destabilise the inactive form of B-RAF, but mimic the phosphorylated, active form of the protein (Wan *et al.*, 2004) thus disrupting native protein function. Data derived from other external resources (including the Catalytic Site Atlas (Porter *et al.*, 2004), PROCOGNATE (Bashton *et al.*, 2008) or dbPTM (Lee *et al.*, 2006)) could be incorporated to widen the focus of SAAPdb with respect to explaining mutations. It may also be beneficial to consider the protein in a wider context, for example its role in known pathways (Kanehisa *et al.*, 2008).

What is entirely missing from SAAPdb currently is the consideration of genomic data. The focus of SAAPdb is the manifestation and effects of genomic mutations at the protein level, primarily with respect to structure; however, there is undoubtedly more information implicit in the raw genomic data (Cargill *et al.*, 1999). For example, are PDs more often transversions (where a purine base (AG) is substituted with a pyrimidine base (CT) or vice versa) and therefore an alteration of the chemical nature of the base, and SNPs more often transitions (mutations between purine bases or between pyrimidine bases), where the chemical nature of the base does not change? Is there any bias in codons targeted by PDs or SNPs, or is there a bias in the particular position in the codon that is mutated? At the very least, estimates of base

change substitution rates, calculated from a basic understanding of biochemistry and mutagenesis mechanisms, could allow protein level data to be 'normalised' such that genomic effects are removed from analysis at the protein level (e.g., Care *et al.* (2007)). For example, arginine has a high rate of mutability (due to deamination of 5'-CpG dinucleotides in the arginine codon); such information could be used to normalise, for example, amino acid frequencies as shown in Figure 7.16 (where, indeed, arginine is one of the most commonly mutated residues).

Further, mutations may have effects in controlling expression or splicing. Such effects have been completely disregarded in this thesis.

8.2.4 Moving onto prediction

Despite the emergence of a biologically rational characterisation of disease mutations, consistent with existing literature, none of the features is strongly predictive on its own. Further, there appears to be a significant 'pathogenicity-differential' in the results of some analyses; that is, some analyses both commonly explain disease associated mutations *and* commonly explain neutral mutations, indicating that there is a differential in the extent to which the mutation affects protein structure. This is most true for the void-creating, hydrogen-bond-breaking, quaternary-structure-affecting and conserved-residue-affecting mutations.

This suggests that careful and considered application of machine learning techniques could exploit the weak predictive power of all of these individual features, resulting in a potentially very sensitive and accurate method for classifying previously unseen mutations as disease-causing or neutral. As mentioned in Section 8.2.1, inter-dependencies between features may exist and the method should be designed to take advantage of informative inter-dependencies while being able to factor out redundant data.

Furthermore, the investigations into kinase mutations described in Section 8.2.1 showed that protein families vary in their explanation profiles. In addition to calling for more research into the extent of such differences and how such protein families might be represented as training data to a predictive model, it is recommended that machine learning methods should at least initially build classifiers for each protein family. One method to predict specifically kinase mutations (Torkamani and Schork, 2008), achieves an MCC of 0.87, higher than any other 'general-purpose' prediction method. Further, several methods use the popular LacI (Suckow

et al., 1996), HIV protease (Loeb *et al.*, 1989) and/or T4 lysozyme (Rennell *et al.*, 1991) mutagenesis datasets, often reporting widely varying performance between these datasets (Ng and Henikoff, 2001; Chasman and Adams, 2001), particularly when the training and test sets are heterogeneous, that is, they are drawn from different mutagenesis datasets (Chasman and Adams, 2001; Krishnan and Westhead, 2003; Needham *et al.*, 2006).

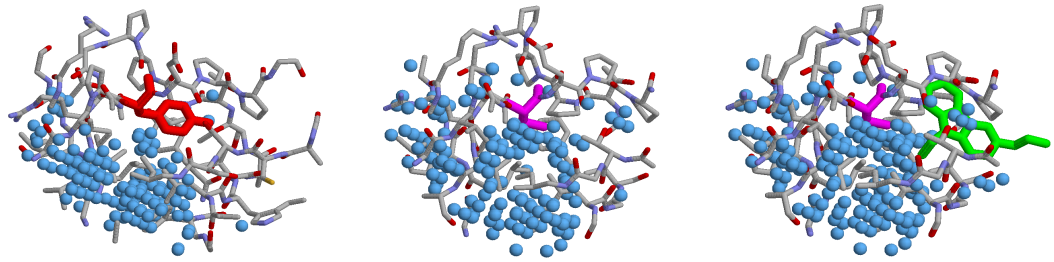
Preliminary work using several unoptimised machine learning methods (both with respect to the method parameters and with respect to the input vector) as provided by WEKA (Witten and Frank, 2005)—including several methods used previously in the literature (e.g., decision trees (Krishnan and Westhead, 2003), SVMs (Krishnan and Westhead, 2003; Tian *et al.*, 2007), 1R (Dobson *et al.*, 2006))—suggests that there is much potential in the current dataset for successful discrimination: default parameterisations of KNN[2], rule learner[3] and decision tree[4] algorithms all exceed the current 'gold-standard' prediction performance of Tian *et al.* (2007) of MCC=0.50. The results are summarised briefly in Appendix [A]. Active collaborations with Professor David Corne (Heriot-Watt University, Edinburgh), Professor Mark Girolami (University of Glasgow) and Professor Giuliano Armano (University of Cagliari) are pursuing further work in this area.

8.2.5 Implications for disease therapies

There is much potential for SAAPdb data to be used in the identification of novel drug targets. If one can characterise the specific reason that a mutated protein is not able to function properly, a counteractive rescue mechanism could be developed.

Recently, Boeckler *et al.* (2008) reported the development of an in-silico screened drug that was shown to rescue the function of a P53 mutant, Y220C. This mutant was known to destabilise the protein by introducing a crevice in the protein structure (compare Figure 8.2(a) with Figure 8.2(b)); incidentally, SAAPdb does successfully identify this mutation as void-creating. Boeckler *et al.*, by way of in-silico screening and multiple NMR spectroscopy experiments, identified a compound (PhiKan083) that bound to the destabilised mutant P53 structure, but not the native P53 structure, and is sufficiently distant from the DNA binding region not to interfere with functionality (see Figure 8.2(c)). Friedler *et al.* (2002) have shown that alternative pharma-

[2] `weka.classifiers.lazy.IBk`
[3] `weka.classifiers.rules.PART`
[4] `weka.classifiers.trees.J48`



(a) P53 native structure (b) P53/Y220C mutant structure (c) PhiKan083 bound to P53/Y220C

Figure 8.2: Stabilising a P53 mutant: a potential cancer therapeutic
The Y220C P53 mutant is the ninth most frequent p53 mutant that is implicated in cancer. An in-silico/NMR screening procedure identified the PhiKan083 compound that binds and stabilises the mutant form of P53 (Boeckler *et al.*, 2008). The relevant sections of the native and mutant P53 structures are shown in Figures 8.2(a) (1tsr, chain B) and 8.2(b) (2vuk, chain B) respectively. The native Y220 residue is highlighted in Figure 8.2(a) in green; the mutant C220 residue is highlighted in Figures 8.2(b) and 8.2(c) in red; the stabilising compound PhiKan083 is shown in Figure 8.2(c) in magenta; voids are represented as blue spheres. Voids are identified by AVP as described in Section 5.3.7.

ceuticals could bind to the functional *native* structure of P53, thus 'chaperoning' the correctly folded structure. Such compounds may form the basis of future P53-deficient cancer therapies, or indeed therapy for any disease caused by structurally-destabilising mutations.

It is therefore encouraging to note that most disease-associated mutations in SAAPdb have been shown to affect protein stability. There is potential for similar stabilising compounds to be identified for other destabilising protein mutations, thus rescuing native protein function and potentially treating disease.

8.3 Final thoughts

Large-scale automated analyses systems like SAAPdb are becoming more standard in bioinformatics: sequencing technologies are improving with respect to reliability, scale and speed, and high performance computational resources are becoming more affordable. The frequency with which disease mutation data are published is increasing (Ding *et al.*, 2008; Sreedharan *et al.*, 2008; Wang *et al.*, 2007a; Stevanin *et al.*, 2007; Mao *et al.*, 2007; The Wellcome Trust Case Control Consortium, 2007) are a few recent examples.

Such methods rely on external data sources maintaining the same data format; maintaining the public interface with the data, and enforcing rigourous standards in how the data are represented. Unfortunately, it is the exception rather than the norm that data formats and interfaces are maintained and that standards are enforced. To cope with such changes within a system like SAAPdb, time that could be dedicated to *analysing* the data is instead required to correct parsers and mirroring systems.

Bioinformatics is a young field, but can benefit from well-established conventions in computer science. Ideally, data representation systems (including UniProtKB, the PDB and dbSNP) should adhere to strict standards of formatting (e.g., valid XML or JSON) enabling fast and reliable extraction of data from *all* records without having to deal with rare exceptions. They should be backwards compatible; that is, parsers written for previous versions of the dataset should also be able to parse newer versions. Should the addition of attributes and values be absolutely necessary, extensible systems of data representation like XML and JSON allow new data to be included without breaking existing parsers, provided existing schemas are not violated. Should the format change, old representations should be deprecated, but gradually phased out rather than removed. Reformatting the UniProtKB description field as described above (see Figure 8.1) requires that *every* UniProtKB parser being used across the world be individually updated: although the change clearly facilitates the extraction of data from the description (DE) lines, any method that expects the data in the previous format will now fail to extract the appropriate data. Ideally the maintainers of such resources would also provide a parser and API to access the data. Should fundamental, defining features of a record change, some feature of the record identifier should also change: currently, if UniProtKB/Swiss-Prot changes the sequence of the protein, this is not expressed in the accession number; the user would have to compare sequence version numbers.

