UNIVERSITY COLLEGE LONDON

UCL Research Department of Structural and Molecular Biology

Analysis of the impact of mutations and prediction of their pathogenicity

Nouf S. Al-Numair

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

Declaration

I, Nouf Al-Numair, 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.

Nouf S. Al-Numair

August, 2014

Abstract

Inherited diseases and cancer are often characterized by single DNA base mutations that can result in altered gene expression, altered mRNA splicing, or changes to the protein structure. The effects of the latter category on protein function and how this is related to disease is the easiest of these to understand. Pathogenic deviations (PDs) are mutations reported to be disease-causing, while true single nucleotide polymorphisms (SNPs) are understood to have a negligible effect on phenotype. With recent developments in biotechnology, the most relevant being the 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 builds on earlier work in the Martin group in which a database (SAAPdb) was developed to map mutation data to protein structure and allow the likely local protein structural effects of a mutation to be evaluated.

In this thesis, a general introduction to the relevant biology (Chapter 1) and bioinformatics tools and resources (Chapter 2) is provided. In Chapter 3, the Single Amino Acid Polymorphism database (SAAPdb) is described and the work done to fix bugs and update the data is outlined. Despite this work, owing to continuous maintenance problems identified when updating the program, the Martin group has now switched to using a 'pipeline' version that no longer relies on any pre-calculated data stored in a database.

Earlier work performed during a Masters project showed that some of the analyses were extremely sensitive to structural details. These analyses have been updated and extended, confirming earlier results. Consequently, some of the analyses were updated to replace Boolean True/False (Good/Bad) assignments with energy or pseudo-energy values. A pseudo-energy potential was developed for evaluating the effects of mutations to-proline or from-glycine (Chapter 4) and a new full-energy method for assessing the effects of side-

chain clashes was evaluated (Chapter 5). A method using the structural analyses data together with random forests to predict whether a mutation will be damaging was then developed (Chapter 6). This method was demonstrated to be better than all competing individual methods. A variation of this approach was used to distinguish between two phenotypes (hypertrophic cardiomyopathy – HCM, and dilated cardiomyopathy – DCM) caused by mutations in the cardiac beta-myosin gene (MYH7, Chapter 7). The thesis finishes with a general discussion and conclusions (Chapter 8).

The final SAAPpred predictor using the updated SAAPdap and the improved analysis outperforms competing methods (for mutations where a structure is available) giving an accuracy of 0.885 and MCC = 0.73 showing that a detailed analysis of structural features is beneficial in predicting the effect of any novel mutation. SAAPpred performed very well when discriminating between pathogenic and neutral SNPs in MYH7 having an accuracy of 0.794 – 0.927 using one PDB structures per mutation and multiple PDB structures respectively. This was followed by creation of a novel predictor which attempts to distinguish between HCM and DCM mutations using SAAP analysis, exploiting feature selection and an additional set of features on structural clustering. This is the first prediction of detailed phenotype and works surprisingly well giving an accuracy of 0.75 and MCC = 0.531.

Abbreviations

AC	:	Accession Code
ASA	:	Accessible Surface Area
ASU	:	ASsymetric Unit
AUC	:	Area Under Curve
BLAST	:	Basic Local Alignment Search Tool
BU	:	Biological Unit
CAGI	:	Critical Assessment of Genome Interpretation
CPD	:	Compensated Pathogenic Deviation
cSNP	:	Coding Single Nucleotide Polymorphism
DAM	:	Disease-Asoociated Mutation
DM	:	Deleterious Mutation
DNA	:	DeoxyriboNucleic Acid
FEP	:	Functionally Equivalent Protein
FN	:	False Negative
FOSTA	:	Functional Orthologues from Swissprot Text Analysis
FP	:	False Positive
FPR	:	False Positive Rate
lpSAAP	:	Low-Penetrance Single Amino Acid Polymorphism
LSMDB	:	Locus-Specific Mutation DataBase
MAE	:	Mean Absolite Error
MCC	:	Matthews Correlation Coefficient
MSA	:	Multiple Sequence Alignment
MUSCLE	:	MUltiple Sequence Comparison by Log-Expectation
ncSNP	:	Non-Coding Single Nucleotide Polymorphism
NMR	:	Nuclear Magnetic Resonance
nSNP	:	Nonsense Single Nucleotide Polymorphism
nsSNP	:	Non-Synonymous Single Nucleotide Polymorphism

OMIM	:	Online Mendelian Inheritance in Man
OOB	:	Out-Of-Bag
PD	:	Pathogenic Deviation
PDB	:	Protein DataBank
PQS	:	Protein Quaternary Structure database
rASA	:	Relative Accessible Surface Area
RF	:	Random Forest
RMSE	:	Root Mean Squared Error
ROC	:	Receiver Operating Characteristic
SAAP	:	Single Amino Acid Polymorphism
SNP	:	Single Nucleotide Polymorphism
sSAAP	:	Silent Single Amino Acid Polymorphism
sSNP	:	Synonymous Single Nucleotide Polymorphism
SVM	:	Support Vector Machine
TDM	;	Torsion Density Maps
TN	:	True Negative
TP	:	True Positive
TPR	:	True Positive Rate
UniProt	:	UNIversal PROTein resource
UniProtKB	:	UniProt KnowledgeBase

Acknowledgements

I would like to express my heartfelt gratitude to Dr. Andrew Martin, who was not only my supervisor and mentor but is also now a dear friend. I could not have asked for better role model; so inspirational, supportive, and patient. Besides my supervisor, I would like to thank the two members of my thesis committee: Dr. Snežana Đorđević and Dr. Joanne Santini, for their encouragement and support. This thesis was co-funded by the King Faisal Specialist Hospital and Research Centre (KPSHRC) and the Ministry of Higher Education - Saudi Arabia (MOHE). I would like to thank both organisations for their generous support.

I could not have contemplated this road if not for my parents, Suliman and Shaikh, who instilled within me a love of success and the determination to achieve my goals. To my parents, thank you. My one and only sister, my shoulder, my biggest fan, my best friend and the greatest gift in this life, Nuha, thank you! My brothers, Khalid, Abdulrahman, Naif and Ahmed, have also been the best of friends to me along this journey; you are my four leaders and with you I cannot be lost on any road. My warm appreciation goes to my second family, Alajaji, auntie Jouhara, uncle Yousif and my lovely cousins Norah, Beesan and Aram.

This thesis would not have been possible without my dear friend Anja, who believed in and supported me from day one, and Jim, who was always there to encourage me and give that extra push. My sincere gratitude is also extended to the ACRM and CATH group members.

I feel a deep sense of gratitude for all of my friends who covered me with infallible love and support, who reminded me that I am not alone in this journey, and have always been my strength; Abeer, Alaa, Hala, Fadwa, Jasmin, Moudhi, Njawa, Nouf, Rana, Samar, Sara, Shoug, Sultana and Yara.

Finally, I am also very grateful for my *Espresso*, for keeping me awake while finishing work well into the night and giving me a special start every morning. We shared thoughts, breaks and many '3AM' moments. I cannot think about a single day without you, thank you!

Nouf S. Al-Numair ♡

Contents

List of Figures

List of Tables

Chapter 1

Biological Introduction

Proteins are one of the basic building blocks of the human body and are essential for nearly everything the body does. A faulty modification of the genetic material of the cell may produce a malfunctioning protein. The study and analysis of the structure of proteins and their mutations will improve the understanding of mutational effects, which may lead to cures for untreated diseases.

Many mutations are related to disease and mutations of single nucleotides may affect the structure and interactions of proteins by means of amino acid substitutions. Recently, there has been increased research into these mutations. In order to understand the molecular mechanisms of disease, it is essential to evaluate the effect of these mutations on the structure and function of proteins.

Andrew Martin's group has gathered information on mutations related to human diseases and Single Nucleotide Polymorphism (SNP) data (which should not be a direct cause of disease), and incorporated it into the Single Amino Acid Polymorphism database (SAAPdb) (Hurst *et al.*, 2009). This is a database of disease-causing and neutral mutations, which have been analysed to determine what effect, if any, they may have on protein structure and function. This PhD is a part of the SAAP project and aims to maintain and expand SAAPdb, introduce the SAAP Pipeline (SAAPdap) and build a SAAP Predictor (SAAPpred). This chapter explores the biological basis of mutations and the effects that they can have on protein structure and function.

1.1 Mutation

Mutation or genomic aberration is one of the most important aspects of disease research. A mutation refers to a change in genetic structure, which may occur spontaneously, by chance, or through damage caused by radiation, mutagenic chemicals, or even viruses (Rothenberg and Chapman, 1989). Mutations do not have a consistent effect on phenotype (McMillan, 2009), although a mutant gene may affect the normal transmission and expression of a trait (Rothenberg and Chapman, 1989). Thus:

- some have negligible or no effect on phenotype;
- some introduce variation in phenotype without compromising health;
- some may offer a phenotypic advantage; and
- some result in a general phenotypic disadvantage:
  - some result in increased susceptibility to disease;
  - some are directly causative of a disease; and
  - some are fatal.

1.1.1 Mutations within genes

A DNA strand is a double helix structure, where the two strands run in opposite (anti-parallel directions). Each strand consists of a sequence of nucleotides or 'bases': adenine (A), guanine (G), cytosine (C) and thymine (T) with a sugar-phosphate backbone. Complementary base pairing between purine (A/G) and pyrimidine (C/T) bases (specifically between A-T and C-G) holds the two helical strands together (Figure 1.1).

The DNA sequence of a gene is constantly undergoing transformation by mutation. Mutations vary in size from a single distinct DNA nucleotide, through to a huge portion of a chromosome or entire chromosome (e.g. Down's syndrome). Here, the focus is on mutations occurring at the gene or 'coding' level, as mutations in coding regions are frequently associated with the development of various genetic diseases.

Approximately 3.2 billion of these sequence base pairs of DNA make up the human genome, which encodes ≈20,000 protein-encoding genes (International Human Genome Sequencing Consortium, 2004), which account for roughly 1.52% of the genome (Lander *et al.*, 2001). A codon is a unit of three nucleotides that encodes a single amino acid. There are 61 codons that define a specific amino acid (known as sense codons)

plus three stop codons, which signal the end of translation of the mRNA message into a protein (Lesk, 2005). The genetic code for nuclear protein-coding genes is universal (given in Table 1.1). These 'coding' regions of the genome are organised into 'genes' (distinct protein encoding units that define individual proteins). Any mutations in these 'coding' regions may therefore alter the structure and/or function of a protein, or alter the quantities of proteins expressed. The remainder of the genome (≈98%), consists of non-coding regions whose functions may include providing chromosomal structural integrity and regulating where, when, and in what quantity proteins are made.

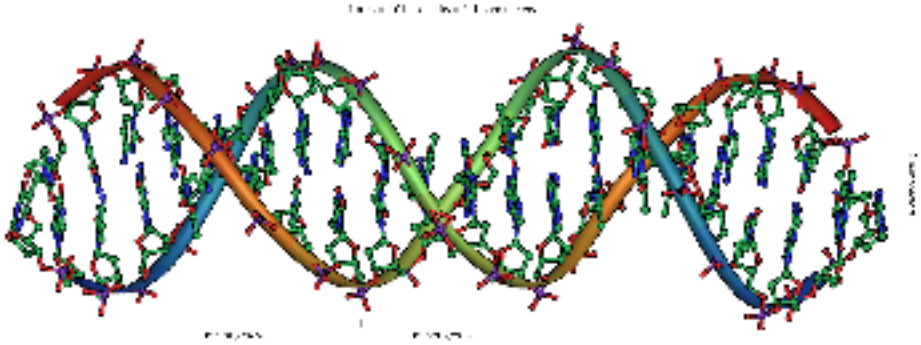


Figure 1.1: Structure of the double-stranded Deoxyribonucleic acid (DNA) and base pairing schema.
The nucleotides are shown here, attached to the sugar-phosphate backbone. *(Obtained from http://en.wikipedia.org/wiki/DNA under Creative Commons license).*

To understand how mutations in DNA can alter the structure and/or function of a protein and potentially alter phenotype, the mechanism of protein synthesis must first be understood. The first stage in protein synthesis involves the copying of one of the strands of DNA into a strand of messenger ribonucleic acid (mRNA) in a process known as transcription (Figure 1.3). Following transcription, mRNA moves out of the nucleus into the main body of the cell, where protein synthesis occurs. A ribosome combines with mRNA at the start of translation where the codon 'AUG' is recognized by an initiator transfer RNA (tRNA). The ribosome then assists in the elongation phase of the process. At this point, the anti-codon on the tRNA is sequentially combined with the correct complementary codon in the mRNA and codes for a particular amino acid (Strachan and Read, 2011).

Table 1.1: The standard 64 genetic code.

First Letter	Second Letter: T	Second Letter: C	Second Letter: A	Second Letter: G	Third Letter
T	TTT (Phe)	TCT (Ser)	TAT (Tyr)	TGT (Cys)	T
	TTC (Phe)	TCC (Ser)	TAC (Tyr)	TGC (Cys)	C
	TTA (Leu)	TCA (Ser)	TAA Stop	TGA Stop	A
	TTG (Leu)	TCG (Ser)	TAG Stop	TGG (Trp)	G
C	CTT (Leu)	CCT (Pro)	CAT (His)	CGT Arg	T
	CTC (Leu)	CCC (Pro)	CAC (His)	CGC Arg	C
	CTA (Leu)	CCA (Pro)	CAA (Gln)	CGA Arg	A
	CTG (Leu)	CCG (Pro)	CAG (Gln)	CGG Arg	G
A	ATT (Ile)	ACT (Thr)	AAT (Asn)	AGT (Ser)	T
	ATC (Ile)	ACC (Thr)	AAC (Asn)	AGC (Ser)	C
	ATA (Ile)	ACA (Thr)	AAA (Lys)	AGA (Arg)	A
	ATG (Met)	ACG (Thr)	AAG (Lys)	AGG (Arg)	G
G	GTT (Val)	GCT (Ala)	GAT (Asp)	GGT (Gly)	T
	GTC (Val)	GCC (Ala)	GAC (Asp)	GGC (Gly)	C
	GTA (Val)	GCA (Ala)	GAA (Glu)	GGA (Gly)	A
	GTG (Val)	GCG (Ala)	GAG (Glu)	GGG (Gly)	G

One by one, amino acids are covalently linked to each other leading to translation into a polypeptide chain according to the sequence encoded in the DNA via the mRNA (Alberts, 2008). In order to form the amino acid monomers into a polymeric chain, amino acids are condensed with one another through dehydration synthesis. This reaction occurs when H_2O is lost between the carboxylic group of one amino acid and the amino group of the next, to form a C-N bond. These polymerization reactions are not spontaneous; however, they occur through the energy-driven action of the ribosome. A stop codon or nonsense codon (UAA, UAC and UGA) will combine with a release factor at the end of the process. This ends the translation process and causes the ribosome to release the complete polypeptide (Manson, 2002) (Figure 1.3).

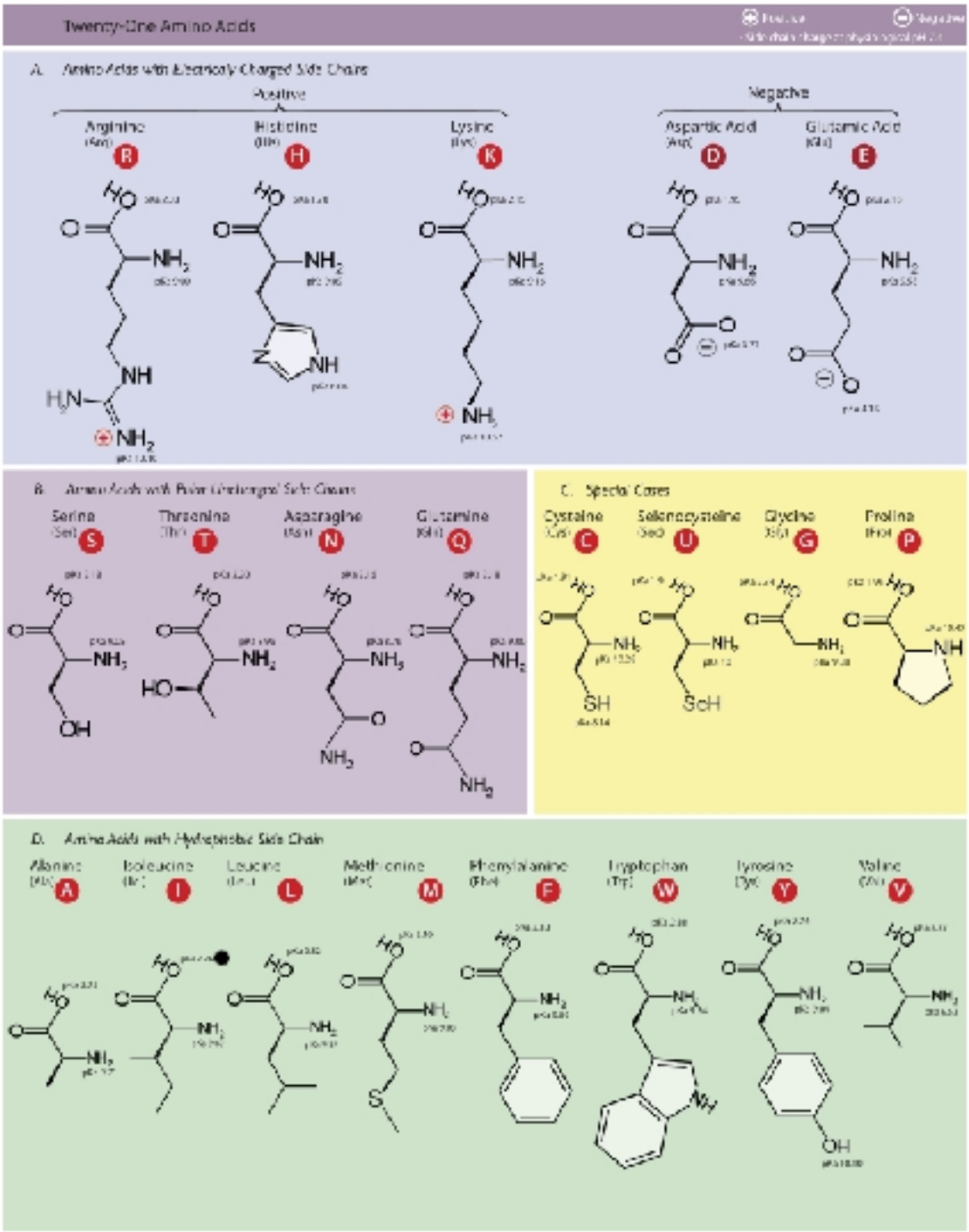


Figure 1.2: The twenty-one common amino acids together with the less common one (selenocysteine).

The twenty-one amino acids, grouped according to their side chains, pKa's and charge at physiological pH 7.4. Note that selenocysteine it is not encoded directly in the DNA. It is encoded in a special way by a UGA codon, usually a stop codon. The UGA codon encodes selenocysteine through the presence of a SElenoCysteine Insertion Sequence (SECIS) element in the mRNA. *(Courtesy of Creative Commons).*

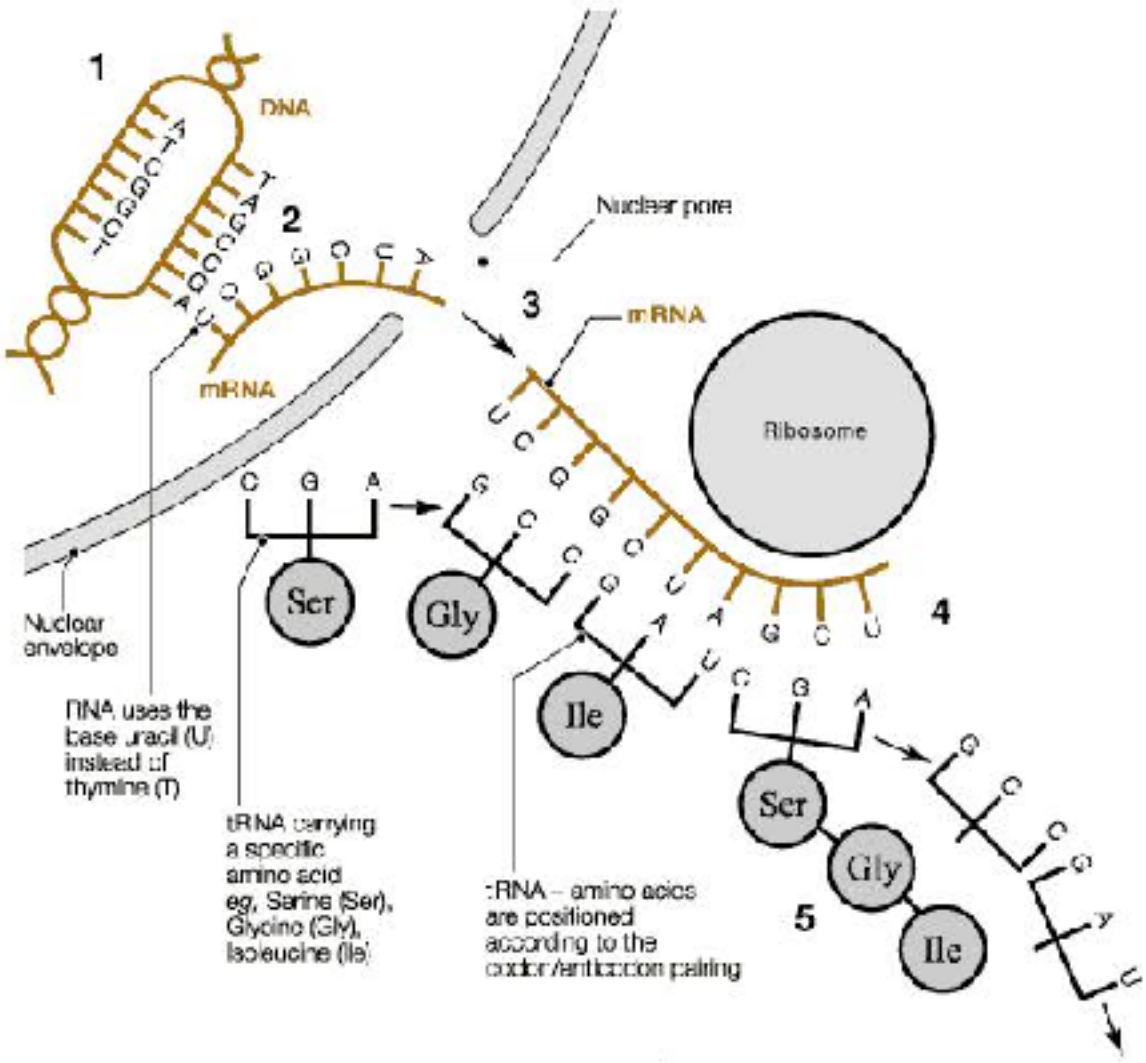


Figure 1.3: A broad overview of protein synthesis.
(1) The DNA double helix unwinds to expose a sequence of nitrogenous bases. (2) A copy of one of the strands is made in a process known as transcription. The copy is made of messenger ribonucleic acid (mRNA) which, following transcription, travels out of the nucleus into the main body of the cell, where protein synthesis occurs. (3) The mRNA couples with the protein synthesis apparatus (the ribosome). Another type of RNA, known as transfer RNA (tRNA), brings free amino acids to the ribosome. (4) The anticodon present on the tRNA recognises the codon present on the mRNA, and the ribosome adds the amino acid to the growing chain of linked amino acids (polypeptides), cleaving it away from the tRNA. This process is known as translation. (5) As the polypeptide chain grows, it folds to form a protein. *(Reproduced from Human Biology and Health Studies, Thomas Nelson, Walton-on-Thames, 1996).*

1.1.2 Mutations at the nucleotide level

Ninety percent of mutations are single base changes (Collins *et al.*, 1998), which can be base substitutions, insertions or deletions. The remaining 10% are insertions or deletions of larger DNA segments, generally as a result of recombination, or changes owing to relocation of mobile genetic elements (Collins *et al.*, 1998). A single base substitution, in which a single nucleotide base is replaced by another nucleotide, is also known as a point mutation. A 'transition' happens when a purine substitutes for another purine, or a pyrimidine replaces another pyrimidine. On the other hand, a 'transversion' occurs when a purine substitutes for a pyrimidine or a pyrimidine replaces a purine (Baird *et al.*, 1981).

Since an alteration in a single nucleotide is easily identified and more easily correlated with its impact on the structure and function of a protein, single base mutations are perfect for computational analysis at both the sequence and phenotype level (Mount, 2004).

There are five types of single base substitution mutation, as shown in Table 1.2. Figure 1.4 shows an overview of the effects of mutation on protein synthesis. There are a total of four mutations in the DNA indicated in purple, green, red and orange. A DNA sequence representing a single strand also highlights these mutations using the same colours. The light blue box at the base of the figure shows the sequence of the native protein that would be synthesised without the presence of the mutation. In the DNA coding section, a purple T>A represents a mutation that is synonymous (silent). This mutation does not affect the protein sequence. The green G>T mutation is known as a nonsense mutation that leads to a premature stop codon. The A>G shaded in red is a missense or non-synonymous mutation. It substitutes the native cysteine with arginine. The orange-coloured T>C is a non-coding mutation and takes place outside the gene.

Of these mutations, a **silent mutation** refers to DNA sequence alterations that have no effect on the final protein product because the same amino acid is inserted as a result of the degenerate nature of the genetic code. For example, when a mutation changes a codon from UCU to UCC, it will still encode a serine residue (see Table 1.1). This type of mutation can only be recognised through gene sequencing and can occur without affecting protein structure or function (Durbin, 1998). Nonetheless such mutation may have an effect on splicing or expression (see below).

A **nonsense mutation** is one in which a codon for an amino acid is replaced with one that codes for one of the stop codons (Figure 1.5a). This leads to the manufacture of prematurely truncated protein, which may function incorrectly. This type of mutation happens in 15-30% of all hereditary diseases including cystic fibrosis, haemophilia, retinitis pigmentosa and Duchenne muscular dystrophy.

A **missense mutation** occurs when the substituted base results in a new codon, which leads to the insertion of a different amino acid in the protein product (Figure 1.5b). The effect of a mutation is dependent on the type of amino acid involved, the position in the sequence and the structural context of the alteration (Khan and Vihinen, 2007). Beyond this chapter, this thesis will only consider single base substitution missense mutations.

Introns (non-coding sequences) must be spliced out of pre-mRNA, so that only exons (coding sequences) remain in the mRNA that is used during translation. Splicing must happen in a very accurate way, which is specified through nucleotide signals that identify specific splice locations. When a **splice mutation** occurs, the signals guiding the process are altered and one or more introns are not removed correctly. If this happens, an incorrect protein will be produced.

In addition to nucleotide substitutions, nucleotide insertions and deletions can also occur (Figure 1.5c and 1.5d, respectively). When this happens, the consequences are usually more serious than with substitution mutations, because, unless a multiple of three bases is inserted or deleted, the whole reading frame downstream of the insertion or deletion event is altered: a **frameshift** mutation (Figure 1.5e). In an **expression mutation**, a mutation occur in a transcription factor binding site of a gene such as a promoter or enhancer and alters promoter function and thereby alters gene expression levels.

1.2 The effect of mutations on protein structure

The previous sections have covered how a gene encodes a series of amino acids that make up a protein and that this sequence of amino acids (the primary structure; Figure 1.6a) can be changed through mutations in the DNA sequence. This section now focusses on the ways in which changes in the primary structure can fundamentally alter the way in which a protein forms and functions.

A polypeptide chain dictates regular geometric shapes in three-dimensional (3D) structures called secondary structure (Figure 1.6b). These are highly regular local substructures. One

Table 1.2: Types of single base mutations and their effect on the protein sequence.

Type of mutation		Expression	Splicing	mRNA Stability	Protein Stability	Protein Function	Termination	Extension
NonCoding		✓	✓	✓				
Coding	Synonymous	✓	✓	✓				
	Nonsynonymous	✓	✓	✓	✓	✓		
	Missense	✓	✓	✓	✓	✓		
	Nonsense	✓	✓	✓	✓	✓	✓	
	Extension	✓	✓	✓	✓	✓		✓

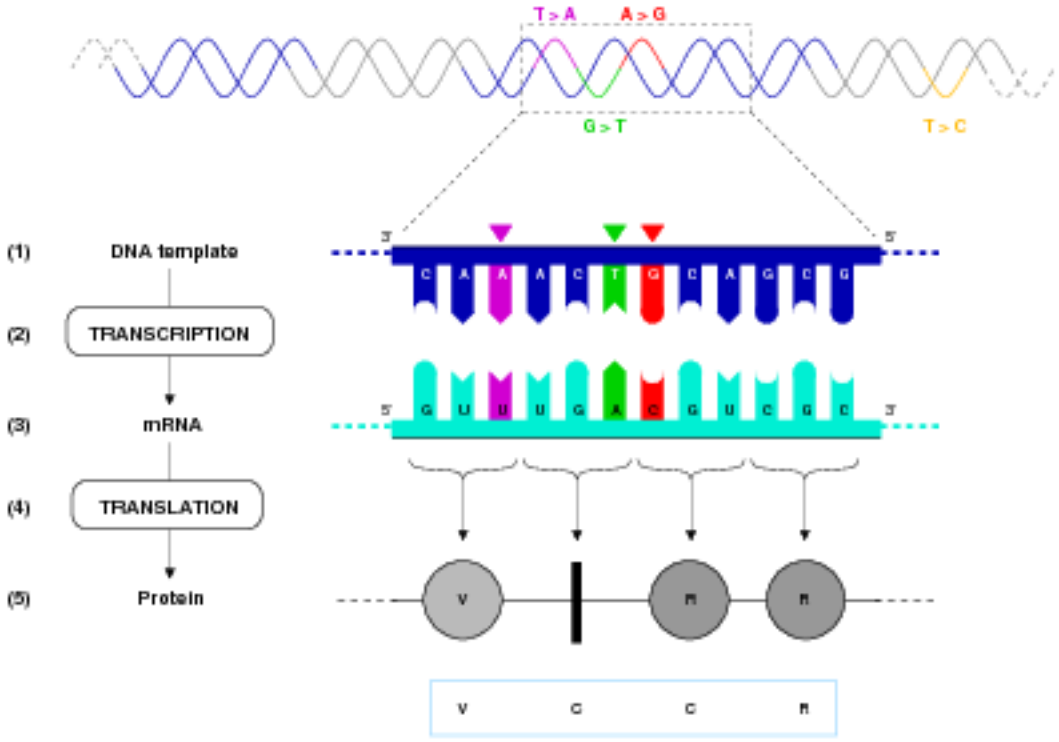
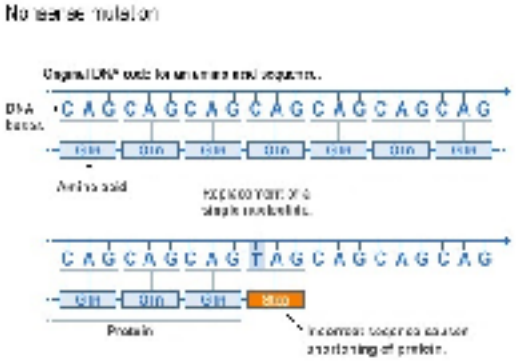
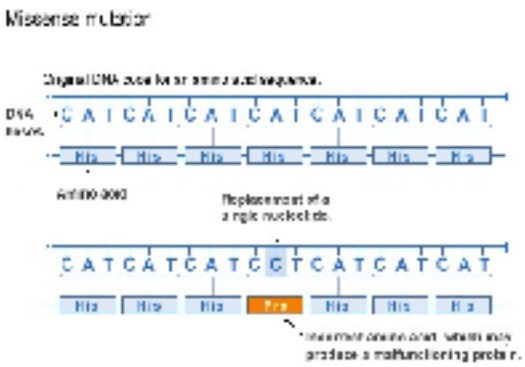


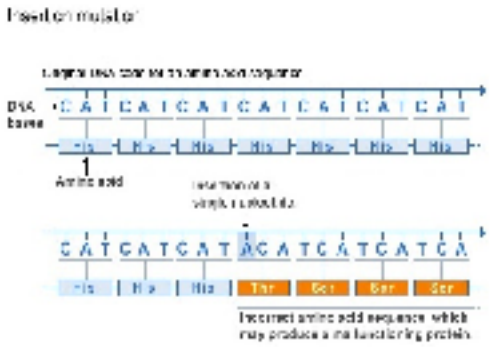
Figure 1.4: A broad overview of the effect of mutation on protein synthesis.
A section of DNA is shown at the top of this figure. Proteins are synthesised from genes and proceed as follows: (1) The double stranded helix is broken to expose a DNA 'template'; (2) The DNA is transcribed (using complementary 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. *(Adapted from McMillan (2009)).*



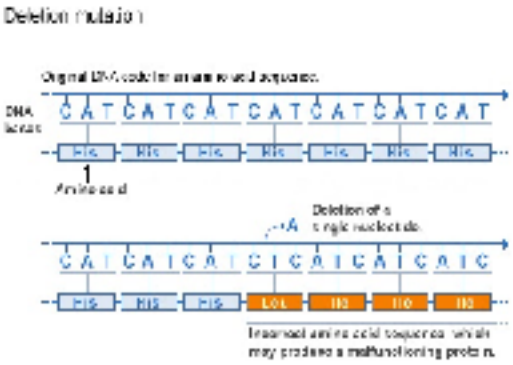
(a) The nucleotide cytosine is replaced by thymine in the DNA code, signaling the cell to shortening the protein.



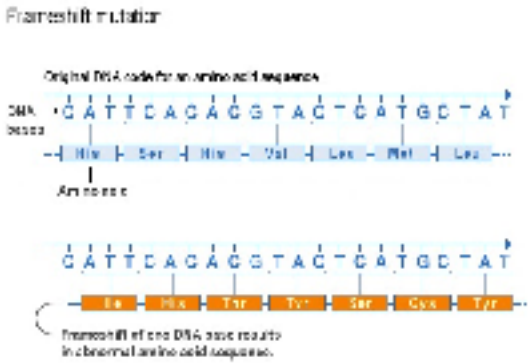
(b) The nucleotide adenine is replaced by cytosine in the genetic code, introducing an incorrect amino acid into the protein sequence.



(c) One nucleotide (adenine) is added in the DNA code, changing the amino acid sequence that follows.



(d) One nucleotide (adenine) is deleted from the DNA code, changing the amino acid sequence that follows



(e) A frameshift mutation changes the amino acid sequence from the site of the mutation.

Figure 1.5: Types of single base substitutions mutation. *(Adapted from Genetics Home Reference, U.S. National Library of Medicine* `http://ghr.nlm.nih.gov/handbook/illustrations`*).*

of the main conformational parameters of the amino acid structure is the value of the phi and psi angles. These angles define the conformation of the polypeptide chain. With repeated special values for these angles, the main chain can adopt conformations such as the α-helix or β-strand (Figure 1.6b). Both of these structures owe their stability to the hydrogen bonds between N-H and O=C atoms. Certain combinations of secondary structure can be observed in folded proteins, which form distinct functional domains or structural motifs such as a helix-turn-helix, leucine zipper, EF-hand calcium binding, and zinc finger domains. These are all referred to as supersecondary structures (Berg *et al.*, 2006).

An intact 3D structure of the polypeptide chain and the arrangement of amino acids so that those far apart in the primary structure come together in space is referred to as the tertiary structure (Figure 1.6c). The stability of this structure is determined by non-covalent interactions and disulphide bonds. Each globular protein ultimately folds into a 3D shape with a distinct inside and outside. The interior of a protein molecule contains a preponderance of hydrophobic amino acids, which tend to cluster and exclude water. The core is also stabilized by Van der Waals forces and hydrogen bonds. In contrast, the exterior of a protein molecule is largely composed of hydrophilic amino acids, which are charged or able to hydrogen-bond with water allowing protein to have greater solubility (Berg *et al.*, 2006).

Many proteins consist of two or more polypeptide chains that are commonly referred to as 'subunits'. Quaternary structure refers to the arrangement of subunits in a multichain protein (Figure 1.6d). Protein stability is determined by noncovalent forces such as hydrogen and ionic bonds, Van der Waals and hydrophobic interactions. Protein chains can associate with other chains to form dimers, trimers and other higher orders of oligomers. Generally, multimers contain 2–6 subunits, which may be chains with the same sequence (homomultimers) or different sequence (heteromultimers).

The function of a protein relies on the precise conformation of the fully folded protein. In turn, the correct folding is dictated by the sequence of amino acids that make up the primary structure. Any change to the amino acid sequence, for example, by the occurrence of a missense mutation, may result in a change in the way in which amino acids interact with each other. Even a slight change in protein 3D structure can alter function, which can have advantageous or deleterious phenotypic consequences. The ways in which amino acids interact with each other is outlined in the next section.

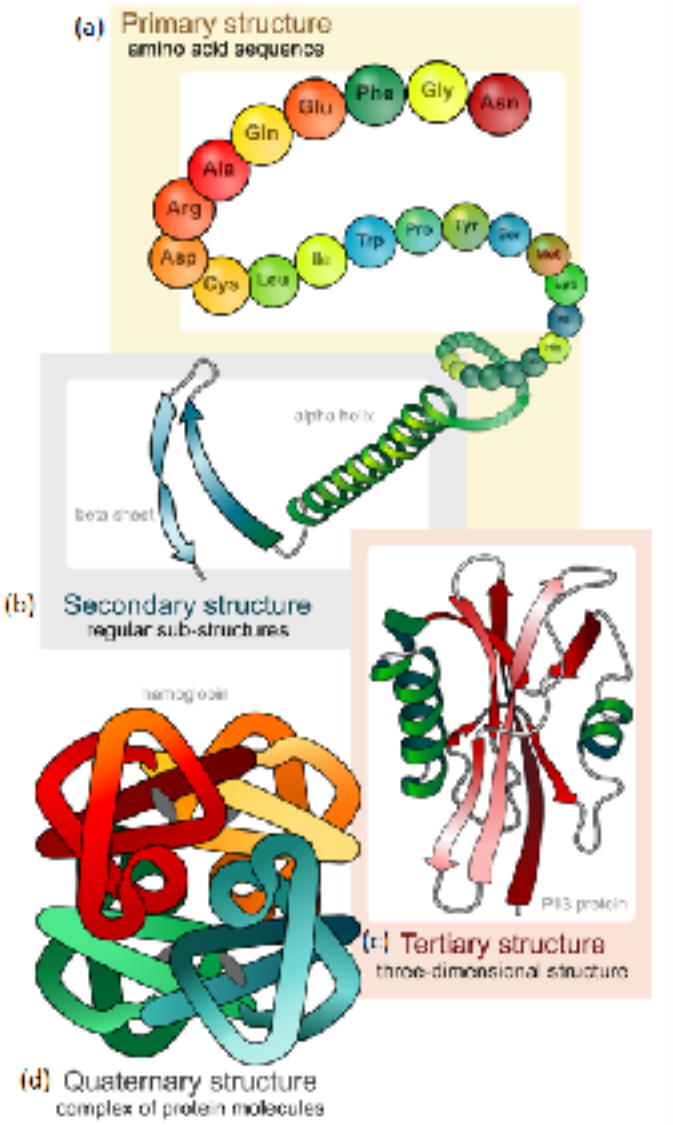


Figure 1.6: Structural hierarchy of proteins.
(a) Primary protein structure is a chain of amino acids. (b) Secondary protein structure α helixes and β sheets occur when the sequence of amino acids is linked by hydrogen bonds. (c) Tertiary protein structure occurs when certain attractions are present between α helix, β pleated sheets and loops. (d) Quaternary protein structure describes a protein consisting of more than one amino acid chain. *(Obtained from* `http://en.wikipedia.org/wiki/File:Main_protein_structure_levels_en.svg` *under Creative Commons license).*

1.2.1 Forces controlling protein structure

The chemical nature of amino acids dictates the specific ways in which they interact within a protein. The structures of the 21 amino acids and their chemical properties are given in Figure 1.2. The following sections detail the type of bonding or interaction in which each of these amino acid types can be involved, relating these to the possible effects of disruption of those interactions.

1.2.1.1 Hydrogen bonds

A hydrogen bond occurs when an electronegative atom interacts with a hydrogen atom that is covalently bonded to another electronegative atom (Baker and Hubbard, 1984). In proteins, this directorial interaction usually shares the hydrogen atom between oxygen and nitrogen atoms (Baker and Hubbard, 1984). The electronegative atom without a hydrogen bound is described as an acceptor atom and the hydrogen atom (or the atom to which it is bound) is described as the donor. Figure 1.7 shows the hydrogen bonding capacity of amino acids.

The vast majority of backbone-sidechain hydrogen bonds are enclosed (inside the protein) indicating that this type of interaction is important in maintaining the stability of the intra-protein structure (Eswar and Ramakrishnan, 2000). Non-local hydrogen bonds (sidechain-sidechain) play an important part in the formation of protein tertiary structure. Although all hydrogen bonds are essential for the proper formation and stability of protein structure, it has been shown that the local bonds provide more stability to a protein than non-local hydrogen bonds (Shi *et al.*, 2002).

Mutation analysis of chemotaxis protein Chey in *Escherishia coli* has shown that replacement of hydrophobic amino acids (valine) with ones that are capable of establishing hydrogen bonds (threonine), increases the stability of the protein structure (Wilcock *et al.*, 1998). Other studies have shown that replacing threonine with residues not capable of hydrogen bonding results in protein destabilization (Alber *et al.*, 1987). More recently, a method was developed to evaluate whether hydrogen bonds can be maintained when mutations occur to residues involved in hydrogen bonding (Cuff *et al.*, 2006). Computational methods such as this could help to identify types of mutations that affect one of the most important inter-atomic interactions in proteomics.

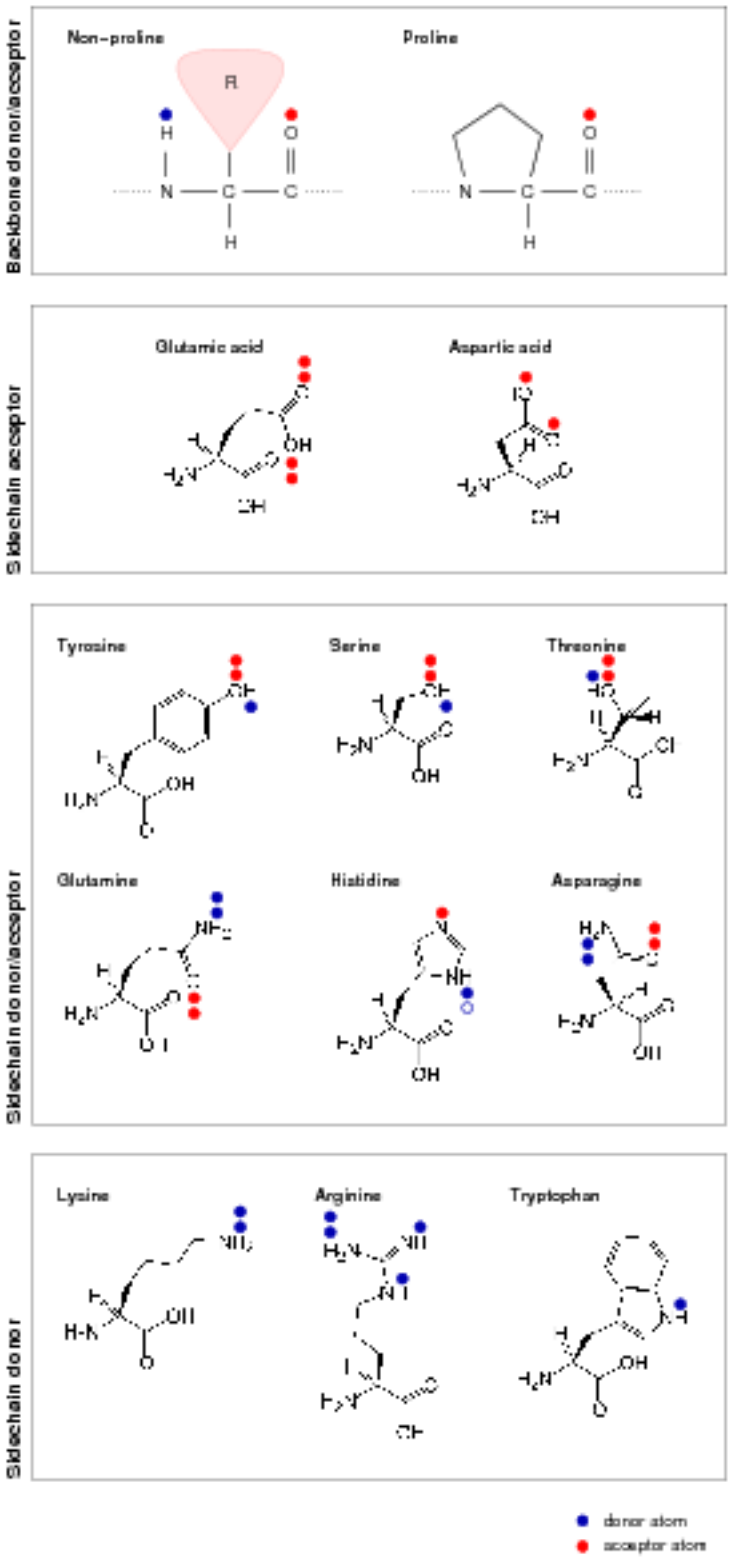
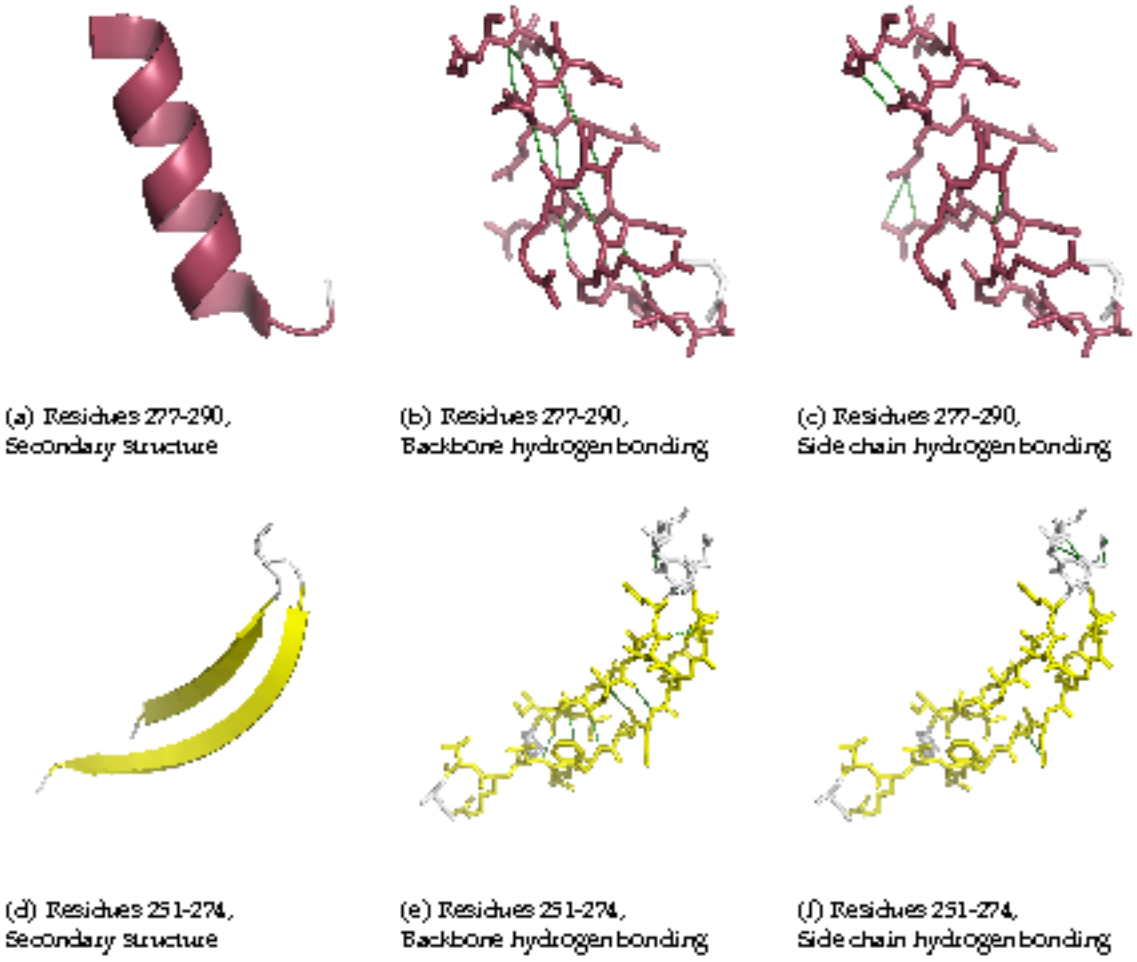


Figure 1.7: Hydrogen bonding in amino acids.
All residues are able to form backbone hydrogen bonds (although proline can only form a backbone hydrogen bond as the hydrogen bond acceptor). In addition, some residues are able to form hydrogen bonds with their side chain. Dots directly above or below an atom indicate that it may act as a donor (blue) or acceptor (red). The empty blue dot indicates that histidine is able to donate two hydrogens when it is positively charged. Residue side chains may form more than one hydrogen bond, and may act both as a donor and acceptor. *(Adapted from McMillan thesis (2009))*.



(a) Residues 277-290, Secondary structure

(b) Residues 277-290, Backbone hydrogen bonding

(c) Residues 277-290, Side chain hydrogen bonding

(d) Residues 251-274, Secondary structure

(e) Residues 251-274, Backbone hydrogen bonding

(f) Residues 251-274, Side chain hydrogen bonding

Figure 1.8: Backbone hydrogen bonding generates α and β secondary structures.
(a) An α helix (residues 277-290) and (d) β sheet (residues 251-274) from the structure of tumour suppressor protein 53 TP53 (PDB ID 2x0u). Hydrogen bonds are indicated by thinner connections (with green indicating backbone bonds ((b) and (e)) and dark blue indicating side chain bonds ((c) and (f)). Residues are coloured by structure (with yellow indicating β structures and pink indicating α structures).

Hydrogen bonds can be divided into local and non-local interactions depending on the distance in linear sequence between interacting partners. The majority of backbone-backbone (Stickle *et al.*, 1992) and backbone-side chain interactions (Eswar and Ramakrishnan, 2000) are local interactions among near-neighbour residues. α-helices are created and maintained by local hydrogen bonding among backbone atoms (Kabsch and Sander, 1983; Wilmot and Thornton, 1988). These make up approximately two thirds (68%) of hydrogen bonds in the protein (Stickle *et al.*, 1992) (see Figure 1.8). McDonald and Thornton, (1994) showed that almost all buried H-bond capable side chains are involved in H-bonding. Hydrogen bonds are fundamental to the proper formation and stability of protein structure, disruption of a buried H-bond caused by a mutation will have destabilising effect on protein.

1.2.1.2 Covalent bonds - disulphide bridges

Covalent disulphide bonds form by oxidation of thiol groups in two cysteine residues (Hazes and Dijkstra, 1988); present on the same or different polypeptides (Murray and Harper, 2000) (see Figure 1.9). Thangudu *et al.* (2008) showed that the majority of those bonds are formed between cysteine residues near to each other in the polypeptide sequence. Distant disulphide bonds (i.e. between cysteine residues that are more than eight residues apart) are less frequent, but play an important rôle in the folding and stability of native protein structures (Abkevich and Shakhnovich, 2000). It has also been shown that the degree of the stability is dependent on (i) protein conformation and (ii) the number of residues between linked cysteines: more residues between the disulphide bridge result in a more stable native structure (Pace *et al.*, 1988).

In 1996, Jeffrey *et al.*'s (1996) calculations suggested that a disulphide bond should give rise to 2.5 - 3.5 kcal/mol of stabilization, depending on the primary sequence separation between the cross-links. Introduction or deletion of disulphides by site-directed mutagenesis has produced varying effects on stability and folding depending upon the protein and location of disulphides in the 3-D structure (Thangudu *et al.*, 2008).

The importance of disulphide bonds to protein stability and function is demonstrated in Parkinson's disease, where mutations in DJ-1 can cause an early-onset form of the disease (Canet-Aviles *et al.*, 2004). Many mutations have been identified, including large deletions and missense mutations, thought to abolish cysteine kinase activity and disulphide bonding at the affected residue (Olzmann *et al.*, 2004; Logan *et al.*, 2010). Restoration

of a disulphide bridge between two opposite subunits has been shown to stabilize several DJ-1 mutants and increased the ability to scavenge reactive oxygen species and block protein aggregation events (Logan *et al.*, 2010). Identification of such destabilizing mutations is important for the identification of protein stabilization strategies that can be used therapeutically. Consequently, disruption of disulphides present in a native structure is likely to have an important effect on protein stability.

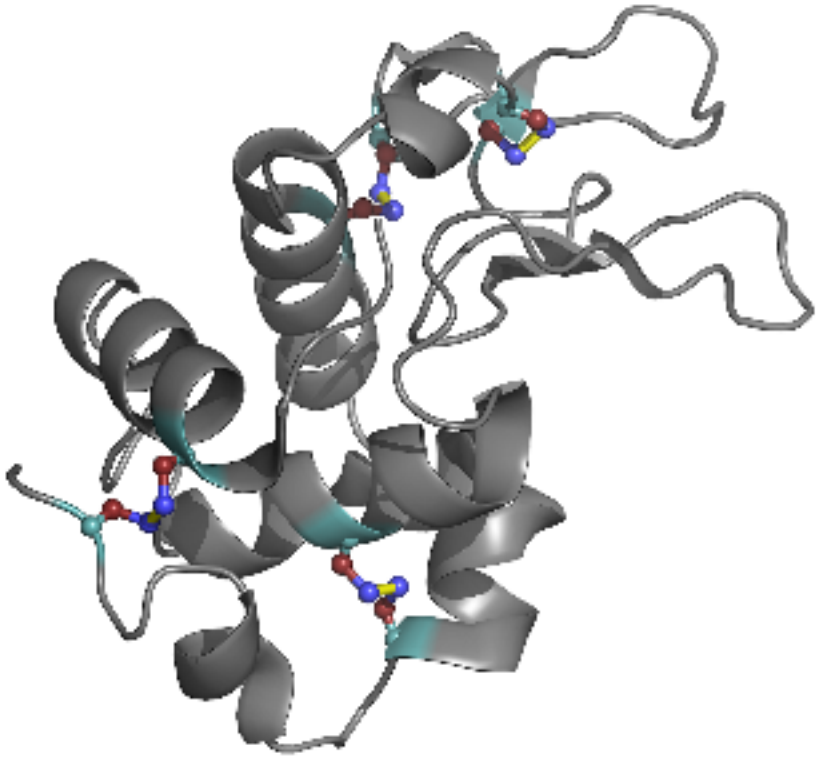


Figure 1.9: Disulphide bonding.
Four disulphide bonds are formed between eight cysteine residues (6-127, 30-115, 76-94 and 64-80) in lysozyme (PDB ID 7lyz). Cysteine residues are highlighted as α-carbon in green, β-carbon in red and γ-sulphur in blue. Disulphide bonds are highlighted in yellow.

1.2.1.3 The hydrophobic effect

The amino acid R group (side chain) is either hydrophilic (Figure 1.2; a polar side chain has a tendency to interact and form hydrogen bonds with water and other polar substances) or hydrophobic (Figure 1.2; a non-polar side chain, thus preferring other neutral and non-polar molecules). Hydrophobic residues often cluster together and their R-groups tend to drive them away from the exterior of proteins and into the interior where they are buried into protein coreforming micelles (Tanford, 1980). This is considered to be the key driving force in protein folding, and restricts the available conformations that proteins can adopt (compare

Figures 1.10a and 1.10b) (Ptitsyn, 1998; Ptitsyn and Ting, 1999; Ting and Jernigan, 2002). A tightly packed hydrophobic core, maximizing favourable van der Waals contacts and minimizing cavities, is crucial for protein tertiary structure and stability of the protein (Levitt *et al.*, 1997; Richards, 1997; Lee *et al.*, 2000; Leiros *et al.*, 2000; Wang *et al.*, 2000; Northey *et al.*, 2002).

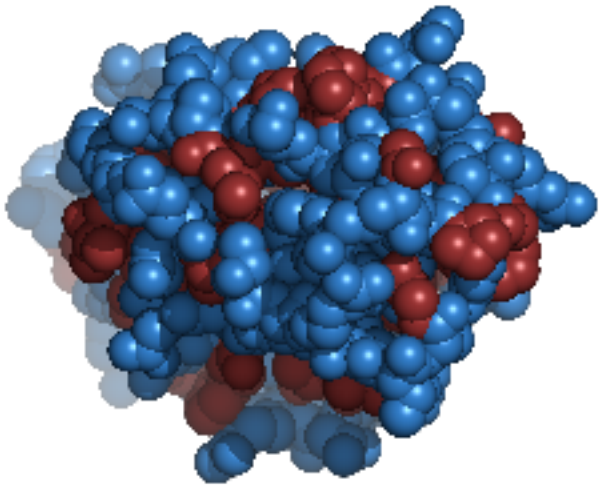
Disruption of the hydrophobic core or exposure of hydrophobic amino acids on the surface can have profound effects on the function of a protein. For example, mutations in the prion protein that cause an increase in exposure of hydrophobic amino acids are thought to be the cause of prion toxicity (Corsaro *et al.*, 2011). Moreover, the hydrophobic core is generally less tolerant of changes that disrupt packing than the solvent-accessible surface (Bowie *et al.*, 1990). In the field of cancer research, 16 independent missense mutations have been identified in the BARD1 protein, which is the heterodimeric partner of the ovarian cancer predisposition gene product BRCA1. It has been suggested that mutations mapping to the hydrophobic core forming the BARD1:BRCA1 interface can prevent formation of the heterodimer and render BRCA1 functionally inactive, thereby predisposing to ovarian cancer (Morris *et al.*, 2002). Consequently introduction of hydrophilic resides into the hydrophobic core, or introduction of hydrophobic residues on the protein surface, is likely to have an effect on protein stability.

1.2.1.4 Van der Waals forces (dispersion forces)

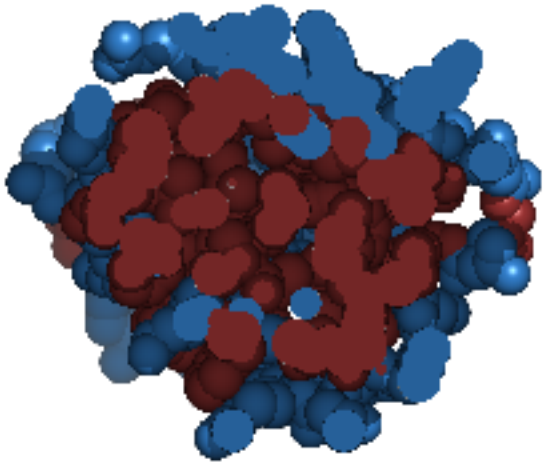
Van der Waals forces are very weak non-covalent interactions (0.01 - 0.2 kcal/mol) and result from interactions between induced dipoles that arise from fluctuations in atomic charge densities giving the attractive component. The repulsive component is the result of the electron-electron repulsion that occurs as two clouds of electrons begin to overlap (Ponder and Case, 2003). These weak interactions stabilize the protein based on the huge number of dispersion forces that occur in protein molecules, and these significantly contribute to protein folding and stability (Eriksson *et al.*, 1992; Chen and Stites, 2001).

The interaction are described by the Lennard-Jones potential Equation 1.1 and Figure 1.11, where E is the potential energy, A and B are constant parameter adjustable based on the interaction atoms and r is the distance between the atoms.

$$E_{vdW} = \frac{A}{r^{12}} - \frac{B}{r^{6}} \tag{1.1}$$



(a) Lysozyme (7lyz)



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

Figure 1.10: The hydrophobic core.
Hydrophobicity in lysozyme (PDB ID 7lyz). Blue indicates hydrophilic residues, red indicates hydrophobic residues. (a) shows the whole protein; (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.

Although it is harder to attribute changes in protein function to specific alterations in van der Waals forces owing to mutation, it is clear that substitution of an amino acid for one with atoms having different atomic numbers will increase or decrease the strength of the dipole effect. However, since van der Waals forces are very weak but numerous, alterations to van der Waals forces resulting from a single point mutation are unlikely to result directly in deleterious consequences. Mutations that results in a clash between atoms will result in a very high van dar Waals energy and lead to disruption of the structure.

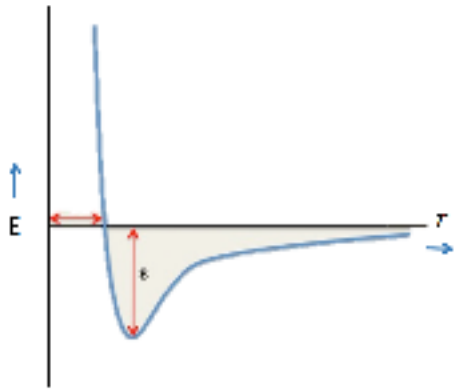


Figure 1.11: Lennard-Jones potential.

1.2.1.5 Electrostatic interactions and salt bridges

Electrostatic forces can be a charge-charge, charge-dipole, or dipole-dipole interactions. The attraction of negatively and positively-charged amino acid side chains (Figure 1.2) can form quite a strong electrostatic force that stabilize protein structure, falling off as the square of the distance between the charged atoms (Mitchell *et al.*, 1992). Interaction strength also depends heavily on the dielectric constant of the medium in which the protein is dissolved. Water and ions can shield electrostatic interactions (as can parts of the protein itself), reducing both their strength and the distance over which they operate. Ionic bond formation depends on the protonation state of the partners and hence on pH. Ionic bonds are local electrostatic interactions of charged atoms over a distance of 4Å or less. Salt bridges are ionic bond interactions between atoms that are also hydrogen bonded (Torshin and Harrison, 2001).

Electrostatic interactions are described by Coulomb's law (Equation 1.2), where q_1 and q_2 are the charges, ϵ is the dielectric constant and r is the distance.

$$E = \frac{q_1 q_2}{\epsilon r^2} \tag{1.2}$$

Long-range electrostatic effects are important not only for stabilizing the tertiary (Heremans and Heremans, 1989; Torshin and Harrison, 2001) and quaternary structure, but also for protein function (e.g. catalysis and ligand binding). A substantial component of the energy involved in protein folding is charge-dipole interactions (Jelesarov and Karshikoff, 2009). This refers to the interaction of ionized R-groups of amino acids with the dipole of the water molecule.

Mutation of any amino acid can affect the shape of the dielectric and the ion accessibility surfaces of the proteins. For example, mutation of residues that are charged at physiological pH (arginine, lysine, glutamic acid, aspartic acid and histidine) to the non-polar alanine cause perturbations in the electrostatic potential distribution of proteins without larger changes to protein structure (Gorham *et al.*, 2011). On the other hand, in phenylketonuria (PKU), a genetic disease caused by mutations in the human phenylalanine hydroxylase (PAH) gene, most of the missense mutations identified result in misfolding of PAH (Pey *et al.*, 2007). Using the protein-design algorithm FoldX, most mutations showed a correlation between energetic impact and residual protein activities and the patient phenotype (Pey *et al.*, 2007). This analysis suggested that decreased protein stability through disruption of electrostatic interactions was the main molecular pathogenic mechanism in PKU.

1.2.1.6 Binding sites in protein structures

A protein ligand is a biomolecule, atom, or ion (e.g. substrates, inhibitors, activators, metals and neurotransmitters), which binds to a specific binding site on a protein and has an effect on its activity, function, or conformation (Figure 1.12). Consequently mutations to residues interacting with ligands are likely to have an effect on protein function. The ligand interacts with its specific protein using the three standard intermolecular forces: (i) electrostatic forces between oppositely-charged ionic or polar groups, (ii) hydrogen bonds and (iii) van der Waals forces as well as through the hydrophobic effect.

A mutation that disrupts ligand binding has been observed in X-linked lymphoproliferative (XLP) syndrome. Missense mutations in the SH2 domain protein SH2D1A or SAP, prevent binding to its physiological ligands including the signalling lymphocyte activating molecule (SLAM) (Li *et al.*, 2003). It is theorized that reduced binding of SAP to SLAM

results in activated SLAM binding to other *SH2* domain proteins, resulting in T cell activation and hyperproliferation of lymphocytes (Nelson and Terhorst, 2000). XLP is a rare genetic disorder characterized by a predilection for fatal or near-fatal Epstein-Barr virus infection, subsequent hypogammaglobulinemia, and a markedly increased risk of lymphoma or other lymphoproliferative disease (Chaganti *et al.*, 2008).

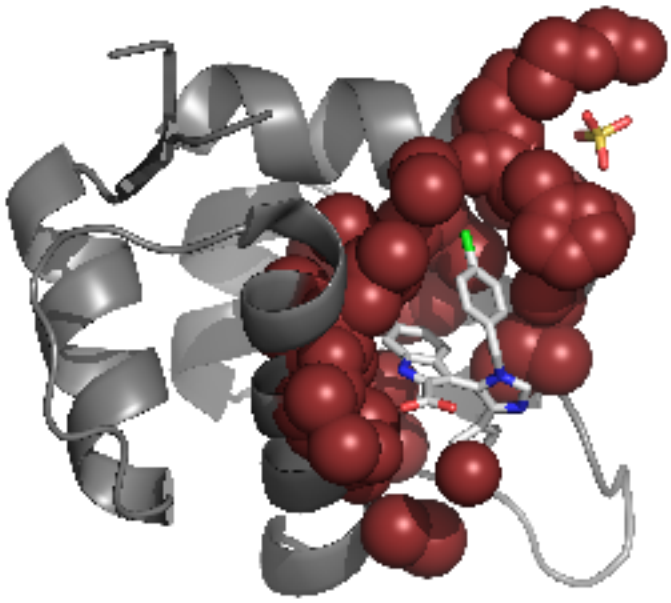


Figure 1.12: Ligand binding.
The structure of a complex between the human MDM2 protein and a small molecule inhibitor (PDB ID 3lbk) which mimics the native ligand. The ligand (inhibitor) 6-chloro-3-[1-(4-chlorobenzyl)-4-phenyl-1H-imidazol-5-yl]-1H-indole-2-carboxylic acid is shown embedded in a binding pocket, and residues within 4.5Å of the ligand are highlighted in red (the rest of the protein is shown in dark grey, with secondary structural elements indicated).

1.3 Phenotypic consequences of mutation

Clearly, an alteration in DNA may change the sequence of a protein such that it leads to a partially or completely non-functional protein. Table 1.2 listed the different effects that single base mutations may have, and all but truly silent mutations may result in disease. Studies on Drosophila melanogaster, show that where a mutation has altered a protein sequence, there is a 70% chance of the mutation being harmful (Sawyer *et al.*, 2007). Conversely, as

described in Section 1.1.1, some mutations are known to confer a genetic advantage, while others have no effect at all. Such mutational consequences are discussed below.

1.3.1 Mutations with phenotypic advantage

Mutations generally have a negative connotation and are assumed to provide nothing but health complications. However, the Lawrence Berkeley National Laboratory at the United States' Department of Energy has published research regarding a rare mutation in proteins that protects humans from cardiovascular disease (Berkeley-Lab, 2002). This discovery is a possible breakthrough in creating more effective medicines for eliminating cholesterol and preventing its accumulation. Damage from oxidation, where free radicals remove electrons from healthy tissues, can be a result of mutations. Diseases such as Alzheimer's, osteoporosis, and atherosclerosis are believed to be caused by excessive oxidation. In atherosclerosis, free radicals tend to withdraw electrons from lipids in artery walls, resulting in plaque formation and blockage of the arteries. However, the Berkeley study showed that the apolipoprotein A-I protein (apoA-I), when it undergoes a particular mutation (Arg173Cys), keeps an antioxidant embodied in the sulphur-based residue, cysteine, that absorbs unpaired electrons and blocks arterial inflammation (Berkeley-Lab, 2002).

The Berkeley research is a response to a paradox that has been puzzling the world of medicine since 1980. At this time, an Italian citizen was sent to Milan's Lipid Centre because of his high level of blood triglyceride; one of the factors that increases the threat of heart ailments. Additional medical checks showed that the patient also had a very low level of protective high-density lipoprotein (HDL), which removes unwanted cholesterol from the coronary arteries and hinders the formation of plaques. However, the patient had not displayed any pathological signs. This patient and others from the same region of Italy were later identified as having a mutated form of the protein apoA-I.

The mutant form known as apoA-I Milano occurs in less than 1 in 50 people and introduces a free cysteine, which possesses a sulfhydryl group. In the Milano mutation, almost 70% of the protein exists as dimers, mainly caused by an inter-chain disulphide bridge. Such pairing prevents the accumulation of HDL, leading to the deficiency found in humans who have this mutation. The remaining 30% of Milano mutant proteins remain as monomers in which the sulfhydryl is unoccupied and is available to perform other reactions including acting as a strong antioxidant. Consequently the mutation has the ability to counter cardiovascular disease by resulting in the elimination of cholesterol, as the reactions that cause its accumulation are prevented.

The next step in the Berkeley project is to utilise these data to develop more effective therapies, such as recent peptide-based cardiovascular disease treatments. Common therapies targeted against apoA-I eliminate cholesterol from the arteries through HDL. However, future treatments could link this with the antioxidant system caused by the mutation, thus both preventing the accumulation of cholesterol and limiting oxidation.

1.3.2 Neutral mutations

Mutations that occur naturally in DNA are normally corrected by DNA repair systems or have no selective advantage or disadvantage (Brctic Kostic, 2005). Mutations that do not affect the phenotype are called neutral mutations (Sunyaev *et al.*, 2000). Where this kind of substitution results in the use of a different amino acid in a protein, the replacement amino acid usually has very similar physicochemical features which have a negligible effect on the protein (e.g. if codon AAA is mutated to AGA, arginine would be used in the resulting protein instead of lysine).

A study by Ma *et al.* (2002) showed that when scanning a coding region of the NLI-IF gene (Nuclear LIM Interactor-Interacting Factor), which is physically near to the tuberculosis-associated gene NRAMP1, three SNPs [204C-A, 402T-C and 472G-A] were identified. None of these mutations in NLI-IF, showed any significant association with human tuberculosis. According to the neutral theory of Kimura (Speicher *et al.*, 2010), the great majority of evolutionary changes at the molecular level are caused by selectively neutral or selectively nearly neutral mutations. Thus, these three SNPs were neutral variants with little or no selective advantage or disadvantage.

1.3.3 Damaging mutations and penetrance

The term 'penetrance' is used in genetics to characterise the likelihood of individuals that carry an allele or genotype, to manifest a particular phenotype. It refers to the proportion of individuals with a disease-associated mutation showing clinical symptoms (Brenner and Miller, 2001), or the chance that a person who carries a mutation will be affected by the disease (Brctic Kostic, 2005; Liao, 2009). As low penetrance mutations rarely develop the symptom or trait with which they have been related at a detectable level, it is hard to disentangle environmental and genetic factors (Brenner and Miller, 2001).

It is well known that defects in the cell cycle, leading to unregulated cell division can cause cancer. For example, variations in DNA located near some genes that control growth can

also increase the risk of skin and brain cancer. Cline (2009) published five studies that show that changes close to the CDKN2A and CDKN2B genes increase the likelihood of some kinds of tumour. Previous research suggests that these genes are 'tumour suppressors'. Two per cent of people with melanoma also have CDKN2A mutations. Complete deletion of CDKN2A and CDKN2B is observed in almost half of all tumours in the brain. These genes have a significant rôle in several of the most basic processes in cells.

Of the five studies by Cline (2009), two were concerned with gliomas, which represent about 80% of all cancers of the brain and generally have a very poor prognosis. The other three studies looked at skin cancers. Two of the three studies related to melanoma while the other concerned basal cell carcinoma. Melanoma represents less than 5% of all skin cancers but accounts for most deaths caused by the disease. On the other hand, basal cell carcinoma is not deadly, but must also be treated carefully.

Mutations close to the genes CDKN2A and CDKN2B were considered independent in the three cancer-related studies. This may indicate that each has its own impact on increasing risk. Changes within and around CDKN2A and CDKN2B may also be important in other diseases; SNPs near those genes are associated with coronary artery ailments and type 2 diabetes (Cline, 2009).

On the other hand, high penetrance mutations are mutations where the carrier usually shows the effect of the defective gene. For example, carriers of BRCA 1 and 2 mutations have a higher than 80% chance of being afflicted with breast and/or ovarian cancer (Warburton, 2008).

Very high penetrance disease-causing mutations or 'pathogenic deviations' (PDs) are easier to identify and study, because the attribute created by the allele will always be apparent. This is called Mendelian inheritance because the attribute manifests itself by genetic transmission that can be accounted for by a distinct gene model (Brenner and Miller, 2001). In addition, Mendelian inherited PDs may demonstrate dominance and co-dominance.

A common example of a Mendelian inherited disease is sickle cell anaemia. When a nucleotide at the 17^{th} position of the gene encoding the β chain of haemoglobin is altered, the codon change leads to an amino acid in the 6^{th} position of the chain being changed from glutamic acid to valine. This results in a change in the quaternary configuration of haemoglobin that has a significant effect on human physiology.

Another example is described by Karkkainen (2000), who found that human hereditary lymphoedema is related to PDs in the vascular endothelial growth factor receptor 3 (eGFR3). An arginine to proline mutation in the extremely conserved structure of the catalytic site causes the protein kinase domain to become inactive; this restricts angiogenesis and vasculogenesis, and eventually causes lymphoedema. Clones of cells with different attributes within the same individual can develop owing to PDs in somatic cells. Most of the experimental data on pathogenic single amino acid polymorphisms relates to non-lethal PDs identified in somatic cells.

1.4 Study aims and objectives

> 'Find out the cause of this effect,
> Or rather say, the cause of this defect,
> For this effect defective comes by cause.'
>
> *William Shakespeare*

Variations in the human genome are a key data source for studies of disease development, potential treatments and understanding evolutionary mechanisms (Venselaar *et al.*, 2010; Studer *et al.*, 2013). Advances in high-throughput sequencing have accelerated the rate at which mutations are identified and exome sequencing (i.e. sequencing of the protein coding part of the genome) is likely to become the most common tool for the identification of Mendelian disease genes in the coming years (Gilissen *et al.*, 2012). Though these sequencing methods are becoming more commonplace, it is still very difficult to predict whether a SNP will cause a disease.

One way to determine the effect of mutation on protein function is by experimental exploration. Such experiments involve site-directed mutagenesis of different residues in different positions, which is time consuming and costly. An alternative to this approach is 3D modelling of side-chain mutations, though these models can only be predictive if they are highly accurate. That one seemingly insignificant change in a side-chain may cause a significant loss of protein function, while another has no effect, makes this type of modelling highly difficult to achieve (Feyfant *et al.*, 2007). Nonetheless, the predictive power of models improves as the quality of the information put into them improves and the number of training points used increases. For example, one model included molecular mechanics energy

terms for bond distances, angles, dihedral angles, peptide bond planarity, and non-bonded atomic contacts to predict the effect of mutation on protein structures (Feyfant *et al.*, 2007). A better understanding of how structural constraints affect protein evolution will help to optimize models of sequence evolution and to determine the consequences of a mutation event (Studer *et al.*, 2013).

Several web servers are available to help interpret mutational effects. Some methods are for the study of very specific mechanisms (e.g. a study of molecular mechanism of kinase activation by cancer mutations (Dixit *et al.*, 2009)), whereas others are developed to predict whether a variation is harmful or benign (Adzhubei *et al.*, 2010). Traditional prediction methods include SIFT (Ng and Henikoff, 2001), PolyPhen (Ramensky *et al.*, 2002), PolyPhen-2 (Adzhubei *et al.*, 2010) and Panther (Thomas *et al.*, 2003), which classify variants according to empirically derived rules (PolyPhen), Bayesian methods (PolyPhen2), or mathematical operations (SIFT, Panther) (Thusberg *et al.*, 2011). While Panther and SIFT are based on evolutionary information, other methods including PolyPhen, and state of the art prediction methods such as SNPs&GO (Calabrese *et al.*, 2009) and MutPred (Li *et al.*, 2009) are based on a combination of protein structural and/or functional parameters and multiple sequence alignment (MSA) derived information (Thusberg *et al.*, 2011). More recently, functional analysis of Hidden Markov Models (FATHMM) has been used to capture position-specific information within a MSA of homologous sequences (Shihab *et al.*, 2013). This system has been shown to out-perform both SNPs&GO and MutPred for the prediction of functional effects of protein missense variants (Shihab *et al.*, 2013). These methods are described in more details in Chapter 2.

This PhD forms part of the Single Amino Acid Polymorphism (SAAP) project and aims to maintain and expand SAAPdb, improve the analysis, introduce the SAAP database Pipeline (SAAPdap) and build a SAAP Predictor (SAAPpred). Having collected the mutation data in SAAPdb (Chapter 3), the database system analyses what effect, if any, mutations may have on protein structure and therefore function. SAAPdb attempts to identify the structural effect and therefore explain the mutation. The development of a conservative, comprehensive structural analysis pipeline with which to analyze SAAPs, is one of the main aims of the SAAP project (Hurst *et al.*, 2009).

However, it is important to realise that there may be more than one structure available for a protein containing a mutation. The protein structure may have been solved with different mutant residues, solved at different resolutions, in different space groups, or simply be multiple chains in a crystal structure. Thus, mutants may or may not be the same as the

mutation being examined. In all of these cases, certain 'explanatory factors' describing the local structural effects of a mutation may or may not be present in the various structures.

Previously (in the MSc project preceding this PhD study) cases where two or more structures of the same protein were available were examined. This was undertaken in order to assess the significance of structure variation on fifteen explanatory SAAPdb analyses, and to understand why the effects differ in the alternative structures. For instance, a mutation might introduce a charge shift in the core of the structure or may cause a clash (i.e. it is too big to fit in the space available) in one structure, but not in another (Al-Numair, 2010). The study showed that several of the analyses that were Boolean in nature (i.e. an effect either was, or was not, present) were very sensitive to precise structural details. This implied that such explanatory factors will be less reliable when low resolution structures or homology models are used for analysis. Consequently, an important aim was to change from a Boolean analysis to real-valued scales.

Overall, this project started by rebuilding the entire SAAPdb, incorporating updated and novel data sources. This was followed by re-analysing cases where multiple structures of mutations are available, to determine the analysis sensitivity to precise structure. The findings will help to assess the significance of factors by looking at the sensitivity of different analysis to alternative structures. This understanding allows us to improve and expand the data analysis spectrum and change the Boolean structural analyses to a continuous variable. These analyses can then be implemented and integrated into the SAAPdb pipeline. The project then focuses on determining rules that will aid in the interpretation of, or making predictions based on, the data by making a distinction between SNPs and PDs, in terms of their impact on protein structure. Hence, this information is useful for predicting whether a novel mutation would result in a disease phenotype and in future for designing novel disease therapies.

Chapter 2

Bioinformatics Resources and Methods

The analysis of the structural effect of mutations requires an understanding of a number of underlying resources and techniques described in this chapter. The Single Amino Acid Polymorphism Database (SAAPdb) is a resource developed in the Martin group that imports mutation data from raw data sources and analyses those data. The SAAP data analysis pipeline (SAAPdap), also requires access to numerous other resources, while the SAAP prediction software (SAAPpred) uses machine learning techniques to predict the pathogenicity (or phenotype) of an novel SAAP. The contributions of this research to the SAAPdap and the function of SAAPpred are described here and in chapter 3.

2.1 Primary information resources

The first step in building SAAPdb (see Chapter 3) was to prepare and import data from external sources. DNA data from the human genome were gathered from GenBank (Benson *et al.*, 2011) and the European Molecular Biology Laboratory (ENA) sequence database (Leinonen *et al.*, 2011). Protein sequence-based information was downloaded from UniProtKB (Consortium, 2011); and protein structures were retrieved from the Protein Data Bank (PDB) (Rose *et al.*, 2011). Single Nucleotide Polymorphisms (SNPs) were acquired from dbSNP (Sherry *et al.*, 2001). Pathogenic Deviation (PD) mutation data were gathered from several resources: the majority were derived from the Online Mendelian Inheritance in Man (OMIM) database (Amberger *et al.*, 2011), and a variety of smaller locus-specific mutation databases (LSMDBs) were also used. The PDBSWS protocol (Martin, 2005) was used to map sequence data onto structural data. These resources and their contents are described in more detail in this section.

2.1.1 GenBank, ENA and the DNA Data Bank of Japan

GenBank (Benson *et al.*, 2011)[1], ENA (Flicek *et al.*, 2011)[2] and the DNA Databank of Japan (DDBJ) (Sugawara *et al.*, 2008)[3] all contain publicly available nucleotide sequences along with supporting bibliographic and biological annotations (Cochrane et al. 2011). GenBank is maintained by the National Centre for Biotechnology Information (NCBI) in the United States. The ENA database is produced by the European Bioinformatics Institute in the United Kingdom. DDBJ is provided by the National Institute for Genetics in Japan. Each database collects a portion of the total sequence data reported worldwide, and the three systems are synchronised on a daily basis through an extensive information exchange, so that each database contains all of the available information.

2.1.1.1 GenBank

GenBank was created in 1982 at the Los Alamos National Laboratory, and development continued at Stanford University in the mid-1980s. By 1992, it had become the responsibility of the NCBI. Most submissions to GenBank come from individual laboratories or via batch submissions from large-scale sequencing projects. These sequencing projects include whole genome, shotgun (WGS), and environmental sampling projects. Sequences are also

[1] http://www.ncbi.nlm.nih.gov/genbank/
[2] http://www.ebi.ac.uk/ena/
[3] http://www.ddbj.nig.ac.jp/

deposited in GenBank by the United States Patent Office. Almost all records enter GenBank as direct electronic submissions. The information is first reviewed for quality assurance after which new entries are assigned an accession code (a unique identifier – usually a combination of one or more letters and numbers, such as a single letter followed by five digits (e.g. U12345) or two letters followed by six digits (e.g. AF123456)).

Each GenBank record must contain contiguous sequence data from a single molecule type. The various molecule types can include DNA, RNA, precursor RNA, mRNA (cDNA), ribosomal RNA, transfer RNA, small nuclear RNA and small cytoplasmic RNA. GenBank records include fields such as: a brief description of the sequence (source organism, gene name/protein name, function); the accession code; a version number (which also includes a GI code in the same field an identifier for the sequence); and keywords (a word or phrase describing the sequence). Also included are descriptions of the source organism; literature references (authors, titles, journal, etc.); features (i.e. information about genes and gene products and regions of biological significance that can include regions of the sequence that code for proteins and RNA molecules); and a number of other features.

Release 196 (15 June 2013) of GenBank held over 165 million sequences from over 380,000 named organisms at the genus level and below. For example, there were more than 570 complete microbial genomes and more than 190 eukaryotic genome assemblies (including the reference human genome). About 12% of GenBank sequences are from humans. In building SAAPdb (Chapter 3), release 183 (11 April 2011) of GenBank was used which held over 135 million sequences.

The entire GenBank dataset or subsets can be downloaded for local use by file transfer protocol (FTP). These databank subsets include taxonomic categories such as bacteria and viruses. Alternatively, parts of the database can be downloaded based on the sequencing strategy used to obtain the data. These sections include expressed sequence tags (ESTs), genome surveys, high-throughput genomics, high-throughput cDNAs and environmental samples.

2.1.1.2 The European Nucleotide Archive (ENA)

ENA (formerly known as EMBL) records are similar to those held in GenBank. They provide (in addition to an identifier), an accession code, description, keywords, organism source and classification, literature reference information, features and the sequence, and database cross-references. Where appropriate ENA entries are cross-referenced to other

databases such as protein sequence databases [e.g. TrEMBL (Consortium, 2011), UniProtKB/SwissProt (Consortium, 2011)], taxonomy databases [e.g. NCBI Taxonomy (Sayers *et al.*, 2011)], species-specific databases [e.g. FlyBase (Tweedie *et al.*, 2009)] and other specialised data collections [e.g. the Eukaryotic Promoter Database (Schmid *et al.*, 2006), TRANSFAC (Matys *et al.*, 2006)], and the literature [e.g. PubMed (Sayers *et al.*, 2011)]. Downloadable information is structured in a similar fashion to GenBank, and subsets of records can be obtained according to taxonomic categories and sequencing approaches.

The latest release (116) of ENA was created on 02 June 2013 and contains over 309 million sequence entries and over 615 million cross-references, of which over 38 million are to UniProtKB/TrEMBL and more than 634,000 refer to structures in the PDB (Rose *et al.*, 2011). ENA contains approximately 32 million records from humans, a further 46 million from other mammals, nearly 32 million metagenomics-based sequences, and 36 million invertebrate sourced entries. The largest taxonomic category is plants, which contains 78 million nucleotide sequences. As the ENA is synchronised with the NCBI and the DNA Data Bank of Japan, these statistics will be very similar in the corresponding databanks. Most additions to the ENA databank are made through direct submissions from individual researchers, groups, genome sequencing projects, and patent applications.

In the version of ENA used for populating SAAPdb (Chapter 3), the databank (release 107 from March 2011) contained over 206 million sequence entries. Of these, almost 29 million were from humans, and nearly 28 million from environmental samples.

2.1.2 The Universal Protein Resource UniProtKB TrEMBL and UniProtKB/Swiss-Prot

The Universal Protein Resource (UniProt) is a collaborative project involving the European Bioinformatics Institute, the Swiss Institute of Bioinformatics and the United States Protein Information Resource. The aim is to provide a protein sequence and functional information resource[1] (Consortium, 2011). It has been under development since 2002 and contains a number of related resources that are created and maintained in the context of the UniProtKB project.

The UniProt Knowledgebase (UniProtKB) is a curated protein information resource that includes information about function, classification and cross references. UniProtKB is composed of two parts: Swiss-Prot and TrEMBL. TrEMBL is an automatically annotated, un-

[1] http://www.uniprot.org/

reviewed database of protein sequences. Swiss-Prot is a high-quality manually annotated database of protein sequences that have been reviewed by UniProtKB scientists. The UniProtKB Archive (UniParc) is a database used to keep track of sequences and their identifiers reflecting the history of all protein sequences stored in the UniProtKB databases.

TrEMBL is made up of all coding regions in the GenBank/ENA/DDBJ databases (TrEMBL can be loosely thought of as translated ENA), proteins from the literature, and those that have been submitted to UniProtKB, but not yet entered into Swiss-Prot. Automatic annotation uses InterPro (Hunter *et al.*, 2009) classifications of predictive protein signatures. Information is transferred from well-characterised entries in Swiss-Prot, to unannotated entries in TrEMBL. Owing to the high volume of data that are deposited in ENA, TrEMBL sequences are released to the public before their entry into Swiss-Prot. This is to avoid a delay while these sequences are processed by UniProtKB staff, and enable researchers to access the very latest information without compromising the quality of information in Swiss-Prot.

An accession number (AC) is allocated to each sequence upon its addition to UniProtKB. The ACs are a string of six alphanumeric characters (starting with A, P, Q or O) and are stable between database releases and are guaranteed always to refer to that particular protein (although the sequence records may be amended). If several UniProtKB entries are merged into one record or deleted, the ACs of all the previous entries are retained as a *secondary* ACs to the new *primary* AC; each record has one *primary* AC and can also have *secondary* ACs. In the example shown in Figure 2.1, the ID (see below) is P53_HUMAN the *primary* AC is P04637 and the *secondary* ACs are: Q15086; Q15087; Q15088; etc. (the *primary* AC is the first AC provided, see lines #1 and #2 in Figure 2.1). The *primary* AC should be cited if an entry has multiple ACs. When working with UniProtKB/Swiss-Prot data, it is important to ensure data integrity by always using *primary* ACs.

The ID "Entry Name" is another unique identifier that is part of UniProtKB records. Each UniProtKB record is described by *both* an identifier ID and AC. The IDs take 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 occasionally revised) vocabulary of species is described and made available at `http://www.uniprot.org/taxonomy/`. IDs are *not* guaranteed to remain the same and it is sometimes necessary to change IDs (e.g. so that related entries have similar names, or if an entry is promoted from TrEMBL to Swiss-Prot). For example, hen egg white lysozyme changed from LYS_CHICK to LYSC_CHICK while PROC_HUMAN used to refer to pyroline-5-carboxylate dehydrogenase and now refers to Protein C.

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 2.1. Records are separated by a line containing only the string '//' (line #35 in Figure 2.1). A more detailed description of the UniProtKB/Swiss-Prot file format can be found at `http://web.expasy.org/docs/userman.html`.

The DR line of UniProtKB/Swiss-Prot provides cross-references between databases. For example P53_HUMAN is cross referenced to ENA records X02469 and CAA26306.1; PIR records A25224 and DNHU53; RefSeq records NP_000537.3 and NM_000546.4; PDB records 1A1U and other databases entries (lines #15-18 in Figure 2.1).

In May 2011 (the time of the last SAAPdb build), Swiss-Prot (Release 2011_04) contained over 526,000 sequence entries. Of these, more than 20,000 were human sequences, more than 16,000 were from mice and more than 10,000 were from *'Aribodspsis.thaliana'*. The majority of Swiss-Prot sequences (62%) were from bacteria, with almost a third from eukaryotes (32%). Most cross references were to the Gene Ontology GO ontology database (Consortium, 2010), followed by InterPro (Hunter *et al.*, 2009), ENA (Flicek *et al.*, 2011) and Pfam (Finn *et al.*, 2010). There were links to a total of 128 different databases.

In June 2013, at the time of writing this thesis, the number of entries in Swiss-Prot had increased. Swiss-Prot (Release 2013 07 of 26 April 2013) contained over 540,000 sequence entries. The representation of sequences from humans and mice were approximately the same (>20,000 and >16,000, respectively), but the number of entries for *'Aribodspsis thaliana'* had increased by 20% (12,000 entries).

In May 2011, TrEMBL contained over 14 million sequences. The most frequently occurring organism was HIV1. This had over 375,000 records listed; a legacy of the continual resequencing of this intensely studied organism. *'O.sativa japonica'* and *'humans'* were the next most populous contributors with over 95,000 and 85,000 records respectively. The distribution of bacteria was very similar to UniProtKB (64% of the total number of records). Eukaryotes constituted 27% of TrEMBL entries. At that point, TrEMBL held cross-references to 129 other databases; most of these were to InterPro, followed by GO, ENA and Pfam.

As with Swiss-Prot, the passage of time increased the number of entries contained within TrEMBL in 2013 to over 39 million sequences from more than 404,000 species. The most frequently occurring organism was still HIV1, with the number of records increasing from 375,000 in 2011 to 519,000 in 2013. *'uncultured bacterium'* and humans were the next most

populous entries with over 185,000 and 114,000 records respectively. At this time, the occurrence of bacteria was slightly more than in UniProtKB (74% of the total number of records). The contribution of eukaryotes had decreased from 27% in 2011 to 20% of TrEMBL entries in 2013. One fewer cross-referenced database was available in 2013 compared with 2011, but the majority were still InterPro, followed by GO, ENA and Pfam.

The UniProtKB databases can be downloaded using FTP. TrEMBL is available as data subsets structured around taxonomic divisions. The entire UniProtKB database is also available in a standard, text-based format, XML or as a FASTA sequence file.

2.1.3 The Protein Databank (PDB)

The Research Collaboratory for Structural (RCSB) PDB contains information about experimentally determined structures of proteins, nucleic acids and complexes (Rose *et al.*, 2011) and is the largest publicly available repository for 3D data describing biological macromolecules (Berman *et al.*, 2000). Three groups make up the RCSB, namely the University of California in San Diego (UCSD), Rutgers University, and the University of Wisconsin-Madison. The RCSB PDB is a member on the Worldwide Protein Data Bank (wwPDB) which is an organization that maintains the archive of macromolecular structure and acts as a deposition, data processing and distribution centres for PDB data. The wwPDB's mission is "to maintain a single PDB archive of macromolecular structural data that is freely and publicly available to the global community".

The wwPDB was started in 2003 by three members: RCSB PDB, the Protein Data Bank in Europe (PDBe) and the Protein Data Bank of Japan (PDBj). In 2006, the Biological Magnetic Resonance Data Bank (BMRB) joined the wwPDB.

Each member's site can accept structural data and process the data. The processed data are sent to the 'archive keeper' at present a rôle fulfilled by the RCSB PDB. This ensures that there is only one version of the data which is identical for all users. The modified database is then made available to the other wwPDB members, each of whom makes the resulting structure files available through their websites to the public. The member sites are more than just mirrors of the archive keeper, because the members offer different tools on their websites for analysing the structures in the database.

```
1  |  ID   P53_HUMAN               Reviewed;         393 AA.
2  |  AC   P04637; Q15086; Q15087; Q15088; Q16535; Q16807; Q16808; Q16809;
3  |  AC   Q16810; Q16811; Q16848; Q2XN98; Q3LRW1; Q3LRW2; Q3LRW3; Q3LRW4;
4  |  AC   Q3LRW5; Q86UG1; Q8J016; Q99659; Q9BTM4; Q9HAQ8; Q9NP68; Q9NPJ2;
5  |  AC   Q9NZD0; Q9UBI2; Q9UQ61;
6  |  DT   13-AUG-1987, integrated into UniProtKB/Swiss-Prot.
7  |  DT   24-NOV-2009, sequence version 4.
8  |  DT   31-MAY-2011, entry version 186.
9  |  DE   RecName: Full=Cellular tumor antigen p53;
10 |  DE   AltName: Full=Antigen NY-CO-13;
11 |  DE   AltName: Full=Phosphoprotein p53;
12 |  DE   AltName: Full=Tumor suppressor p53;
13 |  GN   Name=TP53; Synonyms=P53;
14 |  OS   Homo sapiens (Human).
   |  ...
15 |  DR   EMBL; X02469; CAA26306.1; -; mRNA.
16 |  DR   PIR; A25224; DNHU53.
17 |  DR   RefSeq; NP_000537.3; NM_000546.4.
18 |  DR   PDB; 1A1U; NMR; -; A/C=324-358.
   |  ...
19 |  FT   CHAIN         1    393       Cellular tumor antigen p53.
20 |  FT                               /FTId=PRO_0000185703.
21 |  FT   DNA_BIND    102    292
22 |  FT   REGION        1     83       Interaction with HRMT1L2.
23 |  FT   SITE        120    120       Interaction with DNA.
24 |  FT   CROSSLNK    292    292       Glycyl lysine isopeptide (Lys-Gly)
25 |  FT                               (interchain with G-Cter in ubiquitin).
26 |  FT   CROSSLNK    386    386       Glycyl lysine isopeptide (Lys-Gly)
27 |  FT                               (interchain with G-Cter in SUMO).
28 |  FT   VAR_SEQ       1    132       Missing (in isoform 7, isoform 8 and
29 |  FT                               isoform 9).
30 |  FT                               /FTId=VSP_040833.
31 |  FT   VARIANT       5      5       Q -> H (in a sporadic cancer; somatic
32 |  FT                               mutation).
33 |  FT                               /FTId=VAR_044543.
   |  ...
34 |  SQ   SEQUENCE   393 AA;  43653 MW;  AD5C149FD8106131 CRC64;
   |       MEEPQSDPSV EPPLSQETFS DLWKLLPENN VLSPLPSQAM DDLMLSPDDI EQWFTEDPGP
   |       DEAPRMPEAA PPVAPAPAAP TPAAPAPAPS WPLSSSVPSQ KTYQGSYGFR LGFLHSGTAK
   |       SVICTYSPAL NKMFCQLAKT CPVQLWVDSI PPPGTRVRAM AIYKQSQHMT EVVRRCPHHE
   |       RCSDSDGLAP PQHLIRVEGN LRVEYLDDRN TFRHSVVVPY EPPEVGSDCT TIHYNYMCNS
   |       SCMGGMNRRP ILTIITLEDS SGNLLGRNSF EVRVCACPGR DRRIEEENLR KKGEPHHELP
   |       PGSTKRALPN NTSSSPQPKK KPLDGEYFTL QIRGRERFEM FRELNEALEL KDAQAGKEPG
   |       GSRAHSSHLK SKKGQSTSRH KKLMFKTEGP DSD
35 |  //
```

Figure 2.1: An example of a UniProtKB/Swiss-Prot record.

The above record is for [UniProtKB:P53_HUMAN/P04637], it has been edited to only include data that is relevant to SAAPdb and FOSTA, i.e. ID (the **i**dentifier), AC (the **a**ccession number), DT (the **d**ate field), DE (the **de**scription field), OS (the **O**rganism field), DR (**d**atabase cross-reference line) and FT (annotated features), and SQ (the **S**equnce field); records are terminated by a //; line numbers are given on the left for text references and '...' are used to indicate skipped lines.

PDB coordinate files contain plain text which describe the 3D coordinates of each atom. Residues are described by a simple annotation of each constituent atom using the same residue ID. PDB files also provide details of the method used to solve the structure, related literature, cross-references to other resources (e.g. UniProtKB), specification of ligands, etc.

Although there is usually only one structure per file, a PDB entry can contain more than one model in a single coordinate file mostly in the case of nuclear magnetic resonance (NMR) entries. Files are structured into a number of distinct sections. The 'Title' section contains fields such as the header, title, compound (description of the macromolecular contents of an entry), source (organism, expression system, etc.), keywords, experimental data, submitter information, a primary literature citation and remarks (experimental details, annotations, comments, etc.). The 'Primary Structure' section contains the sequence of residues in each chain of the macromolecule(s) (or other consecutive chemical components covalently linked in a linear fashion to form a polymer), and a field for database cross-references (e.g. GenBank, UniProt). The 'Heterogen' section contains a description of non-standard residues in the entry. The 'Secondary Structure' section identifies the positions of helices, sheets, and turns found in protein and polypeptide structures. The 'Connectivity Annotation' section states the existence and location of disulfide bonds and other linkages. The 'Crystallographic and Coordinate Transformation' section describes the geometry of the crystallographic experiment and coordinate system transformations (e.g. from the database entry to the submitted entry, transformations expressing non-crystallographic symmetry, etc.). The 'Coordinate' section contains the collection of atomic coordinates and model delimiters. Within this section, the ATOM record defines the atomic coordinates for standard amino acids and nucleotides, and the occupancy and temperature factors for each atom. Chain IDs, residue labels and residue sequence numbers are also given for each atom. Non-polymer or other non-standard chemical coordinates (e.g. water molecules, ligands, etc.) are described in a similar way in the 'Coordinate' section using HETATM (rather than ATOM) records.

The RCSB PDB website provides users with a wide range of content, including information to supplement data provided in flat files. These include links to databases describing the enzyme classification code and associated pathways [the KEGG database (Kanehisa *et al.*, 2010), catalytic sites as described in BioCyc (Caspi *et al.*, 2010), ligands e.g. BindingDB (Liu *et al.*, 2007) DrugBank (Knox *et al.*, 2011), etc.]. Additional annotations are provided in PDB records from external resources such as CATH (Cuff *et al.*, 2011), SCOP (Andreeva *et al.*, 2008), Pfam (Finn *et al.*, 2010), GO (Consortium, 2010) and the Structural Biology Knowledge-base (Gabanyi *et al.*, 2011).

The PDB allows users to find information using text-based keyword searching (such as PDB identifiers, literature or UniProt IDs), chemical components, bibliographic information, homology and browsing (i.e. tree traversal). A recently implemented navigational feature combines searching and browsing. Initial search results can be shown as subsets of hierarchies, which can be browsed and searched again. This allows a query to be refined iteratively, based on new information found during the search.

In May 2011 (the time of SAAPdb build) there were over 72,000 structures in the PDB. The vast majority (67,000) were protein structures. There were also more than 2,000 nucleic acid structures, and over 3,000 protein/nucleic acid complexes. Most structures were from humans (>18,800), followed by *E. coli*, mice, and *S. cerevisiae*.

At the time of writing, in November 2013, the RCSB PDB website showed that the most structures had been solved using X-ray crystallography (more than 75,000, representing 88% of all structures). There were also more than 8,700 NMR structures and a smaller number produced by electron microscopy (approximately 500) (Rose *et al.*, 2011).

2.1.4 The PDBSWS protocol

The PDBSWS protocol links UniProtKB to the PDB (Martin, 2005). Reliable mapping between these databases allows the transfer of UniProtKB annotations to PDB chains and residues. SAAPdb uses the UniProtKB-to-PDB mapping in PDBSWS to map sequence residues to their corresponding structural residues. This mapping is performed automatically using cross-references from the PDB to UniProtKB at the chain level (where available) and, for historical reasons, from UniProtKB to PDB at the whole PDB file (not chain) level where available. PDB chains that have not been assigned a UniProtKB AC and which are not short peptides or nucleotides are searched against UniProtKB using a brute force scan based on the sequence from ATOM records. Although the UniProtKB and PDB databases provide cross-references, PDBSWS provides a more complete and accurate link between the two, encompassing the chain and residue levels. It is regularly updated, and uses a consistent form of link, which is not the case in the native databases.

The mapping is performed in a number of stages and is stored in a PostgreSQL relational database. These stages are as follows.

Stage 1 UniProtKB (UniProtKB/Swiss-Prot and UniProtKB/trEMBL) and the PDB databanks are downloaded by FTP and stored locally.

Stage 2 UniProt data extraction. Data are extracted from the UniProt databank to obtain IDs, sequences and modification dates. Mappings are also obtained between UniProt IDs and ACs, between primary and secondary ACs, and mappings between ACs and PDB IDs where available. The method relies on accurate mapping between both UniProtKB IDs and ACs, and primary and secondary ACs.

Stage 3 PDB data extraction. Data are extracted from the PDB. For each chain, a PDB ID and the sequence is collected, and any links to UniProt IDs are extracted from the `DBREF` field. If no UniProtKB AC reference exists in the `DBREF` field, the `REMARK 999` field is parsed in an attempt to find UniProtKB AC references. Some PDB chains are chimeras, i.e. they are composed of regions from two or more UniProt sequence entries. These are handled correctly if they are annotated as such in `DBREF` records.

Stage 4 Corrections to links from the PDB. Links in the PDB to UniProt that use UniProt IDs have the UniProt ID converted to a UniProt AC, using information previously obtained from UniProt in stage 2. Links from the PDB to obsolete UniProt IDs, other incorrect IDs intended to be used as links to UniProt, and deprecated UniProt accessions are all identified. All of the remaining UniProt accessions that are used as links from the PDB to UniProt, are validated to check that they are correct primary ACs.

Stage 5 Addition of cross-links from UniProt. For historical reasons PDB links to UniProt take precedence over links from UniProt to PDB. Links from UniProt to PDB that were not identified in stage 4 (i.e. not given in the PDB) are now collected. In cases where PDB records have multiple chains, sequence alignments are used to check which of the chains correspond to a UniProt sequence. The UniProtKB sequence is aligned with each PDB chain in turn to identify which chain (or chains) are relevant. Stage 4 and 5 may yield multiple matches as a protein sequence can map to multiple PDB structures, and to several chains within a single PDB structure.

Stage 6 Brute-force scan. A FASTA formatted databank of UniProt sequences is created. It is scanned with the remaining unassigned PDB chains, using the FASTA sequence database search algorithm (Pearson, 1991) to find the remaining valid cross-database links. The PDB sequence is reconstructed from the ATOM records rather than SEQRES. 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.

Stage 7 All the PDB chains that were found to have links are aligned against their corresponding UniProt entry (or entries, in the case of a chimera) using the `ssearch33` program (Pearson and Lipman, 1988). The `ssearch33` algorithm implements the (slow but accurate) Smith Waterman sequence alignment algorithm (Pearson and Lipman, 1988) to generate the PDB-UniProtKB *residue-to-residue* mappings. The resulting alignment is mapped onto the PDB structure and stored.

The most recent version of PDBSWS (March, 2014) contains 260324 PDB chains. Of these, 243615 are protein chains (i.e. excluding short peptides and DNA/RNA chains). There are 67525 cross-links obtained from the PDB, 138882 cross-links obtained from SwissProt and 26119 cross-links obtained from the brute-force scan (i.e. an additional 12.15%). As of March, 2014, 95.45% of PDB protein chains are successfully mapped to a UniProtKB sequence.

PDBSWS is available on the `www.bioinf.org.uk/pdbsws` website. It can be queried using PDB IDs (with or without a chain ID), UniProt accession codes and SwissProt IDs. PDBSWS can be downloaded as a dataset at the chain level or the residue level and can be queried using a REST interface. Mutations in PDB files with respect to the UniProt entries can also be downloaded.

2.1.5 Databases of single amino acid polymorphisms

2.1.5.1 dbSNP

The Single Nucleotide Polymorphism Database (dbSNP) was established in 1998, and is hosted by the NCBI in collaboration with the National Human Genome Research Institute (NHGRI) in the United States (Sherry *et al.*, 2001). It is a database of genetic variations, incorporating information about not only single-base nucleotide substitutions (SNPs), but also short deletion and insertion polymorphisms, multinucleotide polymorphisms, microsatellite markers or short tandem repeats, named variants, invariant regions of sequence and heterozygous sequences. SNPs make up the vast majority of this database (>95%). It provides 'neutral' polymorphisms, mappings to protein sequences and only a few disease-causing clinical mutations. In the analysis of SNP data, these disease-associated mutations are removed from the SNP dataset, but retained in the disease dataset. This is based on the assumption that the large-scale genomic scanning technology that is used to identify SNPs, happens to have sequenced the genome of an individual carrying a disease mutation.

Each variation submitted to dbSNP is assigned a unique 'submitted' SNP ID. In cases where submitted SNPs are identical, they are compiled into one reference SNP cluster, which contains data from each. dbSNP accepts information from other public variation databases such as HapMap (Frazer *et al.*, 2007), individual research laboratories, genome centres and industry. In May 2011, the most recent version of dbSNP was 'build 132', which had been available since September 2010. This release was based on almost 244 million submissions, and contained over 87 million reference SNP clusters. Over 30 million of these were human records, with the rest coming from a diverse range of organisms including *M. musculus*, *G. gallus*, *A. gambiae* and *O. sativa*. Just over 29 million of the reference clusters were known to reside within genes. Of the total number of submissions, over 74 million had a phenotype annotation, and more than 35 million had a frequency provided.

The information in a dbSNP record includes the location of the variant, the flanking sequences around the polymorphism, and data on population diversity including variation and frequency by population or individual genotype. Information about the submitter, experimental conditions and the validation status of the variant is also provided. The validation status describes the evidence that supports a variant. These categories of evidence include multiple independent submissions, frequency or genotype data, a submitter confirmation, observation of all alleles in at least two chromosomes, genotyping by HapMap, or that the SNP has been sequenced by the 1000 Genomes project.

dbSNP links to other types of biological databases. These databases include GenBank, various genome databases, the 1000 Genomes project (Durbin *et al.*, 2010), Ensembl (Flicek *et al.*, 2011), RefSeq (Pruitt *et al.*, 2009), PubMed (Mouillet, 2008), OMIM (Amberger *et al.*, 2011), UniGene (Schuler, 1997) and dbSTS (Olson *et al.*, 1989).

The dbSNP database can be downloaded using FTP and is available in multiple formats including a flat file version of the database, a relational database dump, and FASTA, ASN.1 and XML formatted files.

2.1.6 OMIM and LSMDBs

Information about pathogenic deviations (PDs) in SAAPdb is obtained from a variety of sources, of which the Online Mendelian Inheritance in Man (OMIM)[4] database is the largest contributor (Amberger *et al.*, 2011). The Mendelian Inheritance in Man (MIM) catalogue describes human genetic disorders, and was initiated by Dr. Victor McKusick at John Hopkins

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

University. It was first published as a book in 1966 (McKusick *et al.*, 1992), which in 1998 was in its 12th edition. An online version was published in 1985 and is hosted at the NCBI.

The contents of the OMIM database is based on an examination of published literature. The contents of peer-reviewed journals are scanned to identify relevant articles. Particular attention is paid to disease phenotypes, genes with novel biology and genes that do not appear in OMIM. Other online genetics resources are also checked for information and articles that may be relevant. A team of science writers and editors identify, discuss and write up relevant articles, update existing MIM records and create new ones.

The OMIM database is focused on phenotypes and the genes associated with them. In May 2011, it contained more than 20,000 records describing either genes or phenotypes and over 7,000 of these contained a description of a phenotype. OMIM records may include: a detailed description of the gene or phenotype, clinical information (features, synopsis and management), biochemistry, inheritance patterns, map locations, pathogenesis descriptions, diagnosis information, genotype/phenotype correlations, population genetics, molecular genetics, animal models, cloning information, gene names, the gene structure, gene functions, an evolutionary background, allelic variants, polymorphisms, cytogenetic and citations.

Every record in OMIM is assigned a six-digit MIM identifier. The first digit represents the method of inheritance: 1 indicates that the trait is autosomal dominant[2]; 2 denotes autosomal recessive[2]; 3 is X-linked loci or phenotypes; 4 is Y-linked; 5 is mitochondrial; 6 or above is autosomal[3].

Six symbols can precede the six-digit MIM number:

(∗) indicates that a gene annotation may exist.

(#) indicates a descriptive entry (usually of a phenotype), which does not represent a unique locus.

(+) indicates a gene of known sequence and a phenotype.

(%) indicates that the entry describes a confirmed Mendelian phenotype or phenotypic locus for which the underlying molecular basis is not known.

(No symbol) indicates a description of a phenotype for which the suspected Mendelian basis has not been clearly confirmed.

(^) indicates the entry no longer exists.

[2]entries created before May 15, 1994

[3]entries created after May 15, 1994

When the molecular basis for a phenotype is understood, allelic variants are added to the gene entry. Each is assigned a unique four-digit number that is added to the MIM number of its parent entry, with a decimal point between the MIM number and the four-digit number. For example, mutations in the cystic fibrosis transmembrane conductance regulator gene [CFTR (MIM number 602421)] are indicated using 602421.0001 to 602421.0136. In most cases, only certain allelic variants are included, e.g. the first mutation to be discovered, distinctive phenotypes, etc. Most of the variants that are stored represent disease-producing mutations. A few polymorphisms are also included, many of which show a positive correlation with particular common disorders. Because not all allelic variants are described, links to complementary resources are given, including the Human Gene Mutation Database (Stenson *et al.*, 2009), and over 1,500 locus specific mutation databases via the Human Genome Variation Society[5] (Oetting, 2011). In April 2011, over 2,500 gene entries in OMIM contained information about disease-causing mutations. In addition to providing links to a range of external genetics databases, the OMIM database provides links to RefSeq, GenBank, UniGene, Pubmed, and many other resources internal to the NCBI.

However, given that the described mutations are derived from multiple sources and 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. The Martin group automatically maintains SAAPdb using an internal version of OMIM with corrected numbering (this will be discussed in Chapter 3).

As genomic sequencing is becoming cheaper and more reliable, the number of pathogenic deviation identifications is increasing exponentially. Figure 2.2 shows the increase in the content of OMIM. In ten years, the number of disease mutations increased from ~8000 in 1998 (McKusick, 1998) to almost 20,000 in 2008 (Amberger *et al.*, 2009, Figure 2.2). OMIM has over 18,000 allelic variants distributed among 2,494 genes and associated with 4,218 different disorders or susceptibilities (Amberger *et al.*, 2011).

Although OMIM is a rich source of disease-associated information, which can be used to carry out extensive bioinformatic analysis, the pathogenic deviation dataset from OMIM is enhanced by the inclusion of eleven other specialised locus-specific mutation databases (LSMDBs), mutation datasets that are produced and maintained by research groups interested in particular diseases. These resources potentially provide large quantities of high-quality data (George *et al.*, 2008). These specialised resources often hold detailed phenotypic information concerning aspects such as enzymatic function or prognosis. The bioinformatic

[5] http://www.hgvs.org/

analysis of these data may demonstrate effects on protein function that would otherwise be difficult to detect. While this has not been addressed in this thesis, rather than training classifiers on binary classifications (disease causing or neutral), these methods could be trained to predict disease severity.

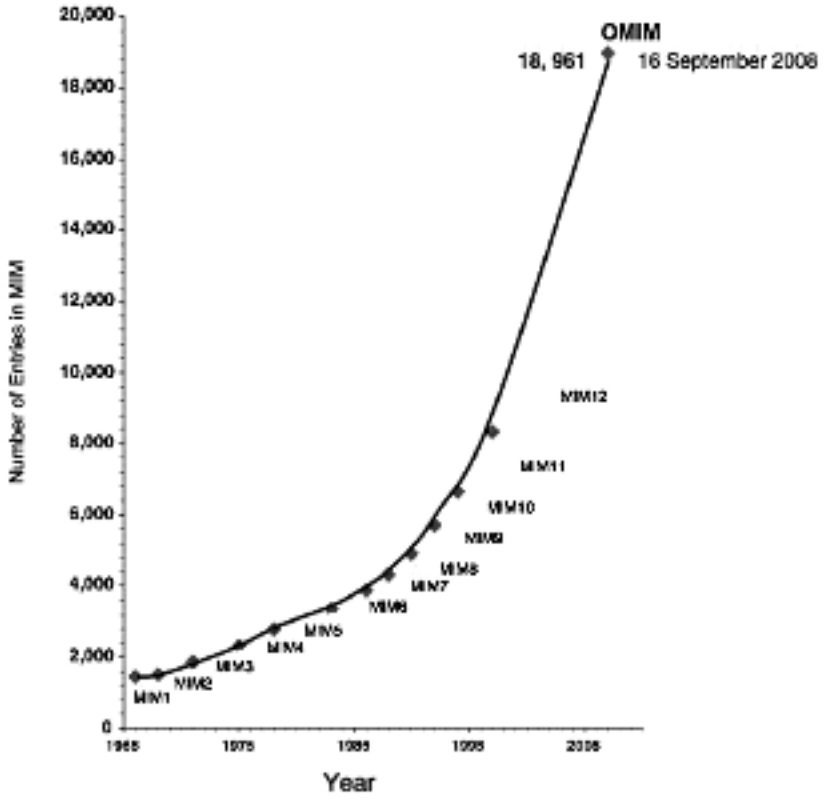


Figure 2.2: (O)MIM growth since 1965.
The size of the printed MIM versions (1-12) are marked with diamonds (Amberger *et al.*, 2009).

A brief description of the eleven LSMDBs as they were at the time of the SAAPdb build (June 2011) is provided below. These LSMDBs were selected and integrated into SAAPdb based on the interests of collaborating groups.

ADAbase [6]

ADAbase is a mutation registry for adenosine deaminase (ADA) deficiency (OMIM:608958) (Piirilä *et al.*, 2006). In June 2011, it contained 72 records. It is maintained at the University of Tampere in Finland. ADA deficiency accounts for about half of the autosomal recessive forms of severe combined immunodeficiency (SCID). In addition to immunological defects, most patients with ADA deficiency have skeletal abnormalities.

[6] http://bioinf.uta.fi/ADAbase/

ALSoD1db [7]

Amyotrophic lateral sclerosis (ALS, also called Lou Gehrig's disease) is a type of motor neuron disease caused by the degeneration of motor neurons (OMIM:147450). Mutations in the gene encoding Cu/Zn superoxide dismutase (SOD1) have been identified in patients with a familial form of ALS. The ALS Online Database (ALSoD) was created to store information about mutations in SOD1 (and now other ALS-related mutations), along with ALS patient information. In May 2011, it recorded 303 mutations in 74 ALS related genes (sporadic and familial) and is maintained at King's College London. It is a complete record of all genotype/phenotype and neutral variations and includes genetic, proteomic, and bioinformatics information associated with the disease. It also contains detailed clinical information, neuropathology data, literature information and data analysis of Genome Wide Association Studies (GWAS).
The PD information extracted for SAAPdb was obtained from the SOD1 mutation records in ALSoD (Abel *et al.*, 2012; Wroe *et al.*, 2008).

G6PDdb [8]

Glucose-6-phosphate dehydrogenase (G6PD) deficiency is an X-linked recessive trait caused by abnormally low levels of G6PD (OMIM:305900) (Beutler *et al.*, 1968). It is characterised by the abnormal breakdown of red blood cells (haemolysis), usually after exposure to certain medications, foods or infections. It affects approximately 400 million people. The G6PD Database integrates mutational and structural data from various genetic and structural databases (Genbank, Protein Data Bank, etc.) and information from the literature. The G6PDdb resource was developed in a collaboration with the Martin group (Kwok *et al.*, 2002) and contained 193 records.

ZAP70base [9]

Zeta-chain-associated protein kinase 70 or ZAP-70 is a 70kDa protein-tyrosine kinase. ZAP-70 is normally expressed in T cells and natural killer cells, and is involved in T-cell signalling. ZAP-70 deficiency (OMIM:176947) is a rare autosomal recessive form of severe combined immunodeficiency. ZAP70base is a mutation registry for ZAP70 deficiency produced at the University of Tampere, Finland (by the same group that maintains ADABase) (Piirilä *et al.*, 2006). In June 2011, it contained 17 records. These included descriptions of alleles, citations, diagnosis information, patient information and other clinical data.

[7] http://alsod.iop.kcl.ac.uk/Als/index.aspx
[8] http://www.bioinf.org.uk/g6pd/
[9] http://bioinf.uta.fi/ZAP70base/index2.html

HADB (HAMSTeRS) [10]

Haemophilia A (HA) is an X-linked hereditary disease, and is the most common form of haemophilia. HA is caused by reduced activity or the amount of factor VIII (OMIM:306700). This protein serves as a cofactor in the coagulation cascade. Deficiency produces clots that take longer to coagulate and are unstable. HADB is a Haemophilia A Mutation Database in which HADB mutations are taken from the literature and electronic submissions. HADB focuses on point mutations, insertions and deletions. It also provides predicted splicing errors and polymorphisms. In January 2007, the Imperial College-run database contained over 1,200 mutations, which had been collated from peer-reviewed literature and electronic submissions (Kemball-Cook *et al.*, 1998).

IARC TP53 [11]

The IARC TP53 Mutation Database is maintained by the International Agency for Research on Cancer (IARC) in Lyon, France (Olivier *et al.*, 2002; Petitjean *et al.*, 2007). P53 regulates the cell cycle and is a tumour suppressor protein (OMIM:191170), in humans encoded by the TP53 gene. A mutation to P53 occurs ~50% of all human cancers (Greenblatt *et al.*, 1994; Sidransky and Hollstein, 1996; Lane and Fischer, 2004). The IARC TP53 database contains all TP53 mutations published in the literature. It includes information about the functional impact of mutations, characteristics of tumours, and demographic data about patients. The initiative provides a variety of information, including a somatic mutation dataset, a germline mutation dataset, polymorphisms, two function datasets, cell-line data, P53 protein structures and a dataset on mouse-models.

The somatic mutation information contains P53 mutations associated with human cancers. Three types of data are provided in relation to mutations: type, prevalence, and prognostic value. In June 2011, the latest release contained over 27,000 mutations. Many of these are repeats of the same mutation in different patients.

The germline (or inherited) mutation information contains data on families that have Li-Fraumeni syndrome (a rare autosomal cancer family syndrome syndrome caused by mutations in the TP53 gene). In June 2011, it included 588 mutations affecting 584 families or individuals.

[10] http://hadb.org.uk/

[11] http://www-P53.iarc.fr/

KinBase [12]

KinBase (Manning *et al.*, 2002) contains information about more than 3,000 protein kinase genes from humans and other organisms (mouse, fly, worm, etc.). The database is maintained by the Salk Institute in San Diego. This highly populated family of proteins is involved in many crucial cellular processes such as signal transduction, cell-cycle regulation and tumourigenesis. As a consequence of these rôles, kinases have been reported to be associated with inherited developmental and metabolic disorders (Lahiry *et al.*, 2010) and also many types of cancer. Due to this association, they are considered as potential targets for therapy (Izarzugaza *et al.*, 2009).

The set of mutations in kinase domains was provided by collaborators (Izarzugaza *et al.*, 2011); a 66 KinbaseDriver (protein kinase domain) of which 26 mapped to protein structure and 66 KinbasePassenger (protein kinase domain) of which only 14 mapped to protein structures.

LDLR FH Database [13]

Familial hypercholesterôlemia (FH) is most commonly a result of variations in the LDLR gene, which encodes the receptor for low density lipoprotein (LDL) cholesterol particles. About 1 in 500 people are affected by pathogenic alterations in the LDLR peptide. These cause increased atherosclerosis and a greater risk of coronary heart disease (Leigh *et al.*, 2008).

In June 2011, the LDLR FH Database at University College London listed 1,741 LDLR allelic variants (1,122 of which were unique). These had been obtained from the literature and included 1,280 DNA substitutions (73.5%), 75 insertions (4.3%), 337 deletions (19.4%), 64 duplications (3.7%), 15 insertion/deletions (0.9%) and 2 inversions (0.1%).

OTC

Ornithine transcarbamylase (OTC) deficiency, is an X-linked disorder caused by mutations in the OTC gene (ornithine carbamoyltransferase, OMIM:300461). It causes hyperammonemia, which is an excess of ammonia in the blood (Gilbert-Dussardier *et al.*, 1996). Although it is a rare metabolic disorder, it is the most common inherited defect in ureagenesis, affecting about 1:16000 children. The OTC dataset used in SAAPdb (Tuchman *et al.*, 2002; Yamaguchi *et al.*, 2006) is produced at the University of Minnesota, with the last major update in 2006. This contained 341 mutations, and an additional 29 non-disease-causing mutations and polymorphisms. Enzyme activities and clinical information are also included.

[12] Greenman C et al. Patterns of somatic mutation in human cancer genomes.Nature 2007, 446(7132):153-8.

[13] `http://www.ucl.ac.uk/ldlr`

PAHdb [14]

Deficiency of phenylalanine hydroxylase (PAH) enzyme function causes hyperphenylalaninemia (HPA) and related forms of phenylketonuria (PKU). PAHdb is a database of mutations in the human PAH gene, and associated phenotypes at the levels of protein, metabolites and organism. This is combined with information about associations of mutations with populations, haplotypes and other features. PAHdb is maintained at McGill University in Montreal (Scriver *et al.*, 2003; Pey *et al.*, 2007).

Mutation data were collated from both published articles and personal communications from 82 investigators from the PAH Mutation Analysis Consortium in 32 countries. In June 2013 the PAHdb held records of 567 mutations (>60% were missense). The alleles are annotated with information such as species, locus, gene, unique identifier number, name and source of information and are flagged as either pathogenic or polymorphic.

STAT3

'Signal transducer and activator of transcription 3' (STAT3), is a human transcription factor. STAT3 is essential for the differentiation of TH17 helper T cells that have been linked with a variety of autoimmune diseases. Loss-of-function mutations in the STAT3 gene lead to hyperimmunoglobulin E syndrome. This syndrome is associated with recurrent infections, and poor development of bones and teeth (Frank, 2007).

Forty-nine mutations in SwissProt entry "P40763" were collated from peer-reviewed literature and electronic submissions and integrated into SAAPdb.

Other sources of mutation data have been considered including HGMD[15] and SwissProt Variants (SwissVar)[16]. However HGMD data are only available to registered users meaning that we have not been able to reproduce their data in our database and SwissVar is not terribly reliable in annotation of disease status (For example, known PDs in G6PD are annotated as 'Natural Variants' of unclassified disease status).

2.1.7 FOSTA

When examining sequence conservation as indicator of the effect of a mutation, it is crucial that the sequences that are aligned have the same function. By definition, proteins that diverge in function will undergo change in functionally critical residues.

[14] http://www.pahdb.mcgill.ca/
[15] http://www.hgmd.cf.ac.uk
[16] http://swissvar.expasy.org

Functional Orthologs from Swiss-Prot Text Analysis[17], FOSTA, is a relational database of automatically generated families of annotated functionally-equivalent proteins (McMillan and Martin, 2008). FOSTA identifies a list of homologues for a given human protein, based on a BLAST search of the database. It then carries out text analyses of the annotations. First this seeks to identify a match in the protein identifier part of the entry name, then the EC number, and finally by matching synonyms at various levels of granularity from the description (DE) field. If a non-human homologue passes any of the three filters, it is marked as a **functionally-equivalent protein**, FEP, and added to the FEP family with the human protein.

McMillan and Martin (2008) showed that Swiss-Prot functional annotation can produce excellent results; they also identified cases where FOSTA correctly assigned FEPs a questionable functional annotation, and others where several families shared the same entry name protein identifier. Examples of such inconsistencies can be found in the HOX proteins and the PROC_HUMAN example in McMillan and Martin (2008). FOSTA is preferred over standard lists of orthologs when highly reliable data are required. This is because when very distant orthologs are gathered using traditional methods, they may diverge in function (e.g. because of mutations in functional residues).

As part of this project, the existing FOSTA code was improved by removing all 'hard-coded' variables (e.g. paths or user-names) into a separate Perl module. This means that running FOSTA on a new machine (or as a new user) requires no editing of the main code, all changes being done in this single module. A Perl script was also written to look at each of the database's tables and give summary counts for the tables, to make it easier to compare different FOSTA runs. Additionally, full instructions on how to run FOSTA from scratch were written; this document has been reviewed and tested by another person within the group to ensure its practical usefulness.

2.1.8 Databases of single amino acid polymorphisms used for prediction work

Section 1.1 showed that there are many ways to divide mutations into types and subtypes. This thesis focuses on protein-level variations and only considers SAAPs: substitutions of one amino acid in the protein sequence at a time. Depending on the effect on the phenotype, SAAPs can be divided into neutral and pathogenic. The main data sources of pathogenic mutations are described in Section 2.1.6, and dbSNP has been presented as a resource that

[17] http://www.bioinf.org.uk/fosta/

provides data on neutral or low-penetrance SAAPs in Section 2.1.5.1. These were all accessed via the SAAPdb resource described in Chapter 3.

In addition, the HumVar and HumDiv dataset `V.2.1.9` was used as a resource. These datasets were developed for evaluation of PolyPhen2[18]. HumDiv consist of 5564 deleterious and 7539 neutral mutations from the same set of 978 human proteins. HumVar consists of 22196 deleterious and 21119 neutral mutations in 9679 human proteins, with no restriction on deleterious and neutral mutations coming from same proteins.

HumDiv, was compiled from all damaging alleles with known effects on molecular function causing human Mendelian diseases, present in the UniProtKB database. These were grouped together, with differences between human proteins and their closely related mammalian homologs of human proteins ($>= 95\%$ sequence identity), assumed to be non-damaging. These assumptions are questionable given the paper on (Compensated Pathogenic Deviations) CPDs (Baresic *et al.*, 2010). HumVar consists of all human disease-causing mutations (except cancer mutations) or mutations resulting in loss of activity/function from UniProtKB. Common human nsSNPs (minor allele frequency $>$ 1%) without annotated involvement in disease, which are treated as non-damaging are also included.

2.2 Data handling

In a large-scale automated system (such as SAAPdb) data integrity must be ensured by appropriate and robust data handling. The system has to handle vast quantities of information, and it must be possible quickly to retrieve and process it. This section describes the fundamental data handling methods: relational databases (Section 2.2.1), XML (Section 2.2.2) and an alternative XML/ASN.1-based representation of the PDB, XMAS (Section 2.2.3).

2.2.1 PostgreSQL relational databases

The first relational database was developed by Edgar Codd at IBM Almaden Research Centre (Codd, 1970). A relational database simply stores information. It consists of tables (or 'relations') which describe the types of data. Tables in turn contain columns or 'fields' which hold records, i.e. the actual data.

[18] `ftp://genetics.bwh.harvard.edu/pph2/training`

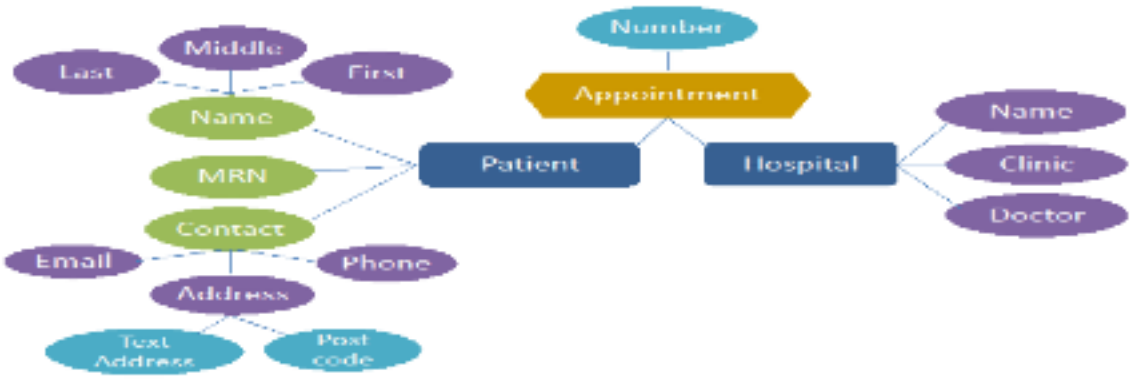
The power of relational databases lies in their ability to 'relate' or join data held in different tables based on a common identifier (the 'foreign key'). This key is recorded in one table and refers to data in other tables. This concept makes it possible to 'normalise' large tables that contain many fields into smaller data structures describing individual concepts.

The first step in constructing a good relational database system is to start with a clear specification. This involves defining the problems and constraints (i.e. any issues or limitations the system will have to handle), the objectives (what the system is going to do and how), and the scope and boundaries (i.e. the information that will be stored). Once the specification is in place, logical, conceptual, and physical design steps follow. The database is created in a database management system (DBMS) and datasets are loaded. Finally, the database is tested and evaluated against the initial specification. Maintenance and evolution are important considerations for any database. These activities fix problems with the system or implement enhancements or new requirements.

The design phase decomposes the problem into its constituent 'entities', 'relationships' and 'attributes' to produce a high-level model of the database structure. Entities describe distinct objects in the dataset. Combining entities using relationships creates entities that are more abstract. Furthermore, both entities and relationships can have attributes that describe the corresponding object. To represent high-level data models, 'entity-relationship' (ER) modelling diagrams clearly define the entities and relationships in the data to be stored.

In this section, a dummy dataset is used to illustrate the main concepts concerning a relational database. The dataset describes hospital doctors, where patients can have multiple appointments at different hospitals with different doctors or clinics. Figure 2.3 describes the entities, relationships and attributes in this example. There are two entities: a patient and a hospital. These entities are joined by the relationship 'appointment' which captures the more abstract or 'associative' appointment entity. Both entities and relationships can have attributes: a patient has a name, contact details and a MRN [Medical Record Number]; a hospital has a name, doctors and clinics; the number of appointments booked at each hospital defines an appointment. In addition, each entity is given a unique identifier (ID). This enables each individual entity (i.e. each patient or hospital) to be uniquely identified.

If attributes have multiple values, it is useful to decompose them into two or more attributes. Relationships between entities are defined with respect to their 'cardinality'. This describes how entities are related to each other. The cardinality may be many-to-many, one-to-many or one-to-one.



(a) An example entity relationship (ER) diagram.
Entities are in square-edged rectangles while attributes are in oval boxes; lines join entities and relationships to their attributes.

Patient	Contact	Hospital	Num
Height K Douglas	7 Victoria Road LIONACLEIT, HS7 7DS , 07927526572, Kerher@aol.com	(UCLH,Card,Dr. Laura) (CMH,Sur,Dr. Khan)	1, 1
Abun L Sreur	67 Stamford Road ANGSRGL, TA3 9QY, 02373649827, Abun@mail.com	(UCLH,Inr,Dr Ivan)	2
Kelrin S Heald	67 Holgate Rd, RASKELF, YO6 1XH, 07927526572, Kel@yahoo.com	(UCLH,Inr,Dr Ivan)	1
Joseph M Connolly	31 Emerson Road KINLOSS, 02 1555 2743, Joseph@Cn.com	(KGH-HC,Car,Dr. Mahab)(KGH-HC,Sur,Dr. Ivan)	1, 1
Marta M Connolly	31 Emerson Road KINLOSS, 07 2223 6571, Marta@Cn.com	(KGH-HC,Car,Dr. Mahab)	1

Patient

ID*	First	Mid	Last	Address^	Contact^
1	Height	K	Douglas	2	1
2	Abun	L	Sreur	4	2
3	Kelrin	S	Heald	1	2
4	Joseph	M	Connolly	3	4
5	Marta	M	Connolly	3	5

Hospital

ID*	Name	Clinic	Doctor
1	KGH-HC	surgical clinic	Dr. Ivan
2	KGH-HC	cardiology clinic	Dr. Mahab
3	CMH	surgical clinic	Dr. Khan
4	UCLH	cardiology clinic	Dr. Laura
5	UCLH	cardiology clinic	Dr. Tom
6	UCLH	immediate clinic	Dr. Ivan

Address

ID*	Post Code	Text Address
1	YO61 3LH	67 Holgate Rd, RASKELF
2	HS7 7DS	7 Victoria Road LIONACLEIT
3	IV36 1LU	31 Emerson Road KINLOSS
4	TA3 9QY	67 Stamford Road ANGSRGL

Contact

ID*	Phone	E-mail	Address^
1	07927526572	Kerher@aol.com	2
2	07526726267	Kel@yahoo.com	1
3	02373649827	Abun@mail.com	4
4	02155527643	Joseph@Cn.com	3
5	07222326571	Marta@Cn.com	3

Appointment

ID*	Patient^	Hospital^	number
1	1	4	1
2	1	3	1
3	2	6	2
4	3	6	1
5	4	2	2
6	4	1	1
7	5	2	1

(b) An example relational database.
The data to be represented is a list of patients appointments, shown in the top half of this diagram; these data can be decomposed into smaller entities (Patient, Hospital, Appointment) and stored in separate tables as shown in the bottom half of the diagram; primary keys are annotated with an asterisk (*), foreign keys are annotated with a caret (^); primary keys in the Patient, Contact, Address and Hospital tables are highlighted with the same colours used in the Delivery table to indicate where these primary keys are used as foreign keys.

Figure 2.3: Using a patient and hospital information to illustrate database design.

Once the Entity-Relationship (ER) diagram is complete, the design of the database is determined by the application of various rules. The most relevant rules are: (i) each entity is represented by a table; (ii) each many-to-many relationship is represented by a table; and (iii) any multiple attributes that have dependencies between sub-attributes should be factored out into another table. The resulting database and its relationship to the original data are shown in Figure 2.3(b).

A fundamental concept in relational databases is the primary and foreign keys. Primary keys are IDs that allow each entry in a table to be identified uniquely. Often they are arbitrary numbers applied to data as they are entered into the database. However, how these keys are allocated is debated by database designers, some of whom argue for the use of real data, particularly if the entry already has a unique identifier (such as SwissProt primary accession codes). Foreign keys are references to primary keys found 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 (^). Furthermore, all foreign keys and the data to which they refer are highlighted with the same background colour to make it easier to identify inter-table references. In a well-designed database the use of foreign keys improves data integrity and facilitates administration, as changes only need to be made in one table.