The fact that data formats do change on a regular basis demonstrates that the problem of representing these data is not conceptualised fully from the outset. More effort and discussion with regards to which data are relevant, how these data are related and what standards should be enforced is necessary before data are collected. Furthermore, biological resources must expect to be interrogated computationally and as such should facilitate data extraction. It is clear, then, that the best representation of data, both with respect to the biology *and* the informatics, can only be achieved by way of collaboration between biologists, computer scientists and bioinformaticians.

Bibliography

Abadie, A. (2002). Bootstrap tests for distributional treatment effects in instrumental variable models. *Journal of the American Statistical Association*, **97:457**, 284–292.

Aguirre, T., Matthijs, G., Robberecht, W., Tilkin, P. and Cassiman, J. J. (1999). Mutational analysis of the Cu/Zn superoxide dismutase gene in 23 familial and 69 sporadic cases of amyotrophic lateral sclerosis in Belgium. *European Journal of Human Genetics*, **7**, 599–602.

Aitkin, M. and Wilson, G. T. (1980). Mixture models, outliers, and the EM algorithm. *Technometrics*, **22**, 325–331.

Akindahunsi, A. A. and Chela-Flores, J., (2005). On The Question of Convergent Evolution in Biochemistry. In Seckbach, J., Chela-Flores, J., Owen, T. and Raulin, F. (eds.), *Life in the Universe: From the Miller Experiment to the Search for Life on Other Worlds*, page 135.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, **215**, 403–410.

Amberger, J., Bocchini, C. A., Scott, A. F. and Hamosh, A. (2009). McKusick's Online Mendelian Inheritance in Man (OMIM). *Nucleic Acids Research*, **37**, D793–D796.

Amores, A., Force, A., Yan, Y. L., Joly, L., Amemiya, C., Fritz, A., Ho, R. K., Langeland, J., Prince, V., Wang, Y. L., Westerfield, M., Ekker, M. and Postlethwait, J. H. (1998). Zebrafish HOX clusters and vertebrate genome evolution. *Science*, **282**, 1711–1714.

Artamonova, I. I., Frishman, G., Gelfand, M. S. and Frishman, D. (2005). Mining sequence annotation databanks for association patterns. *Bioinformatics*, **21 Suppl 3**, iii49–iii57.

Bader, G. D., Donaldson, I., Wolting, C., Ouellette, B. F., Pawson, T. and Hogue, C. W. (2001). BIND–The Biomolecular Interaction Network Database. *Nucleic Acids Research*, **29**, 242–245.

Bader, G. D., Betel, D. and Hogue, C. W. V. (2003). BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Research*, **31**, 248–250.

Baker, E. N. and Hubbard, R. E. (1984). Hydrogen bonding in globular proteins. *Progress in Biophysics and Molecular Biology*, **44**, 97–179.

Bao, L. and Cui, Y. (2005). Prediction of the phenotypic effects of non-synonymous single nucleotide polymorphisms using structural and evolutionary information. *Bioinformatics*, **21**, 2185–2190.

Barešić, A., Rogers, H., McMillan, L. E. M. and Martin, A. C. R. (2009). Compensated pathogenic deviations: analysis of structural effects. *In preparation*.

Bashton, M., Nobeli, I. and Thornton, J. M. (2008). PROCOGNATE: a cognate ligand domain mapping for enzymes. *Nucleic Acids Research*, **36**, D618–D622.

Bejerano, G., Pheasant, M., Makunin, I., Stephen, S., Kent, W. J., Mattick, J. S. and Haussler, D. (2004). Ultraconserved elements in the human genome. *Science*, **304**, 1321–1325.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J. and Wheeler, D. L. (2008). GenBank. *Nucleic Acids Research*, **36**, D25–D30.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. and Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, **28**, 235–242.

Beutler, E., Mathai, C. K. and Smith, J. E. (1968). Biochemical variants of glucose-6-phosphate dehydrogenase giving rise to congenital nonspherocytic hemolytic disease. *Blood*, **31**, 131–150.

Biémont, C. and Vieira, C. (2006). Genetics: junk DNA as an evolutionary force. *Nature*, **443**, 521–524.

Boeckler, F. M., Joerger, A. C., Jaggi, G., Rutherford, T. J., Veprintsev, D. B. and Fersht, A. R. (2008). Targeted rescue of a destabilized mutant of p53 by an in silico screened drug. *Proceedings of the National Academy of Sciences of the United States of America*, **105**, 10360–10365.

Branden, C. and Tooze, J., (1999). *Introduction to Protein Structure*. Garland, 2nd edition.

Brookes, A. J., Lehväslaiho, H., Siegfried, M., Boehm, J. G., Yuan, Y. P., Sarkar, C. M., Bork, P. and Ortigao, F. (2000). HGBASE: a database of SNPs and other variations in and around human genes. *Nucleic Acids Research*, **28**, 356–360.

Cai, Z., Tsung, E. F., Marinescu, V. D., Ramoni, M. F., Riva, A. and Kohane, I. S. (2004). Bayesian approach to discovering pathogenic SNPs in conserved protein domains. *Human Mutation*, **24**, 178–184.

Care, M. A., Needham, C. J., Bulpitt, A. J. and Westhead, D. R. (2007). Deleterious SNP prediction: be mindful of your training data! *Bioinformatics*, **23**, 664–672.

Cargill, M., Altshuler, D., Ireland, J., Sklar, P., Ardlie, K., Patil, N., Shaw, N., Lane, C. R., Lim, E. P., Kalyanaraman, N., Nemesh, J., Ziaugra, L., Friedland, L., Rolfe, A., Warrington, J., Lipshutz, R., Daley, G. Q. and Lander, E. S. (1999). Characterization of single-nucleotide polymorphisms in coding regions of human genes. *Nature Genetics*, **22**, 231–238.

Cavallo, A. and Martin, A. C. R. (2005). Mapping SNPs to protein sequence and structure data. *Bioinformatics*, **21**, 1443–1450.

Cerini, C., Kerjan, P., Astier, M., Gratecos, D., Mirande, M. and Sémériva, M. (1991). A component of the multisynthetase complex is a multifunctional aminoacyl-tRNA synthetase. *EMBO Journal*, **10**, 4267–4277.

Chasman, D. and Adams, R. M. (2001). Predicting the functional consequences of non-synonymous single nucleotide polymorphisms: Structure-based assessment of amino acid variation. *Journal of Molecular Biology*, **307**, 683–706.

Chen, F., Mackey, A. J., Stoeckert, C. J. and Roos, D. S. (2006). OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Research*, **34**, D363–D368.

Chothia, C. (1976). The nature of the accessible and buried surfaces in proteins. *Journal of Molecular Biology*, **105**, 1–12.

Clifford, R. J., Edmonson, M. N., Nguyen, C. and Buetow, K. H. (2004). Large-scale analysis of non-synonymous coding region single nucleotide polymorphisms. *Bioinformatics*, **20**, 1006–1014.

Collins, F. S., Brooks, L. D. and Chakravarti, A. (1998). A DNA polymorphism discovery resource for research on human genetic variation. *Genome Research*, **8**, 1229–1231.

Conover, W. J., (1971). *Practical Nonparametric Statistics*, pages 309–314. Wiley, New York.

Conrad, C. and Rauhut, R. (2002). Ribonuclease III: new sense from nuisance. **34**, 116–129.

Cornette, J. L., Cease, K. B., Margalit, H., Spouge, J. L., Berzofsky, J. A. and DeLisi, C. (1987). Hydrophobicity scales and computational techniques for detecting amphipathic structures in proteins. *Journal of Molecular Biology*, **195**, 659–685.

Couzin, J. (2002). Breakthrough of the year. Small RNAs make big splash. *Science*, **298**, 2296–2297.

Cripps, D., Thomas, S. N., Jeng, Y., Yang, F., Davies, P. and Yang, A. J. (2006). Alzheimer disease-specific conformation of hyperphosphorylated paired helical filament-Tau is polyubiquitinated

through Lys-48, Lys-11, and Lys-6 ubiquitin conjugation. *Journal of Biological Chemistry*, **281**, 10825–10838.

Cuff, A. L., Janes, R. W. and Martin, A. C. R. (2006). Analysing the ability to retain sidechain hydrogen-bonds in mutant proteins. *Bioinformatics*, **22**, 1464–1470.

Cuff, A. L. and Martin, A. C. R. (2004). Analysis of void volumes in proteins and application to stability of the p53 tumour suppressor protein. *Journal of Molecular Biology*, **344**, 1199–1209.

Dantzer, J., Moad, C., Heiland, R. and Mooney, S. (July 2005). MutDB services: interactive structural analysis of mutation data. **33**, W311–314.

Dayhoff, M. O., Schwartz, R. M. and Orcutt, B. C., (1978). *Atlas of Protein Sequence and Structure*, volume 5. National Biomedical Research Foundation.

Ding, L., Getz, G., Wheeler, D. A., Mardis, E. R., McLellan, M. D., Cibulskis, K., Sougnez, C., Greulich, H., Muzny, D. M., Morgan, M. B., Fulton, L., Fulton, R. S., Zhang, Q., Wendl, M. C., Lawrence, M. S., Larson, D. E., Chen, K., Dooling, D. J., Sabo, A., Hawes, A. C., Shen, H., Jhangiani, S. N., Lewis, L. R., Hall, O., Zhu, Y., Mathew, T., Ren, Y., Yao, J., Scherer, S. E., Clerc, K., Metcalf, G. A., Ng, B., Milosavljevic, A., Gonzalez-Garay, M. L., Osborne, J. R., Meyer, R., Shi, X., Tang, Y., Koboldt, D. C., Lin, L., Abbott, R., Miner, T. L., Pohl, C., Fewell, G., Haipek, C., Schmidt, H., Dunford-Shore, B. H., Kraja, A., Crosby, S. D., Sawyer, C. S., Vickery, T., Sander, S., Robinson, J., Winckler, W., Baldwin, J., Chirieac, L. R., Dutt, A., Fennell, T., Hanna, M., Johnson, B. E., Onofrio, R. C., Thomas, R. K., Tonon, G., Weir, B. A., Zhao, X., Ziaugra, L., Zody, M. C., Giordano, T., Orringer, M. B., Roth, J. A., Spitz, M. R., Wistuba, I. I., Ozenberger, B., Good, P. J., Chang, A. C., Beer, D. G., Watson, M. A., Ladanyi, M., Broderick, S., Yoshizawa, A., Travis, W. D., Pao, W., Province, M. A., Weinstock, G. M., Varmus, H. E., Gabriel, S. B., Lander, E. S., Gibbs, R. A., Meyerson, M. and Wilson, R. K. (2008). Somatic mutations affect key pathways in lung adenocarcinoma. *Nature*, **455**, 1069–1075.

Dobson, R. J., Munroe, P. B., Caulfield, M. J. and Saqi, M. A. (2006). Predicting deleterious nsSNPs: an analysis of sequence and structural attributes. *BMC Bioinformatics*, **7**, 217.

Duffy, D. L., Montgomery, G. W., Chen, W., Zhao, Z. Z., Le, L., James, M. R., Hayward, N. K., Martin, N. G. and Sturm, R. A. (2007). A three-single-nucleotide polymorphism haplotype in intron 1 of OCA2 explains most human eye-color variation. *American Journal of Human Genetics*, **80**, 241–252.

Durbin, R., Eddy, S., Krogh, A. and Mitchison, G., (1998). *Biological Sequence Analysis*. Cambridge University Press.

Edgar, R. C. (2004a). MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 113.

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

Feng, Z., Chen, L., Maddula, H., Akcan, O., Berman, H. M. and Westbrook, J., (July 2003). *ACA Program and Abstract Book*, volume 30 of *2nd series*. Northern Kentucky Convention Center. ISSN 0569-4221.

Ferlini, A., Obici, L., Manzati, E., Biadi, O., Tarantino, E., Conigli, P., Merlini, G., D'Alessandro, M., Mazzaferro, V., Tassinari, C. A. and Salvi, F. (2000). Mutation and transcription analysis of transthyretin gene in Italian families with hereditary amyloidosis: a putative novel hot spot in codon 47. *Clinical Genetics*, **57**, 284–290.

Ferrer-Costa, C., Orozco, M. and de la Cruz, X. (2004). Sequence-based prediction of pathological mutations. *Proteins*, **57**, 811–819.

Ferrer-Costa, C., Orozco, M. and de la Cruz, X. (2002). Characterization of disease-associated single amino acid polymorphisms in terms of sequence and structure properties. *Journal of Molecular Biology*, **315**, 771–786.

Fisher, R. A. (1935). The logic of inductive inference. *Journal of the Royal Statistical Society Series A*, **98**, 39–54.

Fitch, W. M. (2000). Homology a personal view on some of the problems. *Trends in Genetics*, **16**, 227–231.

Fitch, W. M. and Margoliash, E. (1967). Construction of phylogenetic trees. *Science*, **155**, 279–284.

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, **13**, 317–322.

Fredman, D., Siegfried, M., Yuan, Y. P., Bork, P., Lehväslaiho, H. and Brookes, A. J. (2002). HGVbase: a human sequence variation database emphasizing data quality and a broad spectrum of data sources. *Nucleic Acids Research*, **30**, 387–391.

Friedler, A., Hansson, L. O., Veprintsev, D. B., Freund, S. M. V., Rippin, T. M., Nikolova, P. V., Proctor, M. R., Rüdiger, S. and Fersht, A. R. (2002). A peptide that binds and stabilizes p53 core domain: Chaperone strategy for rescue of oncogenic mutants. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 937–942.

George, R. A., Smith, T. D., Callaghan, S., Hardman, L., Pierides, C., Horaitis, O., Wouters, M. A. and Cotton, R. G. H. (2008). General mutation databases: Analysis and review. *Journal of Medical Genetics*, **45**, 65–70.

Gilbert-Dussardier, B., Segues, B., Rozet, J. M., Rabier, D., Calvas, P., de Lumley, L., Bonnefond, J. P. and Munnich, A. (1996). Partial duplication [dup. TCAC (178)] and novel point mutations (T125M, G188R, A209V, and H302L) of the ornithine transcarbamylase gene in congenital hyperammonemia. *Human Mutation*, **8**, 74–76.

Greenblatt, M. S., Bennett, W. P., Hollstein, M. and Harris, C. C. (1994). Mutations in the p53 tumor suppressor gene: Clues to cancer etiology and molecular pathogenesis. *Cancer Research*, **54**, 4855–4878.

Greenman, C., Wooster, R., Futreal, P. A., Stratton, M. R. and Easton, D. F. (2006). Statistical analysis of pathogenicity of somatic mutations in cancer. *Genetics*, **173**, 2187–2198.

Gribskov, M., Lüthy, R. and Eisenberg, D. (1990). Profile analysis. *Methods in Enzymology*, **183**, 146–159.

Han, A., Kang, H. J., Cho, Y., Lee, S., Kim, Y. J. and Gong, S. (2006). SNP@Domain: a web resource of single nucleotide polymorphisms (SNPs) within protein domain structures and sequences. *Nucleic Acids Research*, **34**, W642–W644.

Hazes, B. and Dijkstra, B. W. (1988). Model building of disulfide bonds in proteins with known three-dimensional structure. *Protein Engineering*, **2**, 119–125.

Henikoff, J. G., Greene, E. A., Pietrokovski, S. and Henikoff, S. (2000). Increased coverage of protein families with the blocks database servers. *Nucleic Acids Research*, **28**, 228–230.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, **89**, 10915–10919.

Henikoff, S., Henikoff, J. G. and Pietrokovski, S. (1999). Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics*, **15**, 471–479.

Henrick, K. and Thornton, J. M. (1998). PQS: a protein quaternary structure file server. *Trends in Biochemical Sciences*, **23**, 358–361.

Hirosawa, M., Totoki, Y., Hoshida, M. and Ishikawa, M. (1995). Comprehensive study on iterative algorithms of multiple sequence alignment. *Computer Applications in the Biosciences*, **11**, 13–18.

Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., De Castro, E., Langendijk-Genevaux, P. S., Pagni, M. and Sigrist, C. J. A. (2006). The PROSITE database. *Nucleic Acids Research*, **34**, D227–D230.

Hulsen, T., (2004). Benchmarking ortholog identification methods using function similarity. Poster presented at ICS PhD Two-Day Conference. Available online at `http://www.cmbi.ru.nl/~timhulse/documents/orthology_040419.pdf`.

Hulsen, T., Huynen, M. A., de Vlieg, J. and Groenen, P. M. A. (2006). Benchmarking ortholog identification methods using functional genomics data. *Genome Biology*, **7**, R31.

Hurst, J. M., McMillan, L. E. M., Porter, C. T., Allen, J., Fakorede, A. and Martin, A. C. R. (2009). The SAAPdb web resource: a large scale structural analysis of mutant proteins. *Human Mutation*, **In press**.

International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, **431**, 931–945.

Izarzugaza, J. M. G., McMillan, L. E. M., Baresic, A., Valencia, A., Orengo, C. A. and Martin, A. C. R. (2009). Characterising pathogenic deviations in human protein kinases. *In preparation*.

Jabs, A., Weiss, M. S. and Hilgenfeld, R. (1999). Non-proline cis peptide bonds in proteins. *Journal of Molecular Biology*, **286**, 291–304.

Janin, J. (1997). Specific versus non-specific contacts in protein crystals. *Nature Structural Biology*, **4**, 973–974.

Jmoudiak, M. and Futerman, A. H. (2005). Gaucher disease: Pathological mechanisms and modern management. *British Journal of Haematology*, **129**, 178–188.

Jones, D. T., Taylor, W. R. and Thornton, J. M. (1992). The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences*, **8**, 275–282.

Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T. and Yamanishi, Y. (2008). KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, **36**, D480–D484.

Karchin, R., Diekhans, M., Kelly, L., Thomas, D. J., Pieper, U., Eswar, N., Haussler, D. and Sali, A. (2005). LS-SNP: large-scale annotation of coding non-synonymous SNPs based on multiple information sources. *Bioinformatics*, **21**, 2814–2820.

Khan, S. and Vihinen, M. (2007). Spectrum of disease-causing mutations in protein secondary structures. *BMC Structural Biology*, **7**, 56.

Kimura, M., (1983). *The neutral theory of molecular evolution*. Cambridge University Press, Cambridge.

Koonin, E. V. (2005). Orthologs, paralogs, and evolutionary genomics. *Annual Review of Genetics*, **39**, 309–338.

Koski, L. B. and Golding, G. B. (2001). The closest BLAST hit is often not the nearest neighbor. *Journal of Molecular Evolution*, **52**, 540–542.

Kretschmann, E., Fleischmann, W. and Apweiler, R. (2001). Automatic rule generation for protein annotation with the C4.5 data mining algorithm applied on SWISS-PROT. *Bioinformatics*, **17**, 920–926.

Krishnan, V. G. and Westhead, D. R. (2003). A comparative study of machine-learning methods to predict the effects of single nucleotide polymorphisms on protein function. *Bioinformatics*, **19**, 2199–2209.