Additional constraints on the contents of fields in a table can improve data integrity and performance. For example, they can define whether a field must be unique, whether data must exist (i.e. the field cannot be 'null'), or the range of values the field may take. Finally, indexing vastly improves the performance of relational databases. Indexing leads to the generation of a secondary table that enables rapid look-up of the original data. Any field (or combination of fields) this is frequently used should be indexed. Both FOSTA 2.1.7 and SAAPdb (Chapter 3), make extensive use of indexes as without them, the manipulation of very large datasets quickly becomes impractical.

Once the design is in place, the database is built, populated, and queried using structured query language (SQL). Foreign keys are implemented to retrieve related data by 'joining' tables based on a common term or terms. A sample query is shown in Figure 2.4. This requests the total number appointments booked at each hospital. This query demonstrates the basic `SELECT/FROM/WHERE` SQL grammar. It also shows how `GROUP BY` and `ORDER BY` are used to aggregate and sort data, and illustrates the use of `SUM()`, one of the many built-in, standard SQL functions. PostgreSQL also allows the user to define new functions.

Normalisation is the process of efficiently managing and organizing data in a database into smaller data structures describing individual concepts. This involves the elimination of redundant data matches requirements for the first normal forms (1NF) and ensuring that only related data are stored together. The NFs in relational database theory are a series of guidelines written to ensure that databases are normalized to ensure that data are logically stored and provide criteria for determining a table's degree of immunity against logical inconsistencies and anomalies. The lowest form of normalization (1NF) requires that duplicate columns are removed from the same table. The second normal form (2NF) continues to address the problem of duplicated data and requires the creation of relationships between tables containing subsets of data and their parent data tables. The third normal form (3NF) matches requirements of (2NF) and must remove all columns that are not dependent on the primary key. Additional level of normalization are sometimes used.

```
=> SELECT h.Name, h.Doctor, SUM(a.Number)
   FROM   Hospital h, Appointment a
   WHERE  a.Hospital = h.ID
   AND    h.Clinic = 'cardiology clinic'
   GROUP  BY h.Name
   ORDER  BY h.Name;

   Name    | Doctor     | sum
-----------+------------+----
 KFSHRC    | Dr. Moheb  | 2
 UCLH      | Dr. Laura  | 2
 UCLH      | Dr. Tom    | 1
(3 rows)
```

Figure 2.4: An example PostgreSQL query.
Two tables (`Hospital` aliased to `h` and `Appointment` aliased to `a`) are joined on `a.Hospital` and `h.ID`; the data are constrained to those `Hospital/h` with Clinic (`h.Clinic = 'cardiology clinic'`); the aggregate function `SUM` is calculated for each `h.Name` as defined by the `GROUP BY h.Name` clause; results are sorted by h.Name as defined by the `ORDER BY h.Name` clause.

2.2.2 XML

XML (eXtensible Markup Language) is a self-descriptive standard mark-up language used to structure, transmit and store data. XML allows the user to define a personalised set of specific tags and document structure. A Document Type Definition (DTD), which may be implemented in XML-DTD or in XML schema, defines a restricted grammar consisting of elements and attributes that the user can use to define a specialised framework for the representation and storage of their data. The DTD can be declared inline (in an XML document)

or be provided as an external reference. XML data are stored in plain text format, which is independent of any particular software or hardware. Consequently, it is much easier for different applications to share data. When data are represented in a consistent XML format, the same parser can extract specific data required for processing, database population, etc.

Figure 2.5 is an example of an XML file that illustrates its hierarchical structure. The 'mutation' element contains two instance of the element 'protein_data' and 'amino_acid' which followed by multiple item's that contain further subelements ('aa_label', 'wildtype', etc.). Figure 2.6 shows how the corresponding DTD defines such a framework.

```
<?xml version='1.0'?>
<!DOCTYPE ladb SYSTEM 'definition/ladb_xml.dtd'>

<ladb name='OMIM' url='http://www.bioinf.org.uk/omim/omim_sprot.csv'>

<mutation id='100650' supplementary_id='0001' arbitrary_id='0' number_of_records='1'>
<protein_data ac='P05091'>
<amino_acid aa_label='504' wildtype='E' mutant='K' valid='t'>504</amino_acid>
</protein_data>
</mutation>

<mutation id='100690' supplementary_id='0001' arbitrary_id='0' number_of_records='1'>
<protein_data ac='P02708'>
<amino_acid aa_label='262' wildtype='N' mutant='K' valid='t'>262</amino_acid>
</protein_data>
</mutation>

<mutation id='100690' supplementary_id='0002' arbitrary_id='0' number_of_records='1'>
<protein_data ac='P02708'>
<amino_acid aa_label='201' wildtype='V' mutant='M' valid='t'>201</amino_acid>
</protein_data>
</mutation>

<mutation id='100690' supplementary_id='0003' arbitrary_id='0' number_of_records='1'>
<protein_data ac='P02708'>
<amino_acid aa_label='299' wildtype='T' mutant='I' valid='t'>299</amino_acid>
</protein_data>
</mutation>

<mutation id='100690' supplementary_id='0004' arbitrary_id='0' number_of_records='1'>
<protein_data ac='P02708'>
<amino_acid aa_label='198' wildtype='G' mutant='S' valid='t'>198</amino_acid>
</protein_data>
</mutation>

</ladb>
```

Figure 2.5: An example of XML, from storing mutation data.
See Figure 2.6 for the corresponding DTD.

```
<!ELEMENT ladb ( mutation+ ) >
<!ATTLIST ladb name CDATA #REQUIRED >
<!ATTLIST ladb url CDATA #REQUIRED >

<!ELEMENT mutation ( dna_data?, protein_data, occurrence?, patient_data*, references? ) >
<!ATTLIST mutation id CDATA #REQUIRED >
<!ATTLIST mutation supplementary_id CDATA #IMPLIED >
<!ATTLIST mutation arbitrary_id CDATA #REQUIRED >
<!ATTLIST mutation number_of_records CDATA #REQUIRED >

<!ELEMENT dna_data ( gene?, dna_base?, codon? ) >
<!ELEMENT gene ( #PCDATA ) >
<!ELEMENT dna_base ( #PCDATA ) >
<!ATTLIST dna_base wildtype CDATA #REQUIRED >
<!ATTLIST dna_base mutant CDATA #IMPLIED >
<!ELEMENT codon ( #PCDATA ) >
<!ATTLIST codon wildtype CDATA #REQUIRED >
<!ATTLIST codon mutant CDATA #IMPLIED >

<!ELEMENT protein_data ( amino_acid ) >
<!ATTLIST protein_data ac NMTOKEN #REQUIRED >
<!ELEMENT amino_acid ( #PCDATA ) >
<!ATTLIST amino_acid aa_label CDATA #REQUIRED >
<!ATTLIST amino_acid wildtype CDATA #REQUIRED >
<!ATTLIST amino_acid mutant CDATA #REQUIRED >
<!ATTLIST amino_acid valid (t|f|?) #REQUIRED >

<!ELEMENT occurrence ( prevalence_text?, prevalence_count?, prevalence_percentage? ) >
<!ELEMENT prevalence_text ( #PCDATA ) >
<!ELEMENT prevalence_count ( #PCDATA ) >
<!ELEMENT prevalence_percentage ( #PCDATA ) >

<!ELEMENT patient_data ( age?, sex?, race?, external_factors?, phenotype? ) >
<!ELEMENT age ( #PCDATA ) >
<!ELEMENT sex ( #PCDATA ) >
<!ELEMENT race ( #PCDATA ) >
<!ELEMENT external_factors ( #PCDATA ) >
<!ATTLIST external_factors details ( f|t ) #IMPLIED >

<!ELEMENT phenotype (disease_name?, enzyme_activity?,
disease_severity?, disease_class?, disease_onset?, prognosis?,
delta_delta_gee?, melting_point?)>
<!ATTLIST phenotype mendelian ( dominant|recessive|partial|non ) #IMPLIED >
<!ELEMENT disease_name ( #PCDATA ) >
<!ELEMENT enzyme_activity ( #PCDATA ) >
<!ATTLIST enzyme_activity numeric ( 0|1|2|3|4|5 ) #IMPLIED >
<!ATTLIST enzyme_activity percentage CDATA #IMPLIED >
<!ELEMENT delta_delta_gee ( #PCDATA ) >
<!ELEMENT melting_point ( #PCDATA ) >
<!ELEMENT disease_severity ( #PCDATA ) >
<!ATTLIST disease_severity numeric ( 0|1|2|3|4 ) #IMPLIED >
<!ELEMENT disease_class ( #PCDATA ) >
<!ELEMENT disease_onset ( #PCDATA ) >
<!ATTLIST disease_onset numeric ( 1|2|3|4 ) #IMPLIED >
<!ATTLIST disease_onset age CDATA #IMPLIED >
<!ELEMENT prognosis ( #PCDATA ) >

<!ELEMENT references ( citation+ ) >
<!ELEMENT citation ( #PCDATA ) >
<!ATTLIST citation year CDATA #REQUIRED >
```

Figure 2.6: An example of DTD from storing mutation data.

This DTD specifies the format for storing mutation data: which elements with which attributes can exists, what the relationship between elements is, and whether data is required or may be omitted.

2.2.3 An alternative format for the PDB: XMAS

XMAS files represent PDB data using a hybrid XML/ASN.1 format (XMAS comes from the first two letters of XML and ASN.1 and stands for eXtensible Markup with Abstract Syntax). The XMAS format was developed by Dr Andrew Martin while working at Inpharmatica to overcome the problem of PDB files, which do not provide a standard format for adding information (e.g. accessibility, H-bonding, secondary structure assignments etc.). PDB files require a large amount of data cleaning and contain a lot of implicit information (e.g. H-bonding, sequence alignment between SEQRES and ATOM records), that is not explicitly stated and must be calculated for each individual PDB structure.

XMAS files are automatically generated for all new or updated PDB structures and are used extensively in SAAPdb. This is because it is essential to have a standardised format, which can be easily parsed by our structural analysis system. These files must contain all the required PDB data and the results of other calculations. When necessary, XMAS-formatted structures can be easily generated for additional structures using proprietary software.

The conversion from PDB to XMAS format is as follows. Note that, once converted to XMAS format, the following steps can be preformed in any order as the format is self-describing annotating the columns in which the additional data are stored.

1- PDB data: Convert raw PDB data to XMAS format, preforming various data clean-up.

2- Solvent accessibility: Calculate and add atom and residue solvent accessibility statistics using Lee and Richards' (1971) method.

3- Secondary structure: Calculate and add secondary structure assignments for each residue using Kabsch and Sander's (1983) method.

4- Hydrogen bonds: Identify and add any hydrogen bonds in the structure (i.e. protein-protein, protein-ligand and ligand-ligand hydrogen bonds) using the simple Baker and Hubbard (1984) criteria for defining a hydrogen bond.

In addition, the hydrogen bond calculation program identifies and annotates non-bonds and pseudo-hydrogen bonds and annotates them. Non-bonds are non-consecutive residue atom pairs 2.7-3.35Å apart that are not covalently or hydrogen bonded, for example, electrostatic interactions and Van der Waals contacts. Pseudo-hydrogen bonds are atom pairs that satisfy the constraints described in Baker and Hubbard (1984) for hydrogen bonding but one or both atoms do not form strict hydrogen bonds, for example, they are metal ions.

2.3 Machine learning

2.3.1 Introduction

Machine learning is a sub-division of artificial intelligence and aims to train computers to adapt to certain responses and initiate actions (Zhang and Rajapakse, 2009). Machine learning has applications that range from search engines, natural language processing, bioinformatics, medical diagnosis and chemoinformatics to stock market analysis, game-theory and computer vision.

The field of machine learning has grown rapidly, requiring the development of powerful learning algorithms for diverse applications. Machine learning has been a key technique for data mining; the discovery of previously unknown properties in data, which has led to the creation of sophisticated database interfaces. According to Witten and Frank (2005), data mining has three purposes: to understand, explain (in human-readable terms) or predict data features. Machine learning approaches have been used effectively to solve many technological problems and greatly increase the knowledge-acquisition process (Frasconi *et al.*, 2003).

While data are accumulating at a faster rate than ever, data storage is becoming cheaper and more accessible. This has made it possible for the field of knowledge acquisition to expand, to collect data on various naturally-occurring processes and other aspects of human activity. However, vast quantities of data are not useful in their own right, they must be interpreted and learned from to be of any use. Currently, our ability to analyse such large data sets lags behind the rate of data accumulation (Witten and Frank, 2005). Machine learning addresses exactly this issue, allowing the identification of structure in unstructured data either automatically or semi-automatically.

Machine learning is founded on the idea of instances. Typically, each instance has a unique identifier, supplemented by a set of measurable attributes (also termed features). Attributes are assumed to contribute new knowledge to the description of the concept. Each attribute is assigned a value, which can take two forms: it is termed numerical if it can be expressed on a numerical scale; or categorical if it can be defined by a finite set of mutually exclusive categories. These categories can be numerical, but not continuous (for example, binary attributes with possible outcomes of 0 or 1), or a non-numeric description (e.g. helix, strand, coil). These are termed nominal attributes.

2.3.2 Machine learning approaches

Machine learning deals with programs that improve or adapt their performance over time and can be achieved in several ways. The two approaches (learning scenarios) to machine learning that are most frequently applied are termed 'supervised' and 'unsupervised' (Zhang and Rajapakse, 2009).

In a supervised learning approach, a program performs a function (e.g. classification) after training on a data set where inputs (attributes and class values) and desired outputs are provided. This is followed by thorough testing of its accuracy and efficiency on an independent dataset of instances where the attribute values are known, but the class value is not known. As such, the program is fed examples (input attributes) and must predict the output attribute of every next instance based on pre-defined criteria before the answer is revealed by a 'teacher'. Thus, supervised approaches are often also referred to as discrimination or prediction classifications. If the output is categorical, the process is called classification and the attribute predicted by the model is termed the class attribute. All of the models described in chapter 6 and chapter 7are based on categorical outputs. If the outcome is continuous rather than categorical, and the model can be formalised as a numerical function of input variables, it is called a regression model.

In unsupervised learning, the program must determine certain regularities or properties of the instance in the absence of a teacher. Thus, unsupervised learning focuses on relationships between attributes, rather than trying to predict outcomes. In contrast with supervised learning, there is no test dataset, the class labels of the data are unknown, and the output of the model is trains instances that are grouped according to a similarity measure. The main types of unsupervised learning methods are association learning and clustering. Association rules-mining finds associations or a structure among attributes in large sets of data items. Association rules are an essential data-mining tool for extracting knowledge from data, made useful by the good scalability characteristics of the algorithms employed. Clustering on the other hand, aims to generate groups (clusters) of instances without necessarily identifying the underlying structure or associations of attributes within a cluster. Clustering is one of the most utilised data mining techniques and is useful when it is necessary to train a classifier with a small number of samples if the labelling of a large collection of samples is costly.

There are several other types of machine learning approaches. These include semi-supervised learning (learning based on a mixture of labelled and unlabelled instances) and reinforcement learning (where the inputs come from an unpredictable environment, not a training set, and positive feedback is given after a short sequence of learning steps). A description of these methods is outside the scope of this thesis but are described in detail by Mitchell (1997) or Alpaydin (2009).

This thesis uses binary classifications (values can only be 1 or 0) to predict the presence or absence of a single feature. In this case, the feature of a pathogenic deviation (PD) can either exist or not exist for a given instance (i.e., is/is not a PD). The work presented in this thesis led to the construction of two mutation pathogenicity/phenotype classifiers, which are described in detail in chapter 6 and chapter 7. This thesis uses two forms of machine learning (neural network and random forest) which are described below.

2.3.3 Neural networks

There are two ways to understand neural networks. The first is the traditional biological concept associated with the neural system. The second describes interconnected artificial networks that are built according to the principles of biological networks.

The multilayer perceptron model is an example of an artificial neural network model that can carry out concept classification tasks. It is a feed-forward network (i.e. nodes are connected in a non-circular fashion) that builds the class prediction function by training the network on a back propagation algorithm. It minimises learning errors by adjusting the weight of the connections between the network's nodes (Rumelhart *et al.*, 1986). The model consists of input, output and one or more intermediate layers and explains the flow of data from the input to the output layer. The model can be applied to various pattern ranges for classification, prediction, recognition, and approximation.

A minimum representation of the network consists of three layers of interconnected nodes (also termed neurons or perceptrons) with weighted connections. Figure 2.7 shows the architecture of the model, which is divided into the input layer (one neuron for every attribute and in the case of a neural network, these are exclusively numerical), the (usually single) hidden layer, which consists of a user-defined number of hidden nodes, and the output layer, which has a node for every class category. One of the drawbacks of the models is that the data structures they learn, although efficient at prediction, cannot always be easily translated into human-readable terms. While it is useful as a 'black box' prediction model it is not trivial to visualise (unlike rule-based trees).

In mathematical terms, a multilayer perceptron is a function that maps input values to the output class value. Every node transforms input based on a non-linear activation function (Equation 2.1). The model iteratively learns the weights that minimise the error rate on the given instances (the training set). The aim is to reach a global (rather than a local) minimum error rate.

$$O = f(\sum I_i W_i) \tag{2.1}$$

Where I is an input, W is a weight learned during training, and $f()$ is the evaluation function.

Local minima are a problem that can be avoided through the introduction of 'momentum'. This is a small amount of random noise introduced into the system in each epoch (training step). Finding the appropriate ratio between the learning rate and momentum is the key to optimising the model and achieving a good level of both generalisation and specialisation.

The stopping conditions for training are specified by the user, either by defining the number of epochs, or when the error rate has not changed for the last n epochs. Although in theory there is no limit to the number of iterations (this allows the model to sample error space around the minimum) in practice the process is usually stopped soon after the learning rate reaches a plateau. This is to avoid over-fitting the model to the training data (especially when the training set is small), a process that is also known as 'early stopping'.

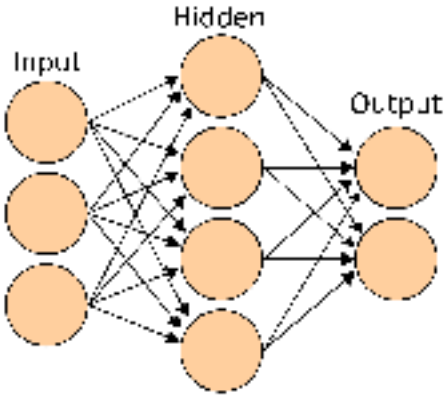


Figure 2.7: Multilayer perceptron schema.
Nodes are organised into three layers: input, hidden, and output layer. The weights on the connections between the nodes are optimised. *Figure obtained from* `http://en.wikipedia.org/wiki/File:Artificial_neural_network.svg` *under Creative Commons license.*

2.3.4 Random forests

The classification of datasets with multiple attributes can be problematic. Typically, a method must be applied to reduce the number of dimensions, as most models do not perform optimally when highly interdependent or irrelevant attributes are mixed with those that are informative. Attribute optimisation is resource-intensive and the standard way to handle it is to use decision trees. In a decision tree algorithm, each iteration evaluates all possible splits of all attributes in order to determine the condition that maximises information gain (Witten and Frank, 2005). This recursive procedure is repeated for every node created by the split, until there is no further information gain or the maximum tree size is reached.

Although single decision trees can perform robustly on high-dimension data, they are often inaccurate when trained on a small dataset and various authors have suggested an obvious improvement. This consists of constructing a set of T trees (usually termed a forest) rather than a single tree. Results generally outperform the single decision tree method (Svetnik *et al.*, 2003) as the final classification is a combination of the predictions made by each tree (often simply a majority vote). The random forest has been shown to be the most efficient method for learning a solution to a problem and can also be used to identify interactions between variables (Pavlov, 2000). In this case, it learns by 'bagging' a decision tree that has not been modified or pruned and randomly identifies features in every split, which are used in the construction of a group of decision trees with controlled variation. The random forest has become a common data-exploration method that represents the combination of individual learning decision trees.

Advantages and disadvantages

The random forest method builds an accurate learning algorithm; consequently the classifier is also accurate. Moreover, it performs efficiently on very large databases and it can handle a large number of input variables with no deletion. It is easy to estimate missing data, and is accurate even if there is a lot of missing data. It provides an estimate of variable types that can be used in the classification and it can provide an internal indiscriminate bound on the generalization error as the forest is built. It provides effective methods for error balancing in unbalanced data sets. The forest can compute prototypes, which makes the relationship between classification and variables easy to identify. The ability to compute proximity between pairs of cases makes it easy to cluster and locate outliers.

Algorithm

A random forest builds a user-specified number of trees (T). Each tree is trained and evaluated on a bootstrapped sample of the initial dataset (i.e. a sample $n_i \leq n$ of the data with replacement gives us $(1 - 1/e)n$ of the original data ($\sim$63%)), while the remainder of the data makes up the 'out-of-bag' dataset. The number of trees is limited by the available computational power; in practice, the more trees a model can produce, the better the performance. Moreover, as the Law of Large Numbers (Breiman, 2001) established that there is an upper bound for the generalisation error, adding trees to the random forest does not lead to over-fitting.

When the decision tree is built, node splits are based on a randomly-chosen subset of m_{try} attributes[19] (sometimes referred to as the random tree algorithm). In this respect, the random forest resembles the bagging algorithm (Breiman, 1996). A split is based on all p attributes, with a clear improvement in performance when $m_{try} < p$. The tree continues to be built until there are no further information-gaining splits and no pruning. Finally, the model is applied to the set of out-of-bag instances and performance is recorded as the 'out-of-bag' (OOB) error (i.e. the $\sim$37% not in each bootstrap set).

According to Breiman (2001), a low m_{try} suggests low correlation between trees (i.e each tree explores a different region of feature space). At the same time, each tree provides less information as it covers a narrower range of attributes in each split. Increasing m_{try} leads to more similar trees, but each tree provides a more accurate prediction. Consequently, optimum performance results from optimizing the value of T and m_{try} ($1 \leq m_{try} \leq p$) (Svetnik *et al.*, 2003).

It is only in extreme cases that the optimal number of trees in a random forest depends on the number of predictors. Despite the official description of the algorithm, which states that the random forest does not overfit and the number of trees is unimportant, at least one author (Segal, 2004) has demonstrated that it could over-fit noisy datasets. There are various methods to obtain the optimal number of trees. In this thesis a simple approach of trying a range of T and m_{try} was used.

[19] m_{try} stands for the number of randomly chosen attributes in every split; T is the number of trees

Performance

Svetnik *et al.* (2003) suggest that random forests have several advantages over single decision trees:

(i) Every tree-split uses a subset of parameters, which significantly reduces the time needed to build the tree;

(ii) There is no need for time-consuming cross-validation, which is avoided by bootstrapping and evaluating the method on the OOB dataset (usually $1/3$ ($\sim 37\%$) of the training dataset). It performs significantly better (in terms of speed) than bagging and some decision trees; and

(iii) Tree-building is simplified by omitting pruning.

Not only are random forests resource-efficient when run on large datasets with many attributes, they perform as well as boosting (Meyer *et al.*, 2003) and decision forests (Tong *et al.*, 2003) and can outperform bagging. Finally, when used as an 'off-the shelf' method with only two parameters, the method is simple to implement. In this case, the number of attributes tested during tree-building and the number of trees can be set as high as computing resources permit.

The method also provides a measure of the importance of each training attribute. Once the tree is created, the misclassification rate for an attribute in the OOB set can be calculated by randomising the attribute's values (Breiman, 2001). The difference between the misclassification rate and the OOB error gives the raw importance of the attribute.

2.3.5 Data sampling

In data sampling, the overall dataset (known attribute and class values) is divided into training and test datasets. However, this has to be done with care. If too many data points are used for training, the model may be excellent, but the test dataset might not be representative, giving a misleading impression of poor performance. In the opposite case, while the model may not be robust owing to the lack of training data, testing will be very thorough. The optimal balance is achieved by iteratively using all instances for both training and testing in a process called cross-validation. Data are divided into N (usually 3, 5 or 10) non-overlapping equal subsets. N models are built, each time using a different fold (iteration) for testing, and all other folds are merged for training. Cross-validation is the average of the scores from all iteration.

An extreme example of cross-validation is the **leave-one-out process** (also known as jack-knifing). In this case, the number of iterations (folds) is equal to the number of data points. In each step all but one of the instances are used for model building and tested on one data point. The drawback of this procedure is that it is very resource intensive, and therefore it only makes sense to use it on a very small sample taken from a small dataset used to build something like a specific disease predictor (Chapter 7).

Both cross-validation and leave-one-out validation are examples of data sampling without replacement. This means that once an instance sampled from the pool of instances, it is removed, and cannot be sampled again. In contrast, data sampling with replacement, also called **bootstrapping**, always leaves the instance in the original pool and simply copies it to the test dataset. In this way, each sampled instance is chosen from the original N instances, which allows repeated sampling of the same instance. Sampling with replacement is performed M times on the dataset of N instances included in the test set. To use a simple example, sampling without replacement would be like dividing a group of children into two football teams, whereas sampling with replacement would be like drawing the names of children winning a prize from a hat, and then returning the name back into the hat, so the same child can win more than once.

The other important classification issue is the ratio of data points in each of the classes. This ratio has to be maintained throughout all partitions of training and testing datasets, in order to avoid creating unbalanced models. For example, if random data partitioning ends up with all instances with one class value in the test set, and only the other class value in the training set, the model will simply predict the latter class value in 100% of cases.

Finally, when the sample is small, the model should perform equally well on the entire population, even if some of the patterns present in the overall population are not present in the training data. Although performance is measured during training, it is more important that the model is a good predictor of future data. If the classifier is over-fitted, it will perform misleadingly well in training, but will perform poorly on slightly different instances.

2.3.6 Missing data

It is often the case that the values for the attributes of some data points are not known. For example, an error could have occurred in the measuring process (e.g. an instrument malfunction), or it did not make sense to record the measurement for a certain data point (e.g. a patient's condition was too severe to perform an expensive test which would have not

helped in their treatment). In these cases, it is usually not possible to repeat the measurement, and modelling must be carried out with missing values.

There are three main strategies for handling missing data: (i) removing a data point, (ii) creating a new category (if the feature is nominal) or, (iii) inputting the value from data points with known values for that attribute. More details of these three strategies can be found in Witten and Frank (2005) or Saar-Tsechansky and Provost (2007). Removing the data point makes sense if the dataset that remains is not so small that it seriously affects the model's performance. For example, Chapter 6 shows that the training dataset was big enough to use this procedure. In the second case in Chapter 7 (when the missing attribute is nominal), a new attribute category ('missing value') was created. This can work in cases where significant bias is not introduced by equating all instances with the missing value. Finally, there are several ways to predict the most likely value for the instance. For a review of missing data imputation, see for example Jerez *et al.* (2010).

2.3.7 Model evaluation

The aim of classification is to use known data to build a model that is able to sort new instances into the correct class. Here, we use the example of the binary classification of a data point. By definition, the test instance has a known true class value (positive or negative), and a predicted class value (again, positive or negative). The four possible combinations of these values are shown in Table 2.1. An instance with a positive class value can correctly be classified as positive (a true positive, TP), or wrongly as negative (a false negative, FN). On the other hand, an instance with a negative class value can correctly be classified as negative (a true negative, TN), or be wrongly labelled as positive (a false positive, FP).

		Predicted class	
		Positive	Negative
Actual class	Positive	TP	FN
	Negative	FP	TN

Table 2.1: Outcomes of a two-class prediction, also termed a confusion matrix.

The 'success' of the classification, or how 'correct' it is, is relative and depends on the purpose of the model. For example, when used as a diagnostic tool, it is very important not to misclassify a positive as a negative (identifying true positives and avoiding false negatives). In the case of the prediction of the structural effects of mutations on protein stability (presented in Chapter 6), it is important that true positives are identified, however, it is acceptable for a (small) proportion of true positives to be missed. This shows that there are various ways to measure model performance. The terms accuracy, precision, sensitivity, and specificity detailed below are all useful factors on which to measure model performance. Table 2.2 defines a number of measures used to assess binary classification predictions and lists the range of values each measure can take.

Accuracy is also termed the 'overall success rate' and measures the proportion of correctly predicted cases compared to all cases. This is in contrast to the error rate, defined as 1−accuracy.

Precision indicates how many instances predicted to be positive really are. In other words, it reflects how likely it is that the model will record a false positive.

Sensitivity indicates the fraction of actual positives identified. It is crucial to avoid low sensitivity when using models for medical research as missing an existing disease could have fatal consequences.

Specificity has the same meaning for true negatives as sensitivity does for true positives.

The **F-measure** is the harmonic mean between precision and sensitivity: it is usually calculated from the equally weighted contribution of the two. While it is a more general measure of accuracy than the first four measures, it does not take account of true negatives. Therefore a more appropriate general performance indicator is the **Matthews correlation coefficient** (MCC). This shows how well the predicted class correlates with the actual class (-1 indicates an inverse correlation, 0 shows no correlation, 1 indicates a positive correlation). The MCC is the only metric that combines all four measures from Table 2.1 into a single value.

Model performance is generally improved by higher values of precision and accuracy and lower error rates. However, when the model is optimised, a performance trade-off usually has to be made between sensitivity and error rates. This is achieved by experimenting with various attribute combinations and adjusting the model parameters until the desired correctness is achieved.

Table 2.2: Binary classification performance measures.

Name	Formula	Range of values
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	$[0, 1]$
Precision (positive predictive value (PPV))	$\frac{TP}{TP+FP}$	$[0, 1]$
Sensitivity (True Positive Rate (TPR), coverage or recall)	$\frac{TP}{TP+FN}$	$[0, 1]$
Specificity (True Negative Rate (TNR))	$\frac{TN}{TN+FP}$	$[0, 1]$
False positive rate	$\frac{FP}{FP+TN} = 1 - \text{specificity}$	$[0, 1]$
F-measure	$\frac{2TP}{2TP+FN+FP}$	$[0, 1]$
Matthews correlation coefficient	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(FP+TN)(FN+TN)}}$	$[-1, 1]$
Root mean squared error	$\sqrt{\frac{\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}{n}}$ *	N/A**
Mean absolute error	$\frac{\sum_{i=1}^{n}\lvert\hat{x}_i - x_i\rvert}{n}$	N/A**

* $\hat{x}_i$ and x_i are the predicted and the actual class values for the i-th instance, respectively

** the scale of values depends on the scale of the numerical class value

There is one more performance measure available in the case where a classifier ranks the outcome, or assigns probabilities or confidence values. This is the receiver operating characteristic (ROC) curve, and the corresponding **area under the curve** (AUC). If the true positive rate is plotted against the false positive rate, the learned model can be compared with the performance of a random model, i.e. a predictor that randomly outputs a class value, regardless of the input value. As shown in Figure 2.8, a random model has AUC = 0.50 (the area under curve 'A'). A perfect predictor with a zero error rate would have AUC = 1.00 (curve 'D' is closest to this ideal scenario), with a true positive rate of 1 for all false positive values. However, researchers such as Hand have expressed doubts about comparing classifiers on the basis of the AUC, as each ROC curve is the result of a different misclassification metric (2009). Therefore a mixture of performance characteristics should be used to evaluate a model and particular attention should be paid to any suboptimal model behaviours that must be tolerated when it is applied in practice.

For models such as neural networks that have numerical outputs (unlike categorical classifiers such as Random Forests), three more performance measures can be applied: root mean square error (RMSE), and mean absolute error (MAE).

Root mean square error (also called the **mean square error**) is the square root of the variance of residuals. The difference between the expected and observed value for each data point is squared, averaged and a square root is calculated.

Mean absolute error is the averaged sum of absolute errors, which are calculated as the absolute difference between the predicted and observed class value for a data point.

2.3.8 Benchmarking

The assessment of data sampling and evaluation strategies must focus on the future performance of the model based on training and testing on a limited set of instances. Benchmarking assumes that testing is independent and transparent. Usually several similar models are tested on a new dataset. Although in theory benchmarking is seen as essential (particularly when the performance of one method is tested against another) it is rare that it happens in practice. This is because it requires a great deal of effort, computing time, and most importantly, a dataset that is both appropriate to the task and has not yet been used.

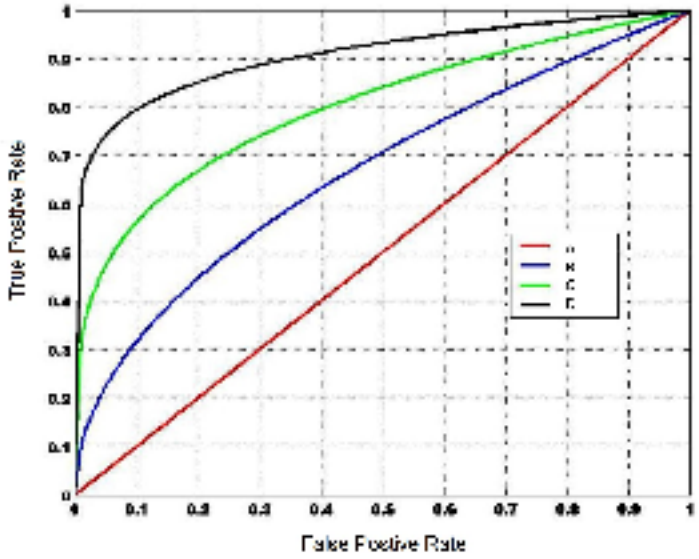


Figure 2.8: Receiver operating characteristic curve.
A is a random model, **B**, **C** and **D** show increasing improvement over random prediction. *Figure obtained from* (Kabari and Nwachukwu, 2012), *with minor modifications.*

2.4 Statistics and data representation

This section describes the basic statistical concepts and tests used in this thesis. It begins by outlining log ratios, as a way to compare datasets *graphically* (rather than statistically). The χ^2 and Fisher's exact test are used to test categorical data for differences in frequency distributions (usually two datasets are tested for the presence or absence of a single feature). A t-test is applied to two populations where features can be measured on a continuous scale; it tests the significance of the difference in the means of the two samples.

2.4.1 Log ratios

Log ratios compare the observed prevalence of a feature with its expected prevalence, as shown in Equation 2.2. A value of 0 indicates that the observed and expected values are the same. For a log_n (i.e. log to n), a value of 1 indicates that the observed value is n times n^1 what is expected, a value of 2 indicates that the observed value is n^2 times what is expected, etc. Similarly, a value of -1 indicates that the observed value is $1/n$ what is expected, etc.

$$logratio = \log_2 \left(\frac{observed}{expected} \right) \tag{2.2}$$

Log ratios are not a statistical test (from which a p-value can be derived). Rather they are a form of descriptive statistics that represent the difference between an observed value and an expected value.

2.4.2 χ^2 test

The Chi-squared test (χ^2 test) (Mood *et al.*, 1974) is a goodness of fit test. This nonparametric test is used on nominal, categorical data and compares the frequency distribution of a sample to a theoretical distribution. It can also be used as a test of independence to compare two samples. In this case the null hypothesis is that data are drawn from the same frequency distribution. Data are divided into n datasets, and k outcome categories with two constraints. First, outcome categories must be mutually exclusive and second, the frequency probabilities for a given dataset, over all categories must sum to 1. On the basis of this definition, the test has $(n-1)\ (k-1)$ degrees of freedom and is calculated as follows:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \text{other end} \tag{2.3}$$

Where O_i is the observed count and E_i is the expected count.

The χ^2 test assumes that the sampled data conform to the χ^2 distribution, which is a special case of the gamma distribution. However, this assumption introduces significant errors when expected counts of five or less appear in a 2×2 contingency table: it increases the χ^2 value and consequently erroneously decreases the p-value. This problem can be partially overcome by the introduction of the 'Yates correction' for continuity (Yates, 1934). This involves subtracting 0.5 from the difference between the observed and expected value in order to increase the p-value, but the procedure can result in an over-correction. Fisher's exact test (described in Section 2.4.3) is the only way completely to overcome assumptions about the distribution of the tested data, but cannot be applied to large datasets.

Where data consist of nominal counts, the χ^2 test (Mood *et al.*, 1974) can be used to indicate whether there is a difference between two datasets. It should be noted that where this thesis reports χ^2 results with percentages, raw counts were used in the χ^2 test. Equation 2.3 shows

how χ^2 statistic is calculated. Although expected values are not always available, they can be estimated from observed data. Wherever possible, this thesis uses known, expected values, rather than estimated values. In other words expected values are calculated from known data rather than estimated from observed data.

Throughout this thesis, the χ^2 test is Yates-corrected in cases where it is carried out on a 2×2 contingency table.

2.4.3 Fisher's exact test

The χ^2 test becomes unreliable where the contingency table is sparsely populated (i.e. where any cell has value of ≤ 5) and where counts are unevenly distributed. On the other hand, the theory behind Fisher's exact test makes it possible to make a robust comparison of datasets of different sizes and can be used to analyse contingency tables with empty cells (Fisher, 1935).

Fisher's exact test (Fisher, 1935) is used instead of a χ^2 test when counts ≤ 5 or when empty fields occur in a 2×2 contingency table. This test provides an *exact* p-value, and removes the difference between the sampling and theoretical χ^2 distribution for small datasets. The `fisher.test` provided in `R` was used with default parameters.

For the example, labelling the counts as shown in Table 2.3 gives the p-value:

$$p = \frac{(x_A + x_B)!(y_A + y_B)!(x_A + y_A)!(x_B + y_B)!}{N!x_A!x_B!y_A!y_B!} \tag{2.4}$$

Table 2.3: Fisher's exact test.

	A	B	
Set X	x_A	x_B	$x_A + x_B$
Set Y	y_A	y_B	$y_A + y_B$
	$x_A + y_A$	$x_B + y_B$	N

A significant limitation of Fisher's exact test is calculation complexity. Large datasets mean that it soon becomes unfeasible to calculate the p-value.

2.4.4 T-test

T-test is a statistical test that measure the significance of the difference in the means of two normally-distributed populations. Although t-tests are often assumed to be synonymous with Student's t-test, in strict terms, Student's t-test assumes that the variances of the two populations are equal. However, Markowski and Markowski (1990) have shown that if two samples are roughly the same size, Student's t-test will still yield accurate results, regardless of differences in the variance between samples. Finally, if the two populations differ in terms of both variance and the size of the dataset, Welch's t-test can be used (Welch, 1947). This calculates the t-statistic (based on the null hypothesis that the means of the two samples are equal) as follows:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \tag{2.5}$$

Where $\bar{X}$ is the mean of the sample, s^2 is the sample variance and N is the number of data points. In this case, the degrees of freedom cannot be calculated and must be approximated using the Welch-Satterthwaite equation (Equation 28 in Welch (1947)). This approximation of the degrees of freedom is based on the linear combination of the degrees of freedom from each of the sample's variances, which is not directly linked to sample size. The `t.test` implemented in the `R` language was used, which by default is equivalent to the two-sided Welch's t-test.

2.4.5 R

The R[20] statistic and data representation system is a powerful programming language and environment for statistical computing and graphics. It is licensed under the GNU license and provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques. Many support packages are also available to end users.

Throughout this thesis, R was used in combination with some additional extra packages, e.g. gplots (Gregory *et al.*, 2010) and plotrix (Jim *et al.*, 2009) to create statistical plots, and HeatMaps matrix.

[20] `http://www.r-project.org` (R Development Core Team, 2008)

2.4.6 PyMOL

PyMOL (Schrödinger, LLC, 2010) is a molecular graphics program[21]. It provides effective visualisation of protein and ligand interactions and can be used to create figures via programming scripts that can be saved in a text file. Throughout this thesis, the PyMOL program was used to create protein structure figures.

2.5 WEKA

Waikato Environment for Knowledge Analysis (WEKA)[22] is a machine learning / data mining software package developed in Java. The main features of the WEKA software are numerous data processing tools, learning algorithms and evaluation methods such as classification and regression, clustering, association rules mining, feature selection algorithms and data visualization techniques.

The software is associated with an extensive Graphical User Interface (GUI) consisting of four major WEKA applications: (i) Explorer (provides an environment to explore the data), (ii) Experimenter (provides a platform to carry out experiments and statistical tests / analysis among various learning algorithms), (iii) KnowledgeFlow (almost has the same function as the Explorer but with a drag-and-drop interface in addition), and (iv) Simple CLI (a simple command-line interface that supports direct execution of WEKA commands). The GUI is the starting point to launch any of the WEKA applications and related tools.

While working with WEKA, the first task is the presentation of data to the software, which is the primary step for data investigation. This step is known as the pre-processing of data, and may be carried out under the Explorer option. In addition, the other tasks are Pre-process (selection of data file), Classify (training/testing of data for classification/regression algorithms), Cluster (data clustering), Associate (discovery of association rules), Select attributes (significant features selection in the data), and Visualize (data visualization in a 1D/2D shape).

WEKA works with flat text files (rectangular table format) that include its own ".arff" (Andrew's Ridiculous File Format), CSV format and C4.5 file formats. A URL or SQL database can also be used as a source for reading data into WEKA.

[21] http://pymol.org
[22] http://www.cs.waikato.ac.nz/ml/weka/index.html

WEKA contains numerous pre-processing tools known as "filters". These filters are widely used for discretization, normalization, transforming, attribute selection, and attribute merging, etc. 'Classifiers' are the actual models used for prediction in WEKA. The learning algorithms used for building such models are decision trees, support vector machines, Bayes' nets etc. Bagging, boosting, stacking, error-correcting output codes and many more are termed as Meta-classifiers.

In a dataset, some features are more important than others. WEKA can use attribute selection mechanism searches to identify those attributes/features, which are more significant and vital for accurate prediction. WEKA has the capability to visualize the data with any number of attributes. To evaluate the performance of different learning algorithms used for classification, prediction and regression problems, exhaustive experimentation is often the best choice. WEKA provides various evaluation options such as cross-validation, learning curve etc.

2.6 Available computational tools to predict damaging mutations

Single nucleotide polymorphisms (SNPs) account for the majority of genetic variation in the human population (Wang *et al.*, 1998). Much of this variation is benign, especially when mutations are synonymous. However, the majority of monogenic (single gene) diseases are mediated by single, non-synonymous base changes (Human Gene Mutation Database (HGMD)). The availability of large-scale, high-throughput SNP genotyping is rapidly increasing the amount of available SNP data. Interpretation of these data in terms of relevance to human disease states requires the modelling of associations between SNP genotypes and resultant phenotypes. Many research groups have developed different predictors for evaluation of the disease-causing potential of DNA sequence alterations.

Many different tools currently exist that predict whether a mutation that changes an amino acid within a protein is likely to increase disease susceptibility or is considered benign. Such tools include MutationAssessor, PolyPhen (1 and 2), SIFT, Condel, FATHMM, V2alignGVGD, Bongo, CanPredict, LS-SNP/PDB, MAPP, nsSNP Analyzer, Panther, Parepro, PhD-SNP, PMut, SNAP, SNPs3D, topoSNP, and others. Below, some of the most common tools are described followed by a summary table 2.4 .

2.6.1 MutationAssessor

MutationAssessor[23] is an online tool for the assessment of evolutionary conservation of amino-acid residues in a protein family. It uses a multiple sequence alignment (MSA) to calculate a functional impact score using the conservation of the region in relation to protein homologues (Castellana and Mazza, 2013). This tool classifies its output into two categories, 'Neutral' or 'Damaging' (Reva *et al.*, 2011). The inputs to MutationAssessor are Uniprot and Refseq protein sequences IDs, which allows users to input mutated sequences. The server then defines the boundaries of the given domain together with the mutated input sequence. It then builds an MSA using Uniprot sequences or those already present in its database, and attempts to distinguish between functional and non-functional mutations in the conserved regions.

MutationAssessor uses numerical estimates to assess the functional impact of a mutation. The estimate is based on a statistical model that displays the similarity of the given sequence to a family of related proteins. The model also makes the assumption that important residues are conserved in the region, which is generally the case in biologically essential genes throughout evolutionary history (Reva *et al.*, 2011). All non-viable mutations are discarded from the analysis. The numerical estimate of functional impact is calculated from the difference in the entropy caused by the occurrence of a particular mutation compared with the entropy of the native structure. This is described using the equation below that calculates a specificity score (S) (Reva *et al.*, 2011):

$$\Delta S_i^v(\alpha \rightarrow \beta) = -ln\frac{n_i(\beta)+1}{n_i(\alpha)} \tag{2.6}$$

Where $n_i(\beta)$ is the number of residues of type β in an alignment column i; $n_i(\alpha)$ is the number of residues of type α in an alignment column i.

A combinatorial score, called the Functional Impact Score (FIS), which assesses the impact of changing an amino acid of type α to type β at a position i, is given by Equation 2.7, ΔS_i correspond to evolutionarily selected specificity residues, i.e. residue distributions constrained at the level of one or more subfamilies.

[23] http://mutationassessor.org/

$$x_i(\alpha \rightarrow \beta) = [\Delta S_i^c(\alpha, \beta) + \Delta S_i^s(\alpha, \beta)]/2 \tag{2.7}$$

Where ΔS_i^c is the family conservation score. ΔS_i^s is the specificity score that quantify the entropy difference resulting from a mutation that affects conserved residue patterns in protein subfamilies.

The validity of the FIS score was tested using the various 'disease-associated' and 'common polymorphism' mutants present in the Uniprot database as a test set (Reva *et al.*, 2011). The evolutionary conservation score is used to distinguish between 19,179 disease-associated and 35,608 polymorphic mutants, thus making it possible to study the impact of mutations.

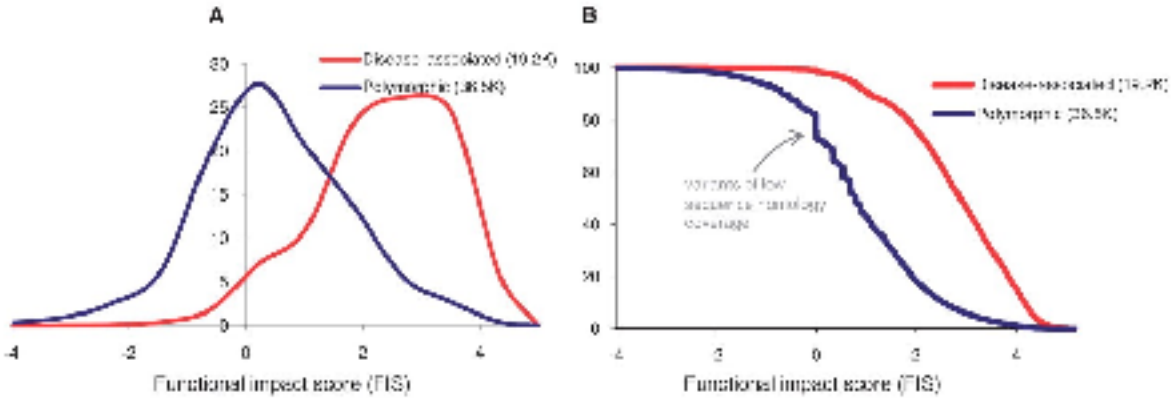


Figure 2.9: The Functional Impact Score and its stability to separate pathogenic and polymorphic variants. (A) Normalized smoothed distributions of the values of the functional impact score as computed for 19179 known 'disease-associated' and 35608 'common polymorphism' variants and mutations annotated in UniProt. (B) The cumulative distributions of the score values computed for disease-associated and polymorphic variants using the same data as in (A) (Reva *et al.*, 2011).

Figure 2.9 B shows that there is a separation (79%) between the two variant classes at a FIS score of approximately 1.9. Around 79% of disease-associated mutants scored higher than this level, and around 79% of all polymorphic mutants scored lower. Further testing of the FIS score was carried out using TP53 mutation information contained in the IARC TP53 database. This resulted in a FIS score that was higher for both mutations that result in 'loss of function' and in cases where there was a 'gain of function' (Reva *et al.*, 2011).

2.6.2 PolyPhen-2 (Polymorphism Phenotyping)

PolyPhen-2[24] is available as a standalone and web-based tool for the prediction of the phenotypic consequences of mutation (Adzhubei *et al.*, 2010). Using naïve Bayesian classification, the tool makes use of sequence and structural conservation to model the position of an amino acid substitution. It then calculates the nature of the substitution and assigns it as 'benign' ($0 \leq p \leq 0.2$), 'possibly damaging' ($0.2 < p \leq 0.85$) or 'probably damaging' ($0.85 < p \leq 1$) depending on the probability intervals obtained (Castellana and Mazza, 2013).

PolyPhen-2 utilizes an algorithm that automatically selects the appropriate features from eight sequence-based and three structure-based options (Adzhubei *et al.*, 2010). The tool first performs a multiple sequence alignment (MSA) against proteins homologous to the query sequence. Homologous sequences are selected for multiple sequence alignment using a clustering algorithm (Camacho *et al.*, 2009), amino acid sequences are aligned using a multiple alignment program MAFFT (Katoh *et al.*, 2002) and then refined using Leon (Thompson *et al.*, 2004). The human alleles forming a pattern of amino-acid replacements are compared to see how distant the deviant is from the wild-type.

The tool then builds a profile matrix using the Position-Specific Independent Counts software (PSIC) (Sunyaev *et al.*, 1999), which in turn provides profile scores, which are logarithmic ratios of the amino acid at that particular position (Sunyaev *et al.*, 1999). The difference in the profile score between the wild and the deviant-type is then calculated. Positive values of high magnitude suggest that the particular amino acid substitution at that position is rarely observed and unlikely to be stable (conserved). PolyPhen-2 then checks for the structural stability of the mutation by conducting a BLAST search on the query against the PDB database. This mapping allows the tool to determine if the substitution in question obliterates a hydrophobic site, alters electrostatic interactions, or influences other structural and interactive components (Adzhubei *et al.*, 2010).

The HumVar and HumDiv datasets were used to train the naïve Bayesian classification used by PolyPhen-2. The HumDiv dataset used contained sequence information on 3,155 alleles with damaging effects on molecular functions, along with 6,321 differences between human-origin proteins and their related non-damaging homologues (Adzhubei *et al.*, 2010). All mutation data listed in HumDiv were retrieved from UniProtKB. Mutations were considered damaging if their annotations contained keywords ("lethal", "complete loss of func-

[24] http://genetics.bwh.harvard.edu/pph2/

tion", "causes", "abolishes", "no detectable activity", "impairs", etc.) implying a causal mutation-phenotype relationship. The version of HumVar used contained 13,032 human disease-causing mutations, along with 8,946 human non-synonymous nsSNPs (nsSNPs). nsSNPs were treated as non-damaging and were compiled from differences in orthologous protein sequences belonging to closely related mammalian species.

When applying a 20% false positive rate (FPR), PolyPhen-2 showed a true positive rate of approximately 92% on HumDiv and 73% on HumVar, respectively (Sunyaev *et al.*, 1999). It is likely that the predictive power of HumVar was lower because of the assumption that the nsSNPs were non-damaging and the fact that HumDiv is more selective in its criteria. PolyPhen-2 uses a 5% / 10% FPR for the HumDiv model and a 10% / 20% FPR for the HumVar model as the limits for its classification into 'benign', 'possibly damaging' and 'probably damaging' categories (Adzhubei *et al.*, 2010). As such, mutations with FPRs at (or below) the lowest FPR value are predicted to be 'probably damaging'; mutations with posterior probabilities related to FPRs at (or below) the higher FPR value are predicted to be 'possibly damaging'; and mutations with estimated FPRs above the second FPR value are classified as 'benign' (Sunyaev *et al.*, 1999).

2.6.3 The SIFT predictor

The Sorting Intolerant From Tolerant (SIFT)[25] tool is a sequence-based homology algorithm that relies on the evolutionary tendency of conserved amino acid positions to be intolerant to substitutions. Thus, this tool does not require structural information (Kumar *et al.*, 2009; Castellana and Mazza, 2013). Using sequence alignment, the query mutation is aligned with orthologous sequences. The tool then creates a score matrix for each position in the alignment and predicts if the variant is damaging or not (Figure 2.10). The score for each possible amino acid substitution is converted to a normalized probability that the substitution would be evolutionarily tolerated (the SIFT score) (Ng and Henikoff, 2003). A score of 0 is considered to be highly damaging, where a score of 1 is neutral. If a score is greater than 0.05, the substitution at the position will be tolerated (Kumar *et al.*, 2009).

To estimate the effect of the substitution, SIFT takes into account the position of the substitution and the nature of the amino acid substituted. The method assumes that functionally and chemically important amino acids are conserved and any substitution with an unrelated amino acid will result in a 'loss of function' (Ng and Henikoff, 2002).

[25] http://sift.jcvi.org/

SIFT aligns the input sequence with related protein sequences using Position Specific Iterated (PSI) BLAST and calculates the probability of each amino acid occurring at a given position, with respect to the most frequent amino acids tolerated (Ng and Henikoff, 2001). The tool then scans each position of the sequence for the probability of occurrence for each of the 20 amino acids. SIFT then builds a probability matrix of the probability of each amino acid at a particular position. Based on the SIFT score, the tool then predicts if the occurrence of a particular amino acid at the given position affects function. A Shannon's entropy conservation value for each position is obtained, which ranges from 0 (the occurrence of all 20 amino acids) to 4.32 (one amino acid) (Ng and Henikoff, 2001).

The SIFT tool was tested using three human variant datasets: (i) annotated substitutions involved in diseases according to SWISS-PROT/TrEMBL; (ii) nsSNPs detected in individuals in studies by the Whitehead Institute; and (iii) putative nsSNPs found in dbSNP (Ng and Henikoff, 2002).

Using the SWISS-PROT/TrEMBL dataset, SIFT predicted that 69% of substitutions would be damaging and this was found to be the lower bound of prediction accuracy (Kumar *et al.*, 2009). In the dataset obtained from the Whitehead Institute, only 19% of mutations were found to be damaging. This demonstrates that the tool can discriminate between neutral and damaging mutations (Kumar *et al.*, 2009). In the third dataset obtained from dbSNP, 25% of mutations were detected as damaging. However, when false positives were accounted for, this value was reduced to 19% (Kumar *et al.*, 2009). SIFT thus predicted changes in amino acids at positions within conserved regions that might influence the function of the protein itself and result in a disease (Ng and Henikoff, 2002). SIFT returned 3,084 (53%) of the 5,780 nsSNPs present in the dbSNP dataset, and predicted that 757 of these would affect protein function. This demonstrates that the accuracy of the SIFT tool depends entirely on the availability of homologous sequences for alignment and the alignment accuracy (Ng and Henikoff, 2001).

2.6.4 Condel

The Condel[26] method (González-Pérez and López-Bigas, 2011) combines the output of five predictive tools for the detection and characterisation of missense SNPs. The tools that Condel combines into a single classification are Log R Pfam E-value (Clifford *et al.*, 2004), MAPP (Binkley *et al.*, 2010; Stone and Sidow, 2005), MutationAssessor (Reva *et al.*, 2011),

[26] http://bg.upf.edu/condel/home/

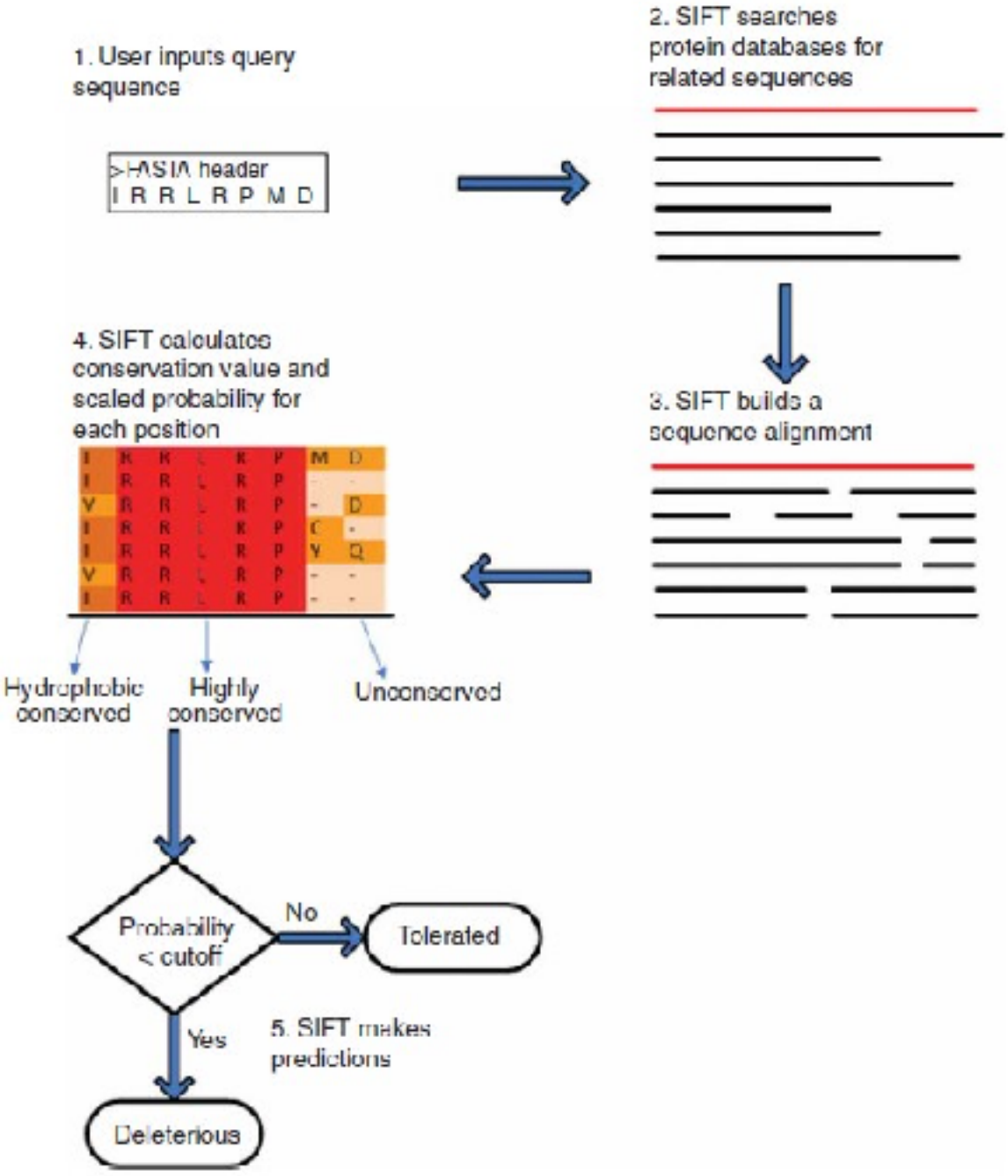


Figure 2.10: The SIFT algorithm uses a sequence query to find the probability of a structural effect owing to a substitution at a particular location using a PSI-BLAST (Kumar *et al.*, 2009).

Polyphen-2 (PPH2) (Adzhubei *et al.*, 2010), and SIFT (Kumar *et al.*, 2009; Ng and Henikoff, 2003). Using a weighted average of the normalized score (WAS), mutations are classified as deleterious or neutral. The WAS weights are calculated using a complementary cumulative distribution of the probabilities of the scores output by each method (González-Pérez and López-Bigas, 2011). The output produced by Condel is superior to that produced by the individual tools, since it combines all data and reports the most likely score.

The Condel authors ran each of the five tools on the HumVar and HumDiv datasets. A complementary cumulative distribution was constructed from the scores for pathogenic and neutral mutations produced by each tool. An internal score, which indicates the probability of an amino acid substitution at a given sequence position is also calculated for each tool. The WAS is calculated as (González-Pérez and López-Bigas, 2011):

$$WAS = \frac{\sum_i S_i * W_i}{\sum_i W_i} \quad \begin{cases} W_i = 1 - Pn_i & if \;\; C_i = 1; \\ W_i = 1 - Pp_i & if \;\; C_i = 0. \end{cases} \tag{2.8}$$

Where C_i is a binary term that takes the value 1 if the i-th tool classifies the mutation as deleterious and 0 otherwise; Si is the normalized score (normalized the internal scores of MAPP, LogR-Pfam, and MutationAssessor to values between 0 and 1 and took the complement of the SIFT probability as the normalized score of this tool); Pni and Ppi are the probabilities of finding a neutral or deleterious mutation, respectively with a score greater than S_i in the given dataset, obtained from the complementary cumulative distribution of the scores produced by the i-th tool.

For a given deleterious mutation, the weight is directly proportional to the score and, for a predicted neutral mutation, the weight decreases with the score.

The WAS method was used to test the recurrence of cancer mutations using four disjointed datasets with increasing recurrence. Mutations were obtained from the Catalogue of Somatic Mutations in Cancer mutation (COSMIC) database (Forbes *et al.*, 2011). The frequency of recurrence subsets were categorized as: (i) mutations that only appeared in a single sample, (ii) mutations that recurred in >24 samples, (iii) mutations that recurred in >59 samples, (iv) and those appearing in >10 samples (González-Pérez and López-Bigas, 2011). Two WAS values were obtained using a complementary cumulative distribution of deleterious and neutral mutations in the HumVar dataset. Mutations that recurred in >10 samples had

an average WAS of 0.706, while neutral variations from HumVar had an average WAS score of 0.236 (González-Pérez and López-Bigas, 2011). The WAS method was then tested to determine if there was a relationship between the WAS score and the biological activity of the altered protein owing to the missense mutation.

Figure 2.11 illustrates the receiver operating characteristics (ROC) curves of the five individual tools and four integrated scores calculated from the aforementioned dataset. The weight average score clearly outperforms the five individual methods in the task of classifying mutations as deleterious or neutral.

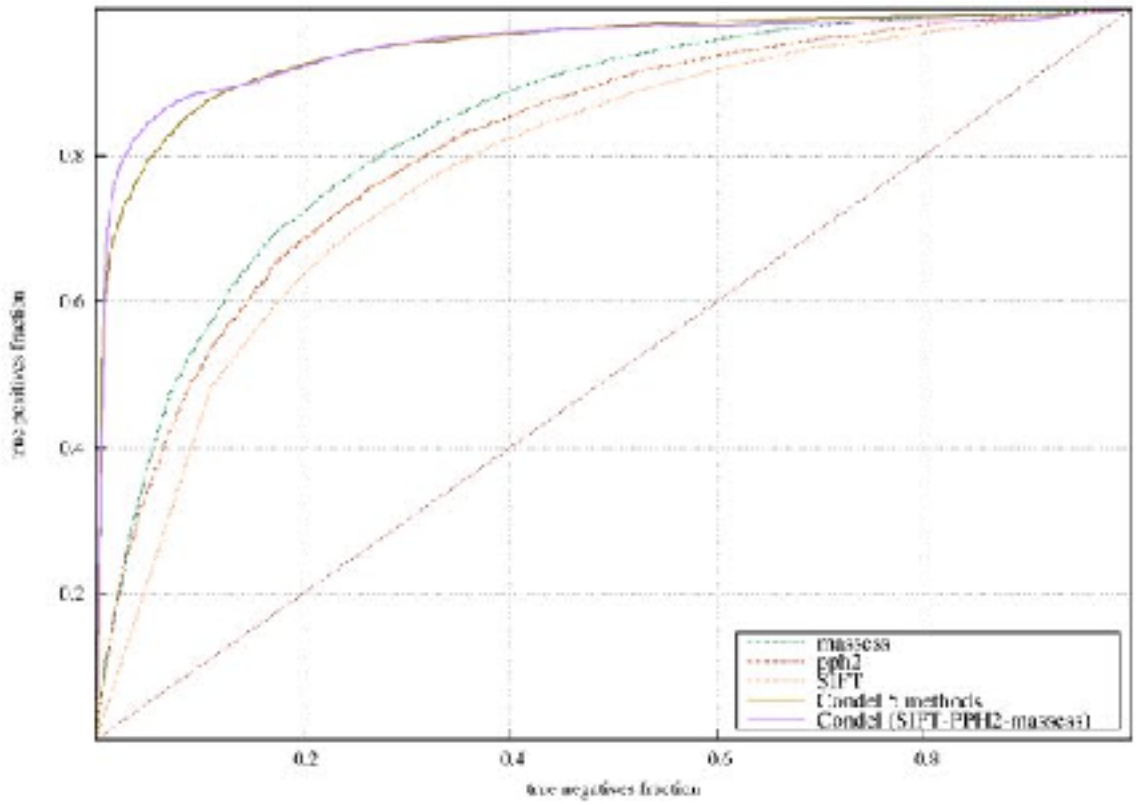


Figure 2.11: ROC curve of the five individual methods and four integrated scores (González-Pérez and López-Bigas, 2011).

2.6.5 FATHMM

Functional Analysis Through Hidden Markov Models (FATHMM)[27] makes use of position-specific information obtained from an MSA of homologous sequences, to find the functional consequences of amino acid substitutions in mutant proteins (Shihab *et al.*, 2013). The MSA is used to build a Hidden Markov Model (HMM) profile. The tool, which is available as a

[27] http://fathmm.biocompute.org.uk/

web-based tool and as a standalone tool uses 'pathogenicity weights' to predict potential deleterious functional effects. These weights are derived from the relative frequencies of amino acid substitutions associated with disease or neutral outcomes in conserved regions of protein structures (Shihab *et al.*, 2013).

The FATHMM tool first searches the UniRef90 database (Apweiler *et al.*, 2004) for homologous protein sequences to construct an initial HMM representing the MSA of the homologous sequences (Shihab *et al.*, 2013). When the amino acid substitution reaches a 'match' state in the HMM, the relevant protein domain information is extracted from the Pfam[28] and SUPERFAMILY[29].

$$\text{UNWEIGHTED} = ln\frac{P_m/(1.0 - P_m)}{P_w/(1.0 - P_w)} \tag{2.9}$$

Where Pw and Pm represent the underlying probabilities for the wild-type and mutant amino acid residues, respectively, and the pathogenicity weights, Wd and Wn, represent the relative frequencies of disease-associated and functionally neutral amino acid substitutions (AASs) mapping onto the relevant HMM, respectively. The pathogenicity weights also include a pseudo-count of 1.0 to avoid a zero divisible term.

A reduction in the probability of an amino acid occurring at a particular location indicates a negative influence on protein function. Conversely, an increase indicates a positive substitution whereby function is improved. Intuitively, large reductions are assumed to have greater impact than smaller reductions (Shihab *et al.*, 2013). The impact is calculated using the formula below, where w is the wild-type and m represents the mutant probability.

$$\text{WEIGHTED} = ln\frac{(1.0 - P_w)(W_n + 1.0)}{(1.0 - P_m)(W_d + 1.0)} \tag{2.10}$$

FATHMM was tested using five on-line datasets; the Human Gene Mutation Database, UniProt, VariBench, SwissVar, and a dataset from a review by Hicks *et al.* (2011). Using the formulation above, substitutions were predicted to be neutral if a score of zero was observed, detrimental if a negative score was observed, and favourable if a positive score was observed.

[28] http://pfam.sanger.ac.uk/

[29] http://supfam.cs.bris.ac.uk/SUPERFAMILY/

To distinguish between disease-associated and functionally-neutral amino acid substitutions, the distribution of the SwissVar dataset was plotted (Figure 2.12). This method showed that the majority of disease-associated substitutions (80%) fell below the threshold, and majority of neutral substitutions (>80%) fell above the threshold. Further testing with a 'blind' dataset is needed to decrease any observation bias in the predictions (Shihab *et al.*, 2013). One limitation of FATHMM is that it is restricted to predicting the effect of substitutions made in the conserved regions of protein sequences, which are present in the Pfam and SUPERFAMILY databases.