Kulikova, T., Akhtar, R., Aldebert, P., Althorpe, N., Andersson, M., Baldwin, A., Bates, K., Bhattacharyya, S., Bower, L., Browne, P., Castro, M., Cochrane, G., Duggan, K., Eberhardt, R., Faruque, N., Hoad, G., Kanz, C., Lee, C., Leinonen, R., Lin, Q., Lombard, V., Lopez, R., Lorenc, D., McWilliam, H., Mukherjee, G., Nardone, F., Pastor, M. P. G., Plaister, S., Sobhany, S., Stoehr, P., Vaughan, R., Wu, D., Zhu, W. and Apweiler, R. (2007). EMBL Nucleotide Sequence Database in 2006. *Nucleic Acids Research*, **35**, D16–D20.

Kunin, V. and Ouzounis, C. A. (2005). Clustering the annotation space of proteins. *BMC Bioinformatics*, **6**, 24.

Kwok, C. J., Martin, A. C. R., Au, S. W. N. and Lam, V. M. S. (2002). G6PDdb, an integrated database of glucose-6-phosphate dehydrogenase (G6PD) mutations. *Human Mutation*, **19**, 217–224.

Kyte, J. and Doolittle, R. F. (1982). A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, **157**, 105–132.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., McKernan, K., Meldrim, J., Mesirov, J. P., Miranda, C., Morris, W., Naylor, J., Raymond, C., Rosetti, M., Santos, R., Sheridan, A., Sougnez, C.,

Stange-Thomann, N., Stojanovic, N., Subramanian, A., Wyman, D., Rogers, J., Sulston, J., Ainscough, R., Beck, S., Bentley, D., Burton, J., Clee, C., Carter, N., Coulson, A., Deadman, R., Deloukas, P., Dunham, A., Dunham, I., Durbin, R., French, L., Grafham, D., Gregory, S., Hubbard, T., Humphray, S., Hunt, A., Jones, M., Lloyd, C., McMurray, A., Matthews, L., Mercer, S., Milne, S., Mullikin, J. C., Mungall, A., Plumb, R., Ross, M., Shownkeen, R., Sims, S., Waterston, R. H., Wilson, R. K., Hillier, L. W., McPherson, J. D., Marra, M. A., Mardis, E. R., Fulton, L. A., Chinwalla, A. T., Pepin, K. H., Gish, W. R., Chissoe, S. L., Wendl, M. C., Delehaunty, K. D., Miner, T. L., Delehaunty, A., Kramer, J. B., Cook, L. L., Fulton, R. S., Johnson, D. L., Minx, P. J., Clifton, S. W., Hawkins, T., Branscomb, E., Predki, P., Richardson, P., Wenning, S., Slezak, T., Doggett, N., Cheng, J. F., Olsen, A., Lucas, S., Elkin, C., Uberbacher, E., Frazier, M., Gibbs, R. A., Muzny, D. M., Scherer, S. E., Bouck, J. B., Sodergren, E. J., Worley, K. C., Rives, C. M., Gorrell, J. H., Metzker, M. L., Naylor, S. L., Kucherlapati, R. S., Nelson, D. L., Weinstock, G. M., Sakaki, Y., Fujiyama, A., Hattori, M., Yada, T., Toyoda, A., Itoh, T., Kawagoe, C., Watanabe, H., Totoki, Y., Taylor, T., Weissenbach, J., Heilig, R., Saurin, W., Artiguenave, F., Brottier, P., Bruls, T., Pelletier, E., Robert, C., Wincker, P., Smith, D. R., Doucette-Stamm, L., Rubenfield, M., Weinstock, K., Lee, H. M., Dubois, J., Rosenthal, A., Platzer, M., Nyakatura, G., Taudien, S., Rump, A., Yang, H., Yu, J., Wang, J., Huang, G., Gu, J., Hood, L., Rowen, L., Madan, A., Qin, S., Davis, R. W., Federspiel, N. A., Abola, A. P., Proctor, M. J., Myers, R. M., Schmutz, J., Dickson, M., Grimwood, J., Cox, D. R., Olson, M. V., Kaul, R., Raymond, C., Shimizu, N., Kawasaki, K., Minoshima, S., Evans, G. A., Athanasiou, M., Schultz, R., Roe, B. A., Chen, F., Pan, H., Ramser, J., Lehrach, H., Reinhardt, R., McCombie, W. R., de la Bastide, M., Dedhia, N., Blöcker, H., Hornischer, K., Nordsiek, G., Agarwala, R., Aravind, L., Bailey, J. A., Bateman, A., Batzoglou, S., Birney, E., Bork, P., Brown, D. G., Burge, C. B., Cerutti, L., Chen, H. C., Church, D., Clamp, M., Copley, R. R., Doerks, T., Eddy, S. R., Eichler, E. E., Furey, T. S., Galagan, J., Gilbert, J. G., Harmon, C., Hayashizaki, Y., Haussler, D., Hermjakob, H., Hokamp, K., Jang, W., Johnson, L. S., Jones, T. A., Kasif, S., Kaspryzk, A., Kennedy, S., Kent, W. J., Kitts, P., Koonin, E. V., Korf, I., Kulp, D., Lancet, D., Lowe, T. M., McLysaght, A., Mikkelsen, T., Moran, J. V., Mulder, N., Pollara, V. J., Ponting, C. P., Schuler, G., Schultz, J., Slater, G., Smit, A. F., Stupka, E., Szustakowski, J., Thierry-Mieg, D., Thierry-Mieg, J., Wagner, L., Wallis, J., Wheeler, R., Williams, A., Wolf, Y. I., Wolfe, K. H., Yang, S. P., Yeh, R. F., Collins, F., Guyer, M. S., Peterson, J., Felsenfeld, A., Wetterstrand, K. A., Patrinos, A., Morgan, M. J., de Jong, P., Catanese, J. J., Osoegawa, K., Shizuya, H., Choi, S., Chen, Y. J. and Szustakowki, J. *et al*. (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921.

Lane, D. P. and Fischer, P. M. (2004). Turning the key on p53. *Nature*, **427**, 789–790.

Lee, B. and Richards, F. M. (1971). The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology*, **55**, 379–400.

Lee, T.-Y., Huang, H.-D., Hung, J.-H., Huang, H.-Y., Yang, Y.-S. and Wang, T.-H. (2006). dbPTM: an information repository of protein post-translational modification. *Nucleic Acids Research*, **34**, D622–D627.

Lee, Y., Sultana, R., Pertea, G., Cho, J., Karamycheva, S., Tsai, J., Parvizi, B., Cheung, F., Antonescu, V., White, J., Holt, I., Liang, F. and Quackenbush, J. (2002). Cross-referencing eukaryotic genomes: TIGR Orthologous Gene Alignments (TOGA). *Genome Research*, **12**, 493–502.

Li, S., Ma, L., Li, H., Vang, S., Hu, Y., Bolund, L. and Wang, J. (2007). Snap: an integrated SNP annotation platform. *Nucleic Acids Research*, **35**, D707–D710.

Lill, M. C., Fuller, J. F., Herzig, R., Crooks, G. M. and Gasson, J. C. (1995). The role of the homeobox gene, HOX B7, in human myelomonocytic differentiation. *Blood*, **85**, 692–697.

Loeb, D. D., Swanstrom, R., Everitt, L., Manchester, M., Stamper, S. E. and Hutchison, C. A. (1989). Complete mutagenesis of the HIV-1 protease. *Nature*, **340**, 397–400.

Lovell, S. C., Davis, I. W., Arendall, W. B., de Bakker, P. I. W., Word, J. M., Prisant, M. G., Richardson, J. S. and Richardson, D. C. (2003). Structure validation by Cα geometry: ϕ, ψ and Cβ deviation. *Proteins*, **50**, 437–450.

Lüthy, R., Xenarios, I. and Bucher, P. (1994). Improving the sensitivity of the sequence profile method. *Protein Science*, **3**, 139–146.

MacArthur, M. W. and Thornton, J. M. (1991). Influence of proline residues on protein conformation. *Journal of Molecular Biology*, **218**, 397–412.

Mao, G., Pan, X., Zhu, B.-B., Zhang, Y., Yuan, F., Huang, J., Lovell, M. A., Lee, M. P., Markesbery, W. R., Li, G.-M. and Gu, L. (2007). Identification and characterization of OGG1 mutations in patients with Alzheimer's disease. *Nucleic Acids Research*, **35**, 2759–2766.

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

Martin, A. C. R., Facchiano, A. M., Cuff, A. L., Hernandez-Boussard, T., Olivier, M., Hainaut, P. and Thornton, J. M. (2002). Integrating mutation data and structural analysis of the TP53 tumor-suppressor protein. *Human Mutation*, **19**, 149–164.

McDonald, I. K. and Thornton, J. M. (1994). Satisfying hydrogen bonding potential in proteins. *Journal of Molecular Biology*, **238**, 777–793.

McKusick, V., (December 1998). *Mendelian Inheritance in Man: A Catalog of Human Genes and Genetic Disorders*. Johns Hopkins University Press, Baltimore, 12th edition.

McMillan, L. E. M. and Martin, A. C. R. (2008). Automatically extracting functionally equivalent proteins from SwissProt. *BMC Bioinformatics*, **9**, 418.

Meyer, A. (1998). Hox gene variation and evolution. *Nature*, **391**, 225–228.

Mood, A., Graybill, F. A. and Boes, D. C., (1974). *Introduction to the Theory of Statistics*, pages 241–246. McGraw-Hill, 3rd edition edition.

Mulder, N. J., Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Buillard, V., Cerutti, L., Copley, R., Courcelle, E., Das, U., Daugherty, L., Dibley, M., Finn, R., Fleischmann, W., Gough, J., Haft, D., Hulo, N., Hunter, S., Kahn, D., Kanapin, A., Kejariwal, A., Labarga, A., Langendijk-Genevaux, P. S., Lonsdale, D., Lopez, R., Letunic, I., Madera, M., Maslen, J., McAnulla, C., McDowall, J., Mistry, J., Mitchell, A., Nikolskaya, A. N., Orchard, S., Orengo, C., Petryszak, R., Selengut, J. D., Sigrist, C. J. A., Thomas, P. D., Valentin, F., Wilson, D., Wu, C. H. and Yeats, C. (2007). New developments in the InterPro database. *Nucleic Acids Research*, **35**, D224–D228.

Needham, C. J., Bradford, J. R., Bulpitt, A. J., Care, M. A. and Westhead, D. R. (2006). Predicting the effect of missense mutations on protein function: Analysis with Bayesian networks. *BMC Bioinformatics*, **7**, 405.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, **48**, 443–453.

Ng, P. C. and Henikoff, S. (2001). Predicting deleterious amino acid substitutions. **11**, 863–74.

Notebaart, R. A., Huynen, M. A., Teusink, B., Siezen, R. J. and Snel, B. (2005). Correlation between sequence conservation and the genomic context after gene duplication. *Nucleic Acids Research*, **33**, 6164–6171.

O'Brien, K. P., Remm, M. and Sonnhammer, E. L. L. (2005). Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Research*, **33**, D476–D480.

Olivier, M., Eeles, R., Hollstein, M., Khan, M. A., Harris, C. C. and Hainaut, P. (2002). The IARC TP53 database: new online mutation analysis and recommendations to users. *Human Mutation*, **19**, 607–614.

Pearl, F. M. G., Bennett, C. F., Bray, J. E., Harrison, A. P., Martin, N., Shepherd, A., Sillitoe, I., Thornton, J. and Orengo, C. A. (2003). The CATH database: an extended protein family resource for structural and functional genomics. *Nucleic Acids Research*, **31**, 452–455.

Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America*, **85**, 2444–2448.

Petitjean, A., Mathe, E., Kato, S., Ishioka, C., Tavtigian, S. V., Hainaut, P. and Olivier, M. (2007). Impact of mutant p53 functional properties on TP53 mutation patterns and tumor phenotype: Lessons from recent developments in the IARC TP53 database. *Human Mutation*, **28**, 622–629.

Piirilä, H., Väliaho, J. and Vihinen, M. (2006). Immunodeficiency mutation databases (IDbases). *Human Mutation*, **27**, 1200–1208.

Porter, C. T., Bartlett, G. J. and Thornton, J. M. (2004). The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Research*, **32**, D129–D133.

Prabhakar, S., Noonan, J. P., Pääbo, S. and Rubin, E. M. (2006). Accelerated evolution of conserved noncoding sequences in humans. *Science*, **314**, 786.

Pupko, T., Bell, R. E., Mayrose, I., Glaser, F. and Ben-Tal, N. (2002). Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. *Bioinformatics*, **18 Suppl 1**, S71–S77.

Ramachandran, G. N., Ramakrishnan, C. and Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, **7**, 95–99.

Ranløv, I., Alves, I. L., Ranløv, P. J., Husby, G., Costa, P. P. and Saraiva, M. J. (1992). A Danish kindred with familial amyloid cardiomyopathy revisited: Identification of a mutant transthyretin-methionine111 variant in serum from patients and carriers. *American Journal of Medicine*, **93**, 3–8.

Rennell, D., Bouvier, S. E., Hardy, L. W. and Poteete, A. R. (1991). Systematic mutation of bacteriophage T4 lysozyme. *Journal of Molecular Biology*, **222**, 67–88.

Reumers, J., Maurer-Stroh, S., Schymkowitz, J. and Rousseau, F. (2006). SNPeffect v2.0: a new step in investigating the molecular phenotypic effects of human non-synonymous SNPs. *Bioinformatics*, **22**, 2183–2185.

Rice, S. B., Nenadic, G. and Stapley, B. J. (2005). Mining protein function from text using term-based support vector machines. *BMC Bioinformatics*, **6 Suppl 1**, S22.

Robinson, A. B. and Robinson, L. R. (1991). Distribution of glutamine and asparagine residues and their near neighbors in peptides and proteins. *Proceedings of the National Academy of Sciences of the United States of America*, **88**, 8880–8884.

Salama, J. J., Donaldson, I. and Hogue, C. W. (2001). Automatic annotation of BIND molecular interactions from three-dimensional structures. *Biopolymers*, **61**, 111–120.

Sandelin, A., Bailey, P., Bruce, S., Engström, P. G., Klos, J. M., Wasserman, W. W., Ericson, J. and Lenhard, B. (2004). Arrays of ultraconserved non-coding regions span the loci of key developmental genes in vertebrate genomes. *BMC Genomics*, **5**, 99.

Sander, C. and Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, **9**, 56–68.

Saunders, C. T. and Baker, D. (2002). Evaluation of structural and evolutionary contributions to deleterious mutation prediction. *Journal of Molecular Biology*, **322**, 891–901.

Sayle, R. A. and Milner-White, E. J. (1995). RASMOL: biomolecular graphics for all. *Trends in Biochemical Sciences*, **20**, 374–374.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, **27**, 379–423,623–656.

Sherry, S. T., Ward, M. and Sirotkin, K. (1999). dbSNP-database for single nucleotide polymorphisms and other classes of minor genetic variation. *Genome Research*, **9**, 677–679.

Shibata, S., Sasaki, M., Miki, T., Shimamoto, A., Furuichi, Y., Katahira, J. and Yoneda, Y. (2006). Exportin-5 orthologues are functionally divergent among species. *Nucleic Acids Research*, **34**, 4711–4721.

Shih, H. H., Brady, J. and Karplus, M. (1985). Structure of proteins with single-site mutations: a minimum perturbation approach. *Proceedings of the National Academy of Sciences of the United States of America*, **82**, 1697–1700.

Shimizu, T., Hozumi, K., Horiike, S., Nunomura, K., Ikegami, S., Takao, T. and Shimonishi, Y. (1996). A covalently crosslinked histone. *Nature*, **380**, 32.

Sidransky, D. and Hollstein, M. (1996). Clinical implications of the p53 gene. *Annual Review of Medicine*, **47**, 285–301.

Smigielski, E. M., Sirotkin, K., Ward, M. and Sherry, S. T. (2000). dbSNP: a database of single nucleotide polymorphisms. *Nucleic Acids Research*, **28**, 352–355.

Sneath, P. H. A. and Sokal, R. R., (1973). *Numerical taxonomy : the principles and practice of numerical classification*. Freeman, San Francisco.

Snow, M. E. and Amzel, L. M. (1986). Calculating three-dimensional changes in protein structure due to amino-acid substitutions: the variable region of immunoglobulins. *Proteins*, **1**, 267–279.

Sreedharan, J., Blair, I. P., Tripathi, V. B., Hu, X., Vance, C., Rogelj, B., Ackerley, S., Durnall, J. C., Williams, K. L., Buratti, E., Baralle, F., de Belleroche, J., Mitchell, J. D., Leigh, P. N., Al-Chalabi, A., Miller, C. C., Nicholson, G. and Shaw, C. E. (2008). TDP-43 mutations in familial and sporadic amyotrophic lateral sclerosis. *Science*, **319**, 1668–1672.

Stabler, S. P., Jones, R. T., Head, C., Shih, D. T. and Fairbanks, V. F. (1994). Hemoglobin Denver [alpha 2 beta 2(41) (C7) Phe→Ser]: a low-O2-affinity variant associated with chronic cyanosis and anemia. *Mayo Clinic Proceedings*, **69**, 237–243.

Stellwag, E. J. (1999). Hox gene duplication in fish. *Seminars in Cell & Developmental Biology*, **10**, 531–540.

Stenson, P. D., Ball, E. V., Mort, M., Phillips, A. D., Shiel, J. A., Thomas, N. S. T., Abeysinghe, S., Krawczak, M. and Cooper, D. N. (2003). Human Gene Mutation Database (HGMD): 2003 update. *Human Mutation*, **21**, 577–581.

Stevanin, G., Santorelli, F. M., Azzedine, H., Coutinho, P., Chomilier, J., Denora, P. S., Martin, E., Ouvrard-Hernandez, A.-M., Tessa, A., Bouslam, N., Lossos, A., Charles, P., Loureiro, J. L., Elleuch, N., Confavreux, C., Cruz, V. T., Ruberg, M., Leguern, E., Grid, D., Tazir, M., Fontaine, B., Filla, A., Bertini, E., Durr, A. and Brice, A. (2007). Mutations in SPG11, encoding spatacsin, are a major cause of spastic paraplegia with thin corpus callosum. *Nature Genetics*, **39**, 366–372.

Steward, R. E., MacArthur, M. W., Laskowski, R. A. and Thornton, J. M. (2003). Molecular basis of inherited diseases: a structural perspective. *Trends in Genetics*, **19**, 505–513.

Stewart, D. E., Sarkar, A. and Wampler, J. E. (1990). Occurrence and role of cis peptide bonds in protein structures. *Journal of Molecular Biology*, **214**, 253–260.

Stickle, D. F., Presta, L. G., Dill, K. A. and Rose, G. D. (1992). Hydrogen bonding in globular proteins. *Journal of Molecular Biology*, **226**, 1143–1159.

Stitziel, N. O., Binkowski, T. A., Tseng, Y. Y., Kasif, S. and Liang, J. (2004). topoSNP: a topographic database of non-synonymous single nucleotide polymorphisms with and without known disease association. *Nucleic Acids Research*, **32**, D520–D522.

Stitziel, N. O., Tseng, Y. Y., Pervouchine, D., Goddeau, D., Kasif, S. and Liang, J. (2003). Structural location of disease-associated single-nucleotide polymorphisms. *Journal of Molecular Biology*, **327**, 1021–1030.

Suckow, J., Markiewicz, P., Kleina, L. G., Miller, J., Kisters-Woike, B. and Müller-Hill, B. (1996). Genetic studies of the Lac repressor. XV: 4000 single amino acid substitutions and analysis of the resulting phenotypes on the basis of the protein structure. *Journal of Molecular Biology*, **261**, 509–523.