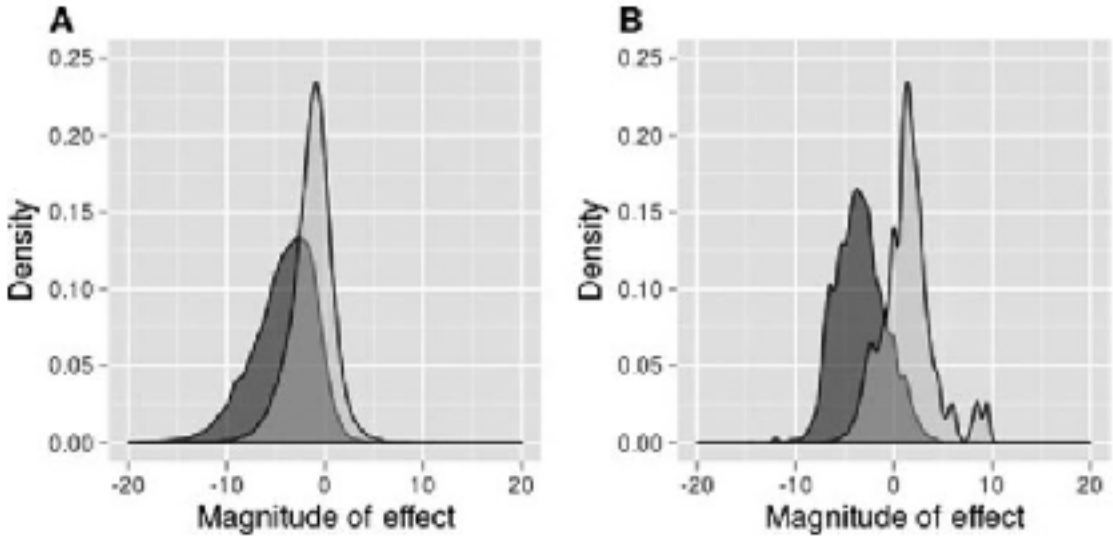


Figure 2.12: Disease-associated (shaded region) and functionally neutral (un-shaded region) amino acid substitutions in the SwissVar dataset using un-weighted and weighted methods (A and B, respectively). A prediction thresholds were calculated at which both specificity and sensitivity were maximized (-3.0 and -1.5, respectively) (Shihab *et al.*, 2013).

2.6.6 Other Methods

MutationTester[30] integrates information from several diverse biomedical databases and uses established analysis tools. Analyses comprise evolutionary conservation, splice-site changes, loss of important protein features and changes that might alter mRNA transcription rates or stability. Test results are then evaluated by a naïve Bayes classifier, which predicts the disease potential. This system is rapid, with a typical query completed in less than 0.3 seconds. Depending on the nature of the alteration, MutationTester chooses between three different prediction models. These are aimed at silent-synonymous or intronic

[30] http://www.mutationtaster.org/

alterations, at alterations changing a single amino acid, or at alterations causing complex changes in the amino acid sequence. To train the Bayes classifier, a dataset of common polymorphisms and known disease-causing mutations was generated using common databases and the literature. The classifier was cross-validated five times, using all three prediction models (Schwarz *et al.*, 2010).

LS-SNP/PDB[31] is a newer web-based source for genome-wide annotation of human nsSNPs. LS-SNP/PDB builds on the existing LS-SNP annotated database, which comprehensively maps nsSNPs onto protein sequences (Karchin *et al.*, 2005). The updated LS-SNP/PDB features fully updated pipeline software with built-in automated build and update functions, and utilizes protein graphics rendered with UCSF Chimera for molecular visualization. Like its parent program, LS-SNP/PDB annotates all human SNPs that produce an amino acid change in a protein structure in the PDB. The local structural environment, putative binding interactions, and evolutionary conservation are all used in the annotation function (Ryan *et al.*, 2009). SNPs can be searched for by using IDs for genes or proteins of interest, or the genomic region (Chen *et al.*, 2009).

Bongo[32] (Bonds ON Graph) is a structure-based approach used to predict both local and global structural effects of nsSNPs. The program uses graph theoretical measures to capture differences in residue-residue interaction networks and to identify residues that are critical for maintaining structural stability. Substituted residues are modelled and the differences in the interaction network are used to define the consequences of single point mutations. Bongo needs a precise protein structure as a starting point for this analysis. Results indicate that structural changes resulting from nsSNPs of key residues are closely related to pathological disease states (Cheng *et al.*, 2008).

AlignGVGD[33] is a web-based program that combines biophysical characteristics of amino acids with protein MSAs to predict where substitutions are so-called 'enriched deleterious' or 'enriched neutral'. The method uses a combination of Grantham Variation (GV), which measures the amount of evolutionary variation at a specific position in an alignment, and Grantham Deviation (GD) which measures the biochemical difference between the wild-type amino acid and the variant (Hicks *et al.*, 2011). The algorithm is very reliant on the quality of the MSAs and list of substitutions that it requires for prediction of mutation effect (Tavtigian *et al.*, 2006; Mathe *et al.*, 2006). The developers of this program suggest that alignments should not be restricted to orthologs, but should also include paralogs to

[31] http://ls-snp.icm.jhu.edu/ls-snp-pdb/
[32] http://www.bongo.cl.cam.ac.uk/Bongo/
[33] http://agvgd.iarc.fr/index.php

account for the phenomenon of functional diversification and increase the accuracy of the algorithms (Tavtigian *et al.*, 2008).

Parepro[34] (Prediction of amino acid replacement probability) is based on a support vector machine (SVM), which is employed to reduce the noise generated from large datasets. As an input, Parepro requires the protein sequence and other protein sequences homologous to it (Tian *et al.*, 2007). Parepro predicts whether genomic nsSNPs have either deleterious or neutral effects, using evolutionary information and properties from the AAindex to determine the differences between the wild-type and mutated amino acid. The predictive tool was trained using the two datasets from the PhD-SNP server (see below). The efficacy of Parepro to predict amino acid variants depends on the number of homologous sequences available, but lack of structural information can be at least partially compensated for by inclusion of 50 different amino acid properties in the attribute list.

PhD-SNP[35] (Predictor of human Deleterious Single Nucleotide Polymorphisms) is based on a combination of SVM-based learning models. As an input, it requires either the protein sequence or the associated Swiss-Prot code, and the position of the mutation. Three slightly different algorithms are available to use. The 'sequence-based' algorithm is the first SVM that classifies mutations as disease-related or neutral, using information regarding residue type and sequence environment. The 'profile-based' algorithm classifies mutations based on a vector of two elements derived from a sequence profile. Finally, the 'hybrid method' combines aspects of these algorithms (Capriotti *et al.*, 2006).

nsSNPAnalyzer[36] is system used to capture the relationship between nsSNPs associated with disease and disease-causing genes. This system requires the input of protein sequences in FASTA format and the provision of SNP data. In addition, the user can provide their own PDB file and chain, allowing the analysis of novel data. The method also uses a Random Forest (Section 2.3.4) to predict the phenotypic effect of nsSNPs. nsSNPAnalyzer calculates three types of information for the assessment of mutation effect. One type of information describes the structural environment of the SNP, and includes solvent accessibility, environmental polarity, and secondary structure. the second is the normalized probability of the substitution in the MSA. Finally, the similarity and/or dissimilarity between the original amino acid and mutated amino acid is calculated using a Blosum matrix (Bao *et al.*, 2005).

[34] http://www.mobioinfor.cn/parepro/
[35] http://snps.biofold.org/phd-snp/phd-snp.html
[36] http://snpanalyzer.uthsc.edu/

MAPP[37] (Multivariate Analysis of Protein Polymorphism) like AlignGVGD combines sequence alignment with amino acid physicochemical characteristics to estimate substitution effects. The difference is that MAPP calculates the physicochemical centroid of each position and the variance between each of the 20 amino acids and that centroid. The input required for MAPPP is an alignment of the protein sequences and a tree describing the distances between the sequences in the alignment. As an output, the user receives a multicolumn table describing the physicochemical characteristics of each position, a listing of neutral and deleterious amino acids, and the MAPP impact score. The score is a continuous variable, for all 20 amino acids at each position (Stone and Sidow, 2005).

Panther[38] (Protein analysis through evolutionary relationships) estimates the likelihood of a particular nsSNP in an exon causing a functional change on a protein by calculating a substitution position-specific evolutionary conservation score (subPSEC). The subPSEC score is a negative logarithm of the probability ratio of wild-type and mutant amino acids at a given position. Values of zero indicate mutations that are neutral whereas those of -10 are more likely to be deleterious. The user is required to input a protein sequence and information about the substitution. subPSEC scores of various values have been shown to correlated with the degree of functional impairment of the mutant protein (Thomas *et al.*, 2003).

SNAP[39] (screening for non-acceptable polymorphisms) is another neural network-based method, which utilizes protein information derived *in silico*. As input it needs a protein sequence only, and returns the output via email. This system benefits from functional and structural annotations, if available. SNAP uses information about residue conservation within sequence families, predicted aspects of protein structure (secondary structure, solvent accessibility), and other relevant information such as biochemical properties (Bromberg and Rost, 2007).

topoSNP[40] is an on-line resource that produces a topographic and interactive visualization of disease and non-disease associated nsSNPs. The method displays geometric location information using the alpha shape method from computational geometry (Stitziel *et al.*, 2004). Geometric locations of SNP structural sites are classified into categories of geometric locations: surface pocket or internal void; a convex region or a shallow depressed region; or buried completely in the interior. A relative entropy calculation using a HMM is used to assess the conservation score (Stitziel *et al.*, 2004).

[37] http://mendel.stanford.edu/SidowLab/downloads/MAPP/index.html
[38] http://www.pantherdb.org/tools/
[39] https://rostlab.org/services/snap/
[40] http://gila.bioengr.uic.edu/snp/toposnp/

CanPredict[41] is a computational tool for predicting cancer-associated mutations. Input data are either a protein AC or a protein sequence together with the mutation(s) to be tested. The impact of each change is measured using two methods: SIFT (as previously described) and the Pfam-based LogR.E-value metric. A third method, the Gene Ontology Similarity Score (GOSS), provides an indication of whether the gene resembles other known cancer-causing genes. Scores from these three algorithms are analysed using the Random Forest method (Section 2.3.4) which predicts whether a change is likely to be cancer-associated (Kaminker *et al.*, 2007).

PMut[42] combines sequence alignment/position-specific scoring matrices (PSSMs) with structural factors to characterize missense substitutions. To accomplish this, the classifier uses a feed-forward neural network using alignment alone or with structural information. The neural network used in the analysis was trained with a large database of disease-associated mutations (obtained from Swiss-Prot) and neutral mutations (observed to be tolerated in human proteins with >95% sequence identity). As input, PMut needs the protein sequence or its Swiss-Prot/trEMBL code. As output, the user receives a confidence index and a binary prediction of 'neutral' vs. 'pathological', represented by pathogenicity index. It is also possible for the user to obtain intermediate information (alignments and Blast (Altschul *et al.*, 1997) and PHD outputs (Rost and Sander, 1993)) used by PMut while generating a prediction. In addition, if the protein structure is available the PMut server can visually display the mutation site within the protein structure using colour-coding to trace the pathogenicity associated with the mutation. This 3D visualisation is obtained as a Rasmol script, for use with Rasmol or the Chime web-browser plug-in (Ferrer-Costa *et al.*, 2005).

SNPs3D[43] is an on-line tool that assigns molecular functional effects of nsSNPs based on structure and sequence analysis using an SVM based algorithm. The tool was trained using a set of disease-causing mutations and a control set of non-disease causing mutations. In jack-knifed testing to asses multivariate data for outliers, the tool identified 74% of disease mutations, with a false positive rate of 15%. This tool relies on the hypothesis that loss of protein structure is a major causative factor in monogenic disease. The SNPs3D website makes the results of the analysis available via the website or gives the option of download. The system can be queried using SNP IDs, protein sequence, or genomic sequence ID (Yue *et al.*, 2006).

[41] http://www.rbvi.ucsf.edu/Outreach/genentech.html
[42] http://mmb2.pcb.ub.es:8080/PMut/
[43] http://www.snps3d.org/

Table 2.4: A summary table for different prediction tools and the methods they used in prediction
For example SAAP uses sequences and a detailed PDB structure analysis (i.e. calculating the void and clash energy, H-bond, hydrophobic etc.) to train a machine learning method to obtain the prediction, where SIFT uses only sequence analysis to give a score for prediction the pathogenicity.

	Sequence	Sequence + Limited Structure	Sequence + Detailed Structure
Score	MutationAssor SIFT FATHMM AlignGVGD MAPP Panther	PolyPhen-2 Condel LS-SNP/PDB cpoSNP	Pmut PhD-SNP
Machine Learning	CanPredict SNAP Parepro	MutationTester Bongo nsSNPAnalyzer SNPs3D	SAAPpred

2.7 Summary

In summary, this chapter has provided detail on the structure and information contained within the primary information sources used to build SAAPdp, and how they have changed since the time of the database-build. Examples of how to store, manage, and interpret these data have also been given, with an emphasis on maintaining data integrity and consistency. Different approaches to machine learning were discussed, all with the common aim of knowledge attainment from large datasets that are yet to be fully characterized. A variety of tools for the assessment of mutation-effect were also presented, each using different methodology to predict the outcome of missense mutations. These were presented to demonstrate the vast array of techniques that can be employed to analyse SNP data and to set the scene for the development of SAAPpred.

Chapter 3

Single Amino Acid Polymorphism Database (SAAPdb)

In this thesis the database of single amino acid polymorphisms (SAAPs) that have been mapped to structure and subsequently analysed to provide hypotheses as to their effect(s), on protein structure was rebuilt and exploited. The resource, named 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. The development of a conservative, comprehensive structural analysis pipeline employing several well established data resources, data handling methods and data analysis methods with which to analyse SAAPs has been one of the main aims of the SAAPdb project. In this chapter, the suite of analyses with which SAAPdb assesses each mutation is described to provide the context for later chapters.

Much of the work described in this chapter was developed by previous members of the group who are identified in each section. My involvement is also indicated. My major contributions have been: (i) to fix a number of bugs in code to populate the database, (ii) to improve portability of the code by removing hard-coded paths and moving them to configuration file; (iii) to update the resource by adding a new database; (iv) to perform some analysis of the data in SAAPdb.

3.1 Introduction

SAAPdb began as a restricted study of mutations in P53 (Martin *et al.*, 2002) and G6PD (Kwok *et al.*, 2002) in which seven structural effects were considered. As the project progressed, other databases of SAAPs were combined and several additional structural analyses were included in the analysis pipeline.

To analyse and visualise mutations and their local structural effects, a new system was introduced by Hurst *et al.*, (2009): the Single Amino Acid Polymorphism Database (SAAPdb). This is a PostgreSQL relational database of SAAPs (an alteration of a single amino acid in a protein sequence, as a result of a missense mutation) providing a range of likely structural effects of SAAPs on structures of human proteins, based on mappings of the mutations to structural data. Andrew Martin's group have developed SAAPdb and its web-server with two main functions: (i) to provide a website that can clearly and effectively display the location of mutated residues within solved protein structures and (ii) to keep a fully automated and up-to-date structural analysis of the mutations that can be accessed through the website[1] (Hurst *et al.*, 2009).

The system begins by gathering data on single nucleotide polymorphisms (SNPs) from dbSNP and maps the data onto the genes to determine whether each mutation is in an exon and, if so, whether it causes a missense mutation in the protein (Cavallo and Martin, 2005). Once this has been determined, the location of the mutation within the protein sequence can be established. Disease-related mutation data, or 'pathogenic deviations' (PDs), from OMIM as well as several locus-specific mutation databases (LSMDBs) are provided at the protein mutation level. Once the mapping of a mutation to a protein sequence has been achieved, if a structure exists in the Protein Databank (PDB) for the protein, the mutant is mapped onto the protein structure and then followed by an automated structural analysis (the SAAPdb analysis pipeline is discussed in Section 3.7). This tests whether the mutant residues will have any local structural effects that may disrupt protein folding, binding, function or stability and therefore may be related to a harmful phenotype.

3.2 Mutation Data in SAAPdb

The raw data on which SAAPdb is based describes two kinds of genomic variation and mutation, classified according to their reported pathogenicity. One type of genomic varia-

[1] http://www.bioinf.org.uk/saap/db/

tion is caused by SNPs, which are assumed to have a negligible effect on protein function, and considered to be neutral. Some SNPs may have a small negative (or positive) effect on human phenotype, but until this is identified, we consider them to be neutral mutations. The second kind of genomic variation are the PDs. These are associated with disease and are therefore thought to have a deleterious effect on protein structure and function. In cases where the same mutation can be found in both datasets (PDs and SNPs) the mutation is removed from the SNP dataset. Section 3.3 discusses the occurrence of PD/SNP overlap in more detail.

Databases used in this thesis to populate the latest version of SAAPdb (released on March, 2011) are shown in Table 3.1. SNPs extracted from dbSNP are all non-synonymous (missense) mutations arising in the coding regions (exons) of the human genome. Mutation data are stored with mappings to sequence data using dbSNP, and to structural data where structures are available. Mappings of PDs are retrieved where available, and then added, verified and/or corrected by an algorithm developed by Martin (2005).

3.2.0.1 SNPs

The term 'Single Nucleotide Polymorphism' (SNP) (Consortium, 2005), is frequently used to pertain to a mutation of any frequency. However, if strictly defined, SNPs are allelic variants where the least common allele occurs in at least 1% of a normal population. According to Wang (2006), a SNP occurs approximately once every 100-300 bases in the genome, introducing a subtle phenotypic variation without causing serious and damaging phenotypic change. However, the reported frequency of SNPs varies as a result of sequencing different fragments of the human genome (Collins *et al.*, 1998; Taillon-Miller *et al.*, 1998; Sherry *et al.*, 2001). Based on the latest dbSNP data, a conservative estimate of SNP frequency is that one occurs every 300-500 base pairs (David, 2005). These mutations are unequally distributed over different regions of the human genome and between exons and introns (Nickerson *et al.*, 1998). Fluctuations in the frequency or locations of SNPs are not relevant for this project and have not been considered further.

The largest publicly available SNP database is dbSNP (Wheeler *et al.*, 2007)[2]. Where relevant, this database provides mapping to the specific residue in a human protein, as well as native and mutated nucleotide and amino acid type. The latest available release of dbSNP at the time of writing (Build 139, Oct 25 2013) includes over 505 million SNP submissions, over 28 million reference SNPs and over 3 million validated SNPs.

[2]Data available at http://www.ncbi.nlm.nih.gov/projects/SNP/

3.2.0.2 Pathogenic Deviations (PDs)

Pathogenic Deviations (PDs) occur with much lower frequency than SNPs and generally create a loss-of-function mutation with a serious adverse effect on phenotype. The term 'pathogenic deviation' is used for any single base change noted to cause a disease.

The PD dataset currently available in SAAPdb is derived mostly from the Online Mendelian Inheritance in Man (OMIM) database[3] (McKusick, 2000). OMIM contains data on a broad spectrum of pathological conditions and protein families. However, it is only a sample of known mutations and is probably biased owing to the specific interests of the scientific community in certain diseases/proteins. Smaller Locus-Specific Mutation DataBases (LSMDBs), are the second group of PD databases. They generally contain data on a single protein or disease, and are maintained by different research groups. The PD dataset is currently augmented by a selection of LSMDBs as described in Section 2.1.6.

Table 3.1 shows the updated OMIM and LSMDBs incorporated into SAAPdb. The PAHdb and STAT3 mutation data are new data sources integrated into SAAPdb for the first time in this thesis.

There are two main problems with publicly available PD data. The first is the diversity of formats of these data; the second is the absence of pathogenicity levels. The latter results in difficulty in comparing the effects on protein structure with the phenotype of the individual. Hurst *et al.* (2009) have discussed the reliability of these sources stating that over 500 LSMDBs are recorded on the Human Genome Variation Society's website, while SAAPdb only includes around 2% of these data. The SAAPdb system has been designed and implemented to facilitate the integration of additional locus-specific data in a simple and straightforward manner; by parsing source data into an XML format that is then loaded into the database.

3.3 SNP/PD overlap

The size of the datasets currently used in SAAPdb and the overlap between them is displayed in Table 3.3. As the central and largest PD resource, as would be expected, OMIM has at least some overlap with all of the other PD datasets. OMIM is at least fourteen times larger than the next largest, the somatic P53 dataset. Within the LSMDBs, the only overlap that exists is between the germline and somatic P53 datasets.

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

Table 3.1: Number of distinct mutations from different sources that have been mapped to protein sequence and deposited in the SAAPdb before and after the update.

Database Name and Location	Mutations before the update	Mutations after the update
dbSNP The Single Nucleotide Polymorphism Database *http://www.ncbi.nlm.nih.gov/SNP/*	34342	48452
OMIM Online Mendelian Inheritance in Man *http://www.ncbi.nlm.nih.gov/entrez/omim_sprot.cgi*	7249 -	9339
ADABase Adenosine DeAminase deficiency *http://bioinf.uta.fi/ADAbase*	30	38
G6PD Glucose-6-Phosphate Dehydrogenase *http://www.bmn.com.ac.uk/g6pd/db*	170	170
Hamsters The Haemophilia A Mutation, Structure, Test and Resource *http://europium.csc.mrc.ac.uk/WebPages/Database/Query/pointl.html*	530	628
IARC_P53_Germline Tumor Protein 53 gene germline mutation in familial cancers *http://www-p53.iarc.fr/Germline.html*	95	138
IARC_P53_Somatic Tumor Protein 53 gene somatic mutations in sporadic cancers *http://www-p53.iarc.fr/Somatic.html*	617	1368
OTC Ornithine TransCarbamylase (OTC) gene *http://www3.interscience.wiley.com/cgi-bin/abstract/89012960/*	148	214
SOD1db SuperOxide Dismutase 1 *http://fester.lif.icnet.uk/Nik/Index.html*	96	125
ZAP70Base Zeta-chain-Associated Protein kinase 70 *http://bioinf.uta.fi/ZAP70base*	5	5
KinbaseDriver / KinbasePassenger Somatic protein kinase driver and passenger mutations *Greenman C et al. Patterns of somatic mutation in human cancer genomes Nature 2007, 446(7132):153-8.*	66 / 66	66 / 66
LDLR Low Density Lipoprotein Receptor *http://www.ucl.ac.uk/ldlr*	515	515
PAHdb Human Phenylalanine Hydroxylase gene *http://www.pahdb.mcgill.ca*	–	337

Table 3.2: Breakdown of the number of mutations in SAAPdb and their mapping to structure. In some cases, several hundred structures are available (e.g. haemoglobin, carbonic anhydrase, porthrombin, transthyretin, insulin, CDK2, lysozyme) and, on average there are approximately two copies of each chain in each PDB file.

Number of Mutations	PDs	SNPs
Mapped to UniProtKB/Swiss-Prot	13,059	48,452
Mapped to PDB	6,527	17,915
Total mapped (to multiple PDBs)	202,566	33,369
Total mapped (to multiple Chains)	405,497	45,699

Encouragingly, very few mutations are simultaneously described as disease-associated and neutral. So far, only six mutations have been identified in PD data and also present in dbSNP. Half of these are common to the dbSNP and OMIM datasets and the other half are common to the dbSNP and P53 somatic datasets. When updating SAAPdb, these mutations are removed from the SNP dataset, but retained in the disease dataset, 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 with a pathological disease.

One important caveat is that the unique complexity of cancer (where multiple mutations are acquired over a short period of time) introduces uncertainty regarding the potential pathogenicity of those mutations found in both the somatic P53 dataset and the dbSNP dataset. Some 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 the three SNPs also described in the P53 somatic dataset are mapped to protein structure and they are therefore not analysed in this research.

3.4 Additional resources

Since collection of SNP and PD data is only one function of SAAPdb, several additional resources are required to process these data to determine their likely effects on protein structure and function. UniProtKB (Section 2.1.2) is required to map gene names to proteins and identify annotated functional residues; ENA and Genbank (Section 2.1.1) are required to map genomic data to protein sequences where mappings are unreliable or absent; and PDBSWS (Section 2.1.4) is used to map protein sequences to protein structures. The functions of each of these resources are discussed more fully in their respective sections.

Table 3.3: 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 in total; the SNP dataset (dbSNP) is separated from the other PD datasets using a double ruled line; 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'), and are further described in Section 3.2.0.2.

	ADABase	G6PD	HAMSTeRS	P53-G	P53-S	OMIM	OTC	SOD1db	ZAP70	dbSNP
ADABase	**38**									
G6PD	0	**103**								
HAMSTeRS	0	0	**526**							
P53-G	0	0	0	**94**						
P53-S	0	0	0	89	**1396**					
OMIM	19	44	135	23	27	**7119**				
OTC	0	0	0	0	0	12	**148**			
SOD1db	0	0	0	0	0	27	0	**96**		
ZAP70	0	0	0	0	0	1	0	0	**5**	
dbSNP	0	0	0	0	3	3	0	0	0	**34081**

- IARC P53 Database - Somatic/dbSNP : 3 mutations are common
  1. LSDB row ID = 16263 / SNP id = rs11540654
  2. LSDB row ID = 3610 / SNP id = rs35163653
  3. LSDB row ID = 12753 / SNP id = rs1800371
- OMIM/dbSNP : 3 mutations are common
  1. LSDB row ID = 19737 / SNP id = rs154001
  2. LSDB row ID = 25610 / SNP id = rs13312740
  3. LSDB row ID = 22441 / SNP id = rs12530380

3.5 Materials and Methods

Two main stages of data processing are employed: (1) importing the SAAP data (PDs or SNPs) and (2) analysing the imported data using the SAAPdb structural analysis pipeline. This processing structure naturally leads to a three-part data division, which applies to data storage and data processing: (1) SNP data; (2) PD data and (3) pipeline data. These three are described in the following sections.

As described above, several other people have contributed to the design, development and maintenance of SAAPdb, including Jacob Hurst, Lisa McMillan, 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.

3.6 The database

3.6.1 Populating reference tables

Three tables in SAAPdb are populated from UniProtKB: Swiss-Prot TrEMBL, gene name map and accession map databases. These tables contain sequence data, mappings between gene names and protein identities and a mapping between secondary and primary accession numbers respectively. UniProtKB is mirrored locally. Before data processing begins, another copy of the UniProtKB mirror is cached locally to ensure that the same version of UniProtKB is used in all relevant SAAPdb analyses (i.e. mirroring does not update the data during processing).

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.

3.6.2 Importing the dbSNP data

▷ *This method was developed by Lisa McMillan.*

The Entrez Programming Utilities (or eUtils)[4] a set of seven server-side programs, are used to provide a stable interface into the Entrez query and database system at the National Cen-

[4] `http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html`

tre for Biotechnology Information (NCBI) and to obtain the most recent dbSNP data. 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 then combined into one XML file and parsed to populate the `snp` and `snp2annotated` tables with dbSNP data.

3.6.3 Mapping the SNPs to protein structure

▷ *This method was 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.

3.6.4 Importing the PDs

The task of mapping and processing the PD data is in some ways more straightforward than that of SNP data. There are less PD data, allowing processing to be sequential. Furthermore, protein sequence mappings are usually provided, avoiding computationally demanding mapping procedures. Instead, the challenges arise when accommodating the different file formats of the source databases. In the following sections, the methods by which the PD data are imported are described.

3.6.4.1 The data-specific wrapper

▷ *These methods were developed by James Allen and Nouf Almunair.*

As mentioned above, PD data are amalgamated from different sources using a variety of file formats. Thus, to permit easy integration of all the data into SAAPdb, it is necessary to represent all the data in the same format. To do this, 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 3.1. This approach requires that each dataset be accommodated by a dataset-specific 'wrapper' which converts the original data into the XML format. This process and the retrieval of the raw data files themselves, are the only manual steps required to import the PD data.

3.6.4.2 Verifying protein sequence numbering

▷ *This method was developed by Andrew Martin.*

The numbering of amino acids provided by the primary datasets must first be verified for correctness. Though OMIM and LSMDBs are curated resources for disease mutation data, the described mutations maybe derived from multiple sources and from the literature. Thus, it is not surprising that there are sometimes inconsistencies in the numbering of amino acids that must be corrected.

A version of OMIM with corrected amino acid numbering is currently automatically maintained by the Martin group. Figure 3.2 shows how the verified OMIM map is derived for each disease dataset. First, a partial sequence is constructed from the native residues described in OMIM (Figure 3.2a). 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 3.2b). The alignment that best matches the named protein sequence is used to calculate an offset value describing how the OMIM numbering should be corrected. In the example given, the offset value is -3 (Figure 3.2c). The offset-rule 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 3.2c), these data are assigned an offset of 0, and flagged as 'probably correct'. In these cases, it is assumed that sequences were submitted to OMIM in a separate batch where correct UniProtKB numbering was used. Some mutations may remain unmapped after these stages. A completed corrected dataset is shown in Figure 3.2d.

Of 2438 OMIM mutations from 221 OMIM entries available in March 2012 (11.3% of all cross-linked-to UniProtKB/Swiss-Prot OMIM entries), 24.1% 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/`.