Sunyaev, S., Ramensky, V., Koch, I., Lathe, W., Kondrashov, A. S. and Bork, P. (2001). Prediction of deleterious human alleles. **10**, 591–597.

Taillon-Miller, P., Gu, Z., Li, Q., Hillier, L. and Kwok, P. Y. (1998). Overlapping genomic sequences: a treasure trove of single-nucleotide polymorphisms. *Genome Research*, **8**, 748–754.

Tatusov, R. L., Natale, D. A., Garkavtsev, I. V., Tatusova, T. A., Shankavaram, U. T., Rao, B. S., Kiryutin, B., Galperin, M. Y., Fedorova, N. D. and Koonin, E. V. (2001). The COG database: new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Research*, **29**, 22–28.

Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., Krylov, D. M., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Smirnov, S., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J. and Natale, D. A. (2003). The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.

Taylor, W. R. and Orengo, C. A. (1989). Protein structure alignment. *Journal of Molecular Biology*, **208**, 1–22.

The ENCODE Project Consortium (2007). Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, **447**, 799–816.

The International Hapmap Consortium (2005). A haplotype map of the human genome. *Nature*, **437**, 1299–1320.

The UniProt Consortium (2009). The Universal Protein Resource (UniProt) 2009. *Nucleic Acids Research*, **37**, D169–D174.

The Wellcome Trust Case Control Consortium (2007). Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, **447**, 661–678.

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

Thompson, J. D., Higgins, D. G. and Gibson, T. J. (1994b). Improved sensitivity of profile searches through the use of sequence weights and gap excision. *Computer Applications in the Biosciences*, **10**, 19–29.

Thorisson, G. A., Lancaster, O., Free, R. C., Hastings, R. K., Sarmah, P., Dash, D., Brahmachari, S. K. and Brookes, A. J. (2009). HGVbaseG2P: a central genetic association database. *Nucleic Acids Research*, **37**, D797–D802.

Tian, J., Wu, N., Guo, X., Guo, J., Zhang, J. and Fan, Y. (2007). Predicting the phenotypic effects of non-synonymous single nucleotide polymorphisms based on support vector machines. *BMC Bioinformatics*, **8**, 450.

Torkamani, A. and Schork, N. J. (2007). Distribution analysis of nonsynonymous polymorphisms within the human kinase gene family. *Genomics*, **90**, 49–58.

Torkamani, A. and Schork, N. J. (2008). Predicting functional regulatory polymorphisms. *Bioinformatics*, **24**, 1787–1792.

Torshin, I. Y. and Harrison, R. W. (2001). Charge centers and formation of the protein folding core. *Proteins*, **43**, 353–364.

Tuchman, M., Jaleel, N., Morizono, H., Sheehy, L. and Lynch, M. G. (2002). Mutations and polymorphisms in the human ornithine transcarbamylase gene. *Human Mutation*, **19**, 93–9107.

Uzun, A., Leslin, C. M., Abyzov, A. and Ilyin, V. (2007). Structure SNP (StSNP): a web server for mapping and modeling nsSNPs on protein structures with linkage to metabolic pathways. *Nucleic Acids Research*, **35**, W384–W392.

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

van Montfort, R. L. M., Congreve, M., Tisi, D., Carr, R. and Jhoti, H. (2003). Oxidation state of the active-site cysteine in protein tyrosine phosphatase 1B. *Nature*, **423**, 773–777.

van Noort, V., Snel, B. and Huynen, M. A. (2003). Predicting gene function by conserved co-expression. *Trends in Genetics*, **19**, 238–242.

Verzilli, C. J., Whittaker, J. C., Stallard, N. and Chasman, D. (2005). A hierarchical Baysian model for predicting the functional consequences of amino-acid polymorphisms. *Applied Statistics*, **54(1)**, 191–206.

Vitkup, D., Sander, C. and Church, G. M. (2003). The amino-acid mutational spectrum of human genetic disease. *Genome Biology*, **4**, R72.

Wagner, A. (2002). Asymmetric functional divergence of duplicate genes in yeast. *Molecular Biology and Evolution*, **19**, 1760–1768.

Wan, P. T. C., Garnett, M. J., Roe, S. M., Lee, S., Niculescu-Duvaz, D., Good, V. M., Jones, C. M., Marshall, C. J., Springer, C. J., Barford, D., Marais, R. and Project, T. C. G. (2004). Mechanism of activation of the RAF-ERK signaling pathway by oncogenic mutations of B-RAF. *Cell*, **116**, 855–867.

Wang, P., Dai, M., Xuan, W., McEachin, R. C., Jackson, A. U., Scott, L. J., Athey, B., Watson, S. J. and Meng, F. (2006). SNP Function Portal: a web database for exploring the function implication of SNP alleles. *Bioinformatics*, **22**, e523–e529.

Wang, X., Reid Sutton, V., Omar Peraza-Llanes, J., Yu, Z., Rosetta, R., Kou, Y.-C., Eble, T. N., Patel, A., Thaller, C., Fang, P. and Van den Veyver, I. B. (2007a). Mutations in X-linked PORCN, a putative regulator of Wnt signaling, cause focal dermal hypoplasia. *Nature Genetics*, **39**, 836–838.

Wang, Y., Addess, K. J., Chen, J., Geer, L. Y., He, J., He, S., Lu, S., Madej, T., Marchler-Bauer, A., Thiessen, P. A., Zhang, N. and Bryant, S. H. (2007b). MMDB: annotating protein sequences with Entrez's 3D-structure database. *Nucleic Acids Research*, **35**, D298–D300.

Wang, Z. and Moult, J. (2001). SNPs, protein structure, and disease. *Human Mutation*, **17**, 263–270.

Weiss, M. S., Jabs, A. and Hilgenfeld, R. (1998). Peptide bonds revisited. *Nature Structural Biology*, **5**, 676.

Westbrook, J., Ito, N., Nakamura, H., Henrick, K. and Berman, H. M. (2005). PDBML: the representation of archival macromolecular structure data in XML. *Bioinformatics*, **21**, 988–992.

Witten, I. H. and Frank, E., (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition.

Wu, C. H., Nikolskaya, A., Huang, H., Yeh, L.-S. L., Natale, D. A., Vinayaka, C. R., Hu, Z.-Z., Mazumder, R., Kumar, S., Kourtesis, P., Ledley, R. S., Suzek, B. E., Arminski, L., Chen, Y., Zhang, J., Cardenas, J. L., Chung, S., Castro-Alvear, J., Dinkov, G. and Barker, W. C. (2004). PIRSF: family classification system at the Protein Information Resource. *Nucleic Acids Research*, **32**, D112–D114.

Wu, H., Xu, H., Miraglia, L. J. and Crooke, S. T. (2000). Human RNase III is a 160-kDa protein involved in preribosomal RNA processing. *Journal of Biological Chemistry*, **275**, 36957–36965.

Yaron, Y., McAdara, J. K., Lynch, M., Hughes, E. and Gasson, J. C. (2001). Identification of novel functional regions important for the activity of HOXB7 in mammalian cells. *Journal of Immunology*, **166**, 5058–5067.

Yu, G.-X. (2004). Ruleminer: a knowledge system for supporting high-throughput protein function annotations. *Journal of Bioinformatics and Computational Biology*, **2**, 615–637.

Yue, P., Li, Z. and Moult, J. (2005). Loss of protein structure stability as a major causative factor in monogenic disease. *Journal of Molecular Biology*, **353**, 459–473.

Yunger, L. M. and Cramer, R. D. (1981). Measurement of correlation of partition coefficients of polar amino acids. *Molecular Pharmacology*, **20**, 602–608.

Appendices

[A] Preliminary predictive work

As described in Chapter 8, some very preliminary, exploratory predictive work has been carried out on the SAAPdb data using several unoptimised methods in Weka (Witten and Frank, 2005). Results are shown in the figure below. Default parameterisations of KNN (`weka.classifiers.lazy.IBk`), rule learner (`weka.classifiers.rules.PART`) and decision tree (`weka.classifiers.trees.J48`) algorithms all exceed the current 'gold-standard' prediction performance of Tian *et al.* (2007) of MCC=0.50.

Several issues with regards to the predictive methods are unresolved. Quite apart from the optimial parameterisation of the individual methods, the issues of dataset sampling and feature vector contents require to be considered.

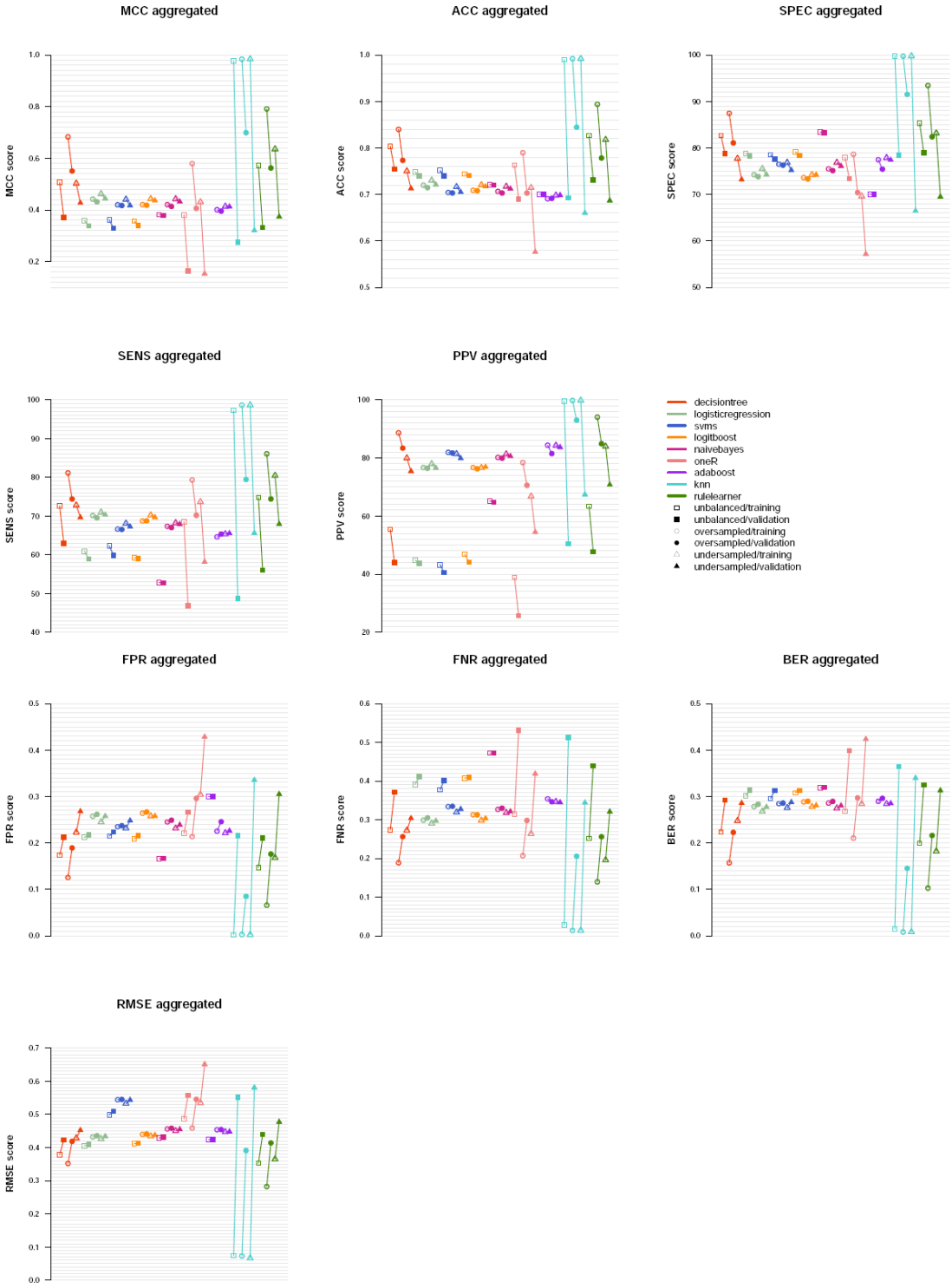
Sampling methods (discussed in Dobson *et al.* (2006)) define how the datasets should be balanced, if at all. In SAAPdb, there are many more PDs mapped to structure than SNPs; oversampling would repeat examples in the SNP dataset so as to balance the size of the datasets, while undersampling would reduce the size of the PD datset to that of the SNP dataset. The results shown in the graphs below suggest that oversampling is more successful than undersampled or unbalanced datasets; however, this requires more investigation and may simply be due to biasing the datasets. Alternatively, some methods may not be sensitive to dataset size and consideration of sampling may not be required.

The feature set used to generate these results was reasonably rudimentary, containing only a binary vector with the results of fourteen analyses (the UniProtKB/Swiss-Prot feature analysis was removed as it was found to be unreliable, see Section 5.11); the relative accessibility, and the native and mutant amino acids, represented as strings. It does not include numerical representation of the amino acids (for example, the BLOSUM/PAM amino acid substitution matrix score, both of which were shown to be statistically significant when comparing the PD and SNP datasets, see Section 7.3.3), nor does it include the apparently powerful 'average dissimilarity' score (Section 7.3.2) for the native and mutant residues.

As seen in Section 7.2.1 some proteins are described by the PDB more than once. It follows that some sequence mutations will be analysed more than once in the analysis pipeline. It is yet to be decided how such multiple results are to be combined to represent the one mutation. It is

undesirable to present each analysis to the predictive method at the training stage, as this will bias the predictor. The results in the figure below describe 'aggregated' counts, where all structural analysis results were aggregated into one vector, where a positive result was assigned to the sequence mutation if *any* of the mapped structures generated a positive result. Also considered was a 'hybrid' counting system, where the sequence mutation was assigned a positive result for an analysis if at least half of the mapped structures generated a positive result. Results were very slightly better for the aggregated results. Most desirable would be to use a machine learning method that can make use of this additional information, without biasing the predictor towards those data that are mapped to multiple structures.

Although very rudimentary, it is clear that significant predictive power lies in these data. With the proper consideration of machine learning approaches and the appropriate choice of feature vector, it should be possible to improve the current gold standard prediction performance.



Results of some very preliminary predictive work Chapter 8 briefly described some very preliminary predictive carried out on the data from SAAPdb; these are the results. Empty symbols describe training errors, closed symbols describe validation errors; paired training and validation errors are joined by a line. Different symbols indicate various data sampling approaches: the unbalanced approach is indicated with a square; the oversampling approach is indicated with a circle; the undersampling approach is indicated with a triangle. Each graph describes a different performance

statistic (from top left to bottom right: Matthew's Correlation Coefficient (MCC); accuracy (ACC); specificity (SPEC); sensitivity (SENS); positive predictive value (PPV); false positive rate (FPR); false negative rate (FNR); balanced error rate (BER); root mean squared error (RMSE). All methods are plotted on the same graph (legend given above).

[B] Biology

[B.i] Amino acid colours (taken from RASMOL 'amino' colour scheme)

These colours are used primarily in Chapter 4 to colour the multiple sequence alignments. They are taken from the Rasmol (Sayle and Milner-White, 1995) 'amino' colour scheme, which colours residues according to traditional amino acid properties (e.g., hydrophobicity and charge).

[C] Amino acid subsitution matrices

[C.i] PAM30

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	J	Z	X	*
A	6	-7	-4	-3	-6	-4	-2	-2	-7	-5	-6	-7	-5	-8	-2	0	-1	-13	-8	-2	-3	-6	-3	-1	-17
R	-7	8	-6	-10	-8	-2	-9	-9	-2	-5	-8	0	-4	-9	-4	-3	-6	-2	-10	-8	-7	-7	-4	-1	-17
N	-4	-6	8	2	-11	-3	-2	-3	0	-5	-7	-1	-9	-9	-6	0	-2	-8	-4	-8	6	-6	-3	-1	-17
D	-3	-10	2	8	-14	-2	2	-3	-4	-7	-12	-4	-11	-15	-8	-4	-5	-15	-11	-8	6	-10	1	-1	-17
C	-6	-8	-11	-14	10	-14	-14	-9	-7	-6	-15	-14	-13	-13	-8	-3	-8	-15	-4	-6	-12	-9	-14	-1	-17
Q	-4	-2	-3	-2	-14	8	1	-7	1	-8	-5	-3	-4	-13	-3	-5	-5	-13	-12	-7	-3	-5	6	-1	-17
E	-2	-9	-2	2	-14	1	8	-4	-5	-5	-9	-4	-7	-14	-5	-4	-6	-17	-8	-6	1	-7	6	-1	-17
G	-2	-9	-3	-3	-9	-7	-4	6	-9	-11	-10	-7	-8	-9	-6	-2	-6	-15	-14	-5	-3	-10	-5	-1	-17
H	-7	-2	0	-4	-7	1	-5	-9	9	-9	-6	-6	-10	-6	-4	-6	-7	-7	-3	-6	-1	-7	-1	-1	-17
I	-5	-5	-5	-7	-6	-8	-5	-11	-9	8	-1	-6	-1	-2	-8	-7	-2	-14	-6	2	-6	5	-6	-1	-17
L	-6	-8	-7	-12	-15	-5	-9	-10	-6	-1	7	-8	1	-3	-7	-8	-7	-6	-7	-2	-9	6	-7	-1	-17
K	-7	0	-1	-4	-14	-3	-4	-7	-6	-6	-8	7	-2	-14	-6	-4	-3	-12	-9	-9	-2	-7	-4	-1	-17
M	-5	-4	-9	-11	-13	-4	-7	-8	-10	-1	1	-2	11	-4	-8	-5	-4	-13	-11	-1	-10	0	-5	-1	-17
F	-8	-9	-9	-15	-13	-13	-14	-9	-6	-2	-3	-14	-4	9	-10	-6	-9	-4	2	-8	-10	-2	-13	-1	-17
P	-2	-4	-6	-8	-8	-3	-5	-6	-4	-8	-7	-6	-8	-10	8	-2	-4	-14	-13	-6	-7	-7	-4	-1	-17
S	0	-3	0	-4	-3	-5	-4	-2	-6	-7	-8	-4	-5	-6	-2	6	0	-5	-7	-6	-1	-8	-5	-1	-17
T	-1	-6	-2	-5	-8	-5	-6	-6	-7	-2	-7	-3	-4	-9	-4	0	7	-13	-6	-3	-3	-5	-6	-1	-17
W	-13	-2	-8	-15	-15	-13	-17	-15	-7	-14	-6	-12	-13	-4	-14	-5	-13	13	-5	-15	-10	-7	-14	-1	-17
Y	-8	-10	-4	-11	-4	-12	-8	-14	-3	-6	-7	-9	-11	2	-13	-7	-6	-5	10	-7	-6	-7	-9	-1	-17
V	-2	-8	-8	-8	-6	-7	-6	-5	-6	2	-2	-9	-1	-8	-6	-6	-3	-15	-7	7	-8	0	-6	-1	-17
B	-3	-7	6	6	-12	-3	1	-3	-1	-6	-9	-2	-10	-10	-7	-1	-3	-10	-6	-8	6	-8	0	-1	-17
J	-6	-7	-6	-10	-9	-5	-7	-10	-7	5	6	-7	0	-2	-7	-8	-5	-7	-7	0	-8	6	-6	-1	-17
Z	-3	-4	-3	1	-14	6	6	-5	-1	-6	-7	-4	-5	-13	-4	-5	-6	-14	-9	-6	0	-6	6	-1	-17
X	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-17
*	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	-17	1