Corrections to amino acid sequence numbering are also applied to LSMDB datasets in an attempt to maximise the amount of correct data extracted.

```
<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 3.1: A sample of the XML format

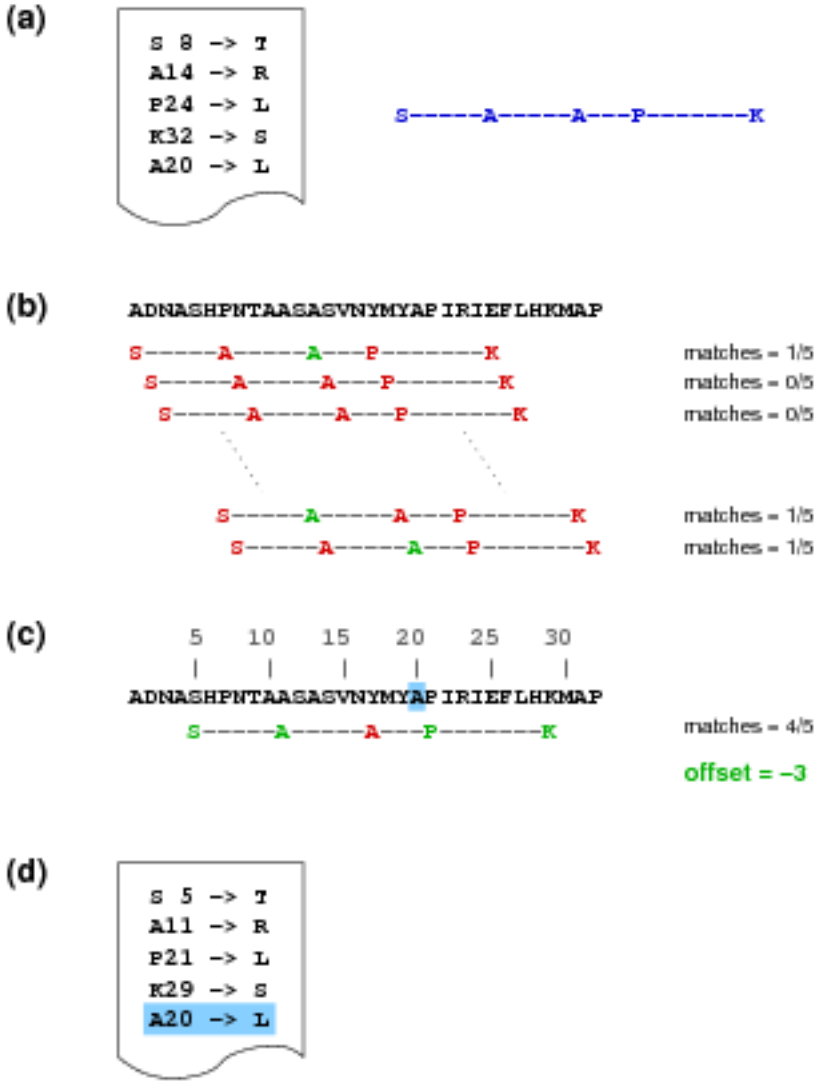


Figure 3.2: (a): a partial sequence is reconstructed from the native residues described in the OMIM record; (b): this partial sequence is slid along the sequence to which it is mapped in OMIM and the number of matches for each position is recorded (matches are shown in green, mismatches are shown in 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). *(Adapted from Lisa McMillan's PhD thesis)* (McMillan, 2009).

3.6.4.3 Pushing the data into the database

▷ *These methods were developed by James Allen, Lisa McMillan and Nouf Alnumair.*

The complete work-flow by which PD data are entered into the SAAPdb is illustrated in Figure 3.3. This includes the data-specific 'wrapper' function highlighted in red. Whilst the wrapper code to convert the raw data into XML format is only written once for each dataset, it is not uncommon for updates to require a wrapper re-write. The pseudocode for the wrapper scripts is shown in Figure 3.4.

The system will attempt to identify the correct AC in cases where the mutation is not already mapped to a UniProtKB/Swiss-Prot sequence. SAAPdb does this by constructing a partial native sequence by combining the wild-type residues from the data and representing all other residues with an 'X'. This partial sequence is then used to search 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 3.3.

Each XML file generated for each dataset is then converted to SQL statements via an XSLT specification (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 3.5) with the appropriate data.

Finally, the imported and verified PDs are mapped to protein structures and the `lsdb_saap` table populated with the mappings (this step requires that the data described in Section 3.6.1 be present in the database). To do this, the UniProtKB/Swiss-Prot accession numbers to which the disease mutations are mapped are updated to their corresponding primary accession number. Primary accession numbers are obtained from PDBSWS. Then, the `lsdb_saap` table is populated with the appropriate sequence and structural data.

Figure 3.3 describes the complete data flow for a single dataset. In reality, processing progresses through the data representations, rather than through each dataset, however, this has been presented for simplicity. Usually, all 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. SQL statements updating the AC numbers and the structural mappings are executed only once all sequence data are in the database.

Figure 3.3: Importing an LSMDB dataset.

In general terms, the work flow is as follows: a wrapper script converts the raw data into valid XML, 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 XSLT); 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 are separated by a dotted blue line.

(a) wrapper inputs

- `data_folder` : the folder containing the raw data
- `xml_folder` : the folder containing the xml

If these are not provided, the default values if '../data' and '../xml' respectively are used.

(b) wrapper process

1. parse the `lsdb_info.txt` file to find the `dbname`, `dburl`, `sprotac`, `rawdatafile`
2. open `rawdatafile` using `cvs.reader()` and the appropriate delimiter
3. check whether a corresponding XML file already exists (if so, exit cleanly without doing anything)
4. identify the `sprotac` using `lsdb_utils.get_ac_number()` unless `sprotac` has been extracted from `lsdb_info.txt`
5. 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`
6. 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 $\geq$ 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
7. write the XML file using the validated data

Figure 3.4: 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 set in `lsdb_utils.correct_residue_number()`; process at line #6a retrieves the sequence from `http://us.expasy.org/uniprot/`.

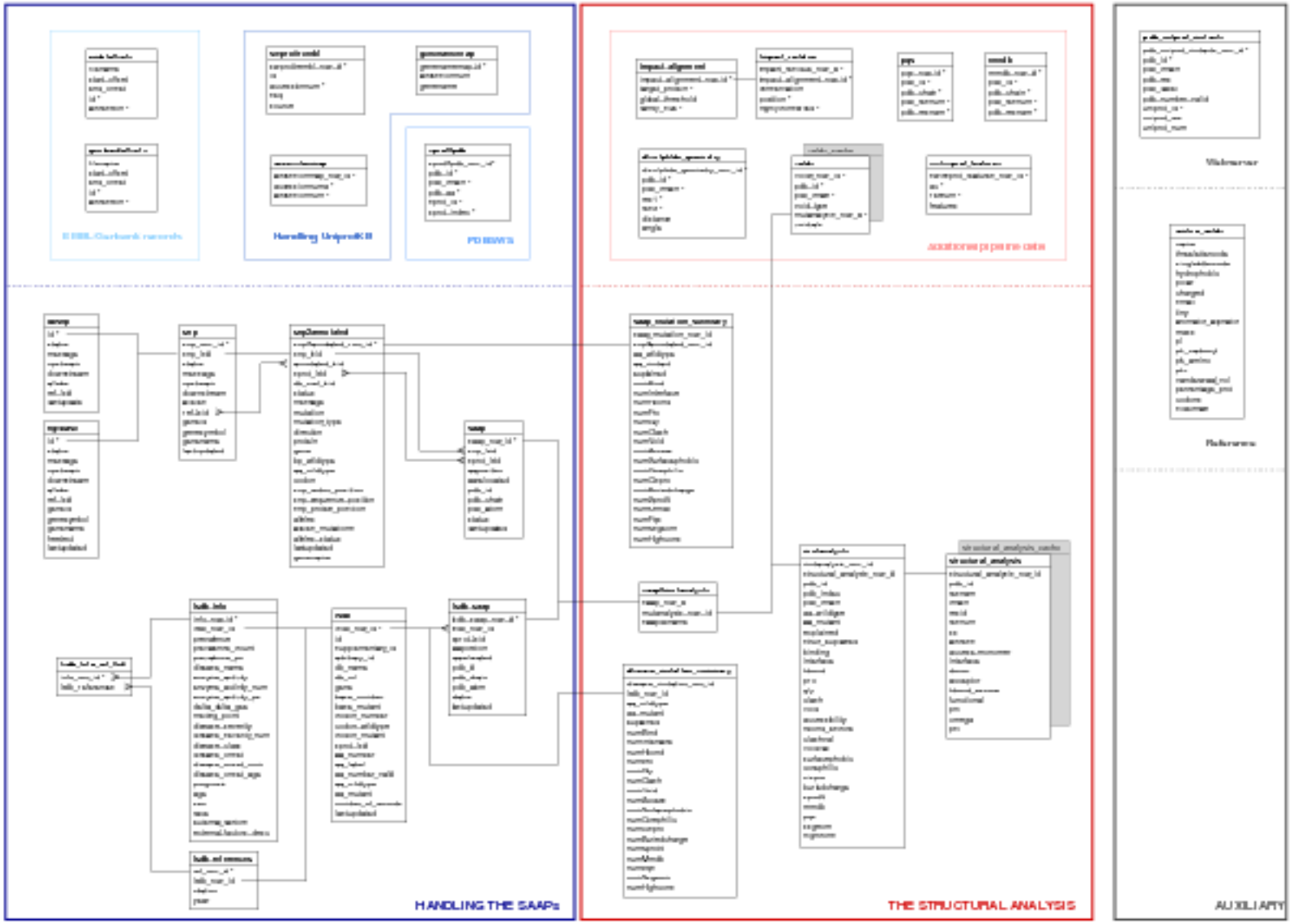


Figure 3.5: SAAPdb schema. Mutation data and mappings to protein sequence and structures are coloured blue, results of structural analyses are shown in red, and additional data in black. *(Adapted from (McMillan, 2009))*.

3.7 The analysis pipeline

▷ *These methods were originally developed by Jacob Hurst and have been extended by Lisa McMillan and Nouf Almunair.*

The purpose of the pipeline is to assess the likely structural effects of the mutation, utilising known structural constraints, interactions and bonding rules. Once the SNP and PD data are mapped to protein structures (i.e. once the saap and lsdb saap tables have been populated), pipeline processing can begin. Eight of the analyses require additional data to be present in the database: hydrogen bonding (Section 3.7.2.1), clash (Section 3.7.2.4), void (Section 3.7.2.5), mutations to binding residues, UniProtKB/Swiss-Prot features (Section 3.7.2.12), sequence conservation (ImPACT) (Section 3.7.2.13), interface (Section 3.7.2.14) and disulphide geometry (Section 3.7.2.7) analyses. Detailed information regarding these analyses, what data are required and how they are derived is available in the respective sections below.

Figure 3.6 illustrates how the pipeline is run and how the data are coordinated. In this figure, the four phases of processing are delineated using dashed lines. Each of the phases are further broken down into a number of sequential processing steps. In step 1 of phase (A), the data from the `saap` and `lsdb_saap` tables are imported into the `mutanalysis` table. In step 2, 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). In step 3, the link between the `mutanalysis` and `structural_analysis` tables is created.

In phase (B), all the necessary pre-processing is carried out for the eight analyses requiring additional data described above. These form step 4, hydrogen bonding; step 5, clash; step 6, void; step 7, interface; step 8, sequence conservation/ImPACT; step 9, MMDB; step 10, UniProtKB features; and step 11, disulphide geometry analyses. Of these, clash, void, interface and ImPACT (steps 5-8) require considerable preprocessing and as such are distributed across the local 20-core grid. Results of all eight analyses are written to the specialist, correspondingly named tables (see Figure 3.5). The `mutanalysis` table is also updated with the results of the clash pre-processing step and therefore carries out the clash analysis. In Figure 3.6, all processing that is distributed is highlighted in grey.

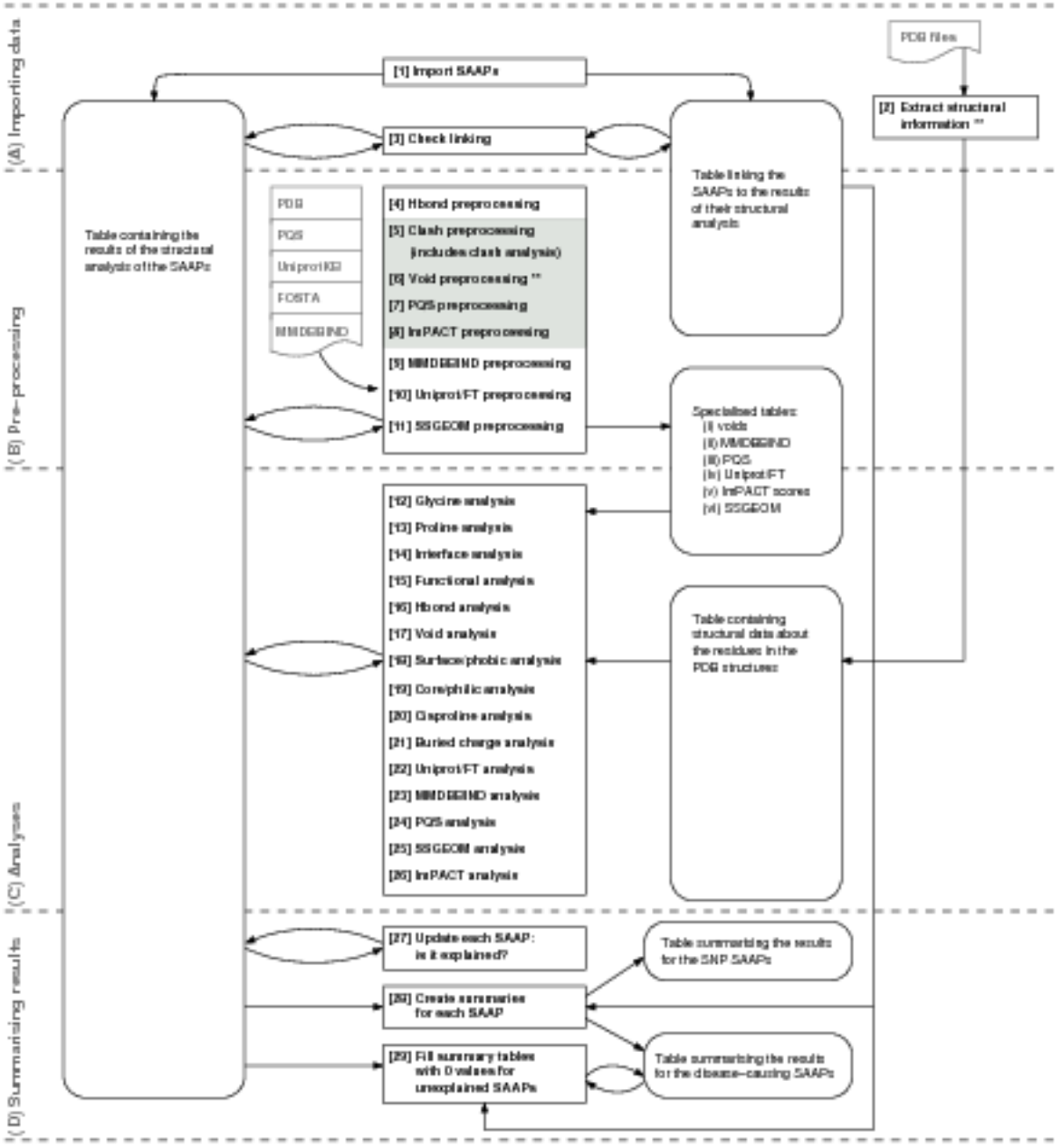


Figure 3.6: Pushing the SAAPs through the structural analysis pipeline.

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, Interface, 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. (*Adapted from Lisa McMillan's PhD thesis* (McMillan, 2009)).

The two most time consuming processing steps in the whole process 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, these tables are named `voids_cache` and `structural_analysis_cache` (these are shown in grey in Figure 3.6). Data from the cached tables are imported if requested, while the original tables are recreated. Processing can then proceed as normal.

Once all additional data are imported into SAAPdb, the remaining analyses can be implemented as SQL queries. These are carried out in phase (C) (steps [12-26]). The results of these analyses are used to update the appropriate columns in the `mutanalysis` table. The purpose of phase (D) is to summarise the results. The first step of phase (D) [step 27] is to annotate each mutation described in the `mutanalysis` table with an indicator of whether it is predicted to have a structural effect or not. This prediction is 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 of all mapped structures, as described in either `saap` or `lsdb_saap`. Finally, any blank entries in the `disease_mutation_summary` and `saap_mutation_summary` tables are replaced by zeros (step [29]).

3.7.1 Generating mutant structures

For the void and clash analyses it is necessary to generate a mutant structure. The MutModel program used to model the mutant residue into the native structure using the 'minimum pertubation protocol' (MPP) (Shih *et al.*, 1985; Snow and Amzel, 1986). MutModel is improved and its performance analyzed and evaluated as part of this thesis and therefore described in details in Chapter 4 - Section 4.3.

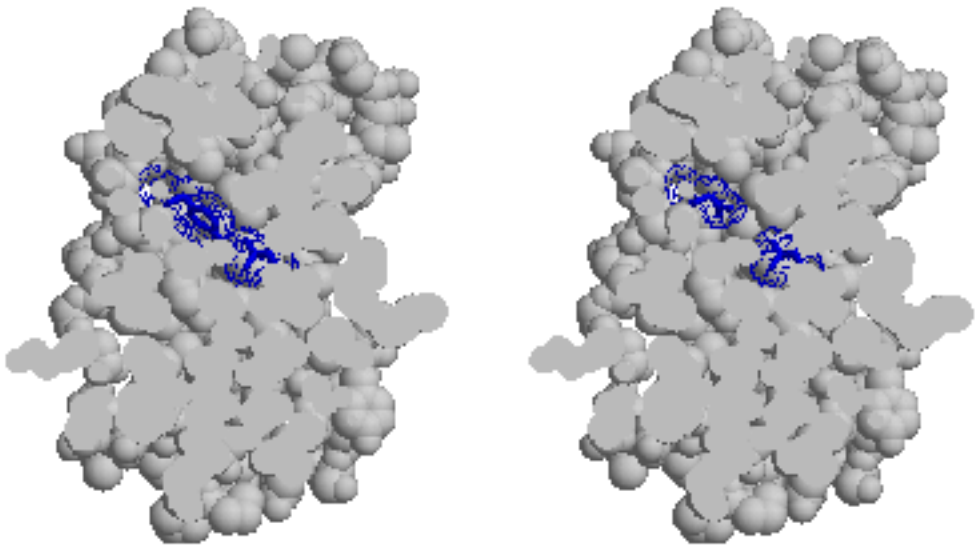
3.7.2 Existing analyses

The analyses described in this section have been previously published in Martin *et al.* (2002), Cuff and Martin (2004) and Cuff *et al.* (2006). These analyses have been used elsewhere to explain disease mutations in disease-specific datasets, including P53 (Martin *et al.*, 2002) and G6PD (Kwok *et al.*, 2002). Here, the existing analyses are described briefly in the context of how they are integrated into the analysis pipeline, as described in Hurst *et al.* (2009).

3.7.2.1 Disrupting native hydrogen bonding

Hydrogen bonding is critical to maintaining the native protein secondary and tertiary structures. Cuff *et al.* (2006) used a grid-based approach to analyse the occurrence and geometry of hydrogen bonds in the PDB for each hydrogen bonding donor and acceptor residue pair. It is then possible to compare hypothetical mutant structures with the observed hydrogen bonding residue profiles to assess whether a hydrogen bond is possible or not using the program `checkhbond`. This program is available for use over the web at `http://www.bioinf.org.uk/hbond/`.

Each mutation must be analysed by `checkhbond`, but the algorithm is designed to be fast and requires only the native structure. 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. A score of 0 implies that it is very unlikely that a hydrogen bond is formed. Mutations that break hydrogen bonds (i.e. those with a pseudo-energy score of 0) are identified between backbone/side-chain and side-chain/side-chain donor and acceptor atoms. At present, this processing is done sequentially by one machine although this strategy is suitable for distributed processing.



(a) Native 2j1w:B (b) Mutant Y236D 2j1w:B

Figure 3.7: Breaking hydrogen bonds.
PDB ID 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 3.7.2.1). Residues 236 and 253 are highlighted in blue in both structures.

An example is shown in Figure 3.7: 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 3.7a. This hydrogen bond is broken in the Y236D mutant structure in Figure 3.7b, as the introduced aspartic acid side-chain 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 is found to cause a de-stabilising internal void.

3.7.2.2 Mutations to proline

Proline exhibits particular spatial constraints owing to the nature of its side-chain. The cyclic nature of the proline side-chain limits the backbone conformations that 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[5] mutations likely to effect the backbone conformation were identified in SAAPdb out-with the region: $-70.0^\circ \leq \phi \leq -50.0^\circ$ and ($-70.0^\circ \leq \psi \leq -50.0^\circ$ or $110.0^\circ \leq \psi \leq 130.0^\circ$). In Figure 3.8, this area is marked in pink.

As part of the work in this thesis, this 'Boolean' method with simple boundaries for allowed conformations has been replaced with a new method introducing an energy evaluation (see Chapter 5).

3.7.2.3 Mutations from glycine

Glycine is the opposite of proline in that is has no side-chain (The R group is a lone hydrogen) 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 out-with the region ($-180.0^\circ \leq \phi \leq -30.0^\circ / 60.0^\circ \leq \psi \leq 180.0^\circ$) or ($-155.0^\circ \leq \phi \leq -15.0^\circ / -90.0^\circ \leq \psi \leq 60.0^\circ$) or ($-180.0^\circ \leq \phi \leq -45.0^\circ / -180.0^\circ \leq \psi \leq -120.0^\circ$) or ($30.0^\circ \leq \phi \leq 90.0^\circ / 20.0^\circ \leq \psi \leq 105.0^\circ$) were identified by SAAPdb. In Figure 3.8, this area is coloured yellow.

Again as part of the work in this thesis, this 'Boolean' method using simple boundaries for allowed conformations has been replaced with a new method introducing an energy evaluation (see Chapter 5).

[5] X is any non-proline amino acid.

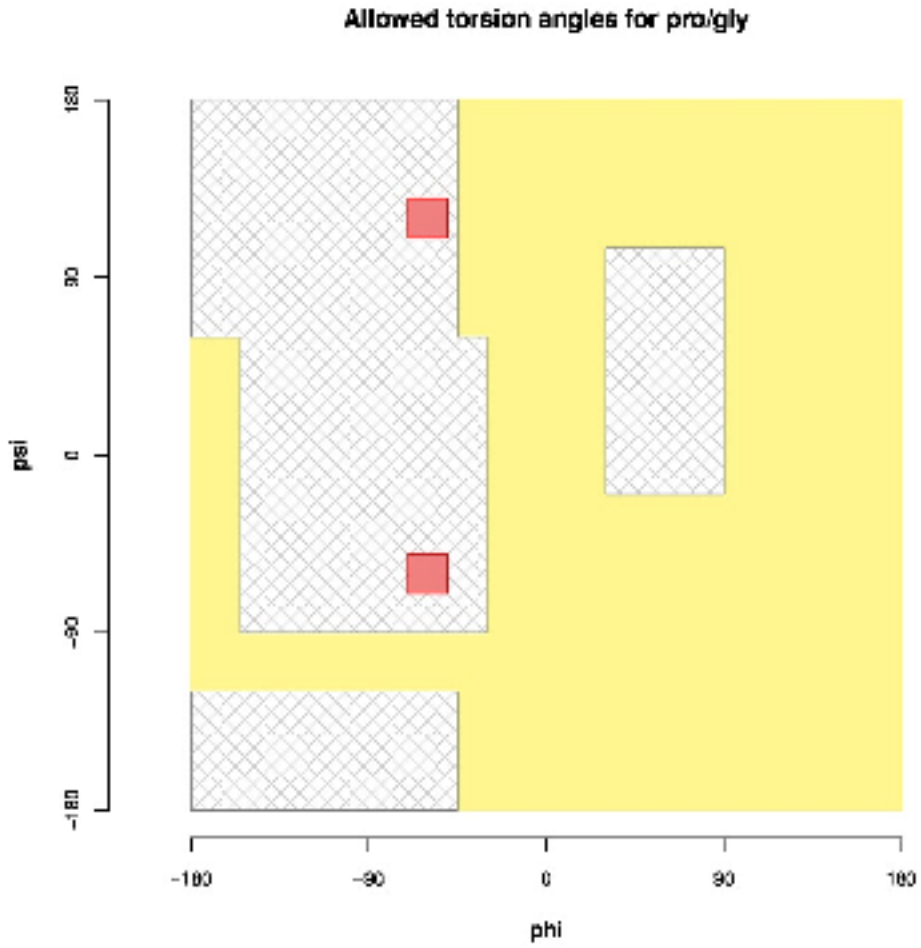


Figure 3.8: Allowed regions for proline and glycine.

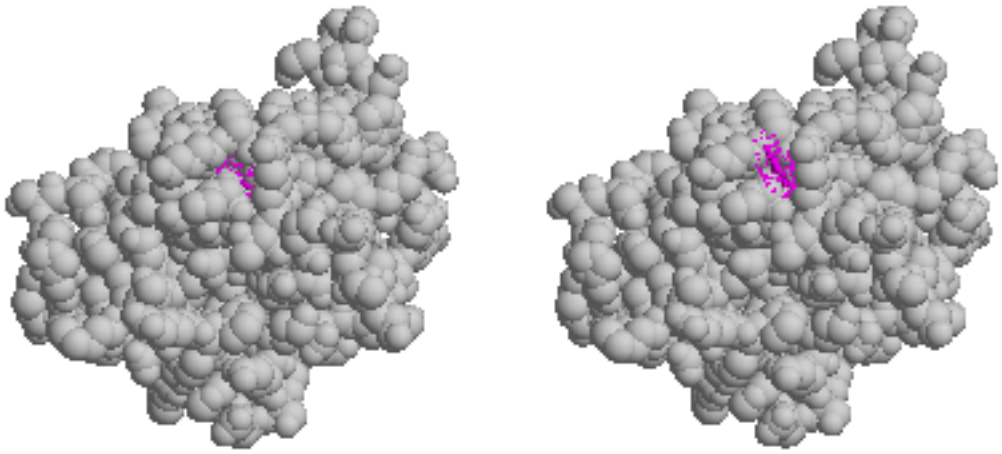
The pink areas mark the restricted conformation for proline residues, the hatched grey areas mark the regions for non-proline, non-glycine residues, and the yellow colour marks the rest of the conformational space, primarily occupied by glycine residues.

3.7.2.4 Mutations that cause steric clashes

Many amino acid substitutions will result in steric clashes. For example, it may not be possible to accommodate a larger mutant residue in the native structure without disrupting the fold, and therefore potentially the function. The MutModel program calculates the number of steric clashes caused by introducing a mutant residue in a protein structure (Section 4.3). Residues that cannot insert into the native structure without clashing with three or more other atoms are identified. The model uses the simple assumption that two residues clash if any atomic centres are within 2.50Å of each other.

A P53 mutant is shown in Figure 3.9. 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 MPP (Shih *et al.*, 1985; Snow and Amzel, 1986), see Section 4.3), the best orientation of the mutant sidechain clashes with 27 other native atoms. Figure 3.9a shows that the native glycine fits neatly inside the structure, while the tryptophan residue in Figure 3.9b protrudes out of the structure, inhibiting the formation of the native fold, thus inducing for the disease phenotype.

As part of work in this thesis, this simple 'Boolean' method counting the number of clashes, has been replaced with a new method using a modified MutModel program which introduces a full evaluation of van dar Waals and torsion energy (see Chapter 4).



(a) Native 2ata:A (b) Mutant G279W 2ata:A

Figure 3.9: Residues found to clash with other existing residues.
PDB ID 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.

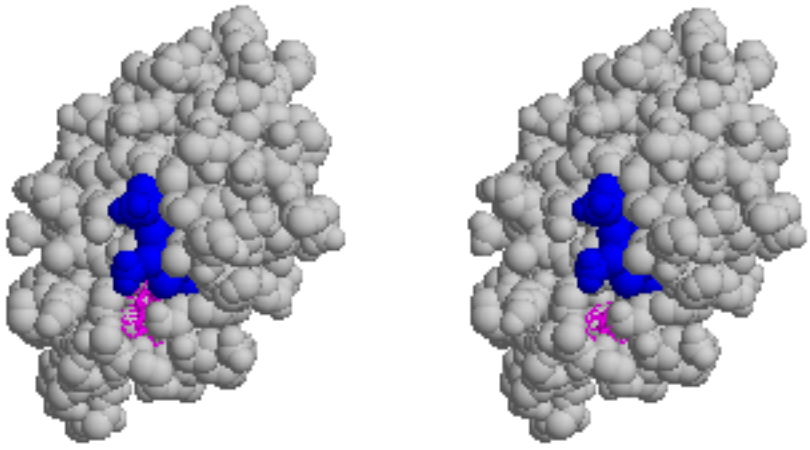
3.7.2.5 Introducing a void in the core

Large to small residue mutations may create a void in the structure. A void is defined as a cavity within a protein structure that is not accessible to bulk solvent. Voids are both enthalpically and entropically unfavourable, thus potentially having an effect on protein stability. The AVP software 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 default probe radii of 1.4Å and 0.5Å used respectively.

To obtain these data, all mutant structures must first be pre-processed using MutModel. AVP is run on each individual structure with a compute time dependent on the size of the protein chain being analysed. This can vary from a few seconds to several minutes.

While the analysis of voids is unchanged in the work described in this thesis, changes to the MutModel program enhance the positioning of side-chains and, therefore, assessment of voids (see Chapter 4).

For example, the mutation P42S in the human haemoglobin Beta chain [UniProtKB:P68871/HBB_HUMAN] is reported to be associated with cyanosis, moderate reticulocytosis and mild anaemia (Stabler *et al.*, 1994). The void analysis shows that this mutation introduces a void (see Figure 3.10b), likely to lead to some collapse of the structure in this region.



(a) Native 1qsh:D (b) Mutant P42S 1qsh:D

Figure 3.10: Creating a void or crevice.
PDB ID 1qsh, chain D. Replacing the native phenylalanine residue at position 42 with a serine residue (as shown on the right) creates an internal void (for details, see Section 3.7.2.5) 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 interface analysis (i.e. affecting binding to the haem ligand).

3.7.2.6 Mutations to binding residues

As a refinement to 'interface' residues, mutations to residues that form hydrogen bonds, (as described by Baker and Hubbard (1984)), and non-bonded interaction with a ligand or another protein chain are identified by parsing the XMAS formatted PDB files. These will be a subset of the 'interface' residues identified from a change in solvent accessibility. '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. As such, non-bonds include Van der Waals forces and electrostatic interactions.

The tumour suppressor protein P53 [UniProtKB:P04637/P53_HUMAN] is mutated in half of human cancers (Greenblatt *et al.*, 1994; Sidransky and Hollstein, 1996; Lane and Fischer, 2004). Chain B of the P53 structure (PDB ID 1tsr (Cho *et al.*, 1994)) is shown in complex with DNA in Figure 3.11 in grey; residues identified as 'functional' by all of the binding, interface and UniProtKB/Swiss-Prot FT 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.

MMDBBIND (Salama *et al.*, 2001)[6] was used in previous builds of SAAPdb and early stages of this thesis. It has not been updated since 2010, and the amount of interfaces deposited in it is fairly small compared with similar databases and, for that reason, this category is considered obsolete and removed from the SAAP analysis.

3.7.2.7 Disrupting disulphide bonding

Disulphide bonds are covalent cross-links that form between cysteine residues in polypeptides and help to stabilise protein structure (Figure 3.12). Mutations that remove disulphide bonding cysteines may alter protein stability and therefore compromise native protein function.

As with the incorporation of binding residue data into the pipeline, a Perl script identifies potential disulphide bonding cysteine residues in PDB files. First, all cysteine residues are identified. Secondly, each pair of cysteine residues is assessed to determine if they are involved in a disulphide bond. To form a disulphide bond, residues must satisfy the following criteria (Hazes and Dijkstra, 1988) as shown in Figure 3.12.

[6] MMDBBIND is an assimilation of the three-dimensional structure information described by Entrez's MMDB database (Wang *et al.*, 2007) and the mmCIF PDB chemical component dictionary (Feng *et al.*, 2003), and is part of the larger BIND database (Bader *et al.*, 2001)

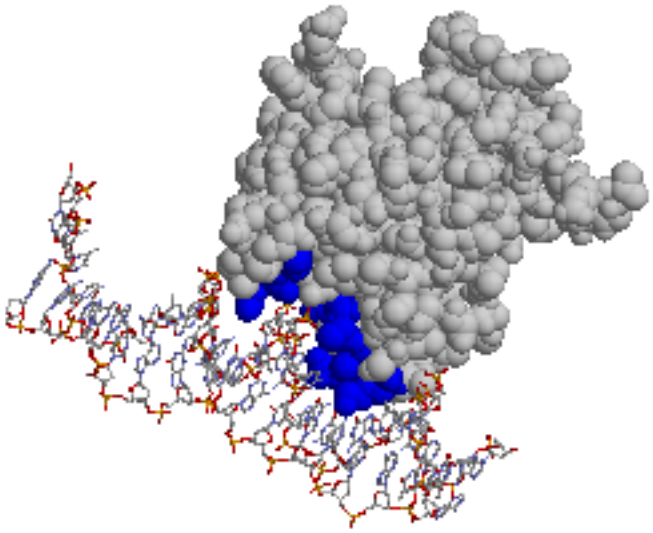


Figure 3.11: Binding residues in P53.
PDB ID 1tsr, chain B (in grey and blue) in complex with DNA. Binding residues, as defined by the binding (see Section 3.7.2.6), interface (see Section 3.7.2.14), ImPACT (high conservation) and UniProtKB/Swiss-Prot FT analyses (see Section 3.7.2.12) are highlighted in blue.

- $S\gamma_1$–$S\gamma_2$ bond length should be $\leq$2.50Å
- $C\beta_1$–$S\gamma_1$–$S\gamma_2$ and $C\beta_2$–$S\gamma_2$–$S\gamma_1$ bond angles should be 104° ± 10%

Standard trigonometry calculations and methods from the Perl `Math::Trig` module are used to calculate distances and angles from PDB coordinates. Each protein structure described in SAAPdb is analysed to identify potential disulphide bonding cysteine residues using isolated PDB chains. Interchain disulphide bonding is identified based on the interface analysis. All candidate sulphur atoms from cysteine residues are extracted from the PDB file and assessed as candidates using the method described above. The script then 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 potential disulphide bonding cysteine.

The example in Figure 3.13 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.

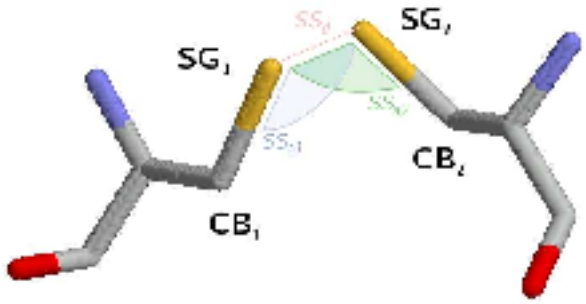
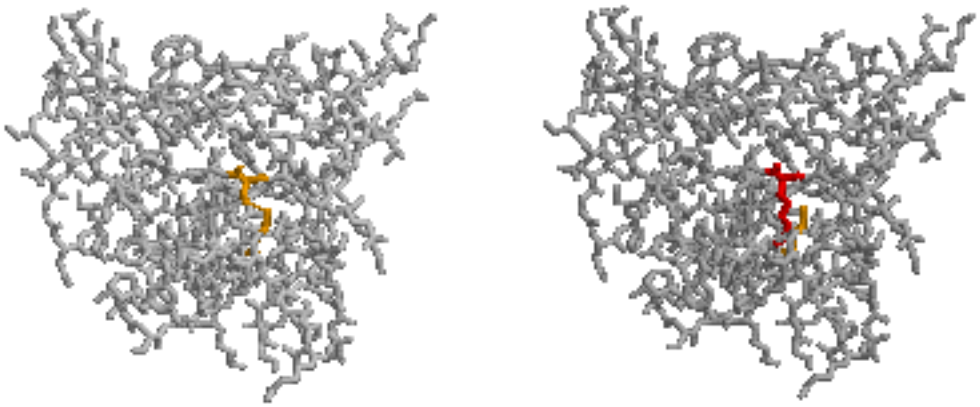


Figure 3.12: A disulphide bond between C6 and C127 of lysozyme (7lyz), showing SS_d, SS_{a1} and SS_{a2}



(a) Native 1hI5:I (b) Mutant C146R 1hI5:I

Figure 3.13: Disrupting disulphide bonding.
PDB ID 1hI5, chain I. A disulphide bond exists between CS7 and C146 in chain I of 1hI5, see native structure on the left and is identified by the SSCEOM analysis (see Section 3.7.2.7). This disulphide bond is also described as UniProtKB/Swiss-Prot features (Section 3.7.2.12, identified by ImPACT and identified by the clash analysis (see Section 3.7.2.4). A mutation replacing C146 with an arginine (see mutant structure on the right), with the mutant arginine highlighted in red.

3.7.2.8 Mutations to cis-prolines

Owing to the partial double bond character of the peptide bond between the carboxylate carbon and the amine nitrogen, rotation around this bond is restricted. Energetically, this favours conformations where the C_α, O, C, N', H', C_α' atoms exist in the same plane. Atoms may be arranged in the *trans* conformation where $\omega \simeq 180^\circ$ or the *cis* conformation where $\omega \simeq 0^\circ$. The vast majority of peptide bonds are found in the trans conformation because 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). 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).

3.7.2.9 Introducing a charge shift in the core

Charged residues are often functional in protein structures as they introduce electrostatic interactions within the protein and between protein and substrate (Torshin and Harrison, 2001). Arginine and lysine, and to a lesser extent histidine, are positively charged residues that often form salt bridges with negatively charged groups. Conversely, aspartic acid and glutamic acid are negatively charged residues that can form salt bridges with positively charged groups. In the protein core, these almost invariably occur as pairs of oppositely charged residues (Torshin and Harrison, 2001). Removing or introducing a charged residue from or into the protein core may therefore destablize or disrupt protein conformation and cause a deleterious phenotype.

Charged residues at the surface of a protein are solvated and therefore do not need to occur as charge pairs. Rather than having a large structural rôle, a charged residue on the surface may interact with other molecules and therefore be critical to protein function. However, these residues should be identified by the binding analyses, which have been described previously. Since charged amino acids at the core are more structurally important, the following section outlines how mutations affecting these are incorporated into the pipeline.

Incorporation of data regarding charged amino acid mutations does not require any additional processing; all required data are parsed from the XMAS files (see Section 2.2.3). A

Table 3.4: 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

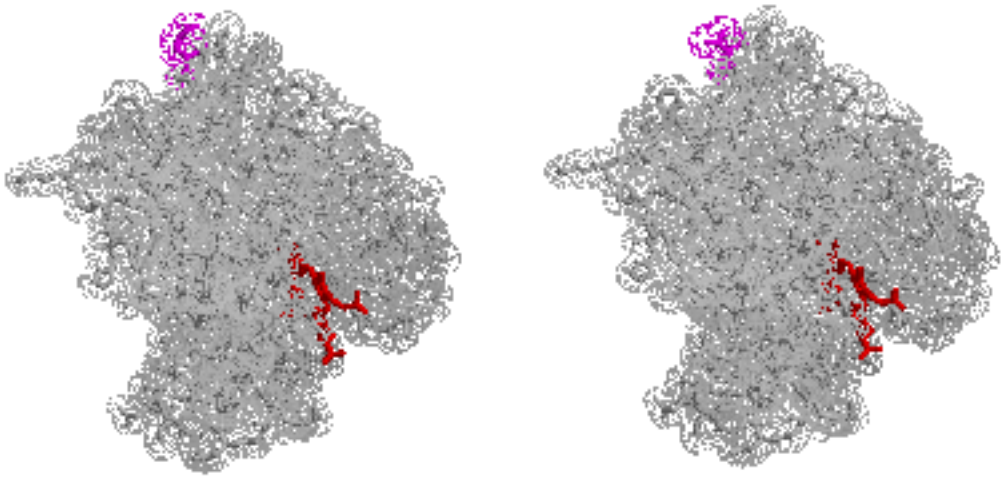
PostgreSQL function calculates the 'charge shift' occurring as a result of a mutation. Table 3.4 shows the charge shift values for mutations between all possible pairs of charged and neutral amino acids. With this PostgreSQL function, it is possible to implement this analysis as a single SQL query, where mutations with a non-zero charge shift occurring in the core (where the relative, monomer accessibility statistic is $\leq 5\%$) are easily identified as introducing a buried, unsatisfied charge.

3.7.2.10 Introducing hydrophobic residues on the protein surface

Hydrophobic residues tend to be concentrated in the protein core, away from the solvent-interacting surface (Branden and Tooze, 1999). Replacing a hydrophilic residue with a hydrophobic one (phenylalanine, isoleucine, leucine, methionine, valine and tryptophan) at the surface could result in protein aggregation or misfolding as well as destabilizing the protein and therefore a deleterious phenotype (for example, the E6V mutation that causes sickle-cell anaemia (Moo-Penn *et al.*, 1977)).

All data required to identify the hydrophobic mutations on the surface, i.e. native/mutant amino acids and accessibility statistics are recorded in the XMAS file which is parsed to populate the structural analysis database table. 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.

An example of introducing a hydrophobic residue on the surface of the protein is shown in Figure 3.14. The mutation seen here is the E6V mutation that causes sickle cell anaemia,



(a) Native 1hbb:D (b) Mutant E6V 1hbb:D

Figure 3.14: Introducing hydrophobic residues on the surface.
PDB ID 1hbb, chain D. A mutation from glutamic acid to valine at residue 6 introduces a 'sticky' hydrophobic residue on the surface of 1hbb. Residue 6 is highlighted in magenta and the haem ligand is highlighted in red. This is the mutation that causes sickle cell anaemia.

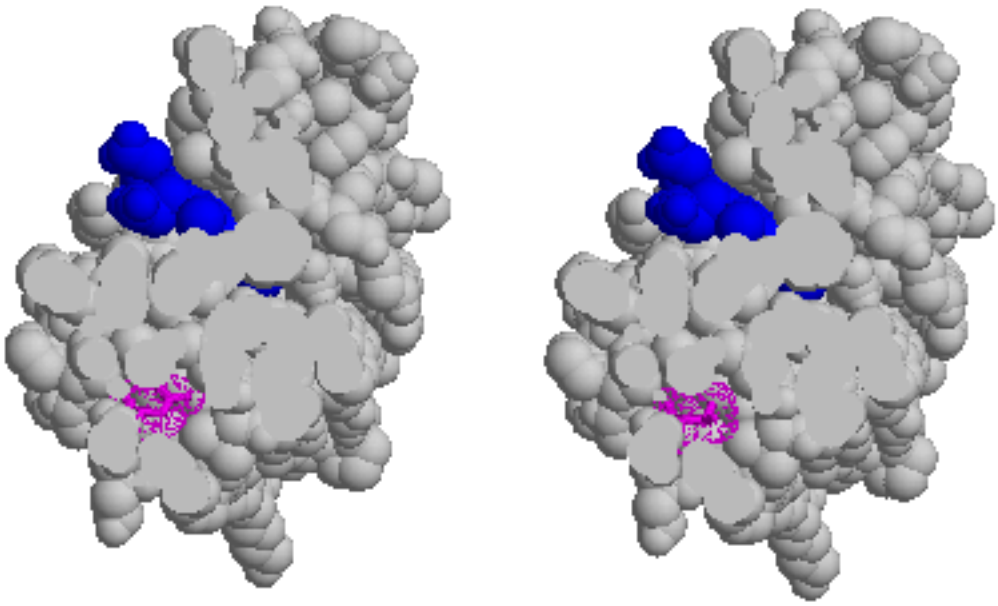
where the 'sticky' hydrophobic patch owing to the mutant valine residue causes aggregation and subsequent deformation of erythrocytes.

3.7.2.11 Introducing hydrophilic residues in the protein core

The vast majority of buried hydrogen bonding capable side-chains do actually participate in hydrogen bonding. Thus replacing a buried hydrophobic residue with a buried hydrophilic residue is likely to destabilise the native protein structure (McDonald and Thornton, 1994).

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

Figure 3.15 shows the crystal structure of human haemoglobin (PDB ID 1rly). Here, the mutation V54D introduces a buried, hydrophilic charge by replacing a hydrophobic valine residue with the negatively charged hydrophilic aspartic acid.



(a) Native 1r1y:D (b) Mutant V54D 1r1y:D

Figure 3.15: Introducing a buried, unsatisfied charge.

PDB ID 1r1y, chain D. The buried charge analysis identifies the V54D mutation in 1r1y as replacing a neutral valine residue in (a) with a negatively charged aspartic acid in (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.

3.7.2.12 UniProtKB/Swiss-Prot features

UniProtKB/Swiss-Prot annotates sequences to describe the function(s) of the protein, any post-translational modifications, domains and sites, structural conformations, associated diseases and sequence conflicts. The database uses a controlled vocabulary and the feature table (FT) to annotate regions of interest in protein sequences. A small number of these annotations are manual, however many more are transferred 'by similarity' from another annotated protein. Many of the annotations provide insight into the residues critical for function or stability and thus indicate which mutations are likely to have adverse effects.

The UniProtKB/Swiss-Prot flat-file data are parsed and residues annotated with FT (feature) tags are identified. As the aim is to explain the effects of mutations, a subset of features that have the potential to affect protein stability or function are relevant. These are described in Table 3.5.

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 3.5). When the start and end number are the same, a single residue is being described. When parsing the UniProtKB/Swiss-Prot data, the two numbering schemes are dealt with accordingly, 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. All feature residues that have been extracted are then stored in the database. To date (22-Jan-2014), 192 776 118 residues are annotated in 225, 339 UniProtKB/Swiss-Prot records. The PDBSWS mapping (Martin, 2005) that is imported into SAAPdb allows these annotations to be mapped to PDB files.

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 data file and stored in the database, this analysis can be implemented by a simple PostgreSQL query.

Upon closer inspection, some feature annotations appear to be unreliable. Figure 3.16 shows the structure of human P53 (PDB ID 1tsr) in complex with DNA (highlighted in red). Residues near DNA (within 10Å) are shown in yellow. The corresponding protein record ([UniProtKB: P04637P53_HUMAN]) describes residues 102-292 as DNA-binding. These residues are shown in dark blue and yellow in Figure 3.16, having been mapped onto

Table 3.5: UniProtKB/Swiss-Prot feature 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

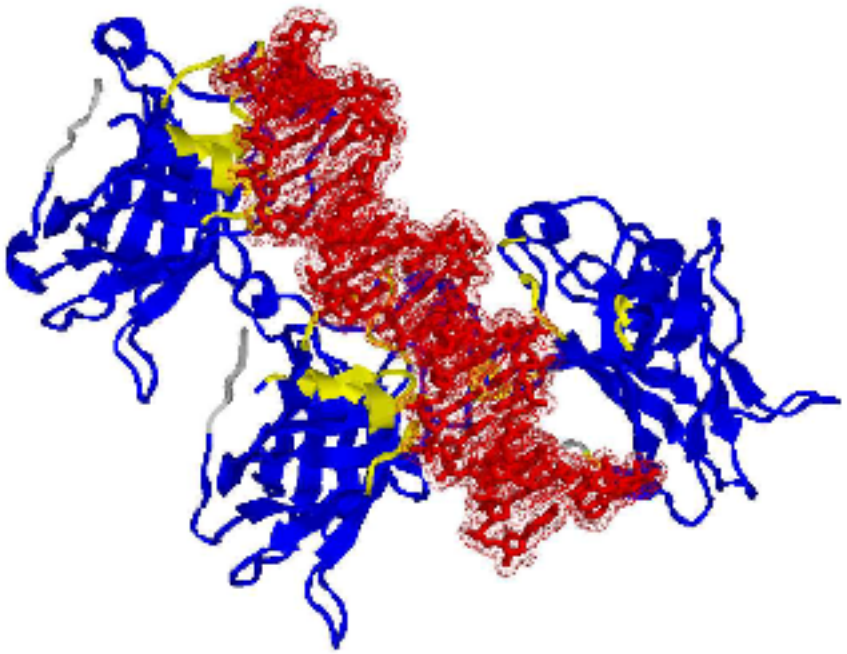


Figure 3.16: An example of coarse-grained UniProtKB/Swiss-Prot FT annotation.
The figure shows the structure of Human P53, PDB ID 1tsr; the DNA to which P53 binds is shown in red with the protein chains shown in blue and grey; the yellow residues indicate those within 10Å of the DNA; the blue residues are those annotated as DNA_BINDing by UniProtKB/Swiss-Prot; even using the very generous distance threshold of 10Å, the UniProtKB/Swiss-ProtDNA_BINDing annotation is very coarse-grained.

the protein structure using PDBSWS (see Section 2.1.4), and comprise most of the protein chain. It is clear from this example that the UniProtKB/Swiss-Prot functional annotation is too coarse-grained having annotated the whole domain and consequently many residues remote from the DNA (i.e. distant by > 10Å) are annotated as `DNA_BIND`ing.

3.7.2.13 Mutating conserved residues

The presence of highly conserved amino acid residues at analogous position in protein sequences indicates that those residues are likely to be critical for function. Where it is not possible to identify the structural effect of a disease mutation, functionality may be inferred from sequence conservation. 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.

Recognition of the importance of this phenomenon led to the development of a novel method (ImPACT) for identifying highly conserved residues, which accounts for species diversity and protein-global conservation patterns (McMillan, 2009).

Using the UniProtKB accession number, all functionally equivalent proteins (FEPs) i.e. orthologous annotated in SwissProt having the same function in UniProtKB/Swiss-Prot are identified using POSTA (McMillan and Martin, 2008). A multiple sequence alignment (MSA) is generated by aligning the FEPs using MUSCLE (Edgar, 2004).

Each protein can be processed independently, allowing ImPACT analyses to be distributed across the local 20-core grid. For each MSA, the ImPACT threshold (McMillan, 2009), target protein and size (i.e. number of sequences) is recorded, and for each residue in each MSA, the position (with respect to the target protein), the species similarity conservation score and whether or not this exceeds the ImPACT threshold for the MSA is recorded.

The original ImPACT code by Lisa McMillan has been corrected and modified, and full instructions on how to update ImPACT's species, similarity matrix have also been written as a part of this thesis.

3.7.2.14 Mutations at the interface

The interface can be any regions between PDB chains, or between chains and ligands. Residues at these interface sites are critical in forming biologically relevant multimers. Thus, mutating these residues is likely to disrupt the native structure and may be dele-

terious. Interface residues are identified by a > 10% ΔASA (accessible surface area) in the monomer state as opposed to the multimer state. The ASA is calculated using a local implementation of the Lee and Richards algorithm (Lee and Richards, 1971) and obtained from the XMAS files.

Figure 3.17 shows the structure of human super-oxide dismutase [UniProtKB:P00441/SODC_HUMAN], (PDB ID 2c9s (Strange *et al.*, 2006). Mutations to 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 analyses are shown in darker blue and red respectively, with their Van der Waals surface indicated with dots. This illustrates interface residues at both the interchain interface and ligand binding sites.

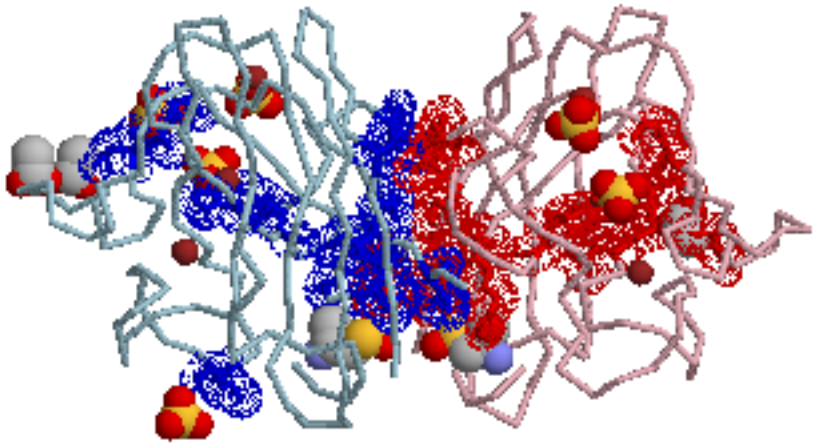


Figure 3.17: Residues identified at the interface.
PDB ID 2c9s, chains A (*in blue and light blue, on the left*) and F (*in red and pink, on the right*); *ligands are shown in spacefill using the CPK colour scheme. Residues identified by the interface analyses (see Section 3.7.2.14) are shown in darker 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.*

The assembly of multiple tertiary protein structures into biologically relevant multimers is described as the quaternary structure. Residues at the quaternary interface will be critical to the native protein fold. The 'interface' and 'binding' analyses (Sections 3.7.2.14 and 3.7.2.6 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 PDB provides comprehensive tertiary structures of proteins, yet it is often misleading in terms of quaternary structure information. The asymmetric unit (ASU) is the smallest

unique unit in a protein crystal structure to which symmetry operations can be applied; however, it is not indicative of the biological unit, i.e. the quaternary structure of the protein as found *in vivo*. Although PDB files sometimes specify the biological units (BUs) provided by the researchers (in headers), this information is scarce and often experimentally unverified.

Henrick and Thornton (1998) developed the Protein Quaternary Structure[7] (**PQS**) server, an automated system that builds BUs from ASUs provided in PDB files. PQS was used in previous builds of SAAPdb and early stages of this thesis, given the discontinuation of PQS in 2010 it was replaced with, Protein Interfaces, Surfaces and Assemblies (**PISA**)[8], developed by Krissinel and Henrick (2007) one of the most widely used automated tool for the prediction of quaternary structure which outperformed PQS. Based on thermodynamic stability calculations, PISA automatically detects macromolecular assemblies in PDB entries and predicts different BUs from PQS for 23% of structures, often resulting in a smaller assembly than PQS.

Figure 3.18b shows the complete hypothetical quaternary structure of the human poliovirus capsid protein (PDB ID 2plv) with the original PDB structure shown in Figure 3.18a. Although *some* of the binding contacts will be recognised by the binding and interface analyses, many will be lost (compare Figures 3.18a and 3.18b with respect to the number of interface surfaces).

3.8 Summary of SAAPdb rebuilding

3.8.1 SAAPdb legacy and update

The last full update of SAAPdb before the current one was in January 2008. Inconsistencies between data-sets were identified during an initial analysis at the outset of this project. To remedy these inconsistencies, a rebuild of SAAPdb and other support databases (i.e. the database of functionally equivalent proteins from SwissProt (FOSTA), and the sequence conservation scoring method that uses a species similarity matrix (ImPACT)) was necessary. This was a labour intensive task requiring substantial testing and rewriting of all the code involved in data collection, database creation, and some structural analysis. Full documentation on the update process were also written.

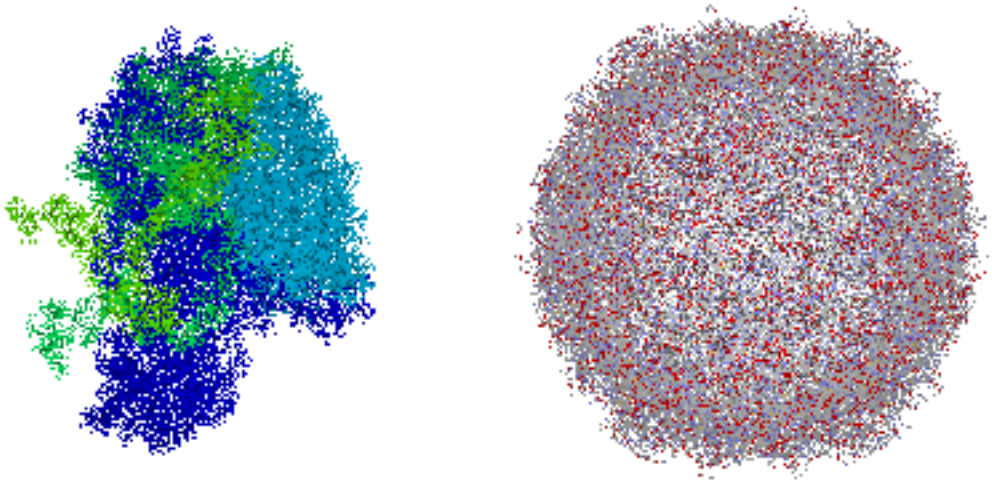
[7] http://www.ebi.ac.uk/pdbe/pqs/

[8] http://www.ebi.ac.uk/msd-srv/prot_int/pistart.html

Table 3.6: SAAPdb categories.
The horizontal line separates structural categories from the sequence-based one.

Category	Effect of mutation
PQS[a]	Affecting residues in the interface with a different protein chain or ligand identified from a PQS file (and therefore more likely to reflect biologically relevant interactions) by a change in solvent-accessibility – PQS replaced with PISA in later stages.
binding[b]	Affecting residues involved in specific binding interactions (a hydrogen bond, salt bridge, or packing interaction) with a different protein chain or ligand.
MMDB[b]	Affecting residues in contact with a ligand, according to the MMDB database.
sprotFT[b]	Residues annotated in SwissProt Feature records as having a functional significance.
proline[c]	Mutations to proline where the backbone angles are restrictive.
glycine[c]	Mutations from glycine where the backbone angles are restrictive.
clash[c]	Causing a clash between atomic radii of the neighbouring residues.
cisproline[c]	Mutations from a *cis*-proline.
hbonding[d]	Causing the disruption of hydrogen bonds between residues.
void[d]	Causing an internal void $\geq$275Å^3 to open in the protein owing to the substitution with a smaller residue.
corephilic[d]	Introducing a hydrophilic residue in the protein core.
surfacephobic[d]	Introducing a hydrophobic residue on the protein surface.
buriedcharge[d]	Introducing an unsatisfied charge in the protein core owing to the substitution with, or of, a charged residue.
SSgeometry[d]	Causing the disruption of a disulphide bridge.
struc_explained	Explained by any of the categories listed above.
highcons[e]	Affecting residue with highly conserved sequence, according to IMPACT (McMillan, 2009)
explained	Explained by any of the categories listed above.

[a]Interface-damaging; [b]Functionally-impairing; [c]Folding (fold-preventing); [d]Instability (destabilizing); [e]Sequence conservation.



(a) The PDB structure of Human poliovirus capsid protein

(b) The hypothetical assembly of Human poliovirus capsid proteins

Figure 3.18: Quaternary structure information from PISA.
(a) shows the PDB representation (PDB ID 2plv) of the Human poliovirus capsid protein which has four chains. The biologically relevant structure, as assembled by PISA is shown in (b).

The legacy system inherited for this project suffered from several weaknesses in its design. These included different software components coded in different programming languages (written by different people at different stages of the development of SAAPdb); untested and malfunctioning computer code, identification of errors and subsequent fixing (debugging); software component version incompatibilities; and inadequate documentation of the system.

There has been considerable effort to improve the code for updating SAAPdb. A summary of the datasets comparing the old and new builds of the database was shown in Table 3.1.

3.8.2 Updated SAAPdb data analysis

The SAAPdb web-server contains fourteen structural analyses and one sequence-based analysis (Martin *et al.* (2002), Cuff and Martin (2004)), shown in Table 3.6, all aiming to show how SAAPs are likely to affect protein structure: in particular interfaces with other proteins, functional sites, folding and stability of the mutated protein. Each analysis is implemented as a separate `SQL`, `Perl` script or `C` program and will output a positive ('likely to affect') or negative ('likely not to affect') result for every SAAP in every category. Therefore the SAAPdb result for a mutation can be viewed as a vector of binary or Boolean values (1

for $\surd$ and 0 for $\times$), as shown in Figure 3.19.

Only mutations mapped to solved protein structures can be assessed, therefore it is not possible to analyse all known mutations. Of the amino acid mutations in OMIM, approximately 65% can be mapped to structure. In addition, approximately 32% of 'valid' SNPs from dbSNP that result in an amino acid change, map to structure. Consequently, the coverage of the analysis is currently somewhat limited. However, clinically relevant proteins tend to be key targets for structural studies, so it is expected that these statistics will improve in the future. In addition some proteins have several resolved structures. In these cases, the effects of the mutations in all available structures are analyzed.

After the task of rebuilding SAAPdb was completed, analysis of the data in SAAPdb shows local structural effects for PDs more often than for SNPs, which are significantly underrepresented in all categories except 'surphacephobic', 'corephillic' and 'cisproline' mutations that are more common in SNPs than in PDs. Figure 3.20 shows a comparison of structural effects seen for SNPs and PDs between the previous SAAPdb build (Hurst *et al.*, 2009) and the current one. In the previous dataset only 'surfacephobic' mutations were more common in SNPs (Figure 3.20). Another notable finding is that structural effects observed for PDs tend to be more 'severe' than those seen for SNPs.

In summary, the number of SNPs in the database has risen by 41% and the number of PDs by 36%. This has been aided by the inclusion of two new sources of mutation data. Other sources of mutation data have been considered including HGMD and SwissProt Variants (SwissVar). However, HGMD data are only available to registered users meaning that the data have not been reproduced in our database. In addition, the annotation of disease status in SwissVar is not very reliable. For example, known PDs in G6PD are annotated as 'Natural Variants' of 'Unclassified' disease status. Other LSMBDs can be easily added (Claustres *et al.*, 2002), but as explained below, the SAAPdap pipeline version of the system is now implemented to allow users to analyse novel mutations. SAAPdap is now regarded as our primary resource.

3.8.3 The future of SAAPdb and The Single Amino Acid Polymorphism Data Analysis Pipeline (SAAPdap)

SAAPdb was designed to be a regularly updated pre-calculated resource. However, the database has proved very difficult to maintain and changes in licensing of OMIM data mean

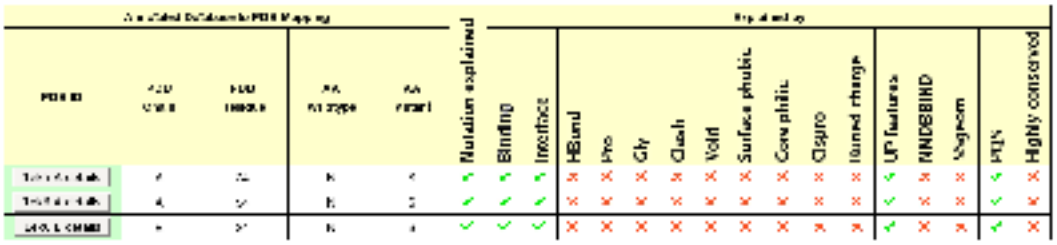
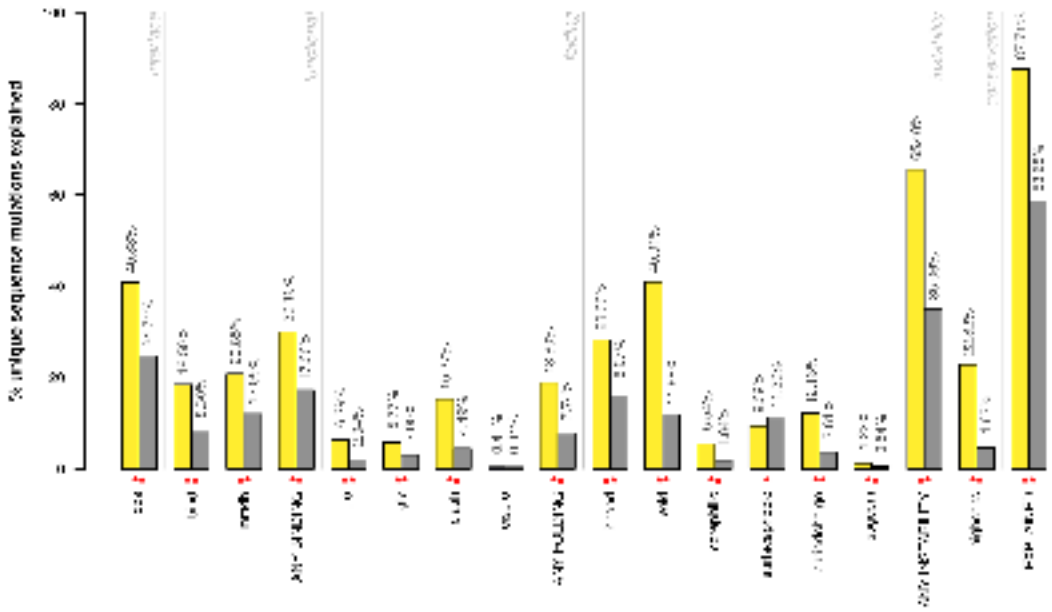


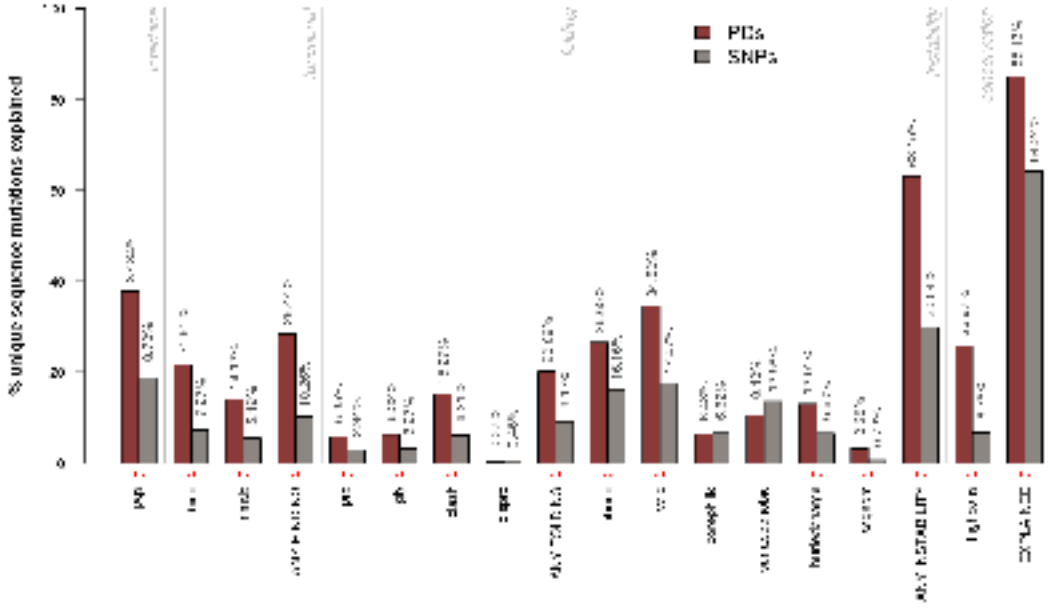
Figure 3.19: Structural effects assigned by SAAPdb to a PD in human UDP-galactose 4-epimerase.
SAAPdb `http://bioinf.org.uk/saap/db/` was queried for accession number Q14376 and N43S was chosen as a representative pathogenic mutation. The mutation is mapped to several residues in different protein structures (only the top three are shown here). The analysis summarised in a vector shows that this mutation is located in a binding and interface site residue, and it carries a functional identifier.

that OMIM may no longer be able to be used as the primary source of PDs. In addition, with the increasing routine use of high- throughput sequencing methods to detect mutations, the analysis of mutations is increasingly undertaken by individuals.

Consequently the value of SAAPdb has diminished and SAAPdap (Single Amino Acid Polymorphism Data Analysis Pipeline) has been implemented as a replacement. SAAPdap is a complete re-write of the SAAPdb pipeline by Andrew Martin using JavaScript Object Notation (JSON) for data storage. JSON is a text-based, publicly available language designed for human-readable data interchange. JSON uses two structures, an ordered list of values, (an array or sequence) and a collection of values or pairs (an object or associative array).



(a) Previous SAAPdb build



(b) Current SAAPdb build

Figure 3.20: Profiling SAAPs with respect to local structural effects.

Table 3.6 explains the individual effects meaning (pqs, bind, mutndb,pro, gly, clash,cisprohbond, void, corephilic, surfacephobic,,buriedcharge, ssgeom, highcons). ANY BINDING: the mutation is positive for bind and/or mutndb; ANY FOLDING: the mutation is positive for pro, gly, clash and/or cis-pro; ANY INSTABILITY: the mutation is positive for hbond, void, corephilic, surfacephobic, buriedcharge and/or ssgeom; EXPLAINED: the mutation is explained by at least one of the above analyses. Different 'classes' of local structural effect (i.e. interface, functional, folding, instability and conservation) are separated by pale grey vertical lines. Yellow and red bars denote results for PDs in (a) and (b) plot respectively, grey bars denote results for SNPs. Precise percentages are given above the corresponding bar. Statistically significant results are denoted with red stars (two where $p < 0.01$ and one where $p < 0.05$).

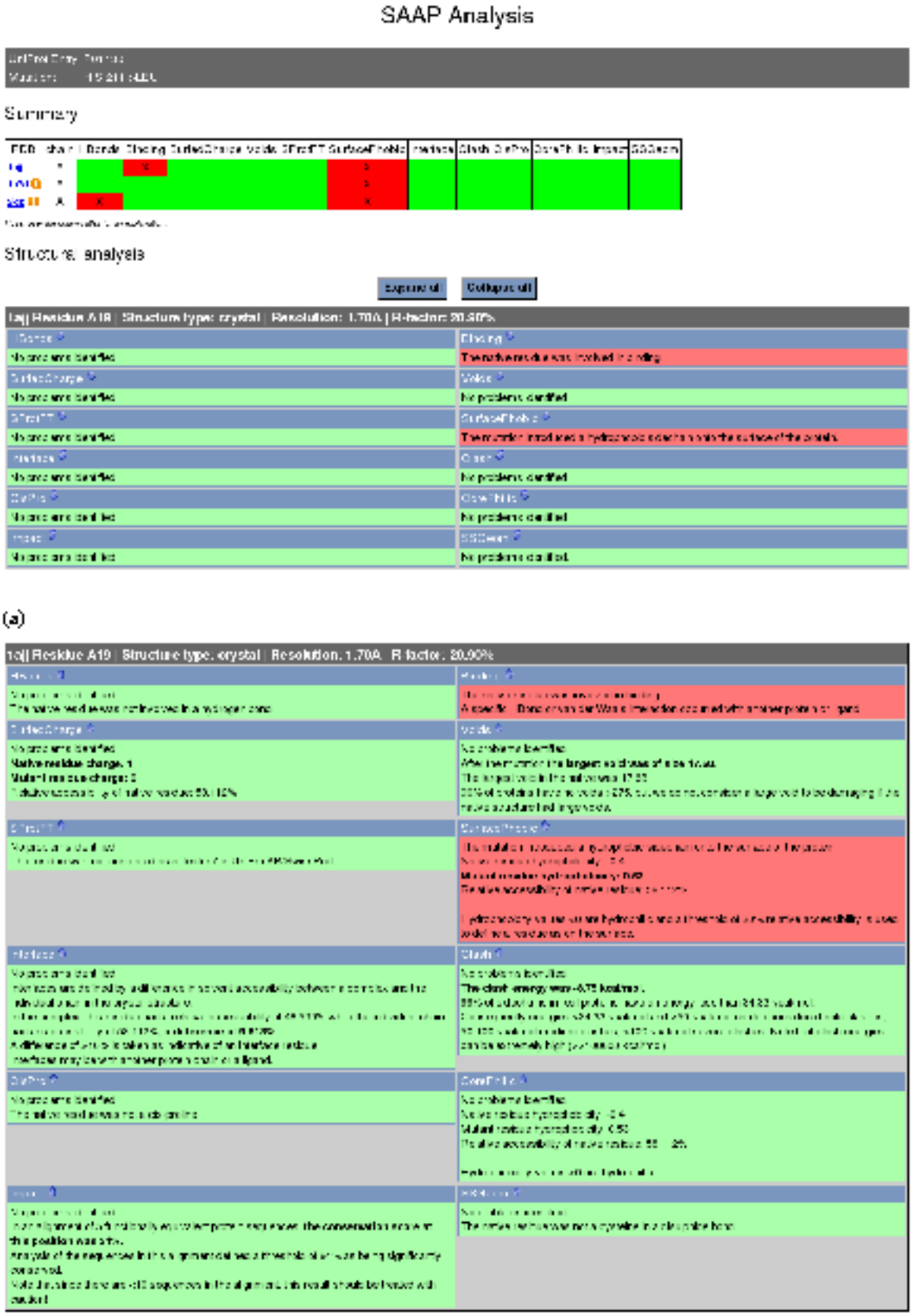
While it is no longer being updated, SAAPdb remains the most extensive database of SAAPs and their structural effects. A large and expanding body of literature exists in the field of protein structure-function analysis in relation to disease phenotypes. SAAPdb contributes to the current understanding of disease-causing mutations and ultimately the treatment of the resulting pathological conditions.

SAAPdap uses a plugin architecture, making use of new non-Boolean analyses (described in Chapters 4 and 5). While SAAPdap still indicates whether a mutation is likely to have a detrimental effect on structure using cut-off values, continuous values are also provided for each of the analyses. Results from the SAAPdap pipeline are presented as shown in Figure 3.21a. Results are summarized at the top where the effects on each structure to which the mutation maps are shown. Below, the analyses of structural effects on each structure are presented and these can then be expanded to provide more detail on the analyses as shown in Figure 3.21b. Analysis descriptions are much more comprehensive than was the case in SAAPdb to make the results easier to understand.

A web interface has been implemented by Andrew Martin to allow users to enter mutations for analysis. Because some of the analyses (especially the analysis of voids) is quite time consuming (taking several minutes), the web interface makes use of AJAX (Asynchronous JavaScript And XML) to update the user with the progress of the analysis. The submission page is available at `http://www.bioinf.org.uk/saap/dap/`.

3.8.4 Single Amino Acid Polymorphism prediction tool (SAAPpred)

SAAPdb data and SAAPdap analysis are used in this thesis to train machine learning methods to predict whether a novel SAAP will disrupt the native protein structure and induce a disease phenotype in a tool known as Single Amino Acid Polymorphism prediction (SAAPpred) (see Chapter 6).



(a)

(b)

Figure 3.21: Results pages from the new SAAPdap pipeline. (a) Summary and brief structural reports — hovering over any of the titles brings up a box to explain the meaning of the effect; (b) Expanded view of full structural analysis.

Chapter 4

Improvements to Clash and Void Analysis

∗ *Some of the work in this chapter has been published (Al-Numair NS, Martin ACR. 2013. The SAAP pipeline and database: tools to analyze the impact and predict the pathogenicity of mutations. BMC Genomics 14.3:1-11).*

In the original version of SAAPdb, all assignments of structural effects are Boolean, that is, any mutation either does, or does not, have a given effect. While Boolean assignment is appropriate in some cases (for example, a residue either is, or is not, annotated as a feature in UniProtKB/Swiss-Prot), in other cases, it relies on some cut-off (for example, energy, void volume, hydrophobicity difference) as described previously (Hurst *et al.*, 2009; Cuff *et al.*, 2006; Cuff and Martin, 2004; Martin *et al.*, 2002).

In earlier work done as part of a Master's degree, I showed that assigning a mutation as either having or not having a structural effect is very sensitive to precise structural details (Al-Numair, 2010) (see also Section 4.2). For example, where multiple structures are available for the same protein, one structure may indicate that a mutation has a value just below a cut-off while another structure has a value just above. This will result in conflicting assessments of whether a mutation is damaging or not. In this chapter and Chapter 5, real-number scores or pseudo-energies are now implemented for each appropriate structural effect. In particular, the analysis of clash and void is enhanced in this Chapter, while pseudo-energies are defined for analysis of from-glycine and to proline mutation in Chapter 5.

4.1 Introduction

This section introduces the definition of clash and void occurring in protein structures and the analysis used in the SAAP analysis to assess their effects.

4.1.1 Mutations that cause steric clashes

When mutations cause steric clashes in a structure owing to a larger mutant residue than in the native structure, this will disrupt or prevent correct protein folding and therefore affect the function of the protein (Martin *et al.*, 2002). Figure 4.1 shows an example of a small to large residue mutation, Gly122→Arg, in the human triosephosphate isomerase dimer structure, PDB ID 1WYI (Kitatani *et al.*, 2006). This mutation is known to increase thermo-sensitivity of the human protein (Mande *et al.*, 1994).

Originally, SAAPdb used very simple, fixed thresholds for defining potentially damaging effects. Both SAAPdb and SAAPdap use the MutModel program (Martin *et al.*, 2002) to calculate the steric clashes caused by introducing a mutant residue in a protein structure. In SAAPdb, a damaging clash was defined as any side-chain that has at least 3 van der Waals overlaps (of any degree) with a distance between atom centres less than 2.5Å (Martin *et al.*, 2002). However, using such a static threshold does not differentiate between two atoms that are slightly overlapping and two atoms that are largely occupying the same space. Using a more informative van der Waals energy calculation would refine the clash analysis and would be expected to improve predictive ability.

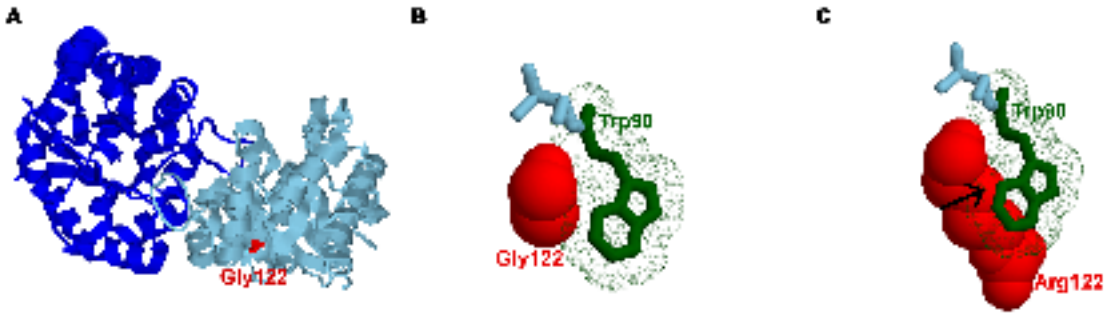


Figure 4.1: A mutation causing steric clashes and potentially affecting folding residue.
A) The position of Gly122→Arg is shown on the human triosephosphate isomerase dimer structure, PDB ID 1WYI. This mutation causes a *clash* and introduces a *buried charge*, and is know to increase the thermo-sensitivity of the human protein. B) Detailed view of the native Gly122 and Trp90 with which it interacts. C) The Gly122→Arg mutation causes atoms to clash, as indicated by the arrow (modelled structure).

4.1.2 Introducing a void in the core

Mutations that replace a large buried amino acid with a smaller one may introduce a void in the protein core. A void is defined as a cavity within a protein structure that is not accessible to bulk solvent. While a void introduces no physical barriers to correct folding, a void reduces the stability of the correctly folded form below that of unfolded or misfolded states (Hurst *et al.*, 2009). Figure 4.2 shows an example of a large to small residue mutation introducing a Phe173→Leu void mutation in glucose-6-phosphate dehydrogenase. In this instance, the void mutation causes neonatal jaundice.