[C.ii] PET91

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	10	-1	0	-1	-1	-1	-1	1	-2	0	-1	-1	-1	-3	1	1	2	-4	-3	1	0	-1	0	0
R	-1	10	0	-1	-1	2	0	0	2	-3	-3	4	-2	-4	-1	-1	-1	0	-2	-3	0	1	0	0
N	0	0	10	2	-1	0	1	0	1	-2	-3	1	-2	-3	-1	1	1	-4	-1	-2	2	0	0	0
D	-1	-1	2	10	-3	0	4	1	0	-3	-4	0	-3	-5	-2	0	-1	-5	-2	-3	3	2	0	0
C	-1	-1	-1	-3	10	-3	-4	-1	0	-2	-3	-3	-2	0	-2	1	-1	1	2	-2	-2	-3	0	0
Q	-1	2	0	0	-3	10	2	-1	3	-3	-2	2	-2	-4	0	-1	-1	-3	-1	-3	0	3	0	0
E	-1	0	1	4	-4	2	10	1	0	-3	-4	1	-3	-5	-2	-1	-1	-5	-4	-2	2	3	0	0
G	1	0	0	1	-1	-1	1	10	-2	-3	-4	-1	-3	-5	-1	1	0	-2	-4	-2	0	0	0	0
H	-2	2	1	0	0	3	0	-2	10	-3	-2	1	-2	0	0	-1	-1	-3	4	-3	0	1	0	0
J	0	-3	-2	-3	-2	-3	-3	-3	-3	10	2	-3	3	0	-2	-1	1	-4	-2	4	-2	-3	0	0
L	-1	-3	-3	-4	-3	-2	-4	-4	-2	2	10	-3	3	2	0	-2	-1	-2	-1	2	-3	-3	0	0
K	-1	4	1	0	-3	2	1	-1	1	-3	-3	10	-2	-5	-2	-1	-1	-3	-3	-3	0	1	0	0

M	-1	-2	-2	-3	-2	-2	-3	-3	-2	3	3	-2	10	0	-2	-1	0	-3	-3	2	-2	-2	0	0
F	-3	-4	-3	-5	0	-4	-5	-5	0	0	2	-5	0	10	-2	-2	-2	-1	5	0	-4	-4	0	0
P	1	-1	-1	-2	-2	0	-2	-1	0	-2	0	-2	-2	-2	10	1	1	-5	-3	-1	-1	-1	0	0
S	1	-1	1	0	1	-1	-1	1	-1	-1	-2	-1	-1	-2	1	10	1	-3	-1	-1	0	-1	0	0
T	2	-1	1	-1	-1	-1	-1	0	-1	1	-1	-1	0	-2	1	1	10	-4	-3	0	0	-1	0	0
W	-4	0	-4	-5	1	-3	-5	-2	-3	-4	-2	-3	-3	-1	-5	-3	-4	10	0	-4	-4	-4	0	0
Y	-3	-2	-1	-2	2	-1	-4	-4	4	-2	-1	-3	-3	5	-3	-1	-3	0	10	-3	-1	-2	0	0
V	1	-3	-2	-3	-2	-3	-2	-2	-3	4	2	-3	2	0	-1	-1	0	-4	-3	10	-2	-2	0	0
B	0	0	2	3	-2	0	2	0	0	-2	-3	0	-2	-4	-1	0	0	-4	-1	-2	10	1	0	0
Z	-1	1	0	2	-3	3	3	0	1	-3	-3	1	-2	-4	-1	-1	-1	-4	-2	-2	1	10	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

[C.iii] BLOSUM62

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	6	-2	-2	-3	-1	-1	-1	0	-2	-2	-2	-1	-1	-3	-1	2	0	-4	-3	0	-2	-1	-1	-6
R	-2	8	-1	-2	-5	1	0	-3	0	-4	-3	3	-2	-4	-3	-1	-2	-4	-3	-4	-2	0	-2	-6
N	-2	-1	8	2	-4	0	0	-1	1	-5	-5	0	-3	-4	-3	1	0	-6	-3	-4	5	0	-2	-6
D	-3	-2	2	9	-5	0	2	-2	-2	-5	-5	-1	-5	-5	-2	0	-2	-6	-5	-5	6	1	-2	-6
C	-1	-5	-4	-5	13	-4	-5	-4	-4	-2	-2	-5	-2	-4	-4	-1	-1	-3	-4	-1	-5	-5	-3	-6
Q	-1	1	0	0	-4	8	3	-3	1	-4	-3	2	-1	-5	-2	0	-1	-3	-2	-3	0	5	-1	-6
E	-1	0	0	2	-5	3	7	-3	0	-5	-4	1	-3	-5	-2	0	-1	-4	-3	-4	1	6	-1	-6
G	0	-3	-1	-2	-4	-3	-3	8	-3	-6	-5	-2	-4	-5	-3	0	-2	-4	-5	-5	-1	-3	-2	-6
H	-2	0	1	-2	-4	1	0	-3	11	-5	-4	-1	-2	-2	-3	-1	-3	-4	3	-5	-1	0	-2	-6
I	-2	-4	-5	-5	-2	-4	-5	-6	-5	6	2	-4	2	0	-4	-4	-1	-4	-2	4	-5	-5	-2	-6
L	-2	-3	-5	-5	-2	-3	-4	-5	-4	2	6	-4	3	1	-4	-4	-2	-2	-2	1	-5	-4	-2	-6
K	-1	3	0	-1	-5	2	1	-2	-1	-4	-4	7	-2	-5	-2	0	-1	-4	-3	-3	-1	1	-1	-6
M	-1	-2	-3	-5	-2	-1	-3	-4	-2	2	3	-2	8	0	-4	-2	-1	-2	-1	1	-4	-2	-1	-6
F	-3	-4	-4	-5	-4	-5	-5	-5	-2	0	1	-5	0	9	-5	-4	-3	1	4	-1	-5	-5	-2	-6
P	-1	-3	-3	-2	-4	-2	-2	-3	-3	-4	-4	-2	-4	-5	11	-1	-2	-5	-4	-4	-3	-2	-2	-6
S	2	-1	1	0	-1	0	0	0	-1	-4	-4	0	-2	-4	-1	6	2	-4	-3	-2	0	0	-1	-6
T	0	-2	0	-2	-1	-1	-1	-2	-3	-1	-2	-1	-1	-3	-2	2	7	-4	-2	0	-1	-1	-1	-6
W	-4	-4	-6	-6	-3	-3	-4	-4	-4	-4	-2	-4	-2	1	-5	-4	-4	16	3	-4	-6	-4	-3	-6
Y	-3	-3	-3	-5	-4	-2	-3	-5	3	-2	-2	-3	-1	4	-4	-3	-2	3	10	-2	-4	-3	-2	-6
V	0	-4	-4	-5	-1	-3	-4	-5	-5	4	1	-3	1	-1	-4	-2	0	-4	-2	6	-5	-4	-1	-6
B	-2	-2	5	6	-5	0	1	-1	-1	-5	-5	-1	-4	-5	-3	0	-1	-6	-4	-5	5	0	-2	-6
Z	-1	0	0	1	-5	5	6	-3	0	-5	-4	1	-2	-5	-2	0	-1	-4	-3	-4	0	5	-1	-6
X	-1	-2	-2	-2	-3	-1	-1	-2	-2	-2	-2	-1	-1	-2	-2	-1	-1	-3	-2	-1	-2	-1	-2	-6
*	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	1

[D] Database queries

[D.i] Generating a list of species pairings from FOSTA

```
SELECT f1.species, f2.species
FROM feps f1, f2
WHERE f1.fosta_family = f2.fosta_family
AND f1.runid = 1
AND f2.runid = 1
AND f1.species != f2.species;
```

[D.ii] Finding the number of proteins in FOSTA for each species

```
SELECT f.species, count(f.*)
FROM feps f
GROUP BY f.species;
```

[D.iii] Finding the FEPs common to both `$speciesA` and `$speciesB`

```
SELECT f1.id, f2.id
FROM  feps f1, feps f2
WHERE f1.id != f2.id
AND f1.fosta_family = f2.fosta_family
AND f1.species = '$speciesA'
AND f2.species = '$speciesB'
AND f1.runid = 1
AND f2.runid = 1;
```

[E] SQL functions

[E.i] Calculating the 'charge shift' of a mutation

```
CREATE OR REPLACE FUNCTION charge_shift(integer) RETURNS int AS '
  DECLARE
    mutanalysisRowID ALIAS FOR $1;
    nativeRes varchar;
    nativeCharge int;
    nativeStatus varchar;
    mutantRes varchar;
    mutantCharge int;
    mutantStatus varchar;
  BEGIN
    SELECT INTO nativeRes,mutantRes m.aa_wildtype,m.aa_mutant
    FROM mutanalysis m
    WHERE m.mutanalysis_row_id=mutanalysisRowID;

    SELECT INTO nativeStatus a.charged
    FROM amino_acids a
    WHERE a.threelettercode=nativeRes;

    nativeCharge := 0;
    IF      nativeStatus = ''positive'' THEN nativeCharge :=  1 ;
    ELSIF   nativeStatus = ''negative'' THEN nativeCharge := -1 ;
    END IF;

    SELECT INTO mutantStatus a.charged
    FROM amino_acids a
    WHERE a.singlelettercode=lower(mutantRes);

    mutantCharge := 0;
    IF      mutantStatus = ''positive'' THEN mutantCharge :=  1 ;
    ELSIF   mutantStatus = ''negative'' THEN mutantCharge := -1 ;
    END IF;

    RETURN ( mutantCharge - nativeCharge );
  END
' LANGUAGE plpgsql;
```