The void calculation method (Cuff and Martin, 2004) calculates the volume of voids, assuming that no movements occur in the protein structure. The AVP software (Another Void Program) is used to identify and measure the size of internal voids in protein structures. AVP allows independent probe radii (default 1.4Å and 0.5Å) for definition of solvent and voids respectively. Obtaining these data requires significant processing: all mutant structures must be generated using the MutModel program 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.

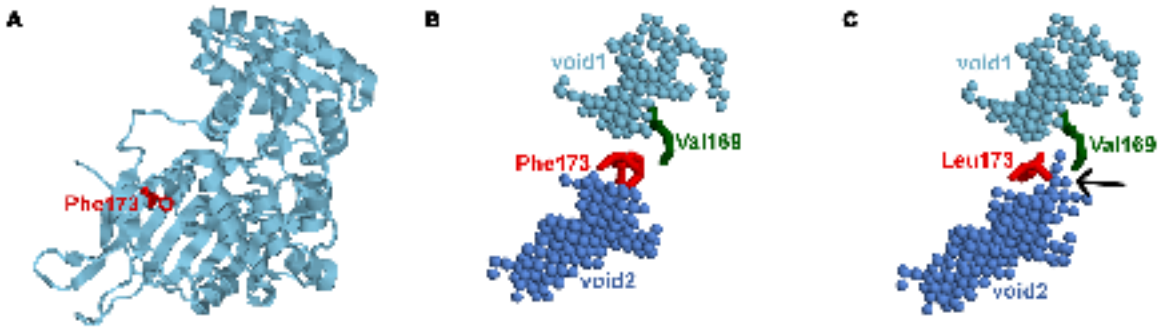


Figure 4.2: A mutation which increase the size of a void.
A) The Phe173 → Leu mutation is shown on the human glucose 6 phosphate dehydrogenase structure, PDB ID 2BH9. This mutation creates a *void* and causes neonatal jaundice. B) The native structure shows a void ('void2' introduced by a set of dark blue spheres) near Phe172. c) After mutation of Phe172 → Leu, the void 'void2' is enlarged to an extent judged to be destabilizing using the threshold described by Cuff and Martin, (2004). The additional void volume is indicated by the small dark blue spheres in 'void2' highlighted with the arrow.

In SAAPdb, a mutation is considered to be a damaging when it causes the creation of voids of volume > 275Å^3. As above, it is likely that movements of side-chains and backbone will occur to fill the void (at least partially). This static threshold was selected based on an

analysis of PDB structures that showed that the largest void in 80% of protein structures is $< 275Å^3$ (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. Considering each protein structure individually and calculating the native structure of voids based on its properties may aid estimation of the maximum void size that may be tolerated. SAAPdb maintains the same threshold for indicating a likely-damaging mutation, but additional void information (top 10 void sizes) is used for prediction in Chapter 6.

4.2 Analysis of sensitivity to structural details

This analysis began as part of a Master's degree (Al-Numair, 2010) and was updated as part of this research using the latest SAAPdb version (described in Chapter 3).

All mutation data analysed in this section were obtained from the SAAPdb PostgreSQL database using Structured Query Language (SQL) queries. The analyses were implemented using several Perl scripts. First, a Perl script used in the examination of the gathered data was written to determine the number of structures to which each mutation maps. This was used for SNPs and PDs[1] across all available mutations imported from SAAPdb, across all proteins. Another Perl script was written for counting mutations classified as unfavourable and mapped to at least 2 structures. The same script was used to calculate the fraction of structures in which a mutation was classified as unfavourable out of the total number of structures to which it maps.

$$F = \frac{n_u}{N} \tag{4.1}$$

Where n_u is the total number of structures in which the mutation is classified as unfavourable and N is the total number of mapped structures.

For example, if the number of structures to which a mutation maps is equal to 10 and the number in which a mutation is classified as unfavourable is 5, then F = 0.5. The same script analysed the mutations classified as unfavourable for each individual SAAP analysis type where at least one structure classified as unfavourable.

[1] As in Chapter 3, the SAAPdb terminology is used here - SNPs are believed to be phenotypically neutral while PDs are known to cause disease.

A further enhancement to the program above was to provide an option (`-nat`) to include no mutant structures, but only native structures from the PDB. To determine whether variation in structural classification frequencies resulted from poor resolution structures, a `-res` option was added to extract the resolution using `getresol`, an external program (ACRM, unpublished). The PDB entries were restricted either into high resolution ($\leq$ 2.0Å when `-res=H`) or low resolution ($>$ 2.0Å when `-res=L`).

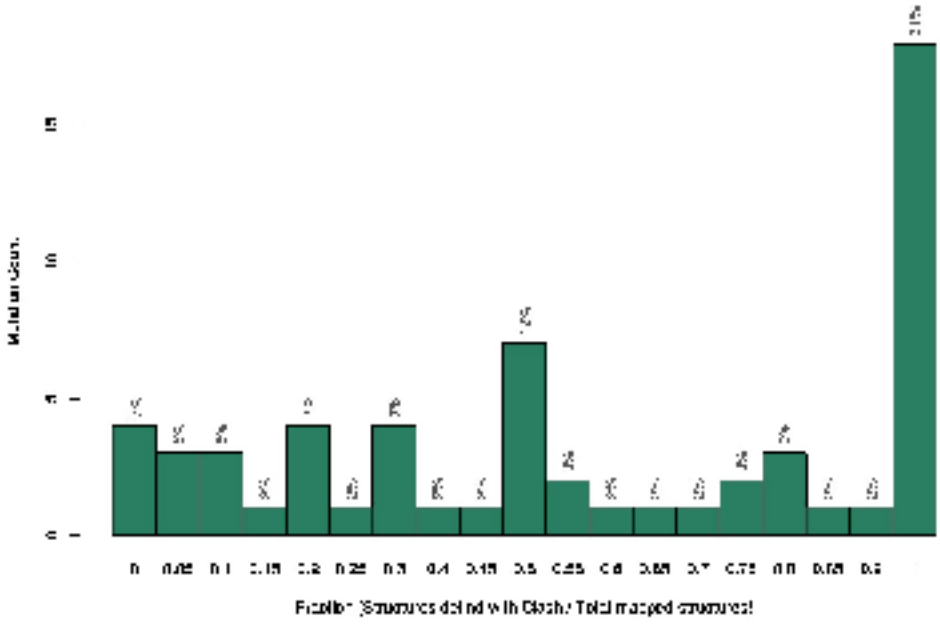
The Analysis of sensitivity to structural details was performed on all 17 SAAP analysis types shown in Table 3.6, where one or more mutations are classified as unfavourable and suggest how SAAPs might affect protein structure. In the previous work (Al-Numair, 2010), all structures (mutant and native) were investigated. After updating the program, the analysis repeated on native structures with high resolution to analyse the effects of mutations but not using mutant structures for this purpose.

4.2.1 Clash and void analysis

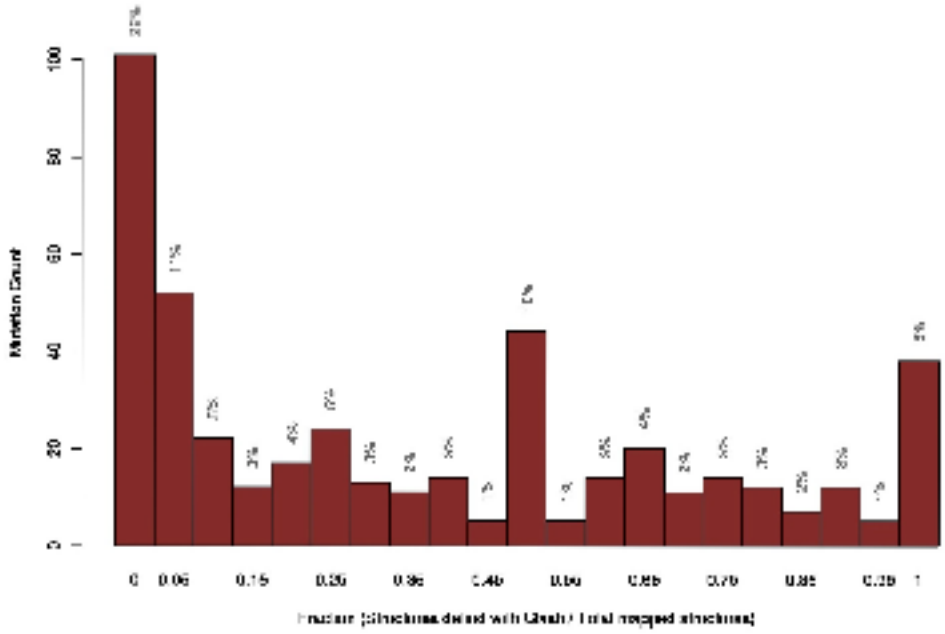
The distribution of fraction of structures in which SNPs and PDs are classified as causing clashes or voids (using the fixed cutoff) was plotted in Figures 4.3 and 4.4 respectively. If these measures were accurate predictors of pathogenicity, independent of precise structural details, one would expect to see the SNP mutation having a peak only at F = 0 (since one would expect these mutations not to have a significant effect on protein structure) while PD mutations would have a peak at F = 1. This is clearly not the case. SNPs causing clashes (Figures 4.3a) show a broad distribution, but surprisingly clearly skewed towards F = 1. This suggests that either the fixed cutoff for clashes is incorrect (classifying too many clashes as damaging) or that the use of a fixed cutoff is misleading or both.

An opposite surprising result is seen for PDs both causing clash and void (Figure 4.3b and 4.4b) where the distribution is skewed toward F = 0 instead of the expected F = 1. PDs causing clashes and SNPs causing voids show a much more even, broad distribution.

All four graphs show a peak around F = 0.5 (probably this is due to an artefact due to a larger amount of entries with just 2 structures) and cases where $0.5 \leq F \leq 0.95$ are likely to be instances where the clashes or voids have values very close to the cutoff and consequently a given mutation in some structures is classified as likely to be damaging while in other structures it is not classified as having any damaging effect. This reinforces the conclusion that a fixed cutoff is very sensitive to precise structure details.

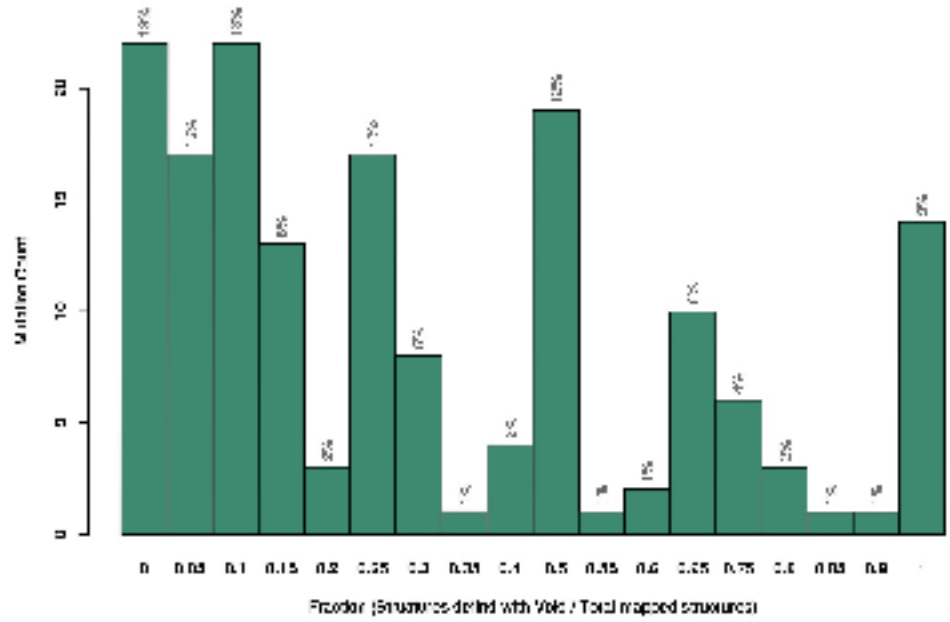


(a) SNP

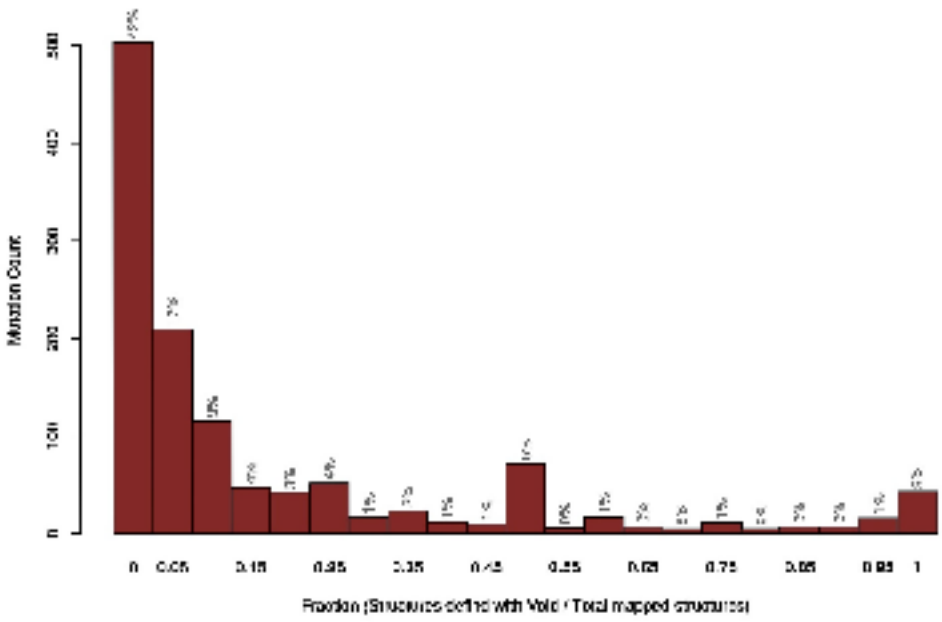


(b) PD

Figure 4.3: Boolean clash analysis
(a) The original Boolean clash method tested on 59 SNP mutations using native structures. **(b)** The original Boolean clash method tested on 453 PD mutations using native structure. A fraction of F = 0 represents no mutations classified as unfavourable, whereas F = 1 represents mutations classified as unfavourable. In **(a)** and **(b)**, each bar less than the next bar label (e.g. Bar 0.25 is $0.25 \leq F < 0.3$ and bar 0.95 is $0.95 \leq F < 1$), the bar labelled with one represents the fraction of structures where all are classified unfavourable.



(a) SNP



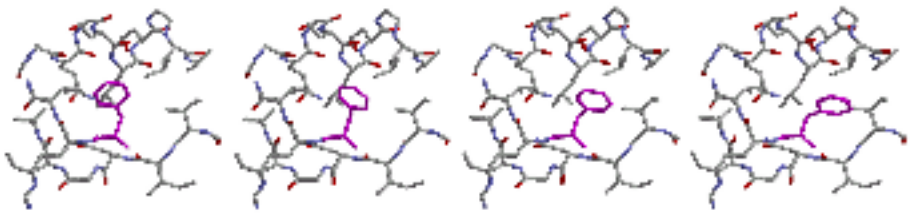
(b) PD

Figure 4.4: Boolean void analysis
(a) The original Boolean void method tested on 164 SNP mutations using native structures. **(b)** The original Boolean void method tested on 1211 PD mutations using native structure. A fraction of F = 0 represents no mutations classified as unfavourable, whereas F = 1 represents all mutations classified as unfavourable. In **(a)** and **(b)**, each bar less than the next bar label (e.g. Bar 0.25 is $0.25 \leq F < 0.3$ and bar 0.95 is $0.95 \leq F < 1$), the bar labelled with one represents the fraction of structures where all are classified unfavourable.

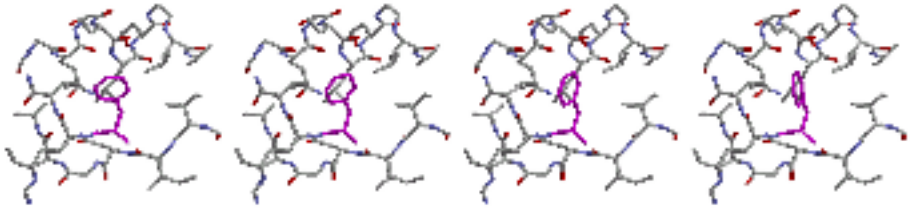
4.3 The MutModel program

MutModel is a program written originally by Dr. Andrew Martin, which performs side-chain replacements using the minimum perturbation protocol (MPP) (Shih *et al.*, 1985). During this calculation, side-chains are replaced and then spun around their Chi1 and Chi2 torsion angles to find the position that makes a minimal number of bad contacts (or 'clashes'). For the void and clash analyses described in this chapter it is necessary to generate a mutant structure. MutModel implements MPP to model a mutant residue into a native structure as follows:

1. The Maximum overlap protocol (MOP) (Snow and Amzel, 1986) is used to replace the side-chain, inheriting torsion angles from the native residue where possible;
2. Neighbouring residues within 8Å of the residue are identified;
3. The side-chain is rotated around χ_1 (Figure 4.5a) and χ_2 (Figure 4.5b) torsion angles recording whether a bad contact is made or not (a bad contact is defined as two atom centres within 2.50Å of each other);
4. If the MOP conformation makes ≤ 1 bad contacts, this conformation is accepted;
5. If a structural rotamer conformation (i.e. staggered X_1 X_2 angles) exists that makes ≤ 1 bad contacts, that is selected;
6. The lowest energy conformation is selected (closest to the MOP conformation if alternatives exist with the same energy).



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



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

Figure 4.5: Using MutModel to model a mutant residue into an existing structure: rotation about the χ angles.

4.4 Improving the clash analysis

The existing MutModel program recorded clashes as Boolean values; therefore, the aim was to provide continuous (rather than Boolean) values to deal with the 'degree of clash'. The processes required to develop this approach are described in the following sections.

4.4.1 Linear Energy

Initially a linear energy was implemented using the equation:

$$E = \begin{cases} 0 & \text{if } d > Max \\ 1 & \text{if } d \leq Min \\ 1 - \frac{distance - Min}{Max - Min} & \text{otherwise} \end{cases} \tag{4.2}$$

The degree of clash is defined by a minimum distance (e.g. Min = 1.5Å) and a maximum distance (Max = 2.5Å). On this basis, a distance greater than 2.5Å gives a clash energy equal to 0, while if the distance is $\leq$ 1.5Å, the clash energy is equal to 1, with a linear scale in-between. This step was preliminary experiment as a step towards the full potential energy.

4.4.2 Full Potential Energy

A full energy calculation to handle the amount of clash was then implemented (by ACRM), this was achieved by incorporating a Lennard-Jones potential and a torsion potential:

$$E = \left(\frac{A}{r^{12}} - \frac{B}{r^{6}}\right) + k(1 + \cos(n\psi + \phi)) \tag{4.3}$$

The Lennard Jones parameters (A and B) depend on the types of the two interacting atoms with parameters coming from the CHARMM forcefield (Brooks *et al.*, 2009; Brooks *et al.*, 1983). The Lennard Jones potential accounts for clashes between atoms of the side-chain being replaced and its surroundings, while the torsional term favours staggered conformations (see Figure 4.6). Then I modified the MutModel code to allow the user to select the evaluation method. Currently the MutModel program evaluates the clash energy using one of the four clash evaluation methods: 1: Boolean; 2: Linear clash; 3: vdW (Lennard-Jones); 4: vdW/Torsion.

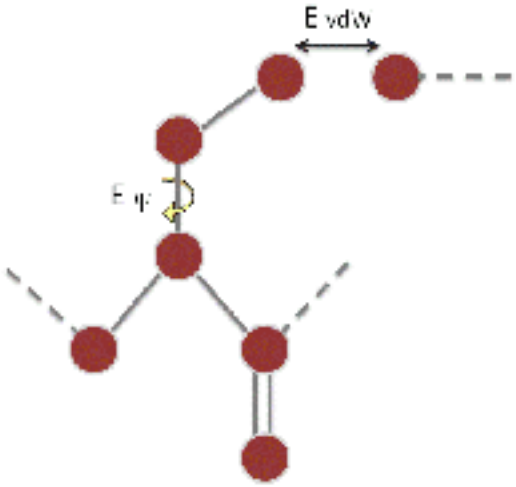


Figure 4.6: Schematic indicating the two new terms used in evaluation of clashes. E_{vdW} is the van der Waals energy evaluated using a standard Lennard-Jones potential while E_{ψ} is the torsion energy.

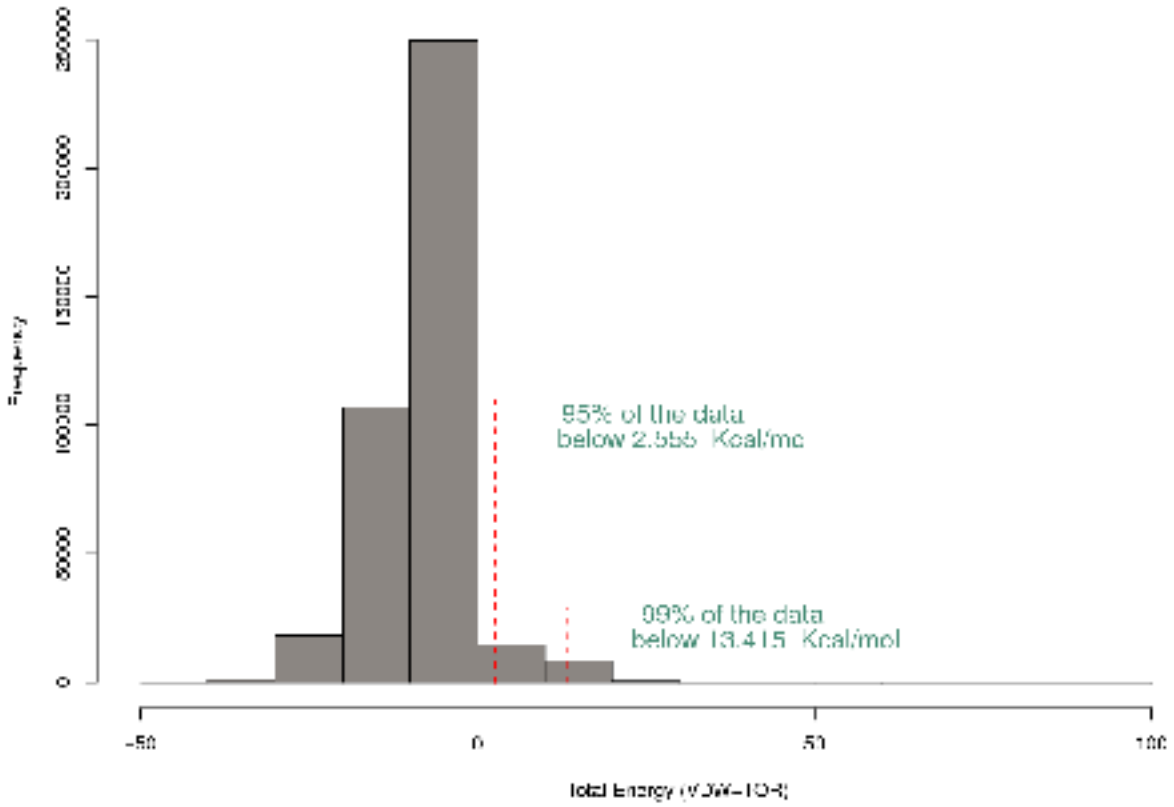


Figure 4.7: Distribution of side-chain clash energies calculated according to Equation 4.3 for high resolution structures amongst CATH O-representatives.

4.4.3 Testing the new method

After incorporating the new analyses into SAAPdap, the vdW/Torsion method was tested on 400,000 residues from CATH O-representatives (domains having no more than 65% sequence identity) of high resolution (< 2.5Å). This showed that 99% of side-chains have an energy of < 13.4 kcal/mol (see Figure 4.7) and the highest energy value is 34.33 kcal/mol. That number was used as a cut-off for defining a damaging clash. Using the new energy evaluation, the performance of the Boolean clash method was assessed. In the Boolean method, no account was made of the degree of clash; overlaps of 0.01Å , or of 1.0Å , were treated as the same and a mutation was classified as damaging if it made 3 or more clashes.

Figure 4.8 shows the energy distribution for side-chain replacements considered to make 0-5 or more clashes by the old method. Looking at side-chain replacements that made no clashes using the old method (Figure 4.8, panel 1), we see that 99% of the data have an energy below 34.33 kcal/mol using the new energy-based method. Panels 2 and 3 show cases evaluated by the original Boolean method as making one or two clashes that would have been classified as non-damaging using the Boolean method. Using 34.33 kcal/mol as an energy cut-off, these graphs indicate that 33.2% and 28.9% of potentially damaging clashes (shaded regions in panels 2 and 3 respectively) were not detected using the old method. Panels 4, 5, and 6 show the energy distributions for side-chain replacements having 3, 4, and 5 or more clashes by the old method, which would have been classified as damaging. However, using the new method, 19.5%, 10.7% and 11.2% of cases (shaded regions in panels 4, 5 and 6 respectively) have energies below the threshold and are therefore unlikely to be damaging.

Overall, approximately 32% of mutations previously classified as not clashing are now found to clash, while approximately 15% of mutations previously classified as clashing are now found to have only minor clashes that could be relieved by very slight movements in the structure. This improved evaluation of side-chain clashes should improve attempts to explain why pathogenic deviations are damaging and will also help to improve machine learning methods for predicting the effects of mutations.

Distribution of energies calculated according to Equation 4.3 for side-chain replacements

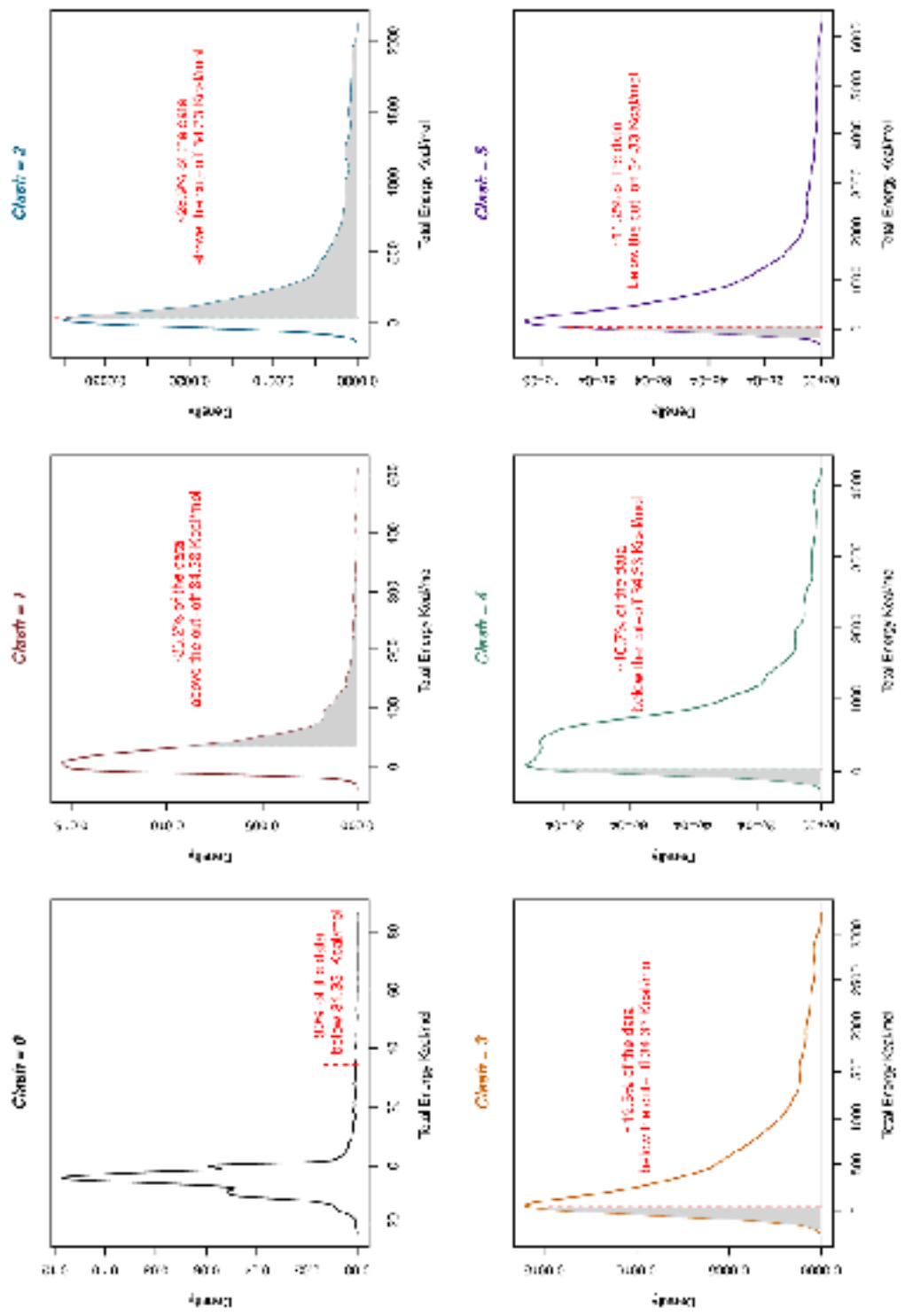


Figure 4.8: Distribution of energies calculated according to Equation 4.3 for side-chain replacements classified as making 0–5 clashes using the original (Boolean) method. In the old method, 0, 1, or 2 clashes were considered not to form a bad clash while 3 or more clashes were considered to be a bad clash. Using the cutoff of 34.33 kcal/mol (see page 164) in each plot, the shaded area shows those residues that were mis-classified according to the new energy-based criterion.

4.5 Improving the void analysis

The aim was not to improve the actual void analysis, but to optimize the side chain replacement by the MutModel program.

Using the CATH (V 2.3) and PDB databases, we first find domains that differ from one another by only one residue and whose structures have been determined by X-ray crystallography were identified with a resolution $\leq$3Å. A Perl program was written to create a list of native proteins with their corresponding mutant by reading through the CATH file and collecting all domains with the same CATH numbers into an array. It then takes each domain in turn, checks if it is a native (i.e. not described in the PDB file as a mutant), and then checks all the other domains in its CATH set and selects those that are mutants and differ from it by only one residue. The list was in the form of the name of the native domain followed by the number of related mutant. The native and mutant details are given on subsequent lines (the positions of the 'changed' residue in both the native and mutant, and the native and mutant residue). The file contains 2,129 native domains with their corresponding mutants from which a total of 19,276 native/mutant pairs were obtained. Another Perl program takes each native-mutation pair from the list created in the previous step, performs MutModel to exchange the appropriate residue, then ProFit is used to fit the new modelled mutant domain against that of the mutant crystal-structure measuring the backbone and side-chain Root Mean Square Deviations (RMSDs). First the structures were compared by fitting on Cα atoms and rejected if the Cα-RMSD was lg 0.5Å using ProFit. This was to ensure that there is no structural rearrangement resulted from the mutation. ProFit was then used to calculate the backbone and side-chain RMSD for the mutated residue.

A further enhancement to the program was made to allow the user to specify the MutModel evaluation method and parameters used, in particular the step-size and tolerance:

1. Evaluation method (1: Boolean; 2: Linear clash; 3: vdW (Lennard-Jones); 4: vdW/Torsion.).
2. Tolerance in energy for accepting the parent (MOP) conformation or a standard rotamer position, rather than the lowest overall energy.
3. Step size of rotation (in degrees).

The evaluation was performed using various step-sizes (1, 5, 15, 30 and 60°), with tolerance of 0, 1, 2 and 3 Kcal/mol clashed for the Boolean and Simple linear scale method (see Tables 4.1 and 4.2) and tolerances of 0, 1, 5, 50, 100, 500 and 1000 Kcal/mol for the vdW (Lennard-Jones) and vdW/Torsion method (see Tables 4.3 and 4.4).

By running the MutModel program on modelled mutations where structures are known with varied combinations of the three parameters and observing the average RMSD, the parameters to be used in calculating the voids with each method were optimised. The best combinations of these parameters is used in the new clash and void analysis integrated into the SAAPdap pipeline. Table 4.5 shows a summary of the best step size and tolerance for the different evaluation methods. Surprisingly all evaluation methods showed the same lowest mean RMSD of 1.30Å, with the two vdW energy methods showing a slightly better standard deviation.

Table 4.1: Exploring different step-sizes and tolerance using MutModel (Method 1: Boolean).

Stepsize	Tolerance	Sum (RMS M*X)	Total number	Mean (RMS)	Standard deviation	Variance (Standard deviation)
1	0	21559.19	15834	1.32	1.17	1.36
1	1	22346.57	15834	1.34	1.19	1.41
1	3	21791.64	15834	1.31	1.17	1.38
1	2	21805.72	15834	1.32	1.17	1.38
5	0	21527.51	15834	1.32	1.17	1.36
5	1	22161.09	15834	1.34	1.18	1.40
5	2	21592.11	15834	1.32	1.17	1.36
5	3	21518.76	15834	1.31	1.17	1.36
15	0	21905.23	15834	1.32	1.18	1.38
15	1	22439.83	15834	1.34	1.19	1.42
15	2	21937.30	15834	1.31	1.18	1.39
15	3	22010.84	15834	1.31	1.18	1.39
30	0	21773.45	15834	1.33	1.17	1.38
30	1	22039.87	15834	1.33	1.18	1.39
30	2	21921.01	15834	1.31	1.18	1.38
30	3	22135.71	15834	1.31	1.18	1.40
60	0	22222.08	15834	1.34	1.18	1.40
60	1	22279.68	15834	1.34	1.19	1.41
60	2	21774.01	15834	1.31	1.17	1.38
60	3	21567.75	15834	1.30	1.17	1.36

Table 4.2: Exploring different step-sizes and tolerance using MutModel (Method 2: Linear).

Stepsize	Tolerance	Sum (RMS M*X)	Total number	Mean (RMS)	Standard deviation	Variance (Standard deviation)
1	0	21645.32	15834	1.32	1.17	1.37
1	2	21990.42	15834	1.31	1.18	1.39
1	1	21929.16	15834	1.32	1.18	1.38
1	3	22418.77	15834	1.31	1.19	1.42
5	0	21758.05	15834	1.33	1.17	1.37
5	1	21857.29	15834	1.32	1.17	1.38
5	2	21745.95	15834	1.30	1.17	1.37
5	3	21965.69	15834	1.30	1.18	1.39
15	0	22110.66	15834	1.33	1.18	1.40
15	1	22018.30	15834	1.32	1.18	1.39
15	2	20888.22	14887	1.32	1.18	1.40
15	3	21118.19	14913	1.32	1.19	1.42
30	0	21053.47	14932	1.36	1.19	1.41
30	1	21112.06	14933	1.34	1.19	1.41
30	2	21179.31	14919	1.32	1.19	1.42
30	3	21243.94	14921	1.32	1.19	1.42
60	0	21352.58	14781	1.36	1.20	1.44
60	1	20565.86	14748	1.33	1.18	1.39
60	2	21755.33	15834	1.30	1.17	1.37
60	3	21674.70	15834	1.30	1.17	1.37

Table 4.3: Exploring different step-sizes and tolerance using MutModel (Method 3: vdW (Lennard-Jones)).

Stepsize	Tolerance	Sum (RMS M*X)	Total number	Mean (RMS)	Standard deviation	Variance (Standard deviation)
1	0	24296.65	15834	1.52	1.24	1.53
1	1	24826.96	15834	1.47	1.25	1.57
1	5	22509.68	15834	1.32	1.19	1.42
1	10	22192.03	15834	1.33	1.18	1.40
1	50	22163.43	15834	1.33	1.18	1.40
1	100	22349.98	15834	1.33	1.19	1.41
1	500	22321.32	15834	1.32	1.19	1.41
1	1000	22397.68	15834	1.32	1.19	1.41
5	0	24317.88	15834	1.52	1.24	1.54
5	1	24882	15834	1.47	1.25	1.57
5	5	22524.97	15834	1.33	1.19	1.42
5	10	22443.83	15834	1.33	1.19	1.42
5	50	22180.18	15834	1.33	1.18	1.40
5	100	22302.54	15834	1.33	1.19	1.41
5	500	22289.92	15834	1.32	1.19	1.41
5	1000	22250	15834	1.32	1.19	1.41
15	0	23747.77	15612	1.52	1.23	1.52
15	1	24309.42	15613	1.47	1.25	1.56
15	5	22109.95	15620	1.32	1.19	1.42
15	10	21901.62	15619	1.32	1.18	1.40
15	50	22100.04	15624	1.33	1.19	1.41
15	100	22523.05	15830	1.33	1.19	1.42
15	500	22384.90	15829	1.32	1.19	1.41
15	1000	22392.70	15830	1.32	1.19	1.41
30	0	22899.81	15831	1.50	1.20	1.45
30	1	23723.77	15830	1.45	1.22	1.50
30	5	21459.78	15830	1.30	1.16	1.36
30	10	23121.80	15278	1.31	1.23	1.51
30	50	21093.90	15274	1.31	1.18	1.38
30	100	21485.75	15243	1.31	1.19	1.41
30	500	21482.62	15253	1.31	1.19	1.41
30	1000	21426.82	15266	1.31	1.18	1.40
60	0	24524.14	15834	1.50	1.24	1.55
60	1	25277.96	15834	1.38	1.26	1.60
60	5	23601.70	15834	1.32	1.22	1.49
60	10	23395.53	15834	1.33	1.22	1.48
60	50	22987.14	15834	1.32	1.20	1.45
60	100	23120.63	15834	1.32	1.21	1.46
60	500	22980.80	15834	1.32	1.20	1.45
60	1000	22925.33	15834	1.32	1.20	1.45

Table 4.4: Exploring different step-sizes and tolerance using MutModel (Method 4: vdW/Torsion).

Stepsize	Tolerance	Sum (RMS M*X)	Total number	Mean (RMS)	Standard deviation	Variance (Standard deviation)
1	0	23883.23	15834	1.46	1.23	1.51
1	1	24301.50	15834	1.42	1.24	1.53
1	5	22317.42	15834	1.32	1.19	1.41
1	50	21940.27	15834	1.32	1.18	1.39
1	10	21923.05	15834	1.32	1.18	1.38
1	100	22121.92	15834	1.32	1.18	1.40
1	500	22217.30	15834	1.32	1.18	1.40
1	1000	22305.66	15834	1.32	1.19	1.41
5	0	23881.62	15834	1.46	1.23	1.51
5	1	24246.59	15834	1.42	1.24	1.53
5	5	22243.86	15834	1.32	1.19	1.40
5	10	22312.50	15834	1.33	1.19	1.41
5	50	21953.47	15834	1.32	1.18	1.39
5	100	22049.08	15834	1.32	1.18	1.39
5	500	22170.77	15834	1.32	1.18	1.40
5	1000	22131.65	15834	1.32	1.18	1.40
15	0	23518.23	15834	1.45	1.22	1.49
15	1	23575.16	15834	1.41	1.22	1.49
15	5	21750.46	15834	1.31	1.17	1.37
15	10	21738.58	15834	1.31	1.17	1.37
15	50	21674.76	15765	1.31	1.17	1.37
15	100	22176.05	15738	1.32	1.19	1.41
15	500	22138.82	15738	1.32	1.19	1.41
15	1000	22164.27	15737	1.32	1.19	1.41
30	0	23600.11	15733	1.47	1.22	1.50
30	1	23262.38	15834	1.40	1.21	1.47
30	5	21306.22	15834	1.29	1.16	1.35
30	10	21314.11	15834	1.30	1.16	1.35
30	50	21569.50	15834	1.31	1.17	1.36
30	100	21382.10	15384	1.30	1.18	1.39
30	500	21450	15352	1.32	1.18	1.40
30	1000	21813.17	15834	1.30	1.17	1.38
60	0	25127.70	15834	1.46	1.26	1.59
60	1	24842.86	15266	1.37	1.28	1.63
60	5	23874.51	15282	1.34	1.25	1.56
60	10	23615.34	15287	1.34	1.24	1.54
60	50	23443.15	15834	1.33	1.22	1.48
60	100	23409.04	15834	1.32	1.22	1.48
60	500	23221.73	15834	1.32	1.21	1.47
60	1000	23123.50	15834	1.32	1.21	1.46

Table 4.5: Summary of the best step size and tolerance for the different MutModel evaluation methods.

Method	Step size	Tolerance	Average RMSD
Boolean evaluation	60	3	1.30
Linear evaluation	5 & 60	2 & 3	1.30
Energy evaluation vdW	30	5	1.30
Energy evaluation vdW/Torsion	30	10 & 100 & 1000	1.30

4.6 Conclusion and discussion

Clash and void analysis in SAAPdb involved Boolean evaluation with defined cut-offs. In analyzing clashes, previous work defined a damaging clash as any side-chain that has at least 3 van der Waals overlaps (of any degree) with other atoms. Similarly, voids were considered damaging when they caused the creation of voids of volume $> 275\text{Å}^3$, assuming no compensatory movement within the protein structure. By looking at the distribution of SNPs and PDs predicted to be damaging, it was clear that the Boolean method did not accurately describe the effect of mutations causing clashes or voids, either overestimating or underestimating damaging effects when values were close to the cut-off.

The new Clash analyses use a continuous energy scale calculation incorporating Lennard-Jones and torsion energies using CHARMM (Brooks *et al.*, 1983) parameters. An energy cut-off representing ≤ 6 of side-chains (i.e. 4–5 structural) in high-resolution structures was selected simply for visual indication that a mutation is likely to be damaging. The actual energy value is used in the machine learning described in Chapter 6. The MutModel program is used in both clash and void analysis and parameters (step-size and tolerance) used in searching side-chain positions were optimised by modelling known mutant structures. Consequently, the evaluation of both clash and void is optimised by using these parameters. No other changes were made to the assessment of voids; the cut-off selected previously is used as a visual indication that a void is likely to be damaging, but as with clash energy actual void sizes are used in the machine learning described in Chapter 6.

Chapter 5

Improvements to Glycine and Proline Analysis

∗ *Some of the work in this chapter has been published in Al-Numair NS, Martin ACR. 2013. The SAAP pipeline and database: tools to analyze the impact and predict the pathogenicity of mutations. BMC Genomics 14.3:111.*

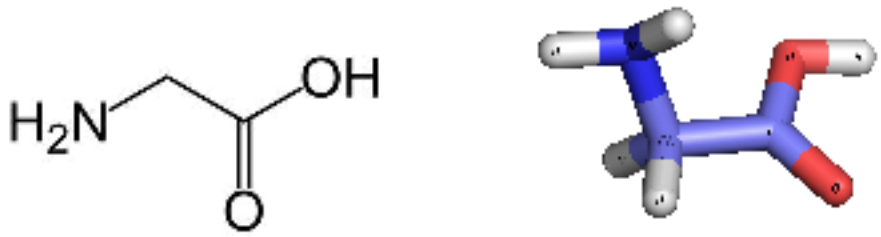
Glycine and proline amino acids both exhibit an unusual Ramachandran distribution. Since glycine has no side-chain, it is able to access a wider range of phi/psi combinations than the other amino acids, while the cyclic side-chain of proline restricts the available phi angles available to it. Consequently, backbone conformational changes may be necessary to accommodate mutations from-glycine or to-proline, which could disrupt protein folding and alter function. The purpose of this chapter is to improve the SAAPdap analysis by developing a pseudo-energy potential based on Ramachandran plots to supercede the simple set of allowed backbone phi/psi angles used in SAAPdb previously.

5.1 Introduction

Glycine and proline are both unusual amino acids as described in the following sections.

5.1.1 Glycine

Glycine (Figure 5.1) is the smallest of the 20 amino acids commonly found in proteins. As described earlier, Glycine has an unusual Ramachandran distribution because it has no side-chain (i.e. its R group is a hydrogen). Thus, glycine is able to access and adopt a wider range of phi/psi combinations than other amino acids which are sterically hindered. Because of this unique capability, when a mutation alters any native glycine residue whose backbone torsion angles are unfavourable for other amino acids, there will be an effect on local protein structure and consequently a potential effect on function.

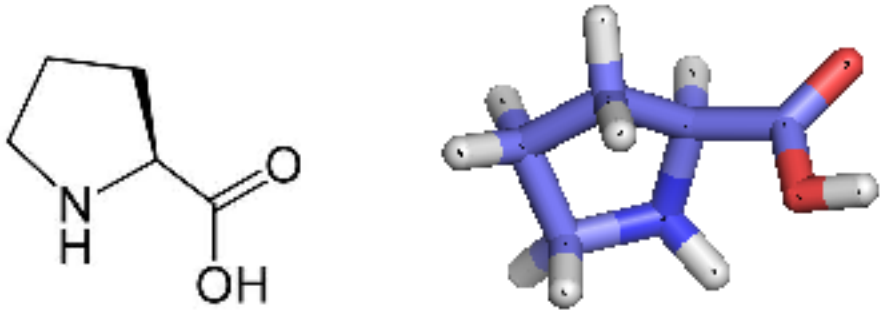


(a) (b)

Figure 5.1: Glycine amino acid (Gly or G),
(a) Glycine molecular formula NH_2CH_2COOH. (b) Glycine structure.

5.1.2 Proline

Proline (Figure 5.2) like glycine, is a 'structural' amino acid (strictly 'imino') that has an unusual Ramachandran distribution. The cyclic nature of the side-chain of the amino acid gives it more conformational rigidity compared with the other 19 amino acids commonly found in proteins. The cyclic side-chain, which links back to the backbone nitrogen, restricts the available phi angles compared with other amino acids (Figure 5.2). Therefore, just as with mutations from-glycine, it is expected that when a mutation introduces a proline at a position where torsion angles are unfavourable, it will affect the local protein structure and potentially protein folding and function.



(a) (b)

Figure 5.2: Proline amino acid (Pro or P).
(a) Proline molecular formula $C_5H_9NO_2$. (b) Proline structure.

The *trans* conformation of amino acids is found in the majority of peptide bonds because in the *cis* conformation, the proximity of the chiral carbons makes the structure approximately 1000 times less stable than the *trans* form. However, *cis* peptide bonds between any residue and proline (*x*-Pro) are only 4 times less stable than the *trans* conformation (Branden and Tooze, 1999). A study by Stewart *et al.*, (1990) showed that only 0.05% of all amide bonds are *cis*, while 6.5% of *x*-Pro amide bonds are *cis*. While these values are considerably lower than the 20% expected from the difference in energy, *x*-Pro *cis* amide bonds are stable enough to occur at a higher frequency than other *cis* amide bonds. Kinetically, *cis-trans* proline isomerization is a very slow process, and is considered the rate-limiting step in protein folding where it occurs (Wedemeyer *et al.*, 2002).

A mutation from a *cis*-proline to any other amino acid is likely to disrupt the protein-fold significantly because an amino acid other than proline is likely to adopt the *trans* conformation. Such a change is very likely to affect function.

In the SAAPdb analysis, mutation of a *cis*-proline to another amino acid is treated as a Boolean analysis, no change is made in this work. However, mutations that cause changes either from-glycine or to-proline that involve changes in the favoured/disfavoured regions of the Ramachandran plot are potentially more subtle in nature. Previously simple ranges of allowed angles were used, but these may be sensitive to precise structural details making Boolean analysis inadequate for outcome prediction.

5.2 Analysis of sensitivity to structural details

This analysis was initiated as part of a Masters thesis (Al-Numair, 2010) and was updated as part of this doctoral research using the latest SAAPdb version (Chapter 3).

As described above, originally the SAAP analysis used a very simple set of allowed boundaries for backbone phi/psi angles to define a damaging mutation. These ranges were introduced during analysis of 1363 point mutations obtained from the P53 mutation databank (Martin *et al.* (2002); `www.bioinf.org.uk/P53/`). In that work, allowed regions for proline of $-70.0^{\circ} \leq \phi \leq -50.0^{\circ}$ and ($-70.0^{\circ} \leq \psi \leq -50.0^{\circ}$ or $110.0^{\circ} \leq \psi \leq 130^{\circ}$) were used and a total of 50 distinct damaging mutations to proline were identified. In Figure 5.3, the allowed regions for proline are shown in pink within the hatched region allowed for other amino acids. These ranges for the allowed regions are very conservative, possibly resulting in false-positive results (i.e an over-prediction of damage caused by a proline mutation). Indeed, some of the mutations studied in P53 were borderline and may be accommodated by a very small structural rearrangement (e.g. L137P).

In the P53 database, Martin *et al.* identified a total of 70 distinct mutations from a native glycine to another residue. The allowed regions of the Ramachandran plot for non-glycine/non-proline residues were, defined as: ($-180.0^{\circ} \leq \phi \leq -30.0^{\circ}/60.0^{\circ} \leq \psi \leq 180.0^{\circ}$) or ($-155.0^{\circ} \leq \phi \leq -15.0^{\circ}/-90.0^{\circ} \leq \psi \leq 60.0^{\circ}$) or ($-180.0^{\circ} \leq \phi \leq -45.0^{\circ}/-180.0^{\circ} \leq \psi \leq -120.0^{\circ}$) or ($30.0^{\circ} \leq \phi \leq 90.0^{\circ}/20.0^{\circ} \leq \psi \leq 105.0^{\circ}$). All non-glycine residues in the P53 crystal structure fell within these limits. In Figure 5.3, the allowed areas for non-glycine/non-proline residues are shaded grey.

Mutation data analysed in this section were obtained from SAAPdb PostgreSQL database. Glycine and proline mutation analyses were implemented using several custom Perl scripts. The first Perl script was used to examine the gathered data in order to determine the number of structures to which each mutation mapped. This method was used for both SNPs and PDs across all available mutations imported from SAAPdb, and across all proteins. Another Perl script was written for counting mutations classified as unfavourable and mapped to at least 2 structures. The same script was used to calculate the fraction of structures in which a particular mutation was classified as unfavourable. Mutations classified as unfavourable for each individual SAAP analysis type, where at least one structure was classified as unfavourable were also identified using this Perl script.

As with analysis of clashes and voids (Chapter 4) an enhancement to the original analysis program was made to introduce an option (`-nat`) to include no mutant structures, but only native structures from the PDB. A `-res` option was also added to restrict the analysis to either high resolution ($\leq$ 2.0Å when `-res=H`) or low resolution ($>$ 2.0Å when `-res=L`) structures. The fraction of structures to which a mutation is mapped that show a structural effect in the Boolean analysis (F) is defined as in Equation 4.1.

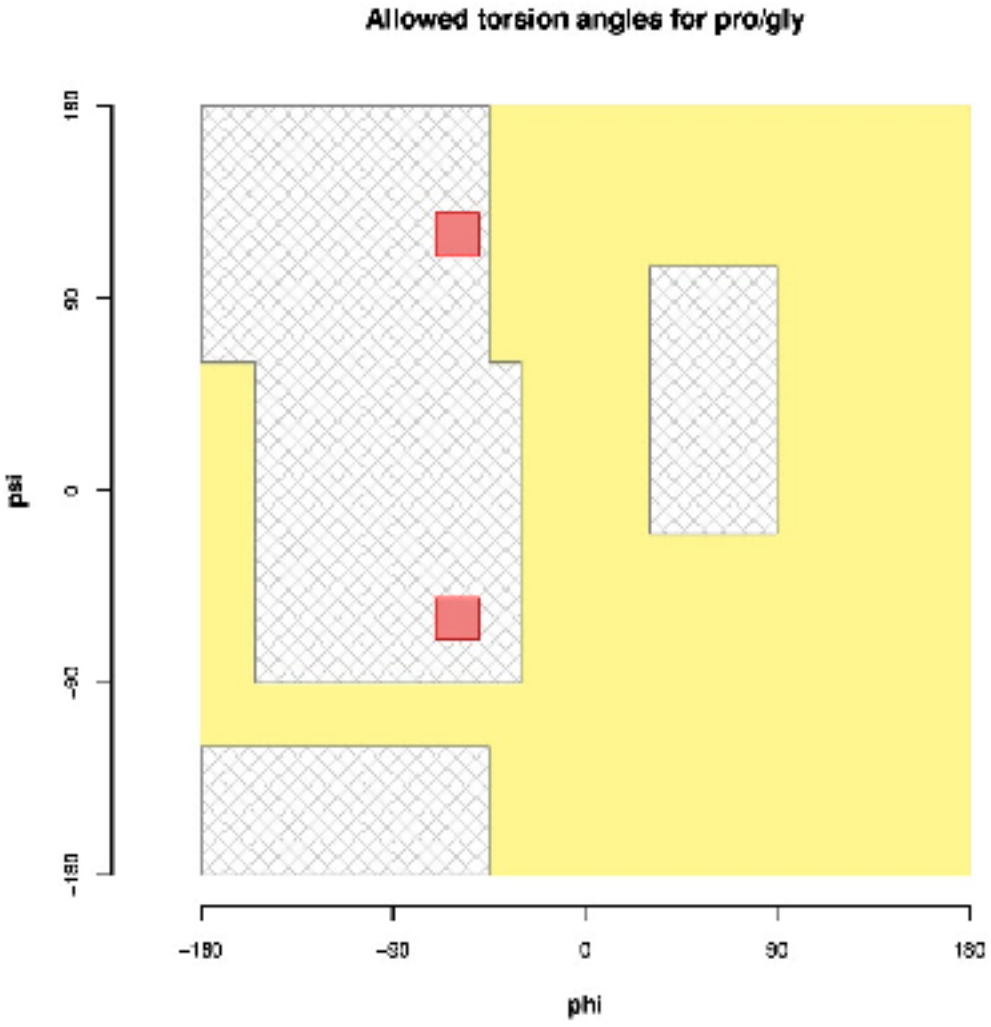


Figure 5.3: Allowed regions for proline and glycine using the previous method.
The pink areas mark the restricted conformation for proline residues used in the Boolean analysis the hatched grey area marks the regions for non-proline, non-glycine residues, and the yellow colour marks the rest of the conformational space, primarily occupied by glycine residues.

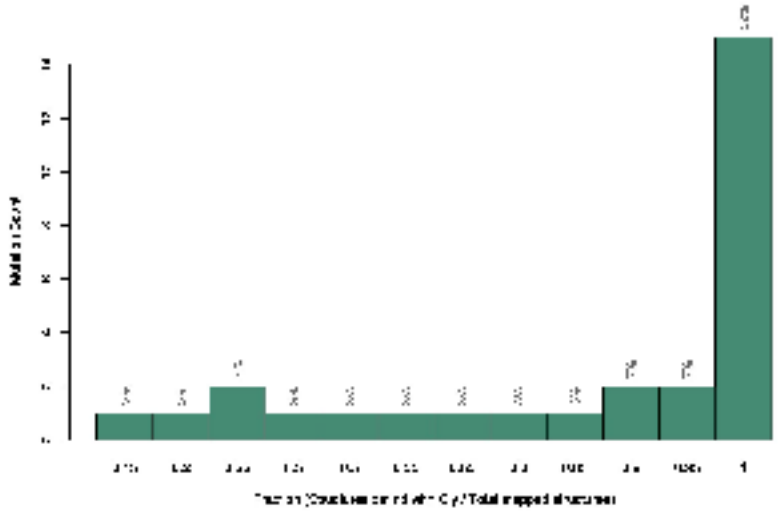
5.2.1 Glycine and proline analysis

In this section, using the old Boolean analysis, the results of mapping SNPs and PDs to native structures with high resolution are presented to show the need for improving the analysis. If the distribution of the fraction of structures (F) in which from-glycine mutations are classified as unfavourable using the rigid ϕ/ψ cut-offs defined earlier (Figure 5.3) for SNPs (Figure 5.4a) is compared with the equivalent analysis of to-proline mutations (Figure 5.5a), we see a similar distribution. Surprisingly the results are clearly skewed to F = 1, whereas, one would expect such mutations not to have any effect and therefore one would expect that the graphs would be heavily skewed towards F = 0. In other words we appear to have a large number of false positive and this clearly indicates the need for a better analysis of these effects.

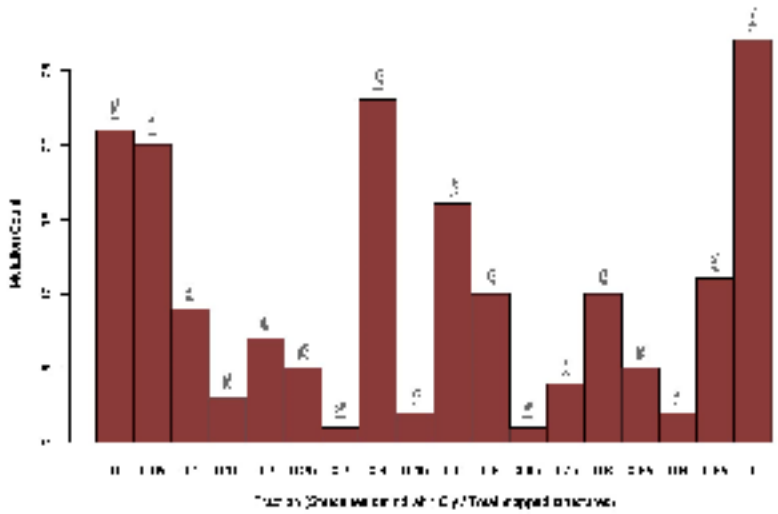
In the case of PDs (Figure 5.4b and 5.5b), a much broader distribution is observed. In these cases, one would hope to see distributions skewed towards F = 1. This is broadly the case for PDs resulting from to-proline mutations with 51% of mutations having $F \geq 0.95$. However, the situation is very different for PDs resulting from-glycine mutations where 21% of mutations have $F \geq 0.95$, but 23% have $F < 0.1$. These broad distributions indicate sensitivity to precise structural details and it is likely that mutations with F = 1 fall well within disallowed regions while other mutations with F < 1 fall on the boundaries of the disallowed regions in the Ramachandran plot (Figure 5.3).

5.3 Calculation of Ramachandran plot pseudo-energies TDMs

This section outlines the development and implementation of a pseudo-energy potential based on Ramachandran plots (Torsion Density Maps (TDMs)) to be used in the new SAAP-dap analysis. Initially a non-redundant set (sequence identity $< 25\%$) of high-resolution protein domains (resolution $\leq$1.8Å, R-Factor $\leq$0.3) was selected from the CATH pdblist (CATH v3.4.0). The number of proteins was insufficient for our requirements, so PISCES, a Protein Sequence Culling Server (`http://dunbrack.fccc.edu`) was used instead to produce the dataset. PISCES selects a subset from the PDB based on specified thresholds for resolution, sequence identity, and Rfactor. All non-X-ray entries were excluded. PISCES determines identities for PDB sequences using the CE structural alignment and uses a Z-score of 3.5 as the threshold to accept possible evolutionary relationships (Wang and Dunbrack, 2005). Table 5.1 shows the specifications and numbers obtained from PISCES for the subset used in these analyses.

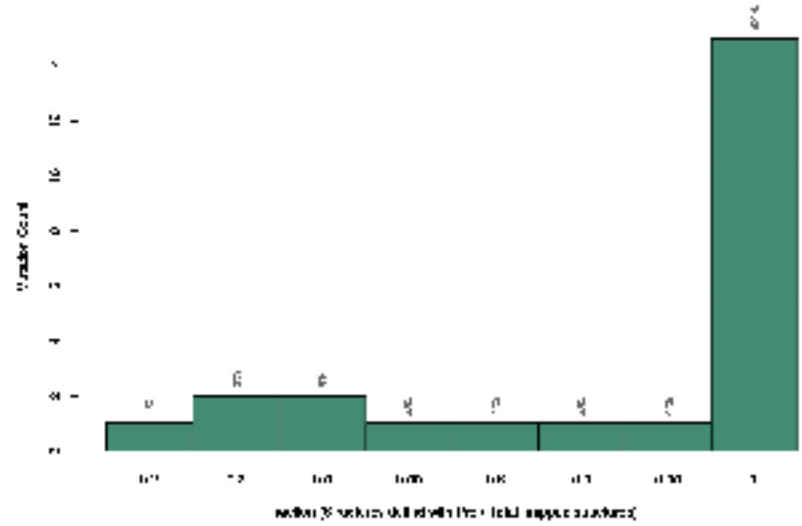


(a) SNP

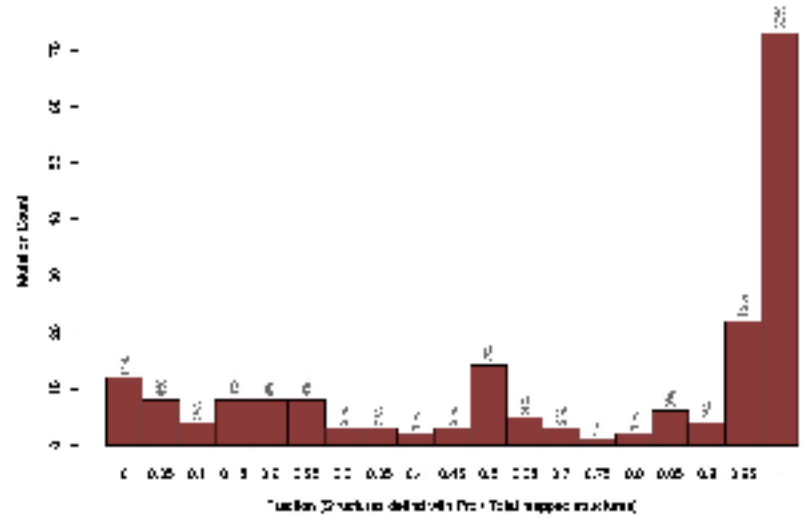


(b) PD

Figure 5.4: Boolean Glycine analysis
(a) The original Boolean glycine method tested on 29 SNP mutation using native structures. (b) The original Boolean glycine method tested on 177 PD mutation using native structures. A fraction of F = 0 represents no mutations classified as unfavourable. whereas F = 1 represents all mutations classified as unfavourable. In (a) and (b), each bar less than the next bar label (e.g. Bar 0.25 is $0.25 \leq F < 0.3$ and bar 0.95 is $0.95 \leq F < 1$), the bar labelled with one represent all the fraction equal to one only (i.e. all the structures are classified as unfavourable).



(a) SNP



(b) PD

Figure 5.5: Boolean Proline analysis
(a) The original Boolean proline method tested on 24 SNP mutation using native structures. (b) The original Boolean proline method tested on 189 PD mutation using native structures. A fraction of F = 0 represents no mutations classified as unfavourable. whereas F = 1 represents all mutations classified as unfavourable. In (a) and (b), each bar less than the next bar label (e.g. Bar 0.25 is $0.25 \leq F < 0.3$ and bar 0.95 is $0.95 \leq F < 1$), the bar labelled with one represent all the fraction equal to one only (i.e. all the structures are classified as unfavourable).

A Perl program was written to generate Ramachandran plots or Torsion Density Maps (TDMs) on a 1-degree grid (360x360 cell matrix) for proline, glycine and 'other' amino acids (non-glycine/proline). The program then extracted the PDB ID and chain for each entry in the dataset, and accessed a local copy of the PDB. The `torsion` and `getchain` programs (ACRM unpublished) were used to calculate backbone torsion angles for each chain. Phi/Psi angles were then rounded to the nearest integer and the count of Phi/Psi combinations was accumulated in the cells of Glycine, Proline and Other TDMs.

The different datasets shown in Table 5.1 were investigated, but throughout this chapter the dataset with 1.8Å resolution is discussed only.

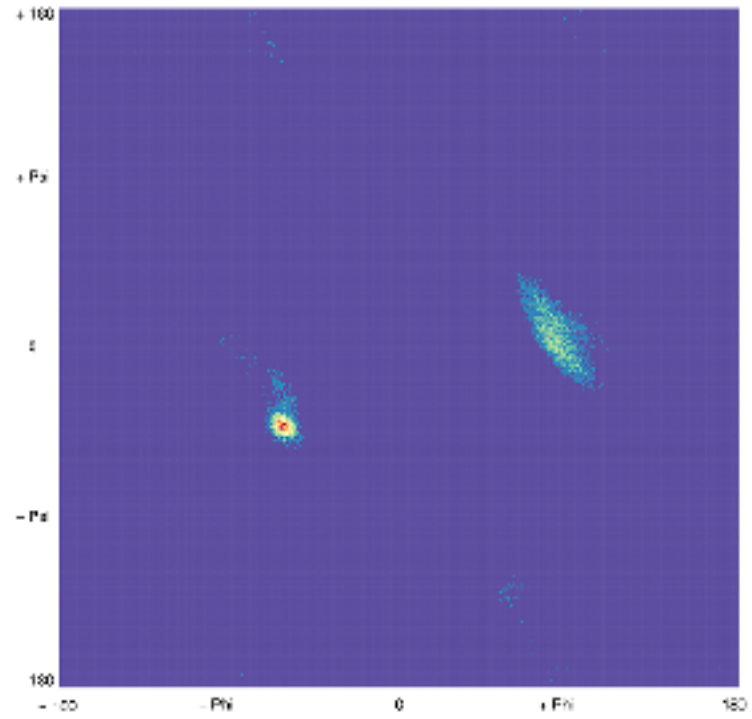
Table 5.1: Protein dataset obtained from PISCES.

Percentage Identity	Resolution	R-value	Domains	Chains
25	≤1.5Å	0.3	111,005	1,689
25	≤1.8Å	0.3	111,028	3,630
25	≤2.5Å	0.3	111,923	6,564

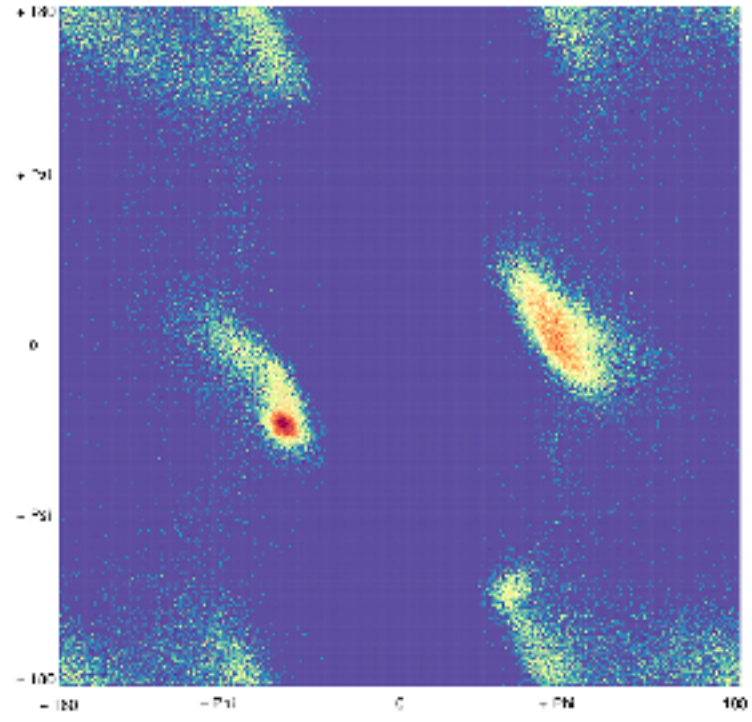
5.3.1 Raw data and log transformation

Ramachandran plots were generated with the total count in each cell (Figures 5.6a, 5.7a and 5.8a). To reveal more information in the plots and transform this to pseudo-energy, the ln (1+count) matrices were then plotted (Figure 5.6b, Figure 5.7b and Figure 5.8b).

This preliminary analysis showed that the data were not smoothly distributed implying the need for smoothing.

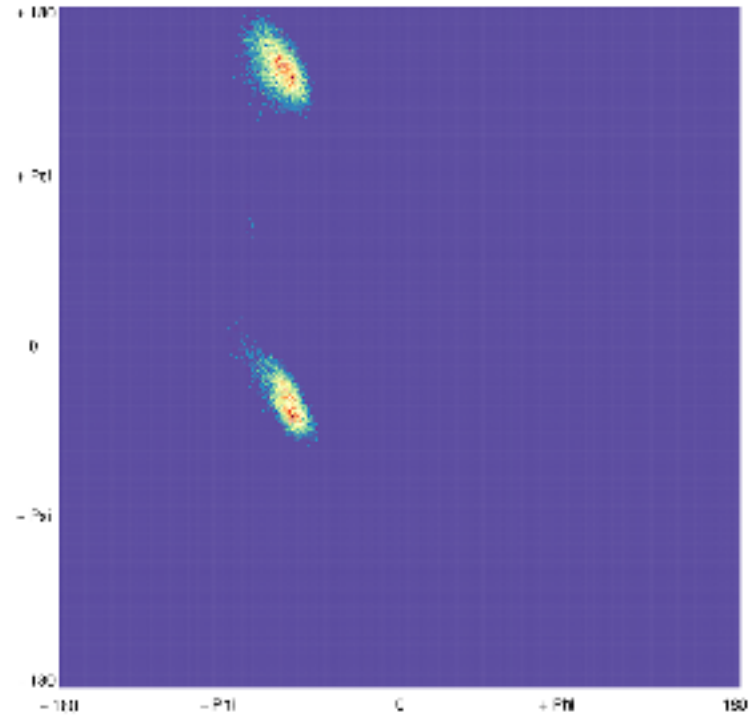


(a)

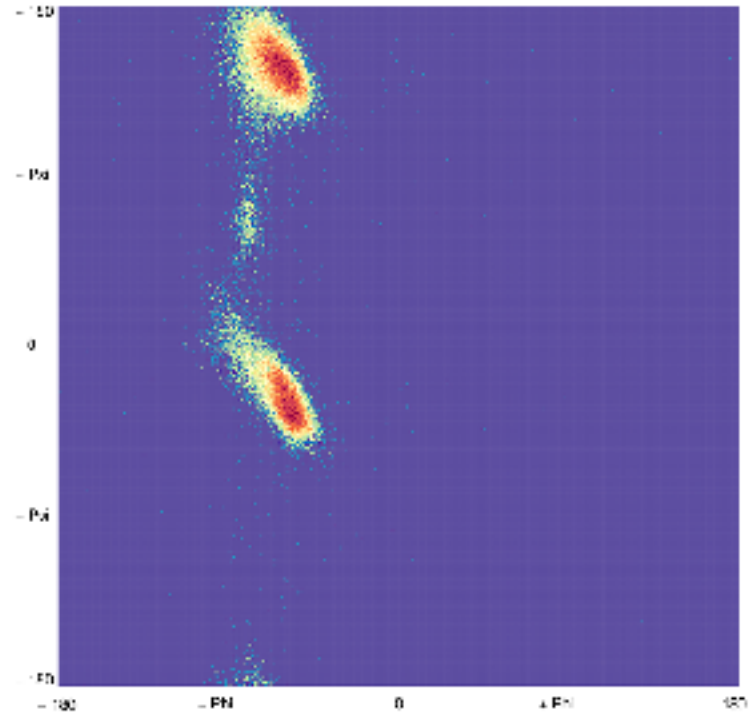


(b)

Figure 5.6: Glycine TDM.
(a) Glycine Ramachandran plot generated with the total observed count in each cell.
(b) Glycine Ramachandran plot generated with the *ln* of total observed count in each cell.
(Dark-purple to dark-red heat map).

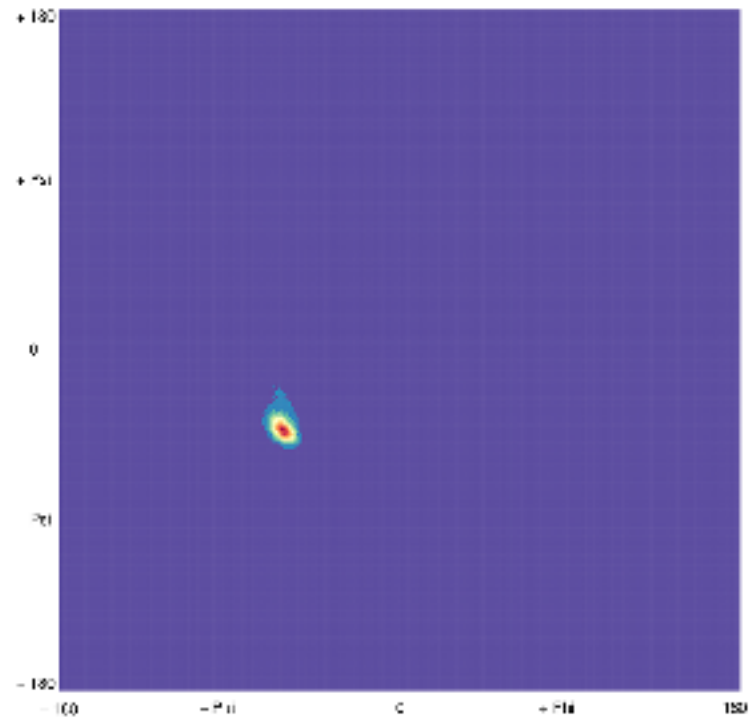


(a)

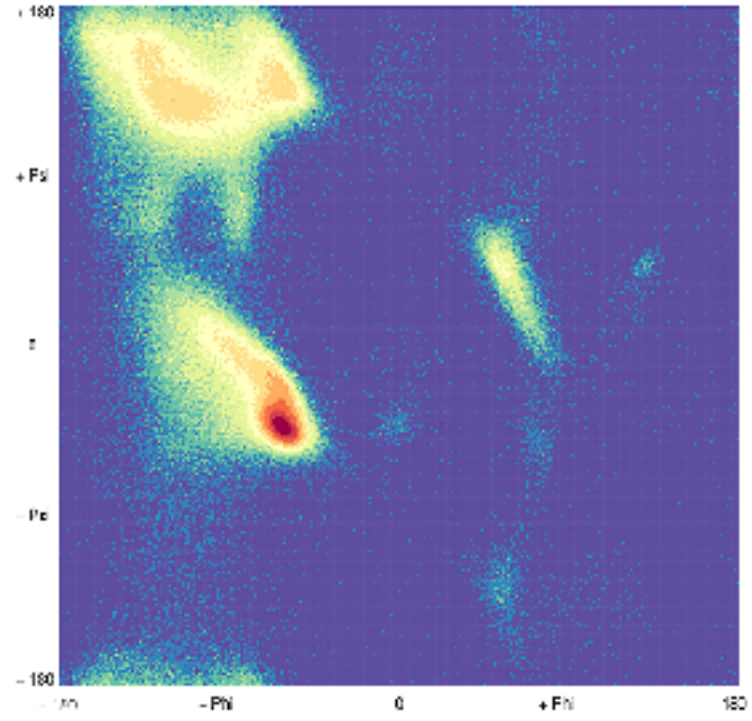


(b)

Figure 5.7: Proline TDM.
(a) Proline Ramachandran plot generated with the total observed count in each cell.
(b) Proline Ramachandran plot generated with the total *ln* observed count in each cell.
(Dark-purple to dark-red heat map).



(a)



(b)

Figure 5.8: Everything except Gly/Pro TDM.
(a) Everything except Gly/Pro Ramachandran plot generated with the total observed count in each cell. (b) Everything except Gly/Pro Ramachandran plot generated with the total *ln* observed count in each cell. (Dark-purple to dark-red heat map).

5.3.2 Cell smoothing

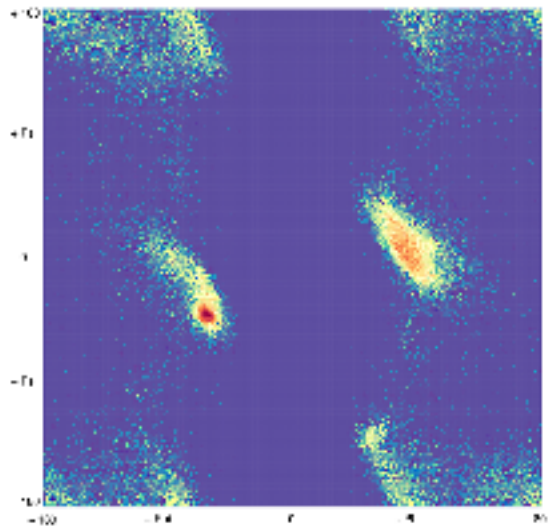
Figures 5.6b, 5.7b and 5.8b showed that the data are not smoothly distributed and there are many isolated points in the plot. The raw observed number of residues with a given phi/psi combination data were smoothed by averaging the data in a cell with neighbouring cells as shown in Equation 5.2. Values of 1, 2, 4, 6 and 8 were used for w, representing smoothing over 3X3, 5X5, 9x9 13X13 and 17X17 regions respectively. An energy value was then calculated for each cell as:

$$E = \ln\left(\frac{1+\text{obs}_s}{1+\text{exp}}\right) \tag{5.1}$$

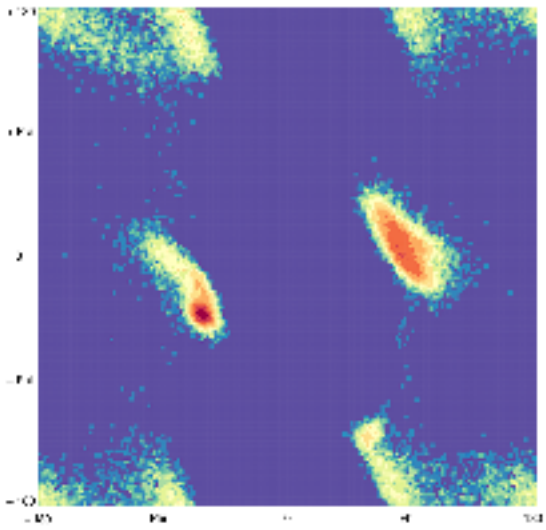
Where obs_s is the smoothed observed count and exp is the expected number calculated as the total number of observations divided by the number of cells.

$$S_{i,j} = \sum_{x=i-w}^{i+w}\sum_{y=j-w}^{i+w} n_{x,y}/N \quad \begin{cases} x = 360 + x & if(x < 0) \\ y = 360 + y & if(y > 0) \\ x = x - 360 & if(x > 360) \\ y = y - 360 & if(y > 360) \\ \end{cases} \tag{5.2}$$

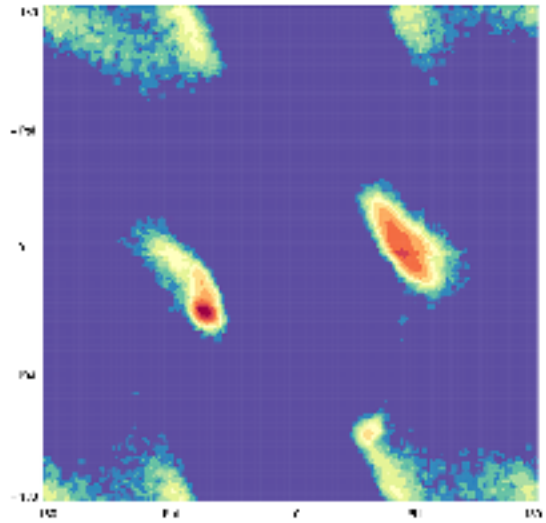
Results of different smoothing are shown in Figures 5.9, 5.10 and 5.11. Of these, the w = 6, (13X13) smoothing was chosen because it showed the best smoothing while retaining all scattered data present in the non-smoothed TDM (e.g. Figure 5.11f vs Figure 5.11e - the 17X17 smoothing loses a favoured region in the middle of the plot).



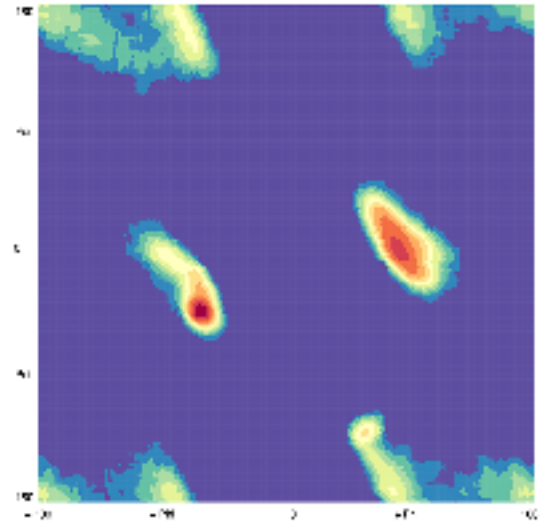
(a) No smoothing



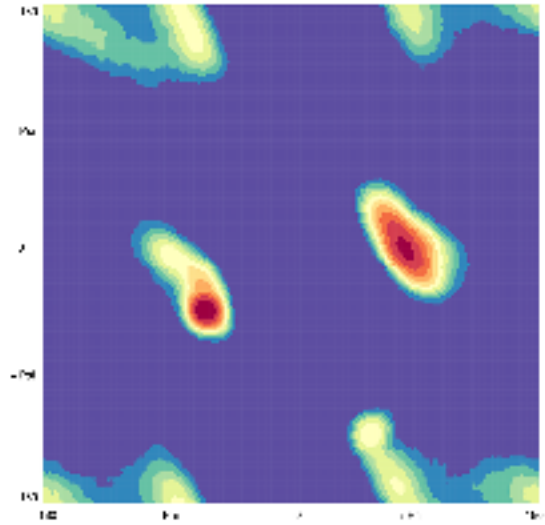
(b) 3x3 cell smoothing



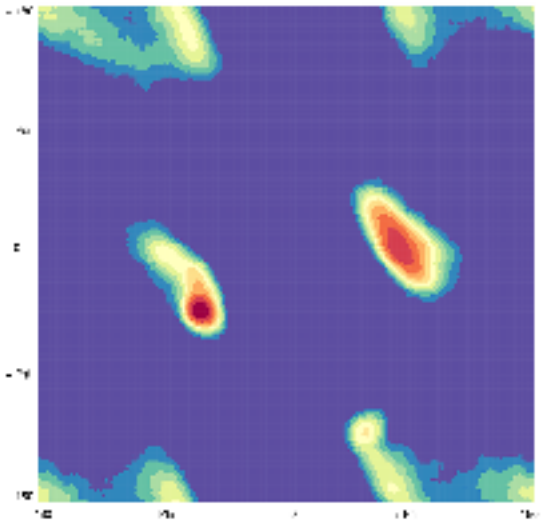
(c) 5x5 cell smoothing



(d) 9x9 cell smoothing

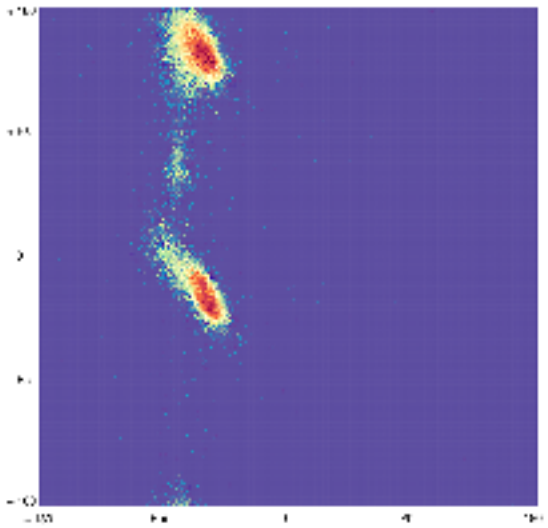


(e) 13x13 cell smoothing

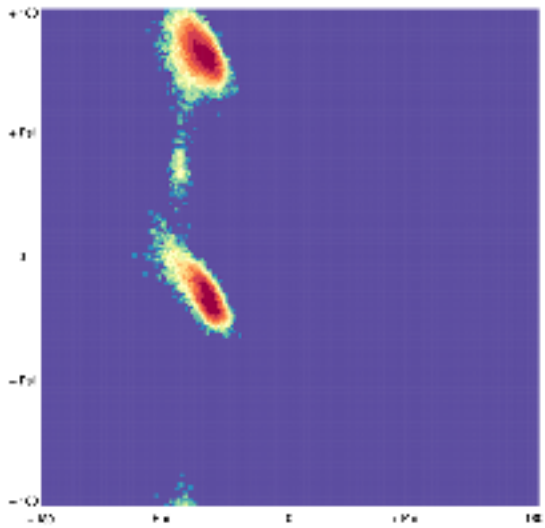


(f) 17x17 cell smoothing

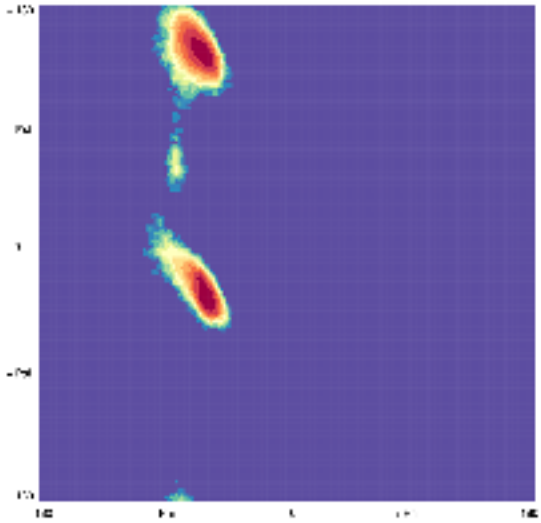
Figure 5.9: Glycine TDM smoothing.
(a) ln(obs/exp), (b)-(f) smoothed ln(obs/exp). (Dark-purple to dark-red heat map)



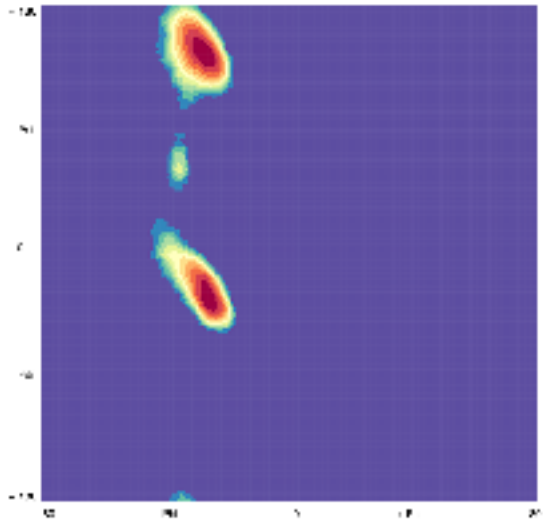
(a) No smoothing



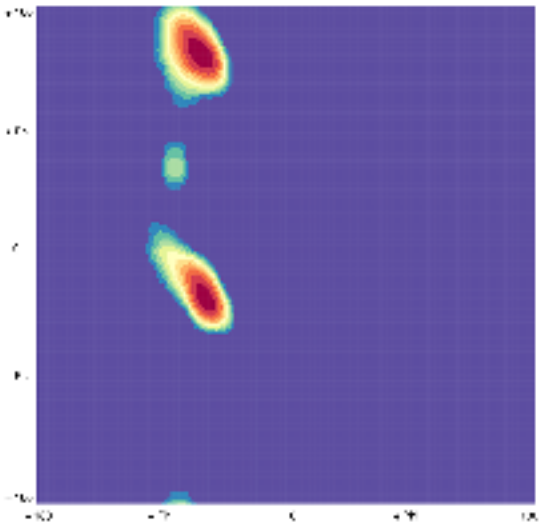
(b) 3x3 cell smoothing



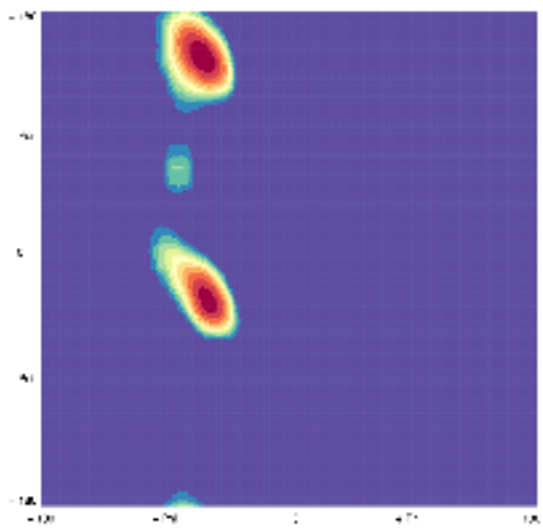
(c) 5x5 cell smoothing



(d) 9x9 cell smoothing

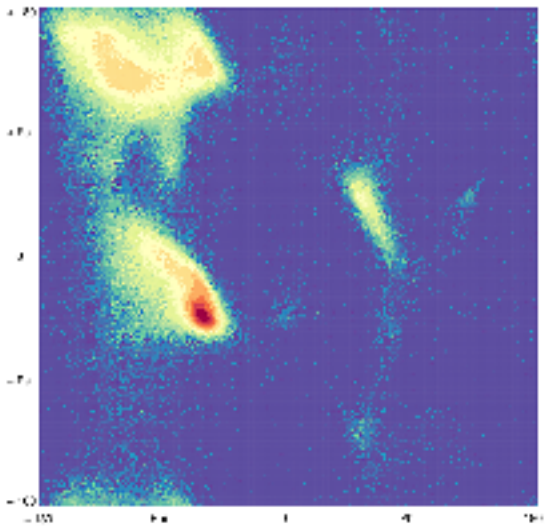


(e) 13x13 cell smoothing

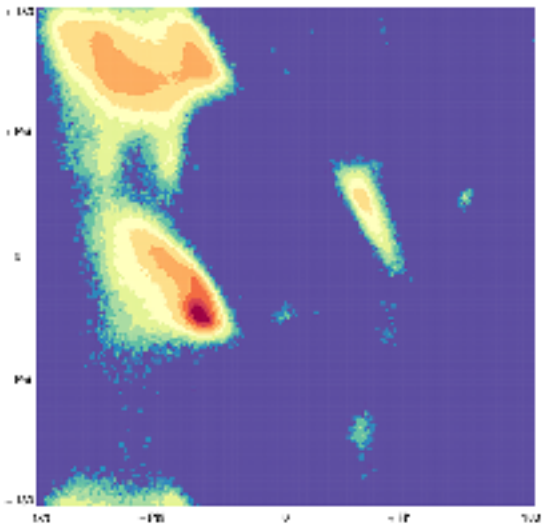


(f) 17x17 cell smoothing

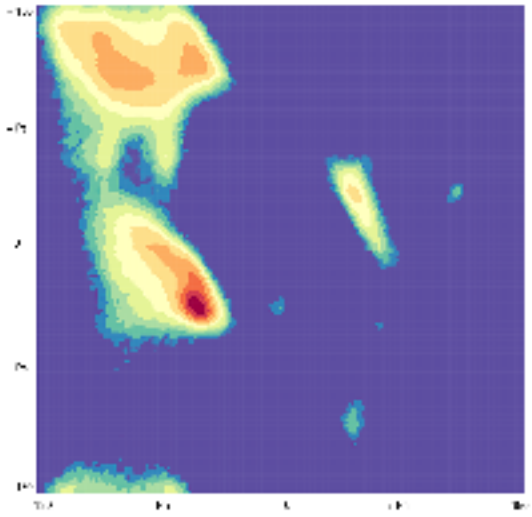
Figure 5.10: Proline TDM smoothing.
(a) ln(obs/exp), (b)-(f) smoothed ln(obs/exp). (Dark-purple to dark-red heat map).



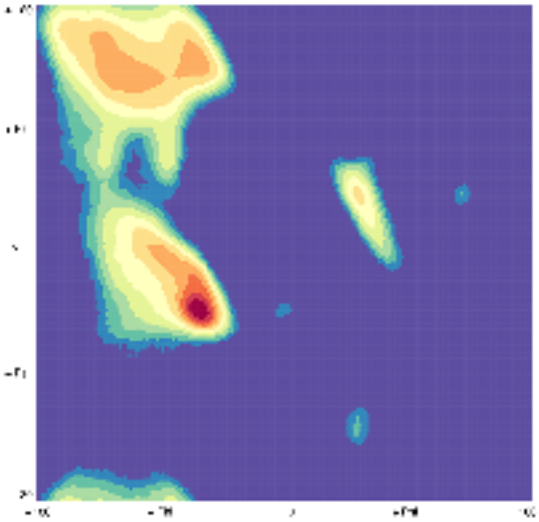
(a) No smoothing



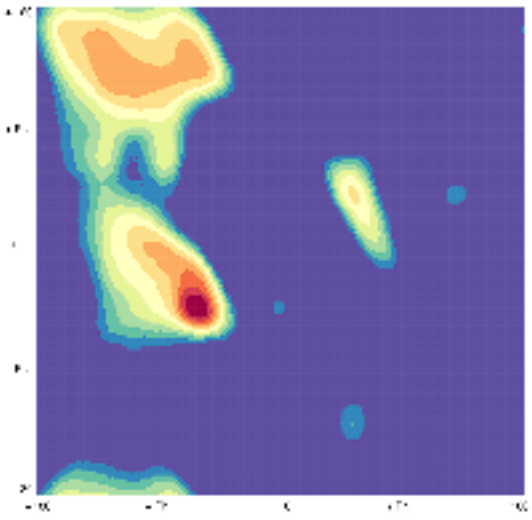
(b) 3x3 cell smoothing



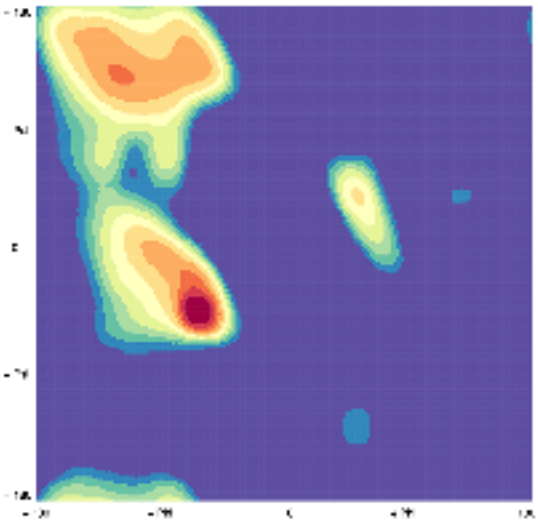
(c) 5x5 cell smoothing



(d) 9x9 cell smoothing



(e) 13x13 cell smoothing



(f) 17x17 cell smoothing

Figure 5.11: Non Glycine/Proline TDM smoothing.
(a) ln(obs/exp), (b)-(f) smoothed ln(obs/exp). (Dark-purple to dark-red heat map).

5.3.3 Threshold selection

The smoothed TDM (Ramachandran plot) created in the previous step was used to select the threshold for a visual indication of whether mutations are likely to be damaging. The energy values are used directly in the machine learning method (Chapter 6). The threshold was based on ~1% of observations in high-quality, non-redundant structures having a worse energy for both Proline and Non-Gly/Pro TDMs and ~2.5% for Glycine TDM (0.35, 0.5 and 1.5 for Glycine, Proline and Non-Gly/Pro TDMs (Table 5.2). Figure 5.12 shows the final TDMs after applying the threshold cutoff.

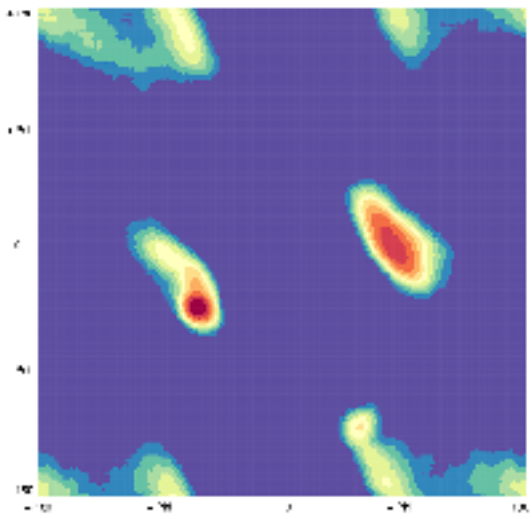
Table 5.2: Threshold selection.
Using 3630 PDB list (Percentage Identity = 25, Resolution = 1.8Å and R-value = 0.3).

	Glycine	Proline	Non-Gly/Pro
Energy threshold	0.35	0.5	1.5
Total count from all the cells	54767	34545	6557157
Expected count in each cell	0.42	0.27	5.06
Out of threshold count	1339 (2.445%)	298 (0.863%)	6073 (0.926%)

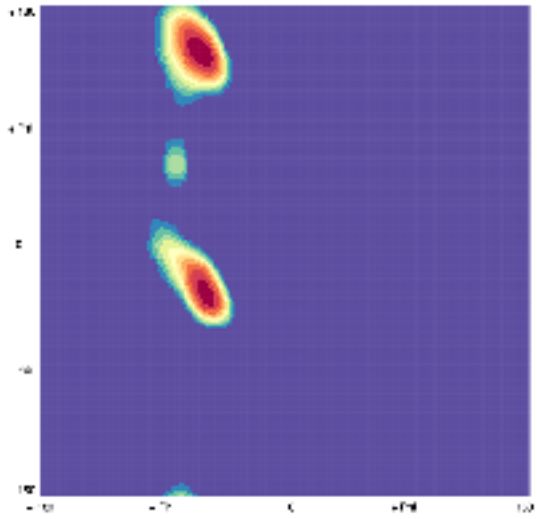
5.3.4 Comparison between the previous and new method

Comparing Figure 5.12b (allowed regions for proline based on 1% of prolines having a worse energy) with the pink regions in Figure 5.3, which represent the regions allowed for proline in the old Boolean analysis (see Figure 5.13b for comparison). The new analysis demonstrates that in the previous method the regions were much too restrictive.

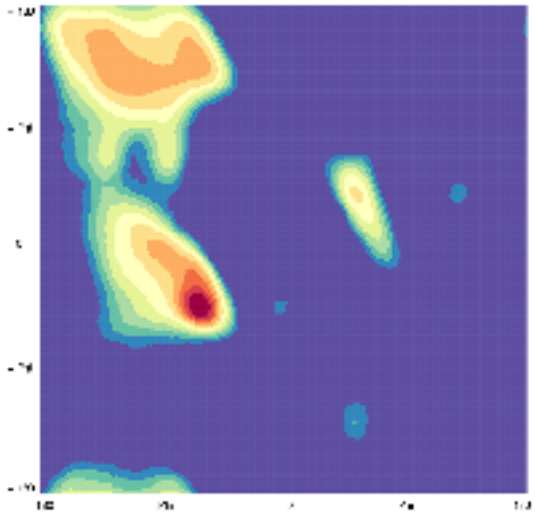
Figures 5.13a and 5.13c shows the same comparison between the allowed regions in the new analysis and the previous Boolean analysis for Glycine and Non-Gly/Pro TDMs.



(a) Glycine

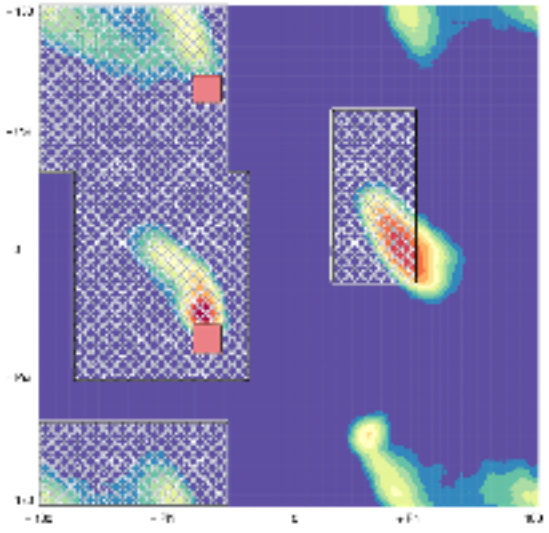


(b) Proline

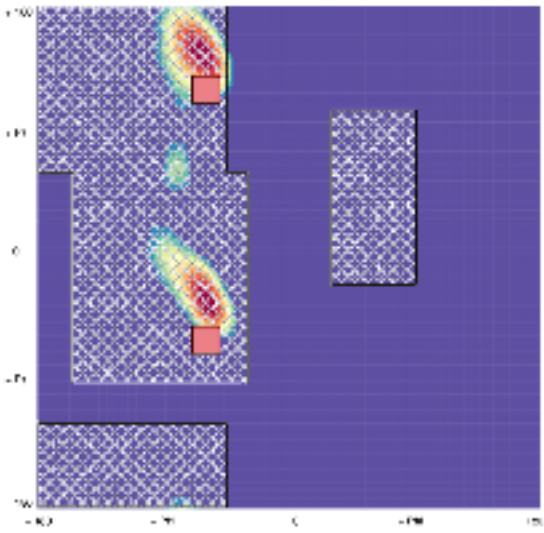


(c) Non-Gly/Pro

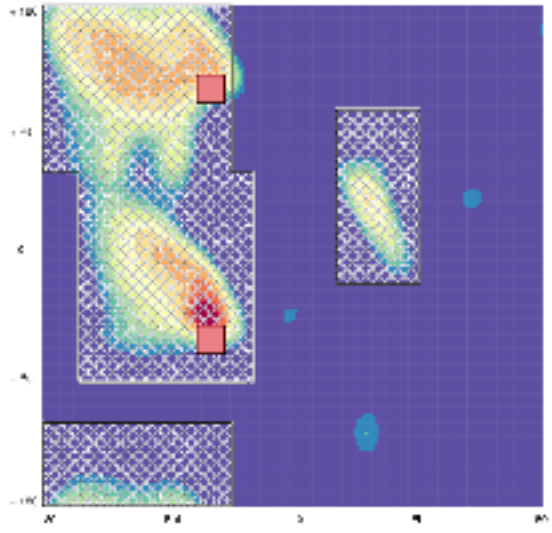
Figure 5.12: Final TDMs after applying the threshold cutoff showing the allowed regions in new analysis. (Dark-purple to dark-red heat map).



(a) Glycine - The allowed regions for glycine using the previous method where non-pink area marks the conformational space, primarily occupied by glycine residues.



(b) Proline - Pink areas mark the restricted conformation for proline residues used in the Boolean analysis.



(c) Non-gly/pro - The hatched grey area marks the regions for non-proline, non-glycine residues residues used in the Boolean analysis.

Figure 5.13: Comparison between the allowed regions in new method and old method.

5.4 Discussion

In summary, the glycine and proline analyses have been improved by moving from simple Boolean decision making with rather arbitrary boundaries to an energy-evaluation approach. Figures 5.13a, b and c clearly shows that the previous allowed regions were inaccurate and, in particular, the allowed regions for proline were incorrect. These analyses have been integrated into the new SAAPdap pipeline and web interface (Chapter 3). Detailed results of these analyses were then used to build the pathogenicity predictor using a machine learning approach (Chapter 6).

While the change from Boolean decision making to pseudo-energy estimation should have improved the analyses of glycine and proline mutations, it is recognised that limitations remain, which could be the subject of future improvements. For example, the cyclic side-chain of proline means that there is no hydrogen bond to the backbone nitrogen.

Consequently, proline cannot satisfy the backbone hydrogen bond donation requirements in α-helices and β-sheets other than at alternate position in the edge strands where hydrogen bonds are not required. This restriction is something that is currently not accounted for in our model, but is an important factor to consider in the future.

Chapter 6

Predicting Damaging Mutations (SAAPpred)

∗ *The method and results in this chapter have been published (Al-Numair NS, Martin ACR. 2013. The SAAP pipeline and database: tools to analyze the impact and predict the pathogenicity of mutations. BMC Genomics 14.3:1-11). However some of the results shown here are more recent after improving the methods.*

The identification of the effect of missense pathogenic mutations has seen great progress through the development of computing applications, the major issue being to separate neutral from pathogenic mutations. Several software tools are currently available to analyse and study these missense mutations (e.g. MutationAssessor - Section 2.6.1, SIFT - Section 2.6.3, PolyPhen2 - Section 2.6.2, Condel - Section 2.6.4 and most recently FATHMM - Section 2.6.5). These tools use various methods to make predictions, such as the conservation of amino acids in homologous sequences (Reva *et al.*, 2011), consensus deleteriousness scores (González-Pérez and López-Bigas, 2011), and position-specific scoring matrices (Adzhubei *et al.*, 2010; Ferrer-Costa *et al.*, 2005).

This chapter describes the main motivation for the project, namely the construction of the SAAPpred predictive tool using the updated Single Amino Acid Polymorphism database (SAAPdb) and Single Amino Acid Polymorphism data analyses pipeline (SAAPdap) including the enhanced analyses described in the previous two chapters. The next chapter (Cardiomyopathy Mutations) describes how the SAAPpred predictive tool was used together with the SAAPdap pipeline on a specific disease-related dataset.

6.1 Introduction

An analysis of the data in the Single Amino Acid Polymorphism database (SAAPdb) (Figure 3.20) shows that there are clear differences in the sequence and structural characteristics of SNPs and pathogenic deviations (PDs): PDs have additional, and more severe, structural effects. This is therefore a clear indicator that these analyses can be used to predict the pathogenicity of a novel mutation.

SAAPdb and SAAPdap currently provide sequence and structural analyses of mutations in structures, deposited in the Protein Data Bank (PDB). Several enhancements were made to the structural analyses (described in previous Chapters 4 and 5) and implemented in SAAPdap in order to provide further information and more detailed structural effects with the aim of creating the SAAPpred tool to be used in predicting the pathogenicity of any novel mutation in the PDB structure.

The following sections describe and discuss the various steps involved in the construction of the SAAPpred tool beginning with preliminary experiments through to a general comparison with other available predictors.

6.2 Preliminary experiments

For this preliminary work, the data and separation of neutral and pathogenic mutation as used by SAAPdb were employed. Later work shows that the boundary between these classes is different from that seen in datasets such as HumVar used in PolyPhen-2. These differences will be discussed in Section 6.2.3.5.

6.2.1 Methods

This section will describe methods used to perform preliminary data selection and preparation, describe how these data were transformed into a machine-readable format to utilise and optimise several machine learning methods finally used to train and build SAAPpred.

6.2.1.1 Data sets

The chapter describing SAAPdb (Chapter 3) describes in detail how an extensive and up-to-date dataset was successfully rebuilt, how raw data were imported from SNP repositories

and the various locus-specific mutation databases (LSMDBs) and how the sequence and structural analyses were obtained including the enhanced and improved structural analyses (Chapters 4 and 5).

In order to import and access SNP and PD mutation data from SAAPdb, a Perl script was written that incorporated Structured Query Language (SQL) queries (Section 2.2.1). An example PostgreSQL query that extracts SNPs from SAAPdb shown in Figure 6.1.

```
SELECT   a.snp_lsid, a.sprot_lsid, a.aa_wildtype,
         a.snp_protein_position, a.alleles_mutations,
         COUNT(DISTINCT s.pdb_id)
FROM     snp2annotated a, saap s
WHERE    a.snp_lsid = s.snp_lsidG
GROUP BY a.snp_lsid,a.sprot_lsid,a.aa_wildtype,
         a.snp_protein_position,a.alleles_mutations;

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

Figure 6.1: An example PostgreSQL query.
Two tables (`snp2annotated` aliased to `a` and `saap` aliased to `s`) are queried with the condition `a.snp_lsid` and `s.snp_lsid`; and the distinct PDB id count for each mutation printed among other information `a.snp_lsid`, `a.sprot_lsid`, `a.aa_wildtype`, `a.snp_protein_position` and `a.alleles_mutations` as defined by the `GROUP BY n.name` clause; all PostgreSQL commands and functions are given in capitals.

The Perl script was enhanced to allow the following command-line parameters:

- `-strtype` which defined the type of PDB structure and could take three values, 'Native', 'Mutant' or 'All' (i.e. both Native and Mutant);
- `-restype` which indicated the resolution of the structure and also took three values: 'High' (equivalent to ≤ 2.0Å), 'Low' (equivalent to > 2.0Å) or 'All'.

The PDB resolution was determined from a PDB file using an external program (`getresol`, ACRM, unpublished). The aim was to divide PDB data into either high resolution or low resolution structures in order to determine whether high resolution structures would be more useful with machine learning method used in later stages.

For initial experiments, from 611,641 pathogenic deviations (PDs) and 71,409 neutral mutations (SNPs) 405,497 PDs (i.e. identified as damaging by at least one of SAAPdap structural/sequence analyses) and 45,699 SNPs (i.e. negatively identified as damaging by at least one of SAAPdap structural/sequence analyses) were used as described in Table 6.1 and Figure 6.2.

Table 6.1: Breakdown of the number of mutations in SAAPdb and their mapping to structure. In some cases, several hundred structures are available (e.g. haemoglobin, carbonic anhydrase, prthrombin, transthyretin, insulin, CDK2, lysozyme) and, on average there are two copies of each chain in each PDB file associated to SNPs and four copies for PDB files associated with PDs.

Number of Mutations	PDs	SNPs
Mapped to UniProtKB/Swiss-Prot	13,059	48,452
Mapped to PDB	6,527	17,915
Mapped to multiple PDBs	202,566	33,369
Mapped to multiple Chains	611,641	71,409

6.2.1.2 Class value

The second step was to run the SAAPdap analyses pipeline (Section 3.8.3) for complete structural and sequence analyses on each mutation using the PDB chain that matched a specified UniProt accession number (obtained from PDBSWS, (Martin, 2005), Section 2.1.4). SAAPdap was used in place of the pre-calculated analysis in SAAPdb because the enhanced analyses (Chapter 4 and 5) have not yet been incorporated into SAAPdb. The SAAPdap pipeline code calls various plugins ('binding', 'buriedcharge', 'cispro', 'clashes', 'corephilic', 'glycine', 'hbonds', 'impact', 'interface', 'proline', 'sprotft', 'ssgeom', 'surfacephobic' and 'voids') each of which implements an individual analysis. Another Perl program `runSAAPdapOnGrid` was written to perform this step. This program processed and batched SNP and PD mutations clean data; created the `.sh` file and submitted jobs to a local computing farm (based on the Oracle Grid Engine)[1]; and recorded any errors. A further enhancement to this program was to use `-recordErrors` which records all errors and categorizes them and re-submits failed jobs to the grid one more time before saving them as an error to be looked at for any further action.

The output of the pipeline program (SAAPdap) is saved in a JavaScript Object Notation (JSON) format file, an example of a JSON file obtained from SAAPdap is shown in Appendix [C]. Appendix [C.i] contains detailed tables that explain of the SAAPdap JSON file output and the assigned class. At this stage another Perl program was written to parse the

[1]Previously known as Sun Grid Engine (SGE) it is an open source batch-queuing system, developed and supported by Sun Microsystems and now by Oracle. SGE is used on a computer farm cluster and is responsible for accepting, scheduling, dispatching, and managing the remote and distributed execution of large numbers of standalone, parallel or interactive user jobs. It also manages and schedules the allocation of distributed resources such as processors, memory, disk space.

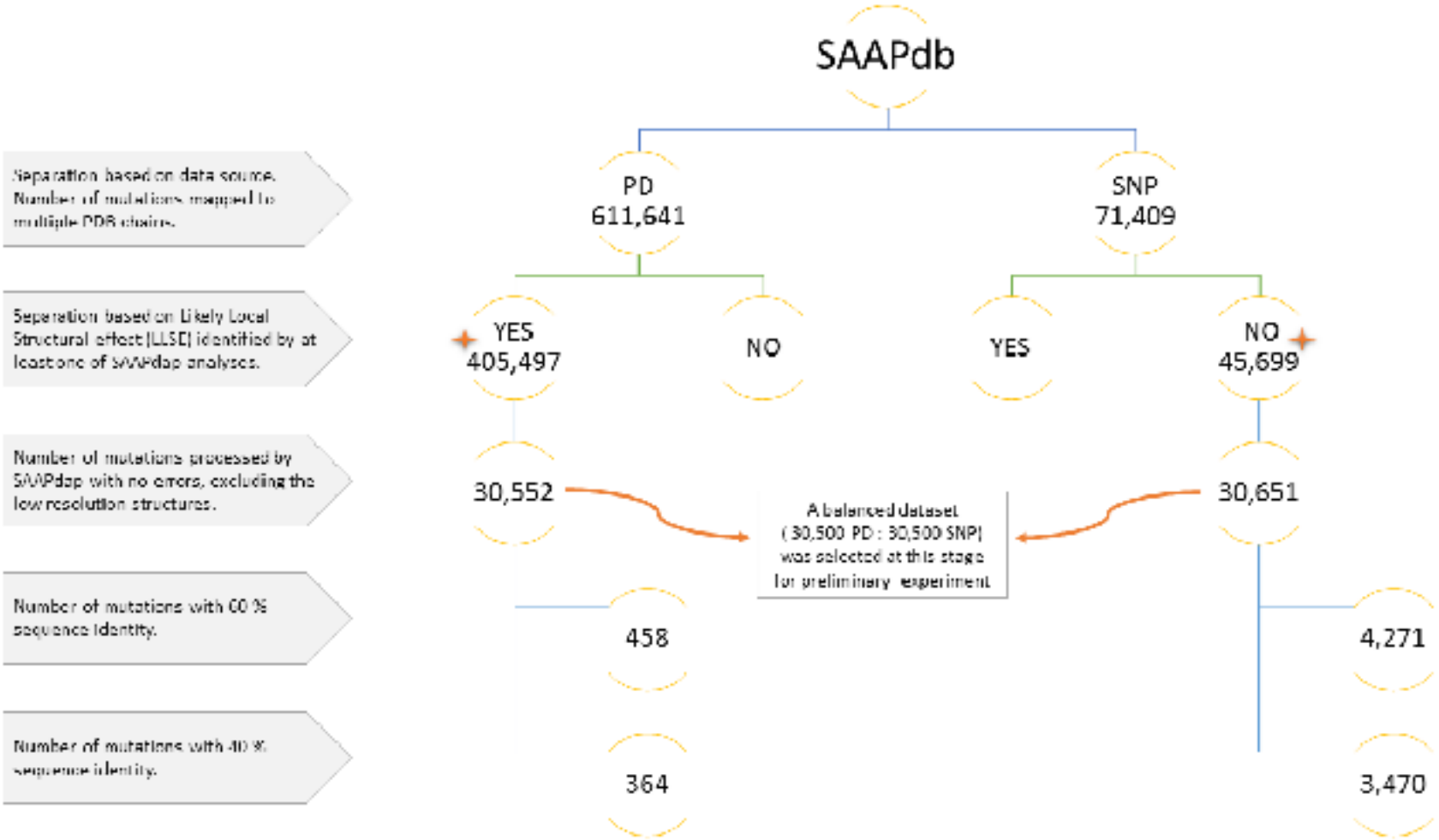


Figure 6.2: Data selection for preliminary experiments.

JSON file, JSON2CSV, which uses several Perl modules to extract structural and sequence results obtained from SAAPdb for each mutation and give it a unique ID (see Figure 6.3) and saves the data into a comma-separated value (CSV) file. JSON2CSV program will list each mutation with its features (analysis results) and a class. A mutation could adopt one of two class values: Neutral (snp), or pathogenic (pd). Figure 6.4 shows example of two entries obtained from the CSV file.

At this stage another Perl program was written, CSV2ARFF (ACRM, unpublished), to convert the CSV file obtained from the previous step to an Attribute-Relation File Format (ARFF) file to be used in subsequent machine learning experiments.

At the end of this step for the preliminary experiments a balanced set of mutation data from SAAPdb was used that consisted of 30,500 SNPs mapped to PDB structures and a random selection of 30,500 PDs (also mapped to PDB structures). This was processed by SAAPdap without any errors or missing structural or sequence analyses results. Where several structures were available for a mutated residue, each was used as an independent data point for machine learning. At this stage the analyses was restricted to high-resolution PDB entries ($\leq$ 2.0Å) (see Figure 6.2).

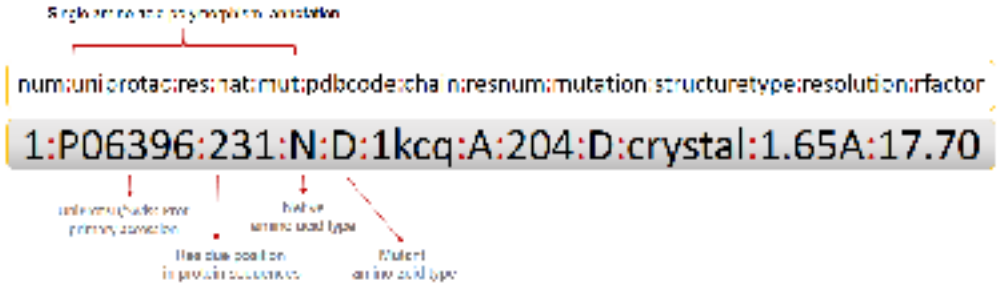


Figure 6.3: An example of a single amino acid polymorphism annotation.
Each mutation is a unique combination of: (i) UniProtKB/Swiss-Prot primary accession number of a protein in which the mutation occurs, (ii) the amino acid type found native in disease unaffected individuals, (iii) residue position in protein sequences, as reported by the UniProtKB/Swiss-Prot, and (iv) the amino acid type found in the mutated genotype.

```
1:P06396:231:N:D:1kcq:A:204:D:crystal:1.65A:17.70%:
0,0,0,0,0,0,0,0,0,0,0,0,0,0,-3.07199999999999,68.876,0.666097558274395,20,0,0.08,
1,0,0,85.624,54.96,47.454,41.197,40.834,37.345,32.599,30.618,24.8,23.758,85.624,
54.96,47.454,41.197,40.834,37.345,32.599,30.618,24.8,23.758,-5.40,-100,-100,0,SNP

2:P31785:222:R:C:2b5i:C:200:C:crystal:2.30A:22.50%:
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24.319,1,20,-1.84,0,1,0,0,100.853,85.731,76.998,
55.629,50.863,47.287,39.666,32.125,31.622,29.827,100.853,85.731,76.998,55.629,
50.863,47.287,39.666,32.125,31.622,29.827,-3.68,-100,-100,0,FD
```

Figure 6.4: An example CSV file.
Obtained from the `JSON2CSV` program consisting of a unique ID (Figure 6.3) and the forty-seven features resulting from the SAAPdap analysis.

6.2.1.3 Feature encoding (Training attributes)

In addition to the class value, each mutation had forty-seven features (attributes) assigned to it. For initial experiments, all forty-seven available features were selected as potential predictors of pathogenicity. Table 6.2 shows the forty-seven features obtained from structural or sequence analyses in SAAPdap.

Table 6.2: The forty seven features obtained from SAAPdap.

Number	Attribute	Variable type	Value type
1	Binding	Binding analyses	Boolean
2-14	SProtFT	The 13 SwissProt features	Boolean
15	Interface	Interface analyses	△ sa
16	Relaccess	The relative accessibility of this residue	△ sa
17	Impact	Conservation score at this position in the alignment	Score
18	HBonds	HBonds	Energy
19	SPhobic	Surface hydrophobicity of mutant residue	△
20	CPhilic	Buried hydrophilic	△
21	BCharge	Buried charge	△
22	SSGeom	SSGeom analyses	Boolean
23	Voids	Voids analyses	Volume
24-33	MLargest	Top 10 voids in the mutant	Volume
34-43	NLargest	Top 10 voids in the native	Volume
44	Clash	Clash analyses	Energy
45	Glycine	Glycine analyses	Energy
46	Proline	Proline analyses	Energy
47	CisPro	CisPro analyses	Boolean

6.2.2 Machine learning

There are many approaches to machine learning and many algorithms have been developed. It is often the case that a researcher needs to experiment with several different options in order to identify the technique that provides the most appropriate solution to their particular problem. With this is mind, software such as WEKA (Hall *et al.*, 2009) and RapidMiner (Mierswa *et al.*, 2006) implement a wide range of tools and techniques. At the same time, they provide a user-friendly interface for the creation, optimisation and evaluation of machine learning experiments.

6.2.2.1 Survey of classifiers using WEKA

WEKA (Hall *et al.*, 2009) (Section 2.5) is an open-source collection of machine learning algorithms written in Java. Machine learning experiments began by selecting the most appropriate machine learning method. All the supervised classifiers implemented in `WEKA_3.6.9` were trained on the dataset (using ALL attributes). In this step, the values of default parameters were not changed. The reason for this was to validate the choice of dataset and attributes.

All of the methods tested had relatively high specificity (~0.9) and lower sensitivity (~0.4-0.5). In other words, classifiers were more likely to miss a prediction of pathogenicity (a false negative) than report a false positive. This is a desirable result when the aim of the classifier is to indicate mutations as damaging with high confidence. The Matthews Correlation Coefficient (MCC) is the best single indicator of a model's performance, as it takes into account both true and false positives and negatives: TP, TN, FP and FN. For more details, see the definition of the MCC given in Section 2.3.7. Most of the methods tested had similar, mediocre MCC scores of ~0.4, indicating that the selected parameters had some (but far from perfect) predictive value, potentially to be improved once various parameters combination are tested and optimized. Furthermore, performance measures indicated that there were no clear benefits in using numerical (regression) models compared to binary (tree and other rule-based model) classification.

Based on the initial survey and other previously published methods, two of the supervised learning methods were tested in more detail using a wider range of parameters and training data set-ups. The results of this step are presented in the next section.

6.2.3 Results and discussion

6.2.3.1 Neural network (NN)

The WEKA implementation of a neural network (`weka.classifiers.functions.MultilayerPerceptron`) was used. The model was trained on normalised attribute values[2] using a sigmoid function, using 5 or 50 hidden nodes over 500 epochs, with all other model variables set to WEKA's defaults. An accuracy (ACC) of 0.862 and a Matthews Correlation Coefficient (MCC) of 0.803 (based on 10-fold cross-validation and 5 hidden nodes) were obtained. Adjusting the value of parameters clearly affect the result and outperformed the initial test run.

6.2.3.2 Random forest (RF)

The other selected method was the Random Forest implemented in the `weka.classifiers.trees.RandomForest` method. This is based on Breiman's Random Forest algorithm (Breiman, 2001). This algorithm creates un-pruned random trees, with no limit on tree depth. As this classifier is unable to take into account missing attribute values, they must be created: typically they are calculated either from mean values (for continuous numerical attributes), or the most common category (for nominal attributes). In these experiments, as the percentage of missing data (attributes) was very small, it was instead decided to remove them from the training and testing datasets when the model was built.

The balanced dataset of 30,500 SNPs 30,500 PDs mutations extracted and prepared from SAAPdb previously were used in training and testing the model using a Random Forest method. All results obtained from WEKA results from 10-fold cross-validation testing.

The common recommendation for the tree-number optimisation is to increase the number of trees until the Out Of Bag (OOB) error[3] stops decreasing. A range from 10-2500 trees per ensemble was tested; the WEKA implementation of the Random Forest method is very memory-demanding and large amount of RAM must be allocated to build Random Forest using up to 2500 trees for the training dataset. Table 6.3 shows the Random Forest's performance while surveying parameter space, starting with different numbers of trees T = 10 - 2500. T = 1000 was selected as an optimum tree-number as increasing the number of trees above that number did not improve the performance with an accuracy (ACC) of 0.946%, a Matthews Correlation Coefficient (MCC) of 0.893 and area under the ROC curve of 0.985.

[2] i.e. all attribute values are adjusted to range between 0 and 1

[3] This is the Out Of Bag error – an internal error estimate of a Random Forest as it is being constructed

Table 6.3: Random Forests performance - surveying the parameter space (SAAPdb).
Performance measures: true positive (TP); false positive (FP); true negative (TN); false negative (FN); accuracy (ACC), precision (PREC), specificity (SPEC), sensitivity (SENS), Matthew's correlation coefficient (MCC), F-measure (F); Area under ROC curve (ROC); m_{try} stands for the number of randomly chosen attributes in every split; T is the number of trees. Best performance highlated in light blue, all scores are averages over 10-folds WEKA of cross-validation and each run was repeated 10 times.

Parameters		TP	FN	TN	FP	ACC	PREC	SPEC	SENS	MCC	F	ROC
T	m_{try}											
10	7	27890	2610	29496	1004	0.941	0.965	0.967	0.914	0.883	0.939	0.977
100	7	28217	2283	29474	1026	0.946	0.965	0.966	0.925	0.892	0.945	0.984
500	7	28243	2257	29470	1030	0.946	0.965	0.966	0.926	0.893	0.945	0.985
1500	7	28232	2268	29484	1016	0.946	0.965	0.967	0.926	0.893	0.945	0.985
2000	7	28236	2264	29485	1015	0.946	0.965	0.967	0.926	0.893	0.945	0.985
2500	7	28235	2265	29495	1005	0.946	0.966	0.967	0.926	0.894	0.945	0.985
1000	5	28213	2287	29314	1186	0.943	0.960	0.961	0.925	0.887	0.942	0.983
1000	7	28232	2268	29493	1007	0.946	0.966	0.967	0.926	0.893	0.945	0.985
1000	15	28289	2211	29715	785	0.951	0.973	0.974	0.928	0.903	0.950	0.989
1000	30	28394	2106	29863	637	0.955	0.978	0.979	0.931	0.911	0.954	0.991
1000	35	28423	2077	29860	640	0.955	0.978	0.979	0.932	0.912	0.954	0.991
1000	40	28431	2069	29858	642	0.956	0.978	0.979	0.932	0.912	0.954	0.991
1000	45	28437	2063	29846	654	0.955	0.978	0.979	0.932	0.912	0.954	0.991

Having selected the number of trees, the next step was trying different numbers of features (attributes) m_{try} ranging from 5 up to 45 (the maximum is 47). Using more features improved the training process, m_{try} = 40 was selected. The best performance using T= 1000 and m_{try} = 40 gave us an ACC of 0.956, MCC of 0.912 and ROC of 0.991 based on WEKA 10-fold cross-validation. Each run was repeated ten times and a summary of average score in Table 6.3.

6.2.3.3 Training SAAPpred on different dataset

At that stage other datasets were examined with the available pathogenicity predictor and the dataset they used. Part of our preliminary experiment was trying to explore training and testing on the PolyPhen dataset (HumDiv: 5564 deleterious + 7539 neutral mutations from the same set of 978 human proteins and HumVar: 22196 deleterious + 21119 neutral mutations in 9679 human proteins), and the separate HumDiv and HumVar datasets using the Random Forest method with WEKA 10-fold cross-validation. Any new dataset for training or testing using machine learning methods was prepared following the same methods described in sections 6.2.1.2 and 6.2.1.3 first by running SAAPdap to obtain the structural and sequence results in JSON format; convert JSON to CSV file then assigning the class values (*snp* or *pd*) and then selecting the feature encoding (Training attributes) and finally producing an ARFF file to be used for training and testing experiments.

6.2.3.4 Summary of preliminary training results

Figure 6.5 shows the ROC curve of the preliminary results and Table 6.4 shows a summary of preliminary experiments performance measures. At that stage no filtering was performed and each entry was used as an independent data point for machine learning. PolyPhen dataset (HumVar and HumDiv) using fully and no balance (*snp*:*pd*) data extracted nor checking for missing results.

From the ROC curve presented in Figure 6.5, training on SAAP data works best presumably because the data set is so large. PolyPhen dataset does not do better than HumVar or HumDiv, presumably because the SNP/PD boundary in HumVar and HumDiv is different, so the training data are less clear. HumDiv, was compiled from all damaging alleles with known effects on the molecular function causing human Mendelian diseases, present in the UniProtKB database, together with differences between human proteins and their closely

related mammalian homologs, assumed to be non-damaging where is HumVar, consisted of all human disease-causing mutations from UniProtKB, together with common human nsSNPs (Minor Allele Frequency > 1%) without annotated involvement in disease, which were treated as non-damaging.

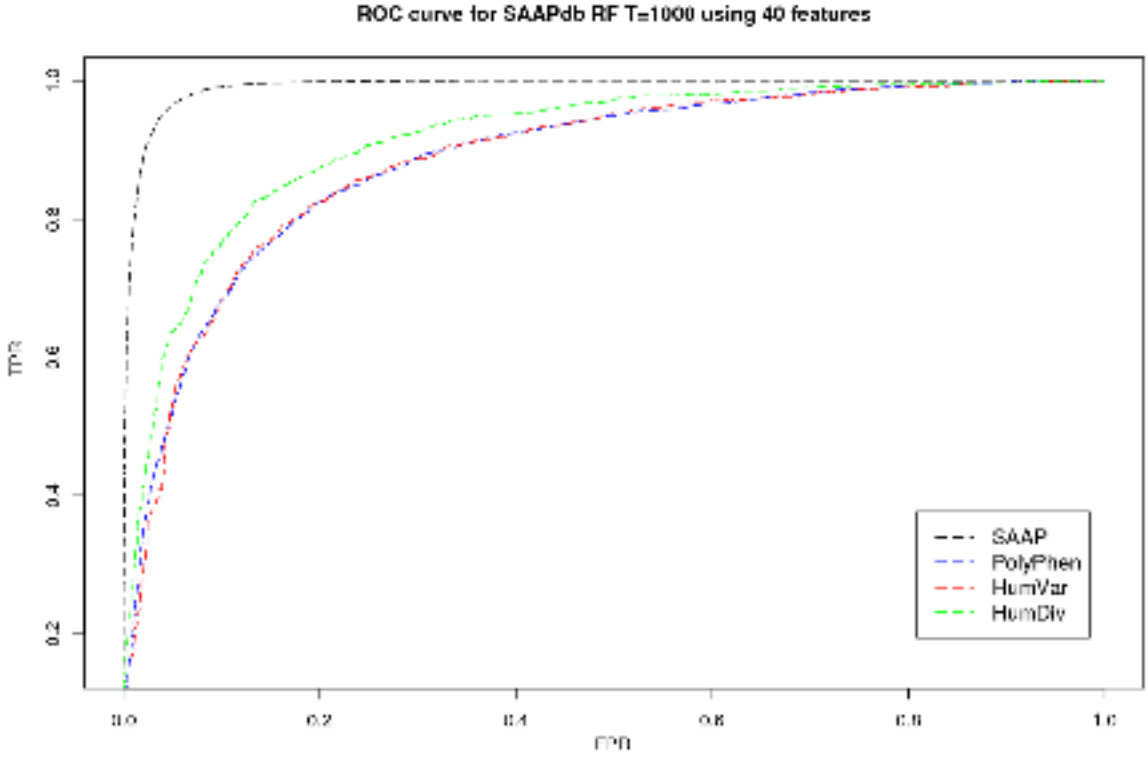


Figure 6.5: ROC curves of SAAPpred predictor training on different dataset SAAPdb, Polyphen (HumVar + HumDir), HumVar and HumDiv dataset. Results from WEKA 10-fold cross-validation.

Table 6.4: Summary of Preliminary experiments.
Performance measures: true positive (TP); false positive (FP); true negative (TN); false negative (FN); accuracy (ACC), precision (PREC), specificity (SPEC), sensitivity (SENS), Matthew's correlation coefficient (MCC), F-measure (F); Area under ROC curve (ROC); m_{try} stands for the number of randomly chosen attributes in every split; T is the number of trees. All scores are averages over 10-folds WEKA of cross-validation.

Traning dataset	method	Parameters		TP	FN	TN	FP	ACC	PREC	SPEC	SENS	MCC	F	ROC
		T	m_{try}											
humDiv	RF	1000	40	1701	280	1384	285	0.845	0.856	0.829	0.859	0.688	0.858	0.916
humVar	RF	1000	40	2048	206	727	384	0.825	0.842	0.654	0.909	0.591	0.874	0.885
SAAPdb	RF	1000	40	28431	2069	29858	642	0.956	0.978	0.979	0.932	0.912	0.954	0.991
		$hidden-nodes$	$epochs$											
SAAPdb	NN	5 or 50	500	24957	5543	27579	2921	0.861	0.895	0.904	0.818	0.725	0.855	0.916

6.2.3.5 Testing SAAPpred (trained on SAAPdb) on HumVar data

WEKA allows you to save trained models as Java binary serialized objects and use them to obtain predictions/classifications via the command line. As mentioned before the HumVar dataset is a popular benchmark dataset composed of pathogenic and common nsSNVs used by PolyPhen2 and CONDEL to benchmark their prediction models for pathogenic SNPs. To test HumVar using the best SAAPpred model built in the previous section (trained on the SAAPdb dataset) the HumVar dataset was prepared as described above to generate an ARFF file similar to the one used in building the classifier model. The actual class (*snp* or *pd*) is contained in the file and the output will contain both the actual and predicted class. An example of the output format is shown in Figure 6.6.

```
Sample output:
=== Predictions on HumVar Test data ===

 inst#     actual      predicted     error     prediction
 1          2:pd        1:snp          +         0.329
 2          1:?         1:snp                    0.886
 3          2:pd        2:pd                     0.587
 4          1:snp       1:snp                    0.947
 5          2:pd        1:snp          +         0.683
 6          1:snp       1:snp                    0.952
 7          2:?         2:pd                     0.729
 8          1:snp       1:snp                    0.54
 ...
```

Figure 6.6: An example predictions output file.
Using the SAAPpred models training on SAAPdb and testing on HumVar dataset. The output will contain test `instance` index, `actual` class index:`actual class value`, `prediction` class index:`prediction` class value, [+], `probability` of prediction class value. If the test class attributes were marked by "?", the "actual" column, which can be ignored, simply states that that instance belongs to an unknown class. The error label will indicate "+" only for those items that were mispredicted. The probability that an instance actually belongs to the positive class is estimated in the second "prediction" column.

Having trained on SAAPdb, testing on 1,540 SNPs and 7,182 PDs from the HumVar dataset that mapped to structure gives an accuracy (ACC) of 0.446 and MCC of 0.135, essentially a random prediction. This appears to be because of the different definition of the 'boundary' between SNPs and PDs. Mutations form a spectrum from completely silent SNPs at one end, to 100% penetrance, Mendelianly inherited PDs at the other end. As shown in Figure 6.7, different datasets use different thresholds to separate the data into two sets or may consider only the extremes. Prediction of the extremes may appear to be a trivial problem, but this is not always the case, some damaging mutations are very hard to predict.

HumVar uses a broader definition of PDs than the SAAPdb data; in contrast, the SAAPdb definition of SNPs is rather wide (anything in dbSNP not annotated as being involved in disease) while the definition in HumVar enforces the requirement that SNPs are present in at least 1% of a normal population. Table 6.5 shows a comparison of SAAPdb and HumVar datasets. While there is an overlap of approximately 50% between PDs in the two datasets, there is virtually no overlap in the SNP datasets.

Table 6.5: Comparison of SAAPdb and HumVar datasets. While there is an overlap of approximately 50% between PDs in the two datasets, there is virtually no overlap in the SNP datasets.

		SAAPdb			
		SNP	PD	Not present	Total
HumVar	SNP	2	24	1,539	1,565
	PD	0	3,411	4,509	7,920
	Not present	17,911	3,092	—	—
	Total	17,915	6,527	—	

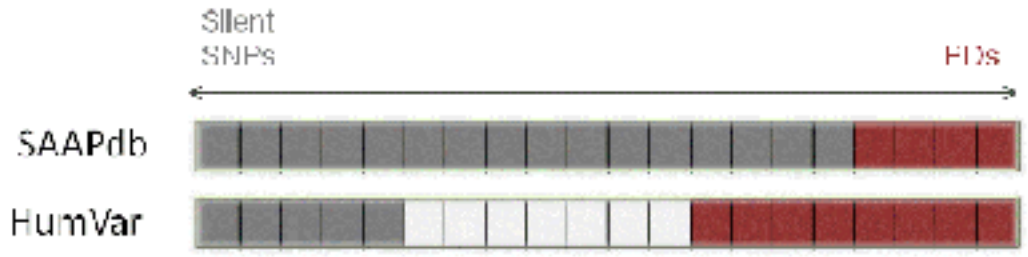


Figure 6.7: The penetrance of a mutation lies on a scale between 'True SNPs' which show no phenotypic effect at one extreme to Mendelianly inherited PDs with 100% penetrance at the other. In SAAPdb, we use a very conservative definition of PDs, but a rather wide definition of SNPs. In contrast, HumVar uses a somewhat broader definition of PDs, but a much more conservative definition of SNPs and does not consider mutations that lie in the middle.

6.3 Main experiments

In the previous experiment, since very different definitions of SNP/PD boundaries were used for training and testing, it is not surprising that a poor performance was obtained. As expected from the different boundary used in SAAPdb and HumVar, SNPs are considerably over-predicted, consistent with SAAPdb's broader definition of SNPs. Consequently, refined models were built and tested with the HumVar dataset.

6.3.1 Data sets

HumVar (Version v2.2.2 - 2011/12) contains 22,196 deleterious mutations and 21,151 neutral mutations of which 7,192 and 1,540, respectively, can be mapped to structure. Consequently, to obtain a balanced dataset, only 3,080 mutations (all 1,540 neutral and 1,540 randomly selected deleterious) can be used (see Figure 6.8).

The strength of a model learned by a classifier, and its ability to generalise and perform on new testing examples, is greatly influenced by the size and quality of the training dataset. Compared with the SAAPdb dataset, HumVar is a small dataset with an even smaller subset that maps to PDB structures. Consequently, it will be a less informative for a machine learning predictor than SAAPdb. Nonetheless it is necessary to use this dataset for valid comparison with other available method. Data were prepared in ARFF format as described previously.

6.3.2 Training and testing on HumVar data

The whole dataset was divided into 10 subsets; each of which used all 1,540 neutral mutations with a random selection of 1,540 deleterious mutations from the total of 7,182. Ten train/test runs were then preformed, each using 10-fold cross-validation and the results from the ten runs were then averaged. Table 6.6 shows the performance of the 10 SAAPpred classifier tried on a balanced HumVar dataset. At this stage we used a unique mutation level filtering, in other words the same mutation (Uniprot:Nat:Num:Mut) does not occur in training and testing sets (see Figure 6.9[1]). If the mutation mapped to multiple PDB structures/chains best PDB/chain (based on resolution) was chosen for each mutation. Although there are no cases of the same mutation in the training and test sets, this was still considered to be partially-cross-validated since there may be a 'structure overlap' between training and testing (i.e the same PDB ID maybe chosen for different munitions). In other

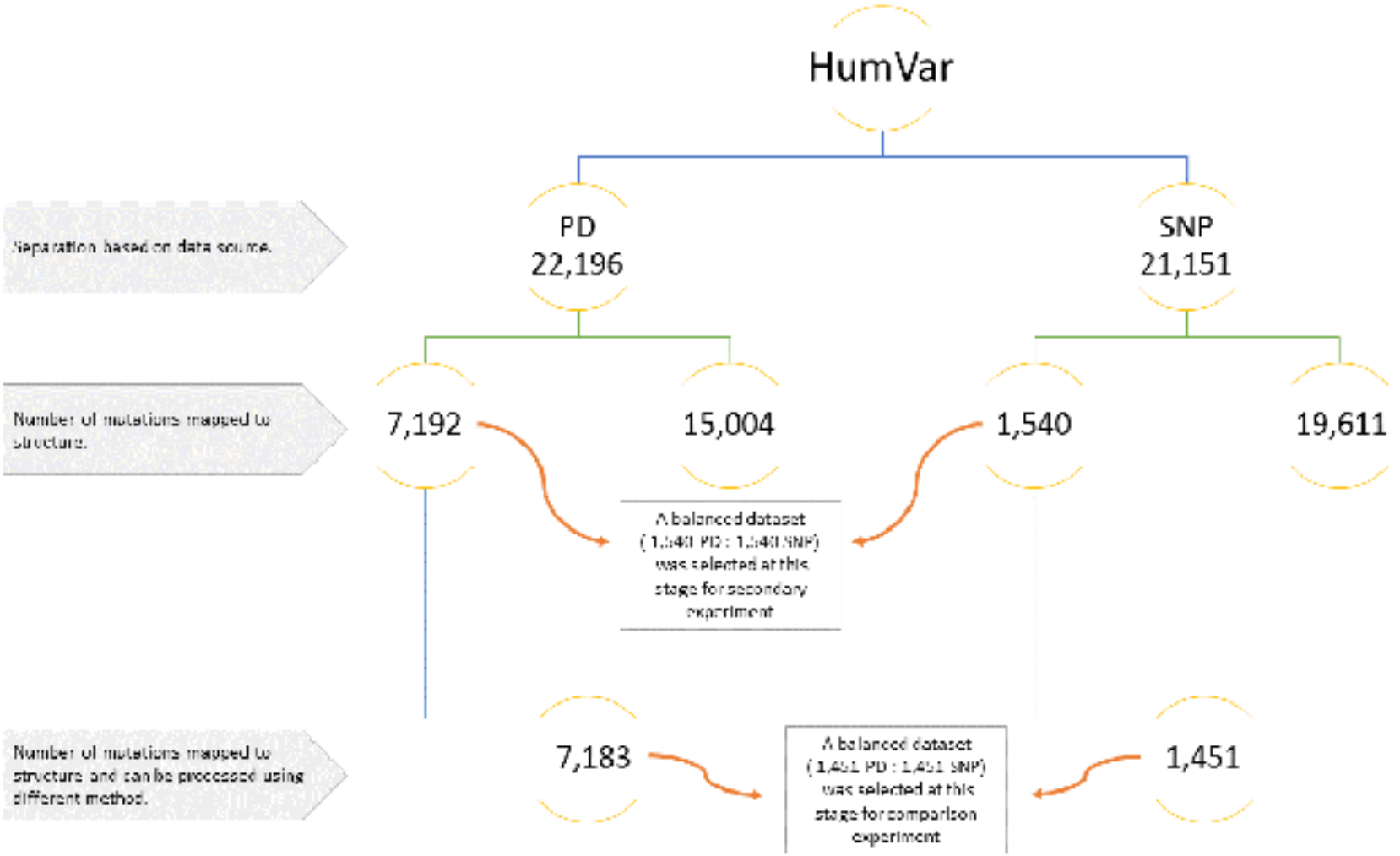


Figure 6.8: Data selection for main experiments.

words while there is no overlap between mutations in training and testing sets, multiple mutations may occur at the same site in the same structure and some of these may be in the training set while others are in the test set. All scores were averaged over 10-folds of WEKA cross-validation and this was repeated 10-times (using different randomly selected sets of deletion mutations) and the results averaged for each model giving an MCC of 0.893 and ACC of 0.944.

To avoid 'structure overlap' between the training and testing data during cross-validation preformed by WEKA (which was present in the previous experiment), a Perl program was written to save all the available PDB structures to which HumVar mutations can map; then each unique mutation (UniProt:Nat:Num:Mut) was read and one PDB structure assigned to it and taken from the list of PDB structures (see Figure 6.9[2]). By using this method it was ensured that there are no cases of the same mutation nor the same site in the same structure 'structural overlap' between training and testing set. The values for the fully cross-validated assessment of SAAPpred were obtained from 10-fold cross-validation performed during the WEKA training and used all 1540 SNPs from HumVar that mapped to structure with a random sample of 1540 of the 7182 PDs that mapped to structure. This was repeated 10-times (using different random sample of the PDs) and the results averaged. Table 6.7 presents the performance of the fully cross-validated classifiers ACC = 0.846 and MCC = 0.692.

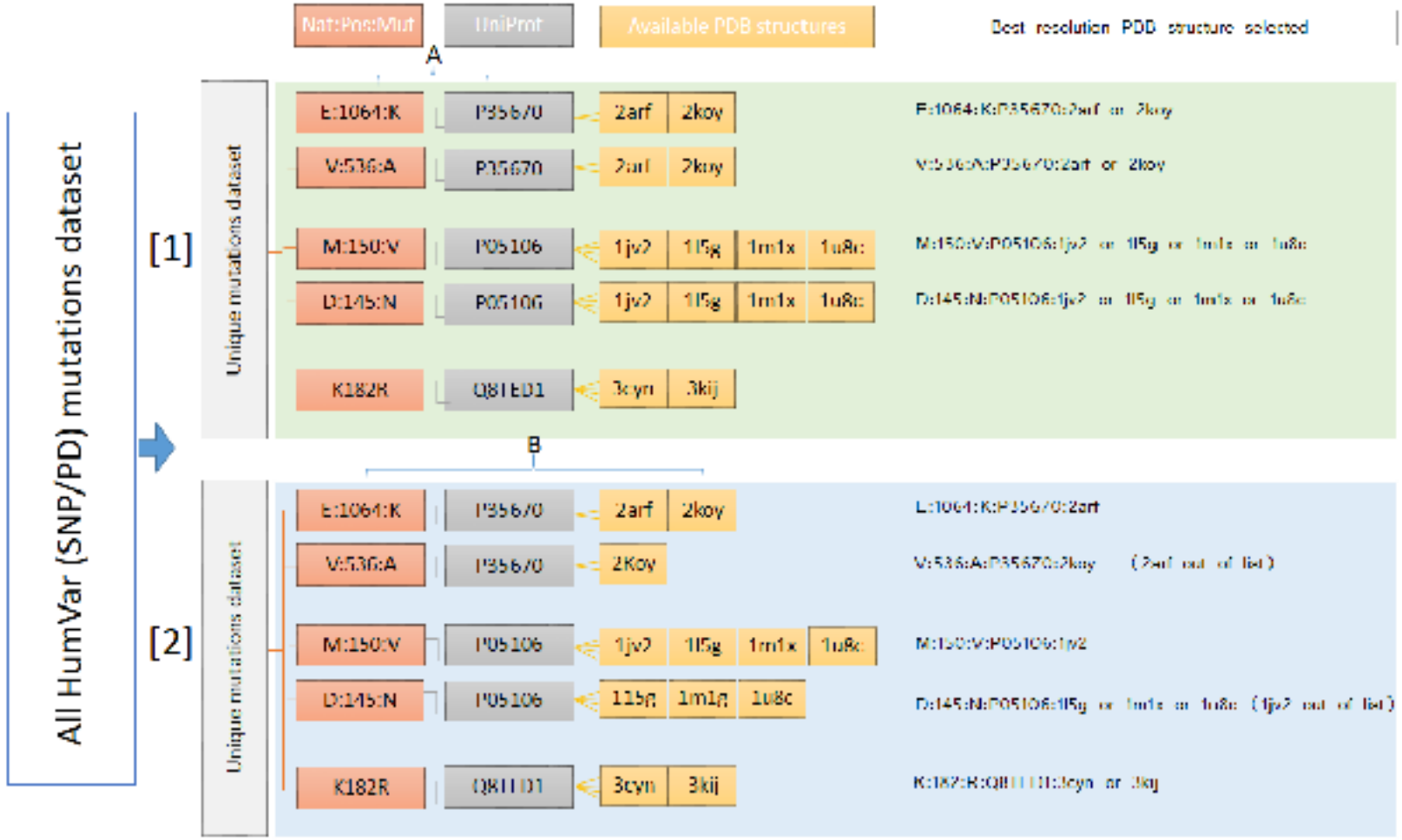


Figure 6.9: HumVar dataset selection for machine learning. At stage [1] we used a unique mutation level filtering at the beginning and assigned the best available PDB structure, in other words the same mutation (UniProt:Native:Number:Mutant) does not occur in training and testing sets but there may be a structure overlap between training and testing (i.e. the same PDB structure representing the same native residue mutated to different residue). In stage [2] the 'structure overlap' between the training and testing data is avoided by using each PDB structure once.

Table 6.6: Evaluation of performance of the 10 SAAPpred classifiers using a balanced dataset of mutations that map to structure extracted from HumVar (1,540 neutral mutations and ten random selections of 1,540 deleterious mutations). Dataset filtered following Figure 6.9[1].
Performance measures: true positive (TP); false positive (FP); true negative (TN); false negative (FN); accuracy (ACC), precision (PREC), specificity (SPEC), sensitivity (SENS), Matthew's correlation coefficient (MCC), F-measure (F); m_{try} stands for the number of randomly chosen attributes in every split; T is the number of trees. All scores are averaged over 10-folds of WEKA cross-validation. This was repeated 10-times and the results averaged for each Model.

SAAPpred classifier	T	m_{try}	TP	FN	TN	FP	**ACC**	PREC	SPEC	SENS	**MCC**	F	ROC
Model (1)	1000	40	1360	180	1539	1	0.941	0.999	0.999	0.883	0.888	0.938	0.999
Model (2)	1000	40	1371	169	1539	1	0.945	0.999	0.999	0.891	0.895	0.942	0.999
Model (3)	1000	40	1375	165	1538	2	0.946	0.999	0.999	0.893	0.897	0.943	0.999
Model (4)	1000	40	1371	169	1539	1	0.945	0.999	0.999	0.890	0.895	0.942	0.999
Model (5)	1000	40	1369	171	1540	0	0.944	1.000	1.000	0.889	0.894	0.941	0.999
Model (6)	1000	40	1354	186	1540	0	0.940	1.000	1.000	0.879	0.886	0.936	0.999
Model (7)	1000	40	1362	178	1540	0	0.942	1.000	1.000	0.885	0.891	0.939	0.999
Model (8)	1000	40	1372	168	1539	1	0.945	0.999	0.999	0.891	0.895	0.942	0.999
Model (9)	1000	40	1371	169	1540	0	0.945	1.000	1.000	0.890	0.895	0.942	0.999
Model (10)	1000	40	1372	168	1540	0	0.945	1.000	1.000	0.891	0.896	0.942	0.999
Average							0.944	1.000	1.000	0.888	0.893	0.941	0.999

Table 6.7: Evaluation of performance of the 10 SAAPpred classifiers using a balanced dataset of mutations that map to structure extracted from HumVar (1,540 neutral mutations and ten random selections of 1,540 deleterious mutations). Dataset filtered following Figure 6.9[2].
Performance measures: true positive (TP); false positive (FP); true negative (TN); false negative (FN); accuracy (ACC), precision (PREC), specificity (SPEC), sensitivity (SENS), Matthew's correlation coefficient (MCC), F-measure (F); m_{try} stands for the number of randomly chosen attributes in every split; T is the number of trees. The highest and lowest scores are coloured blue and red, respectively, all scores are averaged over 10-folds of WEKA cross-validation. This was repeated 10-times and the results averaged for each Model.

SAAPpred classifier	T	m_{try}	TP	FP	TN	FN	ACC	PREC	SPEC	SENS	MCC	F	ROC
Model(1)	1000	40	1299	241	1321	219	0.851	0.856	0.858	0.844	0.701	0.850	0.926
Model(2)	1000	40	1296	244	1305	235	0.844	0.847	0.847	0.842	0.689	0.844	0.921
Model(3)	1000	40	1290	250	1305	235	0.843	0.846	0.847	0.838	0.685	0.842	0.923
Model(4)	1000	40	1282	258	1320	220	0.845	0.854	0.857	0.832	0.690	0.843	0.915
Model(5)	1000	40	1303	237	1315	225	0.850	0.853	0.854	0.846	0.700	0.849	0.919
Model(6)	1000	40	1292	248	1304	236	0.843	0.846	0.847	0.839	0.686	0.842	0.917
Model(7)	1000	40	1297	243	1310	230	0.846	0.849	0.851	0.842	0.693	0.846	0.921
Model(8)	1000	40	1276	264	1314	226	0.841	0.850	0.853	0.829	0.682	0.839	0.915
Model(9)	1000	40	1313	227	1325	215	0.856	0.859	0.860	0.853	0.713	0.856	0.924
Model(10)	1000	40	1275	265	1310	230	0.839	0.847	0.851	0.828	0.679	0.837	0.910
Average							0.846	0.851	0.853	0.839	0.692	0.845	0.919
Model(1)	1000	4	1312	228	1429	111	0.890	0.922	0.928	0.852	0.782	0.886	0.941
Model(2)	1000	4	1286	254	1426	114	0.881	0.919	0.926	0.835	0.764	0.875	0.939
Model(3)	1000	4	1318	222	1421	119	0.889	0.917	0.923	0.856	0.780	0.885	0.944
Model(4)	1000	4	1308	232	1422	118	0.886	0.917	0.923	0.849	0.775	0.882	0.934
Model(5)	1000	4	1305	235	1397	143	0.877	0.901	0.907	0.847	0.756	0.873	0.945
Model(6)	1000	4	1302	238	1405	135	0.879	0.906	0.912	0.845	0.759	0.875	0.936
Model(7)	1000	4	1336	204	1401	139	0.889	0.906	0.910	0.868	0.778	0.886	0.943
Model(8)	1000	4	1339	201	1410	130	0.893	0.912	0.916	0.869	0.786	0.890	0.942
Model(9)	1000	4	1313	227	1413	127	0.885	0.912	0.918	0.853	0.772	0.881	0.940
Model(10)	1000	4	1308	232	1422	118	0.886	0.917	0.923	0.849	0.775	0.882	0.937
Average							0.885	0.913	0.919	0.852	0.773	0.882	0.940

To enhance the predictor, different numbers of features (attributes) were explored using the same dataset used in Table 6.7. A range between 10 and 45 features was selected. Figure 6.10 shows the ROC curve performance of the different predictions demonstrating that using a small number of features with the humVar dataset actually improves the performance. Figure 6.11 shows the ROC curve performance of 10 different Models using $m_{try} = 40$ and another 10 models using m_{try} of 4 with a fix number of trees $T = 1000$. The optimized and final results are shown in Table 6.7 with an average ACC of 0.885 and MCC of 0.773. This is rather worse than training and testing with the SAAPdb data, simply because the size of the HumVar dataset that can be mapped to structure is much smaller than the SAAPdb dataset.

Table 6.8 and Figure 6.12 show the effect of dataset size on training and testing using subsets of the SAAPdb data. The same procedure described above was used to avoid structural overlap between training and testing sets during cross-validation. The graph clearly shows that the smaller datasets perform considerably worse. The HumVar training used 3080 samples (1540 PDs and 1540 SNPs) so it is expected that performance will increase substantially at least until the dataset triples in size.

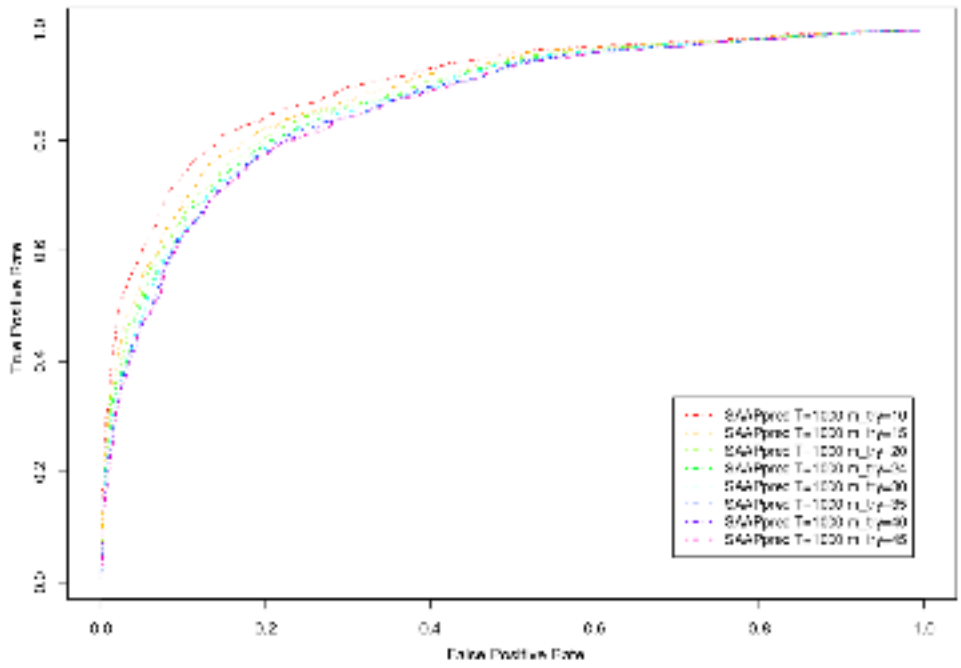


Figure 6.10: ROC curves of SAAPpred trained on HumVar dataset using a 10-fold cross-validation. The SAAPpred based predictor trained using $T = 1000$ and different m_{try}.

Table 6.8: Performance of the machine learning method trained on different sized sets of data from SAAPdb. In each case, a balanced dataset of the required size was extracted at random from the SAAPdb dataset of mutations mapped to protein chains and Random Forests were trained (using m_{try} =40 and T = 1000) and tested using WEKA 10-fold cross-validation, each case was repeated 10 times and the average below.
Performance measures: accuracy (ACC), precision (PREC), specificity (SPEC), sensitivity (SENS), Matthew's correlation coefficient (MCC), F-measure (F); ROC area (ROC);m_{try} stands for the number of randomly chosen attributes in every split; T is the number of trees. The highest and lowest score in every column are coloured blue and red, respectively, all scores are averaged over 10-folds of WEKA cross-validation.

Size of data	TP	FP	TN	FN	ACC	PREC	SPEC	SENS	MCC	F	ROC
100PD:100SNP	79.8	23.2	76.9	20.2	0.783	0.775	0.768	0.798	0.566	0.786	0.868
500PD:500SNP	414.0	62.0	437.7	86.0	0.852	0.870	0.876	0.828	0.705	0.848	0.914
1000PD:1000SNP	842.2	94.8	905.3	157.8	0.874	0.899	0.905	0.842	0.749	0.870	0.931
5000PD:5000SNP	4460.8	252.2	4747.8	539.2	0.921	0.946	0.950	0.892	0.843	0.919	0.968
20000PD:20000SNP	18446.3	541.0	19459.0	1553.7	0.948	0.972	0.973	0.922	0.896	0.946	0.985
30500PD:30500SNP	28443.5	642.3	29858.0	2055.5	0.956	0.978	0.979	0.933	0.913	0.955	0.991

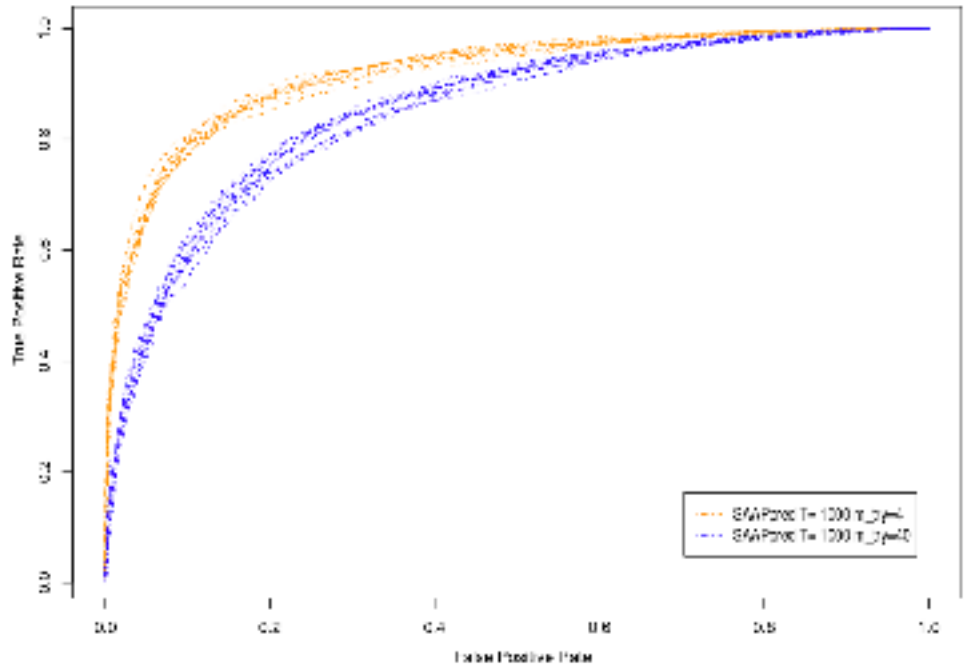


Figure 6.11: ROC curves of SAAPpred trained on HumVar dataset using a 10-fold cross-validation. The SAAPpred based predictor trained using $T = 1000$ and $m_{try} = 4$ and 40.

6.4 Comparison of performance with other predictor methods

The goal of this section is to compare the SAAPpred method developed in this chapter with already existing methods, in a clear and reproducible way. An independent evaluation of methods should be performed using the same HumVar dataset used in Section 6.3.

The results from SAAPpred fully cross-validated (i.e. with no structural overlap between test and training sets) trained and tested on a subset of HumVar mutations that map to structure considerably outperform other well-known individual methods where there may be overlap between testing and training data including SIFT, PolyPhen2, MAPP (Binkley *et al.*, 2010) and MutationAssessor as reported by González-Pérez and López-Bigas (2011) (accuracies between 0.690 and 0.771). Their consensus method (Condel) gives an ACC = 0.882. Our preliminary experiment (using the larger SAAPdb dataset) gives an ACC = 0.956 and a value of ACC = 0.944 and ACC = 0.884 partially and fully cross referenced respectively (using the HumVar dataset) is considerably better. However these results are still not directly comparable with the other methods as those methods are evaluated on the complete HumVar dataset and it may be argued that the subset of mutations for which

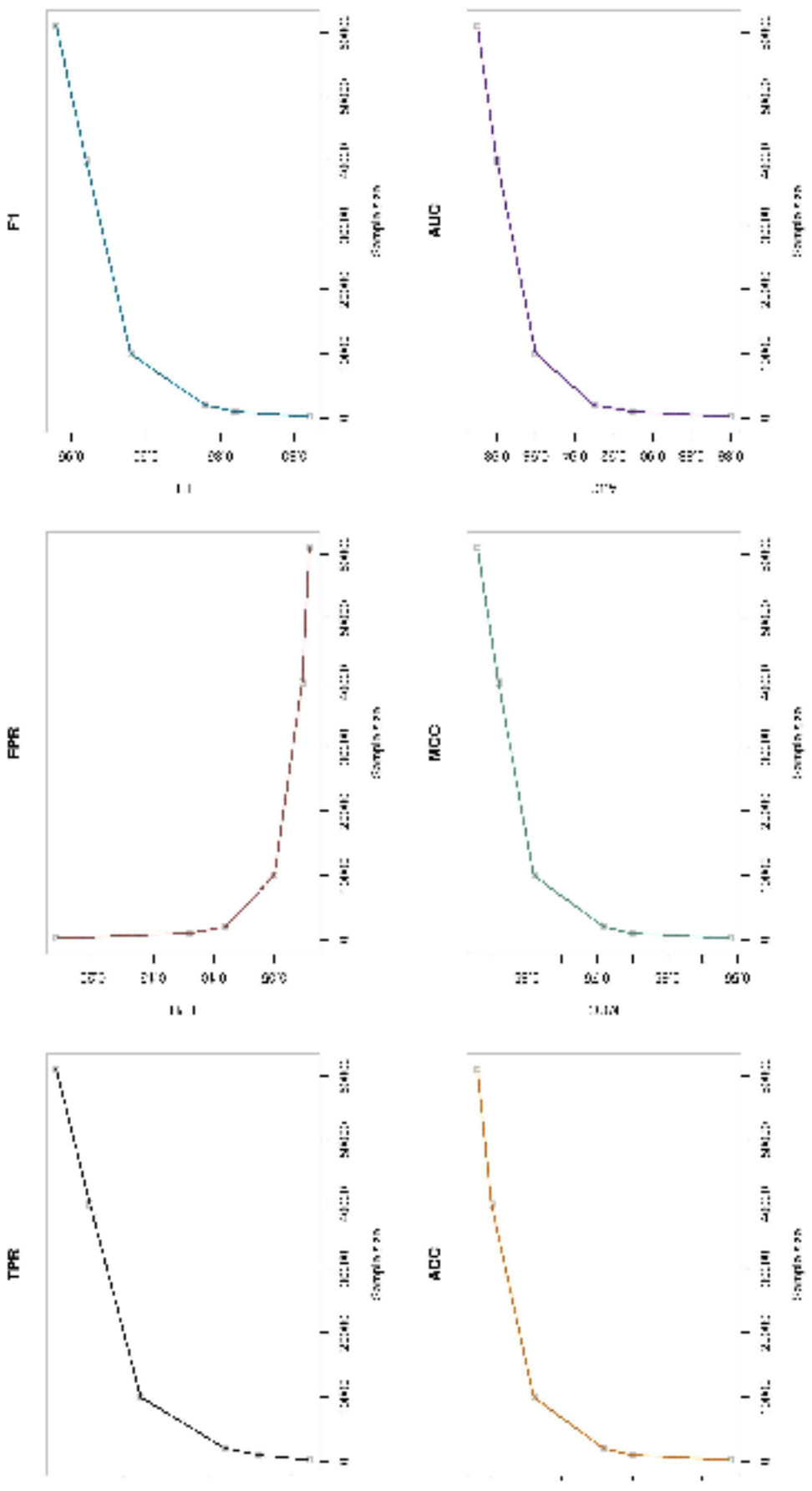


Figure 6.12: Performance of the machine learning method trained on different sized sets of data from SAAPdb. In each case, a balanced dataset of the required size was extracted at random from the SAAPdb dataset of mutations mapped to protein chains (Table 6.8) and Random Forests were trained and tested using 10-fold cross-validation. The graph clearly shows that performance drops as the dataset size decreases, showing a marked drop in performance with datasets below 10,000 samples in size (5,000 SNPs and 5,000 PDs).

structures are available somehow outperform those for which structures are not available in these other methods. For example, PolyPhen2 makes limited use of structural data where these are available, and may be unfairly penalized on the mutations for which structures are not available. Consequently the performance of different pathogenicity predicting methods including PolyPhen2, SIFT, MutationAssessor, Condel and FATHMM was evaluated on the same dataset used for testing SAAPpred.

Balanced datasets (1,451 neutral mutations and ten random selections of 1,451 deleterious mutations) were used. Note that only 1,451 rather than 1,540 mutations could be used since the remaining 89 PDs failed in at least one of the other predictors. In fact this gives PolyPhen2 a significant advantage since it is trained on HumVar leading to an overlap between the training data and our test set. It is not clear precisely what data are used to train SIFT; in their latest paper, Sim *et al.* (2012) state that SIFT was originally trained and tested on LacI, Lysozyme and HIV protease, and refer to the original SIFT papers, but they do not state whether the training has since been modified. MutationAssessor does not appear to use a training set *per se* (Reva *et al.*, 2011).

Partial cross-validated results were performed by using a slightly smaller set of 1451 SNPs that mapped to structure and could be assessed by all the other methods together with a random sample of 1451 PDs that could be assessed by all methods. Again this was repeated 10-times, and the results averaged. The partial-cross-validated values for SAAPpred give the fairest comparison with the public version of PolyPhen2 which is trained on the HumVar dataset.

The results are summarized in Table 6.9 where it can be seen that the results from fully cross-validated (i.e. with no structural overlap between test and training sets) training and testing on HumVar mutations that map to structure considerably outperforms other well-known individual methods where there may be overlap between testing and training data including SIFT, PolyPhen2, MAPP (Binkley *et al.*, 2010) and MutationAssessor as reported by González-Pérez and López-Bigas (2011) (Accuracies between 0.690 and 0.771 – as reported by Condel and 0.676 and 0.785 from our evaluation on HumVar. Their consensus method (Condel) gives an ACC of 0.882) while our preliminary value of ACC of 0.935 (using SAAPdb dataset) and a value of ACC of 0.944 (using HumVar dataset) is considerably better. If we allow overlap in our own set (the fairest comparison) then we outperform PolyPhen2 (the best of the competing methods) by an even larger margin. The partial-cross-validated values for SAAPpred give the fairest comparison with the public version of PolyPhen2 which is trained on the HumVar dataset. Evaluation of The recent predictor

FATHMM (Shihab *et al.*, 2013) on the same dataset shows a performance of ACC = 0.836, MCC = 0.671. While approaching our cross-validated performance, it is likely that some of the HumVar data were included in training FATHMM.

6.5 Conclusions

As previously stated, the main motivation behind this project was to build a pathogenicity predictor using the SAAPdap structural analyses to give us more information about the effects of any novel mutation. SAAPdb was conceived for the understanding of pathogenicity all along, so after redesigning the SAAP pipeline (Chapter 3) and replacing individual Boolean analyses with real values (Chapters 4 and 5), this step was the final phase of a project.

The values for the cross-validated assessment of SAAPpred were obtained from 10-fold cross-validation performed during the WEKA training and used all 1540 SNPs from HumVar that mapped to structure with a random sample of 1540 of the 7182 PDs that mapped to structure. This was repeated 10-times and the results averaged for both 'fully cross validated' experiments where is are no structure overlap between training and testing and 'partially cross validated' where the training and testing sets may contain different mutation at the same potion in the same structure. A comparison with other methods performed by using a slightly smaller set of 1451 (SNPs that mapped to structure and could be assessed by all the other methods) shows SAAPpred results clearly outperform other well-known individual methods including SIFT, PolyPhen2, MAPP, MutationAssessor, Condel and FATHMM giving an accuracy of 0.885 and 0.944 for fully cross validated and partially cross validated (Table 6.9). MutationAssessor-1 was particularly bad as it over-predicts pathogenicity and while the popular SIFT was worst in terms of sensitivity. The 'partial cross validated' values for SAAPpred give the fairest comparison with the public version of PolyPhen2 which is trained on the HumVar dataset.

We learn from the high performance of SAAPpred that the structural information is very important in predicting the pathogenicity of any novel mutation. However predictions based on structural information limit the range of mutations that can be covered by this predictor. There are many reasons in many steps for a failure of structural predictor: missing PDB structures from our database, PDB structure is available but mapping the mutation. This will become less of an issue as protein structures become available for more proteins over time.

Table 6.9: Performance of different prediction methods using a balanced dataset of mutations that map to structure extracted from HumVar. The values for the cross-validated assessment of SAAPpred were obtained from 10-fold cross-validation performed during the WEKA training and used all 1540 SNPs from HumVar that mapped to structure with a random sample of 1540 of the 7182 PDs that mapped to structure. This was repeated 10-times and the results averaged. Partial cross-validated results were performed by using a slightly smaller set of 1451 SNPs that mapped to structure and could be assessed by all the other methods together with a random sample of 1451 PDs that could be assessed by all methods. Again this was repeated 10-times and the results averaged. The partial-cross-validated values for SAAPpred give the fairest comparison with PolyPhen2 which is trained on the HumVar dataset. It is unclear exactly what data were used in training the most recent version of SIFT so there may be some overlap between training and test sets while MutationAssessor has no training set *per se*.
ACC=accuracy, PREC=precision, SPEC=specificity, SENS=sensitivity, MCC=Matthews' correlation coefficient, F=F-measure. The highest and the lowest score in every column are shown in blue and red, respectively.

Method	Cross-validated	ACC	PREC	SPEC	SENS	MCC	F
SAAPpred	Yes	0.885	0.913	0.919	0.852	0.773	0.882
SAAPpred	Partial	0.944	0.999	1.000	0.890	0.894	0.840
PolyPhen2(HumVar-Classifier)	No	0.785	0.759	0.737	0.833	0.572	0.795
PolyPhen2(HumDiv-Classifier)	No	0.762	0.715	0.654	0.869	0.536	0.785
SIFT	?	0.763	0.746	0.730	0.797	0.528	0.771
MutationAssessor-R1.0	N/A	0.676	0.607	0.381	0.977	0.445	0.749
MutationAssessor-R2.0	N/A	0.698	0.633	0.456	0.940	0.453	0.757
FATHMM	?	0.836	0.835	0.836	0.835	0.671	0.836

The ratio of neutral to pathogenic mutations is debatable in two aspects. First, identifying truly negative cases can be questioned as low penetrance pathogenic effect may not have been identified. In other words, they may not be truly neutral, just that we are not aware of any effects based on current annotation. Second, there are more solved PDB structured linked to PDs because protein involved in disease are a natural target for structural studies.

Using a dataset of annotated neutral mutations (such as HumVar) is bound to give a much smaller dataset than the border definitions in SAAPdb and prediction of pathogenicity clearly benefits from large amount of data. So the optimum should be to start with a large set and carefully filter it, until a a sufficiently large experimentally-confirmed SNP dataset becomes available. To make use of all the available mutation information in training the predictor, multiple predictors consisting of 10 predictors, were used followed by a jury vote.

There are multiple ways to achieve further improved prediction performance and results: (i) Incorporating more data in the training process, once they become available; (ii) Investigate the features used in the training and select the most effective ones (to help with the relatively small HumVar dataset size); (iii) Feature combination and construction (e.g. subtracting native void sizes from mutant void sizes); (iv) Feature normalization (e.g. taking the log of some feature values to improve the distribution of values); (v) Using the growing number of structures in the PDB that mapped to the mutations used in training our prediction and (vi) Combining into a meta predictor.

Other approaches include enhancing the predictor by developing methods to make more complete use of unbalanced datasets, especially when there is a smaller dataset in the training stage. SAAPpred only works when there is a PDB structure, starting to combine methodologically-different pathogenicity predictors in meta-predictors such as Condel (González-Pérez and López-Bigas, 2011) (a gatekeeper to dispatch to CONDEL if no structure is expected to improve performance further. The field is currently saturated with predictors of pathogenicity meta predictors are needed. Combining several good predictors will always outperform a single predictor, it is important to choose highest performing predictors, with least overlap in attributes used to predict. Clearly from benchmarking done in this chapter these ones should be combined with SAAPpred, potentially solving the gap SAAPpred has for mutations where no structures are available.

In the next Chapter 7 the predictive power in discriminating between pathogenic and neutral SNPs in MYH7, then create a novel prediction which attempts to distinguish between

HCM and DCM mutation using SAAP analysis, exploring the feature selection, construction, normalisation and an additional set of features on structural clustering.

SAAPpred is now ready to be published on the web for the public to upload any mutation with an available PDB structure to predict its pathogenicity based on SAAP structural and sequence analysis.

Chapter 7

Cardiomyopathy Mutations

Understanding the impact of single nucleotide variations in the beta-myosin heavy chain (MYH7 - UniProt accession code P12883) uncovers new genotype-phenotype relationships in the cardiomyopathy. Unusually, mutations in cardiac beta-myosin heavy chain can lead to two opposite phenotypes: hypertrophic cardiomyopathy (HCM) where the heart wall becomes thicker and dilated cardiomyopathy (DCM) where the heart wall becomes thinner.

In this chapter, the pathogenicity predictor developed in chapter 6 is first applied to predict effects of variants occurring in MYH7, then a novel predictor is created which attempts to distinguish between HCM and DCM mutations using SAAP analysis together with an additional set of features describing structural clustering.

7.1 Introduction

Inherited heart muscle diseases are a major cause of sudden death in the young and an important cause of heart failure at all ages (Hughes and McKenna, 2005). As a set of diseases, they are very heterogeneous in both their genotype and phenotype. For example, over 1000 individual mutations in more than 15 genes have been implicated in hypertrophic cardiomyopathy (HCM) (Seidman and Seidman, 2011) and mutations in 26 genes of 30 chromosomal loci have been identified by OMIM in autosomal dominant dilated cardiomyopathy (DCM) (Ruti Parvari, 2012). Radically different cardiomyopathy phenotypes (e.g. dilated, non-compaction, hypertrophic) have been observed resulting from mutations in the same two sarcomeric genes: beta-myosin heavy chain (MYH7) and troponin T (TNNT2) (Arad *et al.*, 2002).

The beta-myosin heavy chain (P12883) is part of the force-generating molecular motor of the sarcomere (Figure 7.1). It is also the sarcomeric protein for which a larger part of the structure has been solved. The MYH7 gene encodes the beta-myosin heavy chain and is, together with MYBPC3 (the gene encoding myosin binding protein C), the gene where the greatest number of mutations causing HCM have been identified. Finally, and contrary to MYBPC3, the large majority of variants detected in MYH7 are missense, which presents a different challenge for determination of pathogenicity compared with other variants that are expected to cause mRNA and protein truncation (Carrier *et al.*, 1997; Richard *et al.*, 2003).

As in the general case of understanding mutations, an important knowledge gap exists with respect to the relationship between genotype and phenotype. Furthermore, most of the initial phenotypic associations shown for specific genes or individual mutations related to cardiomyopathy have not been replicated in multiple studies. This challenges the effective clinical utilization of genetic data to guide therapy, counselling, sudden death risk assessment, and prognosis. It has recently been hypothesized that such genotype-phenotype variability could be explained by modifying gene-gene interactions (involving common and/or rare variants), gene-environment interactions, or epigenetics (Marian, 2002). Also, different functional consequences may depend on the specific domain/region where the variant is localized. Nevertheless, the hypothesis that the structural impact of a missense variant could influence phenotype, disease severity and outcome has never been directly tested.

While programs such as PolyPhen2 (Adzhubei *et al.*, 2010), SIFT (Sim *et al.*, 2012) and CONDEL (González-Pérez and López-Bigas, 2011) can be used to predict the effect of a missense variant, the performance of the tools is unknown when applied to specific diseases. This is because the programs were not designed for clinical use and have not been validated against phenotype datasets. Moreover, different tools frequently give opposite results that create interpretation challenges. There is one example of a disease- (or protein-) specific tool which is designed specifically for analysing the effect of mutation in voltage-gated potassium channels (Stead *et al.*, 2011). Generally the problem with examining a particular protein is in gathering a large enough data set for training machine learning methods. In addition available data tend to be heavily un-balanced, it being particularly difficult to obtain reliable data on neutral mutations. As described previously, another major limitation of existing prediction software is the fact that methods only make use of limited structural information. SAAPdap and SAAPpred use a combination of rule-based structural measures to assess whether a modification is likely to alter or destroy the function of a protein. The SAAP software has already been used to study structural differences between disease-causing mutations and neutral polymorphisms, and in particular to analyse mutations in glucose-6-phosphate dehydrogenase (Kwok *et al.*, 2002) and in the tumour suppressor P53 (Martin *et al.*, 2002).

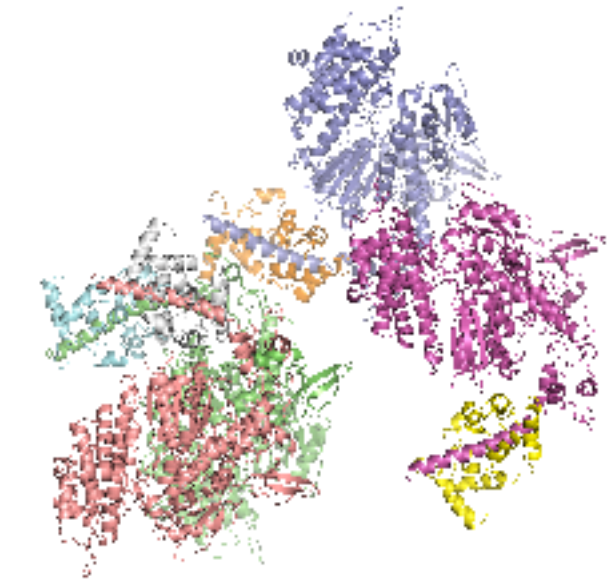
Here the hypothesis that the evaluation of the structural impact of missense variants, using SAAPdap and SAAPpred, will improve the accuracy of predicting pathogenicity compared with the most commonly used *in silico* prediction software (SIFT and PolyPhen) is tested. Further, the possibility of using the same approach to investigate genotype/phenotype relationships at a more detailed level by attempting to distinguish mutations that cause HCM from those that cause DCM is investigated.

7.2 Methods

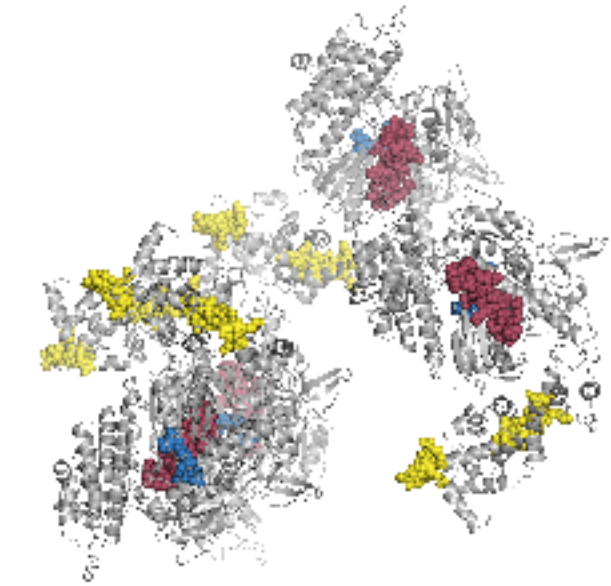
7.2.1 Dataset of variants

A dataset of beta-myosin heavy chain variants detected in a cohort of consecutively evaluated un-related HCM patients was studied and screened[1]. To increase the number of variants analysed, the data were enriched with other established disease-causing or likely pathogenic variants in MYH7, for which phenotypic data are available in HGMD (Stenson *et al.*, 2002).

[1]The data were collected and screened by Prof. Perry Elliott (The UCL Heart Hospital)



(a)



(b)

Figure 7.1: Cardiac human myosin S1dC, beta isoform complexed with Mn-AMPPNP PDB ID 1br4. (a) The structure of 1br4 showing the 8 chains in different colours. (b) actin-binding site (residues 655-677) coloured in ruby; the ATP-binding region (residues 178-185) coloured in blue; myosin light chain binding regions (residues 788-801 and 814-827) coloured in yellow.

7.2.2 Prediction of in silico pathogenicity

Prediction of mutation pathogenicity was performed using Polyphen-2, SIFT, and SAAP-pred as described in Chapter 6.

7.2.3 Manual analysis

The association between each of three phenotype parameters (maximum wall thickness [MLVWT], age at presentation, and HCM vs. other phenotypes) with SAAPdb features were tested using a χ^2 test. These features were a) effect of each of the SAAPdap predicted structural features; b) effect of each of the predicted SAAPdap structural features versus absence of any structural effect; c) number of SAAPdap structural features affected; d) damage prediction by SIFT or Polyphen2 and e) structural domain affected.

7.3 Results and discussion

7.3.1 MYH7 mutation data analysis

MYH7 mutations associated with various cardiomyopathy phenotypes are shown in Table 7.2. A total of 403 mutations were identified in the MYH7 gene. More than two-thirds of them are previously published in the literature as being associated with disease and the others were novel variants. Of the total mutations, 396 were unique, 235 mapped to at least one PDB with a total of 806 mappings to (multiple) PDB structures. Table 7.1 lists five PDB structures from which a model (PDB ID 1ik2) was eliminated at the start. More mutations were associated with HCM (n = 298), whereas all other phenotypes were associated with fewer than 50 mutations each, including DCM with the next highest number of mutations (n = 46). The majority of mutations in both HCM and DCM were unique (292 and 46 respectively). Since mutations related to these phenotypes were the most abundant, further analyses were conducted, looking specifically at HCM and DCM and grouping the remaining phenotypes as 'Others' for most purposes.

The distribution of the variants amongst the structural and functionally-annotated domains of the beta-myosin heavy chain protein were analysed. All of the variants were located in the myosin globular head domain or the neck region. From all variants, 51% were located in functionally annotated domains: 18% mapped to an actin-binding site (residues 655-677); 4% mapped to the ATP-binding region (residues 178-185); 3% were located in the essential and regulatory myosin light chain binding regions (residues 788-801 and 814-827)

(see Figure 7.1b PDB ID 4db1); and 27% mapped to the MYBPC3 binding region (residues 839-964, mapped to different PDB ID 2FXM and 2FXO)).

Table 7.1: PDB structures for UniProt accession code P12883.

PDB ID	Description
1ik2	Model
2fxm	Structure of the human beta-myosin S2 fragment
2fxo	Structure of the human beta-myosin S2 fragment
4db1	Cardiac human myosin S1DC, beta isoform complexed with Mn-AMPPNP
3dtp	Taratula heavy meromysin obtained by flexible docking to tarantula muscle thiCk filament cryo-EM

The expected number of mutations for each residue (E_a) was calculated based on $E_a = N_M \times R_a/N_T$, where E_a is the N_M = total number of mutated residue, R_a = total number of interested amino acids of type a, and N_T = total number of residues in the structure. The $log_2(N_M/E_a)$ was calculated and plotted in Figure 7.2 showing that the predominantly mutated amino acid was arginine (85 variants, and the expected ∼23.53), followed by methionine (20 variants, expected ∼10.84) and Glycine (26 variants, expected∼14.73) (see Table 7.3).

Using SIFT and Polyphen2 prediction software, the 396 unique mutations were analysed (see Table 7.2), of which 69.51% were predicted to be damaging by SIFT, and 90% were predicted to be pathogenic by Polyphen-2. Analysing the same dataset with SAAPdap shows that a total of 175 variants were classified as likely to be damaging by at least one SAAPdap analysis. For 55 variants, no significant structural effect was detected by SAAPdap analysis and 166 failed to be analysed by SAAPdap (i.e. they did not map to a PDB structure) (see Table 7.4). The most frequent features affected were: mutation of a highly conserved residue (impact) occurring in 138 variants; the mutation of an interface amino acid (interface) occurring in 48 of the variants and those disrupting H-bonds occurring in 42 of the variants. Other significant mutations effects occurred less frequently, with mutations causing voids or disrupting disulphide bonds not occurring at all. A significant association was detected between the mutation of a conserved residue detected with the (impact) analysis and the presence of a DCM/LVNC phenotype instead of HCM using a χ^2 test (90% vs. 53%, p = 0.029). In addition, the number of variants with annotated features in UniProt was significantly higher in the presence of a DCM/LVNC phenotype versus an HCM phenotype (20% vs. 0%, p = 0.020). Furthermore, an association was found between the predicted mutation of an (interface) amino acid and the structural domain to which the residue mapped. For example a mutation affecting interface amino acids tended to affect the MYBPC3 binding

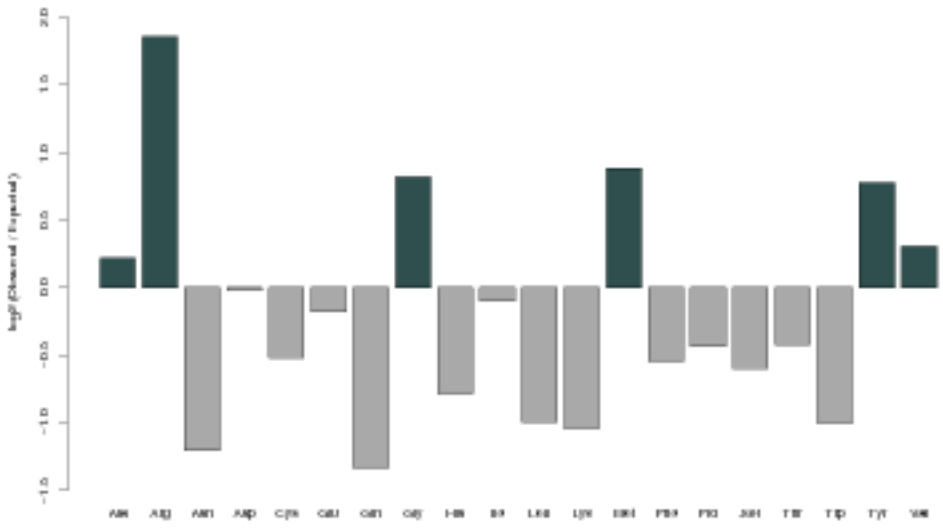
Table 7.2: Numbers of MYH7 mutations dataset per phenotype. Abbreviations: DCM, **D**ilated **C**ardio**M**yopathy; HCM, **H**ypertrophic **C**ardio**M**yopathy; RCM, **R**estrictive **C**ardio**M**yopathy; LVNC, **L**eft **V**entricular **N**on**C**ompaction; ASD, **A**trial **S**eptal **D**efect.

Disease (Phenotype)	Total mutations	Unique mutations	Mutations mapped to PDB
HCM	298	292	188
DCM	46	46	21
RCM	1	1	1
LVNC	17	17	1
LVNC/ASD	1	1	1
DCM/Endocardial Fibroelastosis	1	1	1
DCM/LVNC	3	3	2
HCM/LVNC	1	1	1
HCM/DCM/LVNC	2	2	2
HCM/DCM	3	3	3
HCM/RCM/DCM	2	2	2
HCM plus myopathy central	1	1	1
Myopathy distal de Laing	1	1	1
Ebstein	5	5	1
Myopathy distal	3	3	1
Cardiomyopathy and distal myopathy	2	2	1
Myosin storage myopathy	3	3	1
Hyaline body myopathy	1	1	1
No recorded phenotype	11	11	5
Total	403	396	235

region (56% in the MYBPC3 binding region vs. 0% in the actin-binding domain, p = 0.001), this might be just because the crystal structure only has an interface in that domain or has a large interface. Finally, a tendency was observed for an association between the mutation of an interface amino acid or a binding amino acid and a higher MLVWT (22±4 vs. 19±4mm, p = 0.051 and 25±5 vs. 20±4mm, p = 0.052, respectively) using the *t*-test.

7.3.2 Pathogenicity prediction

Initial pathogenicity prediction was performed using the SAAPpred predictor trained on HumVar[2] as described in Chapter 6. Ten pre-built models were used and the performance results were averaged. Note that TN and FP couldn't be calculated since the dataset did not include any true negatives – all mutations were damaging and consequently, the MCC could not be calculated. Table 7.5 shows the summary of results from the initial predictive model. Initially accuracy for all phenotypes (HCM/DCM/Other) was ~0.970 when using one PDB chain and was reduced to ~0.838 when using all PDB chains. In a later stage, PDB ID 3dtp file was identified as a human/chicken fusion protein and was removed from the dataset. Removing the file improved prediction accuracy to 1.0 for both datasets (mapped to one PDB structure and mapped to multiple PDB structures).



(a)

Figure 7.2: The log_2 of (expected number of mutations for each residue / total number of mutated residue). The dark green bars shows the over-expressed residues while the gray bars shows the under-expressed ones.

[2]Run without -norm option for normalization

Table 7.3: Number of mutated amino acids from MYH7 data compared with the total and expected number of mutations at each amino acid in the protein. Expected values calculated as $E_a = N_M \times R_a/N_T$, where E_a is the N_M = total number of mutated residue, R_a = total number of interested amino acids of type a, and N_T = total number of residues in the structure.

Amino Acid	Mutated	Total number in protein	Expected number of mutations
Ala	40	168	34.38
Arg	85	115	23.53
Asn	8	90	18.41
Asp	21	104	21.28
Cys	2	14	2.86
Glu	46	254	51.98
Gln	10	123	25.17
Gly	26	72	14.73
His	4	34	6.9
Ile	17	89	18.21
Leu	22	214	43.79
Lys	20	201	41.13
Met	20	53	10.84
Phe	8	57	11.66
Pro	5	33	6.75
Ser	13	96	19.64
Thr	13	85	17.39
Trp	1	10	2
Tyr	14	40	8.18
Val	21	83	16.98

Table 7.4: SAAPdap Structural Analysis for MYH7.

SAAPdap Structural Analysis	Number of mutation
Failed (no PDB structure available)	166
No significant structural effect	55
At least one significant structural effect	175
• Hbonds	42
• Buriedcharge	31
• SProtFT	2
• Interface	48
• Clash	14
• Proline	2
• Impact	138
• Binding	20
• Void	0
• SurfacePhobic	15
• Glycine	8
• CisPro	1
• CorePhilic	26
• SSGeom	0

Table 7.5: Summary of SAAPpred performance on MYH7 mutations. The result are an average performance of 10 SAAPpred Model using one PDB per variance and using multiple PBD. The 10 SAAPpred models detailed results can be found in Appendix [D] for all MYH7 mutations, HCM, DCM and Others. Others: LVNC, ASD, Endocardial Fibroelastosis, RCM, Myopathycentral Core, Miopatoa Distal De Laing, Ebsein, Mysin Strong Miopathy, Distal Myopathy. ACC–accuracy, SENS–sensitivity, F–F-measure. Note TN and FP 0 because the data does not include negatives and therefore MCC (Matthew's correlation coefficient) cannot be calculated.

	one PDB			Multi PDB		
	SENS	F1	ACC	SENS	F1	ACC
Run without -norm option for normalization and including PDB ID 3dpt						
HCM	0.966	0.983	0.966	0.836	0.909	0.836
DCM	0.972	0.986	0.972	0.595	0.736	0.595
Other	0.982	0.991	0.982	0.597	0.743	0.597
All	0.971	0.985	0.971	0.839	0.911	0.839
Run with -norm option for normalization and excluding PDB ID 3dpt						
HCM	0.914	0.955	0.914	0.795	0.883	0.795
DCM	0.991	0.995	0.991	0.789	0.878	0.789
Other	0.967	0.983	0.967	0.797	0.884	0.797
All	0.927	0.962	0.927	0.794	0.882	0.794

After this analysis, it was realized that the weighting was incorrect. Scaling of the input parameters on the test sets was not the same as that used in the training and building of the models. The `CVS2ARFF` program used to convert and scale data for WEKA was modified to introduce a `-norm` option to allow the scaling and normalization used to be stored and reused in the test set.

Analyses were performed on data (excluding PDB ID 3dpt, the chicken fusion structure) using the fixed normalization option for each model and the results were averaged. A summary of these analyses is provided in Table 7.5. In this summary table, it can be seen that accuracy for all phenotypes was greater when using one PDB chain versus all PDB chains, it is clear that normalization increased the SAAPpred predictor performance for DCM and other phenotypes when using all PDB chains, but decreased performance with all chains and the HCM phenotype. When using one PDB chain, accuracy was comparable with and without normalization. To compare SAAPpred performance with SIFT and Polyphen2 prediction software, 235 of unique mutations that mapped to at least one PDB structure were analysed (see Table 7.2), of which 92.7% were predicted to be damaging by SAAPpred, 69.51% were predicted to be damaging by SIFT, and 90% were predicted to be pathogenic by Polyphen-2.

7.3.3 A machine learning approach to predict MYH7 phenotype

As a starting point, any mutations associated with multiple phenotypes were discarded. Perl code was written to limit the size of each class by selecting examples at random. For example for HCM, if the HCM class size was limited to 60, the other classes are retained but only 60 mutations were selected at random from the HCM class. Then, WEKA was trained using the Random Forest method. This random selection process was repeated 10 times to provide a representative sample of the HCM class and the results were averaged. Using the same class size limit, DCM phenotypes were also examined and comparisons of phenotypes were made; HCM vs DCM vs 'other' (i.e. pooling CMDM, Ebstein and LVNC).

It was not expected that neural networks (ANNs) as the ML method would work very well. After limiting class size, approximately 150 training examples were obtained. In practice, this was reduced owing to 10-fold cross-validation – i.e. the training holds 10% of the data back and trains on the remaining 90% and then tests on the reserved 10%. It does this 10 times over rotating the test set and averages the results. This means that training actually uses 135 examples. Mutation are represented by a total of 47 'features' from the structural

analysis. Of these, 13 were found to be redundant (i.e. they had the same value for all examples in the dataset), thus reducing the number to 34 features. An ANN consists of layers of 'perceptrons' - an input layer (N_i) in which the observed features are encoded; a 'hidden layer' (N_h); and an output layer (N_o) that encodes the output class. Each perceptron in the input layer is connected to all those in the hidden layer and all in the hidden layer are connected to all in the output layer. So, the number of links is (N_i x N_h) + (N_h x N_o). The ANN learns patterns by adjusting the weights on these links. Consequently, if we have N_i=33, say N_h=10 and N_o=2, then we have 350 weights in the network. A good rule of thumb is that you need 3x the number of training patterns i.e. 1050. In this training set, we only have approximately 12% of the optimal number of examples.

7.3.4 HCM vs. DCM Predictor

The major problem encountered in this analysis was the unbalanced nature of the dataset; many more mutations existed for HCM than for DCM. Because the available dataset was limited in size, it was desirable to use mapping to multiple structure. Thus the same mutation appearing in the training and testing data cannot be avoided during, cross-validation could not be performed by WEKA. For that a Perl program was written to split the 189 HCM and 22 DCM unique mutations with available PDB structures into 10 sets of approximately the same size. Each of these 10 sets in turn was chosen as a test set and enlarged with all the available PDB/chain structures (see Figure 7.3). The remaining 9 sets were used for training by randomly drawing balanced datasets of different sizes from the mutations as mapped to protein chains (see Table 3.2.0.2). This manual cross-validation ensures that there are no cases of the same mutation in the training and test sets but from different PDB chains. Models were built using all DCM and 22 randomly selected HCM mutations, results came from averaging 10 models built using different subsets of HCM and DCM.

7.3.5 Exploring the number of features and number of trees

The initial attempt was performed before the problems with PDB ID 3dtp were identified. Consequently the parameter space around the best result was again explored having removed PDB ID 3dtp. As shown in Table 7.6, the best results were obtained using 1000 trees with 20 features.

Figure 7.3: HYM7 (HCM/DCM) dataset selection for machine learning. A unique mutation level filtering is used, where the same mutation (UniProt:Native:Number:Mutant) does not occur in training and testing sets. This was achieved using a manual cross-validation that splits the dataset into 10 sets, each one in turn was chosen as testing set and the remaining 9 were used for training purpose.

Table 7.6: Exploring the number of features and number of trees
T is the number of trees; m_{trg} stands for the number of randomly chosen attributes in every split. Performance measures: accuracy (ACC) and Matthew's correlation coefficient (MCC). All scores are averaged over 10-folds of WEKA cross-validation.

Number of models	T	m_{trg}	ACC	MCC
10	1000	5	0.7547	0.2734
10	1000	10	0.7655	0.3269
10	1000	15	0.7654	0.3339
10	1000	20	0.7649	0.2721
10	1000	25	0.7549	0.2649
10	1000	30	0.7622	0.2668
No PDB ID 3dtp				
10	1000	10	0.6229	0.2463
10	1000	15	0.6750	0.3590
10	1000	20	0.7000	0.4103
10	1000	25	0.6916	0.3851
10	50	20	0.6833	0.3681
10	100	20	0.6916	0.3872
10	500	20	0.6937	0.4023
10	1000	20	0.7000	0.4103
10	2000	20	0.6812	0.3686
10	5000	20	0.7000	0.4005

Table 7.7: χ^2 tests performed to investigate which features were most informative.
* χ^2 tests were not calculated as the Boolean SAAPdap analysis gave the same result for all mutations analyzed and therefore was not informative.

Feature	χ^2
Binding	4.9
Surface Phobic	0.15
Clash	0.89
Interface	N/A*
Core Philic	0.09
Glycine	11.9
Relative accessibility	N/A*

Feature	χ^2
Buried Charge	0.625
Proline	0.03
Impact (conservation)	19.9
Disulphide	N/A*
CisPro	0.9
Hbonds	N/A*
Voids	0.199

7.3.6 Exploring the most informative features

A simple cut-off for each of the 14 major features was used to suggest whether they were damaging using SAAPdap results. χ^2 tests were performed to investigate which features were most informative (Table 7.7).

The highest χ^2 values were obtained for highly conserved 'impact' mutations, followed by mutations to glycines. These results indicate that mutation of residues affecting these features confers a high probability of a pathogenic phenotype. Mutation of binding residues is also associated with pathogenicity, whereas mutations to proline and introducing hydrophilic residues in the core confers the lowest risk.

7.3.7 Clustering features

MYH7 mutations fall into two distinct regions that map to different PDB files (DCM and HCM mutations). For the more C-terminal structure (PDB ID 2fxm) there are only 2 DCM mutations (compared with 35 HCM), indicating that DCM mutations are rare in this domain. For the N-terminal structure (PDB ID 4db1), there are 16 DCM and 116 HCM mutations. Anecdotal evidence had suggested that HCM and DCM mutations tend to cluster in different areas of the MYH7 N-terminal domain. Consequently the addition of location into the feature vectors was performed as follows.

For the N-terminal domain, the Cα positions of the mutated residues were clustered using single linkage hierarchical clustering. For each of 2...10 clusters a χ^2 test was preformed to see how well the clustering separated HCM from DCM mutations.

- 2 clusters: Significant at the 0.4384 level
- 3 clusters: Significant at the 0.0003755 level
- 4 clusters: Significant at the 0.001256 level
- 5 clusters: Significant at the 0.002577 level
- 6 clusters: Significant at the 0.005057 level
- 7 clusters: Significant at the 0.01013 level
- 8 clusters: Significant at the 0.01778 level
- 9 clusters: Significant at the 0.03044 level
- 10 clusters: Significant at the 0.03116 level

Apart from 2 clusters, these are all clearly significant at the <0.05 level. However, as the number of clusters gets larger one needs to take care with the significance levels, because no more than 20% of expected should be <5 and none <1 (significance will be over-estimated if either of these is true). For ≥ 3 clusters the first of these fails and for 6+ clusters the second also fails. However, between 3 and 6 clusters the significance is so good, that (while it will be over-estimated) it is probably still better than 0.05 and 3 clusters is clearly the most significant result. Consequently we do seem to have clusters of residues that are over/under populated with DCM and HCM mutations compared with what is expected.

Figure 7.4 illustrates the 3 clusters on PDB ID 4db1, colouring the clusters red, green and blue for HCM and orange, yellow and cyan for DCM. Note that the clustering was done on one chain and the results are then shown on the two chains in the 4db1 crystal structure. In particular, DCM is highly over-represented in the third (blue/cyan) cluster. DCM mutations in clusters 1 and 2 (orange and yellow) are hardly visible and therefore mostly buried. On the other hand the DCM mutations in cluster 3 (cyan) are largely on the surface.

To use this information in machine learning, the centroid of each cluster was calculated and the feature vector for each mutation was expanded by the addition of the distances from the C-alpha of the mutated residue to each of the three centroids.

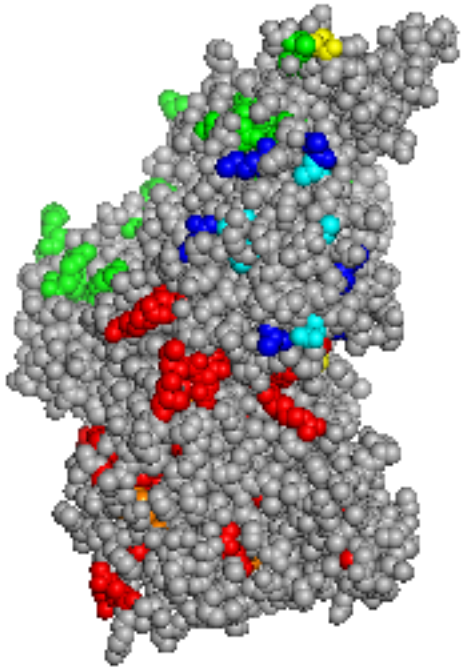


Figure 7.4: Clustering MYHC mutation on PDB ID 4db1 human myosin structure. Colouring the clusters red, green and blue for HCM and orange, yellow and cyan for DCM.

7.3.8 Number of models

Previously, training used 10 models with the prediction results averaged across the 10. Using a larger number of models will allow us to exploit more of the HCM data in each model (while maintaining balanced datasets). Using 20 models, only one unique DCM mutation can be held back from training for test purposes. However, the number of models is not limited to 20 because it is possible to hold one DCM back and then build several models using different sets of HCMs.

After determining the optimum number of features and trees, together with exploration of the most informative feature subsets, different numbers of models were also investigated (5, 10 and 20 models). Addition of the 'clustering' feature described above was also explored. The different sets were defined as follows[3]:

'All'

Refers to the standard set of 33 features of the 47 obtained from SAAPdap. The 13 SwissProt SwissProt features and SSGeom analyses were uninformative: (BCharge, Binding, CPhilic, CisPro, Clash, Glycine, HBonds, Impact, Interface, MLargest1, MLargest10, MLargest2, MLargest3, MLargest4, MLargest5, MLargest6, MLargest7, MLargest8, MLargest9, NLargest1, NLargest10, NLargest2, NLargest3, NLargest4, NLargest5, NLargest6, NLargest7, NLargest8, NLargest9, Proline, Relaccess, SPhobic, Voids).

'Top 5 voids'

Uses only the top 5 native and mutant voids instead of 10 plus the rest of features: BCharge, Binding, CPhilic, CisPro, Clash, Glycine, HBonds, Impact, Interface, MLargest1, MLargest2, MLargest3, MLargest4, MLargest5, NLargest1, NLargest2, NLargest3, NLargest4, NLargest5, Proline, Relaccess, SPhobic, Voids.

'Delta Voids'

Uses the differences between mutant and native voids instead of actual values plus the rest of the features: BCharge, Binding, CPhilic, CisPro, Clash, Glycine, HBonds, Impact, Interface, Largest1, Largest10, Largest2, Largest3, Largest4, Largest5, Largest6, Largest7, Largest8, Largest9, Proline, Relaccess, SPhobic, Voids.

[3] Table 6.2 defined the forty seven features obtained from SAAPdap.

'Set1'

Uses the most informative features based on χ^2 tests: Binding, Relaccess, Impact and Glycine.

'Set2'

A WEKA randomly selected dataset: Binding, Relaccess, SPhobic, Cphil, Voids, MLargest1, NLargest1, Clash, Proline, CisPro.

'Set3'

A WEKA randomly selected dataset: Binding, Interface, Relaccess, Impact, Hbonds, Bcharge, Voids, Largest1, Largest2, Largest3, Largest4, Largest5, Clash, Glycine.

Summary results are presented in Table 7.8 showing that 11 models gave the best performance together with 'Set2' plus clustering features giving an ACC = 0.75 and MCC = 0.531.

Table 7.8: Summary results of machine learning performance using different features of HCM/DCM dataset.

Number of models	Features used	Number of trees	Features per tree	ACC	MCC
5	All	1000	25	0.576	0.152
5	All + Clustering	1000	25	0.648	0.311
5	Top 5 voids + Clustering	1000	25	0.681	0.368
5	10 delta void + Clustering	1000	25	0.608	0.205
11	All	1000	25	0.682	0.429
11	All + Clustering	1000	25	0.608	0.220
11	Top 5 voids + Clustering	1000	25	0.699	0.427
11	10 delta voids + Clustering	1000	25	0.676	0.521
11	Set1 + Clustering	1000	5	0.625	0.314
11	Set2 + Clustering	1000	5	0.750	0.531
11	Set3 + Clustering	1000	5	0.699	0.520
21	All	1000	25	0.631	0.357
21	All + Clustering	1000	25	0.623	0.293
21	Top 5 voids + Clustering	1000	25	0.627	0.374
21	10 delta voids + Clustering	1000	25	0.560	0.133

7.4 Conclusions

The predictive power of the SAAPpred approach was examined in discriminating between pathogenic and neutral SNPs in MYH7, the dataset gives a final prediction results (using all PDB structures) of All MYH7 mutations: ACC = 0.7934, DCM: ACC = 0.789 and HCM: ACC = 0.7951. Some of the incorrect prediction for HCM are mapped to multiple PDB chains, and that affected the prediction result; if the best PDB structure for each mutation was chosen the performance increased to: All MYH7 mutations: ACC = 0.927, DCM: ACC = 0.991 and HCM: ACC = 0.914.

This test was then followed by creation of a novel predictor which attempts to distinguish between HCM and DCM mutations using SAAP analysis. Feature selection, construction, normalisation and an additional set of features based on structural clustering were explored.

In conclusion, the best performance currently achieved for distinguishing HCM and DCM mutations is an ACC = 0.75 and MCC = 0.531. This predictive performance was achieved by averaging 10 models using feature Set2 (Binding, Relaccess, SPhobic, Cphil, Voids, MLargest1, NLargest1, Clash, Proline, CisPro + Clustering) and using 1000 Trees with 5 features. By removing models that perform particularly badly, we can reach ACC = 0.79 and MCC = 0.61. However, the reason for removal of these badly performing models must be justified from a protein structural rather than a prediction perspective. Some of the falsely predicted PDB structures for HCM are mapped to multiple chains, and that affected the prediction result (i.e some structures appear to make the performance worse).

Chapter 8

Conclusions and Discussion

This thesis has described the SAAPdb database, a resource that collates information on single amino acid polymorphisms (SAAPs), SAAPdap, a sequence and structural analysis pipeline to identify the effects of disease mutations by providing hypotheses as to how they might disrupt structure and/or function and SAAPpred, a method for predicting damaging mutations, as well as specialized version of SAAPpred designed to distinguish phenotypes of MYH7 mutations.

The SAAP project is a unique resource. Several other resources collate SAAPs and some calculate SAAP effects. However, (1) there is no other resource that takes a predominantly structural perspective of protein structure perturbation (that is, attempts to assess the mutations in terms of the effects they are likely to have on protein function, stability, folding and interactions); (2) SAAPdap is built to allow easy extension of processing and caches all results; (3) although SAAPdb is no longer updated, it is still a powerful SAAP resource which provides a straightforward but powerful graphical interface to examine SAAPs in protein structures; (4) SAAPdb and SAAPdap takes careful, conservative and sophisticated approach to examine the likely structural effect of SAAPs. These data allowed SAAPpred to be built. The original motivation for the SAAPdb project was the collation of SAAP structural explanations for use in analyzing previously unseen SAAPs. Work presented in this thesis started with rebuilding SAAPdb, improving the structural analysis, introducing SAAPdap and finally training and testing SAAPpred. The work will continue in this direction within the Martin group.

There are numerous other ways in which the resource could be used and the data exploited. These potential applications can be used in protein structure: where experimentalists need-

ing to design a stable mutant protein structure for use in an experiment they will be able to consult SAAPdap and SAAPpred to assess whether a mutation is likely to be damaging; reducing the time taken to devise an appropriate experiment. On a wider and more theoretical scale, the structural integrity of all the structures could aid in the understanding of protein structure in general: currently, the precise mechanisms that are responsible for structural stability are not well understood. These mechanisms could be examined by considering deleterious mutations as 'perturbations' of the usual structural 'system', much in the same way that experimental assays are designed. SAAPdap can describe thousands of disease-associated mutations, both with respect to sequence and structure. These data could be exploited to examine pharmacogenomic variation within populations, specifically, understanding the precise mechanistic reasons for variation in pharmacological response in different populations. In addition, the SAAP analysis data could aid in understanding a disease process. It may also be possible to characterise gain of function and loss of function PDs differently with respect to their analyses, and as such devise different strategies for their treatment.

Chapters 1 and 2 provided details of the structure and information contained within the primary information sources used to build SAAPdb and SAAPdap, and how they have changed since the time of the database build. Examples of how to store, manage, and interpret these data have also been given, with an emphasis on maintaining data integrity and consistency. Different approaches to machine learning were discussed, all with the common aim of knowledge attainment from large datasets that are yet to be fully characterized. A variety of tools for the assessment of mutation-effects were also presented, each using different methodology to predict the structural effect of missense mutations. These were presented to demonstrate the vast array of techniques that can be employed to analyse SNP data and to set the scene for the development of SAAPpred.

8.1 The analysis of disease mutations

Chapter 3 discussed SAAPdb and other support databases (i.e. FOSTA, the database of functionally equivalent proteins from SwissProt, and ImPACT, the sequence conservation scoring method that uses a species similarity matrix), how these were rebuilt and how this was a labour intensive task requiring substantial testing and rewriting of all the code involved in data collection, database creation, and some structural analyses. SNP data were extracted from the XML format dump of dbSNP (Sherry *et al.*, 2001) obtained from the NCBI. Non-synonymous, 'valid' human SNPs (i.e. those annotated with validation

strings 'by frequency', 'by 2hit 2allele', or 'by hapmap'), were extracted and combined into a single XML file. Any mutations not annotated as having disease involvement were assumed to be neutral. PDs were obtained from Online Mendelian Inheritance in Man (OMIM, http://www.ncbi.nlm.nih.gov/omim/) and a number of locus-specific mutation databases (LSMDBs), see Table 3.1. All mutations were then mapped to protein sequences and thence to structure.

The SAAPdb web-server contains fourteen structural analyses and one sequence-based analysis (Martin *et al.* (2002), Cuff and Martin (2004)), shown in Table 3.6, all aiming to show how SAAPs are likely to affect protein structure: in particular interfaces with other proteins, functional sites, folding and stability of the mutated protein. Only mutations mapped to solved protein structures can be assessed, therefore it has not been possible to analyse all known mutations. Of the amino acid mutations in OMIM, approximately 65% were mapped to structure. In addition, approximately 32% of 'valid' SNPs from dbSNP that result in an amino acid change, map to structure. Consequently, the coverage of the analysis is currently somewhat limited, but is expected to improve in the future. After rebuilding SAAPdb, the number of SNPs in the database has rose by 41% and the number of PDs by 36%. However, SAAPdap is now regarded as our primary resource.

SAAPdb was designed to be a regularly updated pre-calculated resource. However, the database has proved very difficult to maintain. Consequently the value of SAAPdb has diminished and it has been replaced with SAAPdap (Single Amino Acid Polymorphism Data Analysis Pipeline). A large and expanding body of literature exists in the field of protein structure-function analysis in relation to disease phenotypes and SAAPdb and SAAPdap contribute to the current understanding of disease-causing mutations and ultimately the treatment of the resulting pathological conditions.

SAAPdap uses a plugin architecture implemented by Andrew Martin, making use of the new non-Boolean analyses (described in Chapters 4 and 5). While SAAPdap still indicates whether a mutation is likely to have a detrimental effect on structure using cut-off values, continuous values are also provided for each of the analyses. Because some of the analyses (especially the analysis of voids) is quite time consuming (taking several minutes), the web interface makes use of AJAX (Asynchronous JavaScript And XML) to update the user with the progress of the analysis. The submission page is available at http://www.bioinf.org.uk/saap/dap/. Results from the SAAPdap pipeline are presented as shown in Figure 3.21.

In conclusion, the data in SAAPdb have been updated, the analyses have been improved (see the following section) and integrated into the new SAAPdap pipeline and web interface.

8.2 Improving and extending the pipeline

In the original SAAPdb all assignments of structural effects are Boolean; that is, any mutation either does, or does not, have a given effect. While Boolean assignment is appropriate in some cases (for example, a residue either is, or is not, annotated as a feature in UniProtKB/Swiss-Prot), in other cases, it relies on a critical cut-off value (for example, energy, void volume, hydrophobicity difference) as described previously (Hurst *et al.*, 2009; Cuff *et al.*, 2006; Cuff and Martin, 2004; Martin *et al.*, 2002). In this thesis it was found that assigning a mutation as (not) having a structural effect is very sensitive to precise structural detail. Wherever appropriate, real-number scores or pseudo-energies for each effect have now been implemented. In particular, the analysis of clashes and torsion angles has been enhanced to provide energy values.

In analyzing clashes, previous work defined a damaging clash as any side-chain that has at least 3 van der Waals overlaps (of any degree) with other atoms. Similarly, voids were considered damaging when they caused the creation of voids of volume $> 275\text{Å}^3$, assuming no compensatory movement within the protein structure. By looking at the distribution of SNPs and PDs predicted to be damaging, it was clear that the Boolean method did not accurately describe the effect of mutations causing clashes or voids, either overestimating or underestimating damaging effects when values were close to the cut-off.

The new clash analyses use a continuous energy scale calculation incorporating Lennard-Jones and torsion energies using CHARMM (Brooks *et al.*, 1983) parameters. The actual energy value is used in the machine learning described in Chapter 6. The MutModel program is used in both clash and void analysis and parameters (step-size and tolerance) used in searching side-chain positions were optimised by modelling known mutant structures. Consequently, the evaluation of both clash and void is optimised by using these parameters. No other changes were made to the assessment of voids; the cut-off selected previously is used as a visual indication that a void is likely to be damaging, but as with clash energy actual void sizes are used in the machine learning described in Chapter 6.

Glycine and proline analyses have been improved by moving from simple Boolean decision making with rather arbitrary boundaries to an energy-evaluation approach. Figure 5.13 clearly shows that the previous allowed regions were inaccurate and, in particular, the allowed regions for proline were incorrect. These analyses have been integrated into the new SAAPdap pipeline and web interface (Chapter 3). Detailed results of these analyses were then used to build the pathogenicity predictor SAAPpred (Chapter 6).

There are many potential structural effects of SAAPs that are currently not assessed by SAAPdap, as with the analysis of the kinase domain where oncogenic mutations are known not only to destabilise the inactive form of B-RAF, but also to 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 enhanced analysis of the likely effect of mutations and consequently improve predictions further.

It may also be beneficial to consider the protein in a wider context, for example its rôle in known pathways (Kanehisa *et al.*, 2008). Consideration of genomic data is another area to explore. The focus of SAAPdap 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 Chapter 7 (MYH7 protein) (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, but are being investigated by another member of the group.

8.3 Moving onto prediction

As previously stated, the main motivation behind this project was to build the pathogenicity predictor using the SAAPdap structural analyses to give us more information about the pathogenicity of any novel mutation. SAAPdb was initially conceived for the understanding of pathogenicity, so after redesigning the SAAP pipeline (Chapter 3) and improving individual binary analyses to ranged values (Chapters 4 and 5), this step was the final phase of the project.

SAAPdb data with SAAPdap analysis were used to train machine learning methods to predict whether a novel SAAP will disrupt the native protein structure and induce a disease phenotype in a tool known as Single Amino Acid Polymorphism prediction (SAAPpred) (see Chapter 6). The SAAPdb database of SNPs and PDs was later replaced with the HumVar dataset for comparison with other methods. SAAPdb and SAAPdap perform fourteen analyses, from these analyses (using software written in Perl and C); 47 features are derived that are used for machine learning.

Application of machine learning techniques exploit the predictive power of all of these individual features, resulting in a very sensitive and accurate method for classifying previously unseen mutations as disease-causing or neutral. The prediction results are summarised in Table 6.9. A comparison with other methods, performed by using a slightly smaller set of 1451 SNPs that mapped to structure and could be assessed by all the other methods, shows SAAPpred results clearly outperform other well-known individual methods including SIFT, PolyPhen2, MAPP, MutationAssessor, CONDEL and FATHMM giving an accuracy of 0.885 and 0.944 for 'fully-cross-validated' and 'partially-cross-validated' (Table 6.9), respectively. The performance of MutationAssessor-1 was particularly bad as it over-predicts pathogenicity and very popular SIFT was worst in terms of sensitivity. The fully-cross-validated reflects performance on a novel mutation/protein for which no training have been done, partial-cross-validated is still fully cross validated in the conventional sense (see section 6.3.2. Performance on PolyPhen2 is not cross validated at all with a full overlap between training and testing set trained on the HumVar dataset), for that partial-cross-validated values for SAAPpred give the fairest comparison with PolyPhen2.

We learn from the performance of SAAPpred, that structural information is very important in predicting the pathogenicity of any novel mutation. However predictions based on structural information limit the range of mutations that can be covered by this predictor. There

are many reasons in many steps for a failure of a structural predictor: for example missing PDB structures; PDB structure is available but the mutated residue is missing; additions such as missing atoms, etc. This will become less of an issue as protein structures become available for more proteins over time.

The ratio of neutral to pathogenic mutations is debatable in two aspects. First, identifying truly negative cases is either questionable when we define them as mutations that merely do not have a known pathogenic effect at the moment (the current definition in SAAPdb) as this does not mean they are truly neutral, just that we are not aware of any effects based on current annotation. Also, there are more solved PDB structures linked to PDs. But a dataset of annotated neutral mutations (such as HumVar) is bound to be much smaller; pathogenicity prediction clearly benefits from large amounts of data. So the optimum should be to start with a large set and carefully filter it, until sufficient experimentally-confirmed SNP data become available. To make use of all the available mutation information in training the predictor, multiple predictors were used and a jury vote was taken.

8.4 Implications for disease therapies

There is much potential for SAAPdap and SAAPpred 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. 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 and SAAPdap successfully identifies this mutation as void-creating. Boeckler et al., used *in silico* screening and multiple NMR spectroscopy experiments, and 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.

Alternatively, Friedler et al. (2002) have shown that alternative pharmaceuticals 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.

In Chapter 7 the predictive power of the SAAPpred approach was examined in discriminating between pathogenic and neutral SNPs in MYH7. This gave, for all MYH7 mutations an accuracy of 0.7934, for DCM an accuracy of 0.789 and for HCM an accuracy of 0.7951.

This test was then followed by creation a novel predictor which attempts to distinguish between HCM and DCM mutation using SAAP analysis, exploring the feature selection, construction, normalisation and an additional set of features on structural clustering.

In conclusion, the best performance we can currently achieve for distinguishing HCM and DCM mutations is an accuracy of 0.75 and MCC = 0.531. This predictive performance was achieved by averaging 10 models using feature Set2 (Binding, Relaccess, SPhobic, Cphil, Voids, MLargest1, NLargest1, Clash, Proline, CisPro) + Clustering and using 1000 Trees with 5 features. By removing models that perform particularly badly, we can reach ACC = 0.79 and MCC = 0.61. However, the reason for removal of these badly performing models must be justified from a protein structural rather than a prediction perspective.

8.5 Future prospects

The Martin group has plans to improve and expand SAAPdap including analysis of mutations in non-coding regions. These features will be used to improve the machine learning training.

There are multiple ways to achieve further improved prediction performance and results: (i) Incorporating more data in the training process, once they become available; (ii) Investigate the features used in the training and select the most effective ones (to help with the relatively small HumVar dataset size); (iii) Feature combination and constructions (e.g. subtracting native void sizes from mutant void sizes); (iv) Feature normalization (e.g. taking the log of some feature values to improve the distribution of values); (vi) Using the growing number of structures in the PDB that mapped to the mutations used in training our prediction; (v) Combining SAAPpred into a meta predictor (CONDEL-style) with other methods such as PolyPhen2 and SIFT; (vi) SAAPpred only works when there is a PDB structure, starting to combine methodologically-different pathogenicity predictor by using a gatekeeper to see if there is an available structure for a particular mutation and using CONDEL-style meta-predictor (González-Pérez and López-Bigas, 2011) which employs SAAPpred as one of its elements, or if there is no available structure, using normal CONDEL for sequence based prediction and (vii) Enhancing the predictor by developing methods to make more

complete use of unbalanced datasets, especially with a smaller dataset in the training stage when applied to specific problems such as phenotype prediction.

The field is currently saturated with predictors of pathogenicity, more meta-predictors are needed. Combining several good predictors will always outperform a single predictor, it is important to choose the highest performing predictors, with least overlap in attributes used to predict. Clearly from benchmarking done in Chapter 6 these ones should be combined with SAAPpred, potentially solving the gap SAAPpred has for mutations where no structures are available.

While the coverage of the method is currently somewhat limited by the need for a structure of the protein, investigation of the use of modelled structures is also planned. However, currently it is not known how well this will work given the detailed structural analysis (e.g. of hydrogen bonds) that the method performs. It is proposed that different predictors would be trained for different sets of models having different ranges of sequences identity with the templates used in modelling (i.e. $< 30\%$, 30–50%, 50–70% and $> 70\%$) – a gatekeeper would then select the appropriate predictor. However clinically relevant proteins tend to be key targets for structural studies, and as more structures become available, the number of mutants mapped to structure will increase, improving the coverage of the method. In addition, more structural data will allow the machine learning methods to be trained and tested with more data. Consequently, we expect performance to increase further.

SAAPpred is now available to be published on the web for the public to upload any mutation with an available PDB structure to predict its pathogenicity based on SAAP structural and sequence analysis.

8.6 Summary

In summary, this thesis has improved the analysis of the likely structural effects of mutation and has used these analysis, present in SAAPdap pipeline, to train a prediction able to distinguish between pathogenic and neutral mutations. SAAPpred, clearly outperform all other individual predictors and when assessed by partial cross validation (still full cross validation by other terminology) outperform CONDEL (Accuracy is 0.944 for SAAPpred compared with 0.882 for CONDEL). A method for distinguishing between phenotypes resulting from MYH7 mutation has also been developed. while the performance is low compared with the general pathogenicity prediction, it outperforms older method which simply predict pathogenicity (such as MutationAssessor).

Bibliography

Abel O, Powell JF, Andersen PM, Al-Chalabi A. 2012. Alsod: A user-friendly online bioinformatics tool for amyotrophic lateral sclerosis genetics. Human Mutation 33.9:1345–1351.

Abkevich VI, Shakhnovich EI. 2000. What can disulfide bonds tell us about protein energetics, function and folding: Simulations and bioninformatics analysis. Journal of Molecular Biology 300:975–985.

Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR. April 2010. A method and server for predicting damaging missense mutations. Nature methods 7.4:248–249.

Aguirre T, Matthijs G, Robberecht W, Tilkin P, Cassiman JJ. 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.

Al-Numair NSA, 2010. Analyzing the effects of protein mutation in diseases. Master's thesis, UNIVERSITY COLLEGE LONDON, UCL Research Department of Structural and Molecular Biology.

Alber T, Dao-pin S, Wilson K, Wozniak JA, Cook SP, Matthews BW. 1987. Contributions of hydrogen bonds of thr 157 to the thermodynamic stability of phage t4 lysozyme. Nature 330:41–46.

Alberts B, 2008. *Molecular biology of the cell*, volume Volume 1. Garland Pub., 1989, 2, illustrated edition.

Alpaydin E, 2009. *Introduction to machine learning*. The MIT Press, 2nd edition.

Altschul SF, Madden TL, SchÃd'ffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. 1997. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Research 25.17:3389–3402.

Amberger J, Bocchini C, Hamosh A. 2011. A new face and new challenges for Online Mendelian Inheritance in Man (OMIM). Human Mutation 32.5:564–567.

Amberger J, Bocchini CA, Scott AF, Hamosh A. 2009. McKusick's Online Mendelian Inheritance in Man (OMIM)). Nucleic Acids Research 37.

Andreeva A, Howorth D, Chandonia JM, Brenner SE, Hubbard TJP, Chothia C, Murzin AG. January 2008. Data growth and its impact on the SCOP database: new developments. Nucleic Acids Research 36.suppl 1:D419–D425.

Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LL. 2004. Uniprot: the universal protein knowledgebase. Nucleic Acids Research 32.suppl 1:D115–D119.

Arad M, Seidman J, Seidman CE. 2002. Phenotypic diversity in hypertrophic cardiomyopathy. Human Molecular Genetics 11.20:2499–2506.

Bader GD, Donaldson I, Wolting C, Ouellette BF, Pawson T, Hogue CW. 2001. BIND–The biomolecular interaction network database. Nucleic Acids Research 29:242–245.

Baird M, Driscoll C, Schreiner H, Sciarratta GV, Sansone G, Niazi G, Ramirez F, Bank A. 1981. A nucleotide change at a splice junction in the human beta-globin gene is associated with beta 0-thalassemia. Proceedings of the National Academy of Sciences of the United States of America 78.7:4218–4221.

Baker EN, Hubbard RE. 1984. Hydrogen bonding in globular proteins. Prog Biophys Mol Biol 44:97–179.

Bao L, Zhou M, Cui Y. 2005. nsSNPAnalyzer: Identifying disease-associated nonsynonymous single nucleotide polymorphisms. Nucleic Acids Research 33:W480–W482.

Baresic A, Hopcroft LE, Rogers HH, Hurst JM, Martin AC. 2010. Compensated pathogenic deviations: Analysis of structural effects. Journal of Molecular Biology 396.1:19 – 30.

Bashton M, Nobeli I, Thornton JM. 2008. PROCOGNATE: a cognate ligand domain mapping for enzymes. Nucleic Acids Research 36:D618–D622.

Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. January 2011. GenBank. Nucleic Acids Research 39.suppl 1:D32–D37.

Berg JM, Tymoczko JL, Stryer L, 2006. *Biochemistry*. W. H. Freeman, 6th edition.

Berkeley-Lab , 2002. A rare protein mutation offers new hope for heart disease patients. http://www.lbl.gov/Science-Articles/Archive/LSD-Milano-Bielicki.html.

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

Beutler E, Mathai CK, Smith JE. 1968. Biochemical variants of glucose-6-phosphate dehydrogenase giving rise to congenital nonspherocytic hemolytic disease. Blood 31:131–150.

Binkley J, Karra K, Kirby A, Hosobuchi M, Stone EA, Sidow A. 2010. ProPhylER: a curated online resource for protein function and structure based on evolutionary constraint analyses. Genome Research 20:142–154.

Boeckler FM, Joerger AC, Jaggi G, Rutherford TJ, Veprintsev DB, Fersht AR. 2008. Targeted rescue of a destabilized mutant of p53 by an in silico screened drug. Proceedings of the National Academy of Sciences U.S.A 105:10360–10365.

Bowie J, Reidhaar-Olson J, Lim W, Sauer R. 1990. Deciphering the message in protein sequences: tolerance to amino acid substitutions. Science 247.4948:1306–1310.

Branden C, Tooze J, 1999. *Introduction to Protein Structure*. Garland, 2nd edition edition.

Brctic Kostic k. 2005. Neutral mutation as the source of genetic variation in life history traits. Genetics Research 86.01:53–63.

Breiman L. 1996. Bagging predictors. Machine learning 24:123–140.

Breiman L. 2001. Random forests. Machine learning 5:5–32.

Brenner S, Miller J, 2001. *Encyclopedia of genetics*, volume 1. Academic Press.

Bromberg Y, Rost B. 2007. Snap: predict effect of non-synonymous polymorphisms on function. Nucleic Acids Research 35.11:3823–3835.

Brooks BR, Brooks CL, Mackerell AD, Nilsson L, Petrella RJ, Roux B, Won Y, Archontis G, Bartels C, Boresch S, Caflisch A, Caves L, Cui Q, Dinner AR, Feig M, Fischer S, Gao J, Hodoscek M, Im W, Kuczera K, Lazaridis T, Ma J, Ovchinnikov V, Paci E, Pastor RW, Post CB, Pu JZ, Schaefer M, Tidor B, Venable RM, Woodcock HL, Wu X, Yang W, York DM, Karplus M. 2009. Charmm: The biomolecular simulation program. Journal of Computational Chem-

istry 30.10:1545–1614.

Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M. 1983. Charmm: A program for macromolecular energy, minimization, and dynamics calculations. Journal of Computational Chemistry 4:187–217.

Calabrese R, Capriotti E, Fariselli P, Martelli PL, Casadio R. 2009. Functional annotations improve the predictive score of human disease-related mutations in proteins. Human Mutation 30.8:1237–1244.

Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden T. 2009. Blast+: architecture and applications. BMC Bioinformatics 10.1:421.

Canet-Aviles RM, Wilson MA, Miller DW, Ahmad R, McLendon C, Bandyopadhyay S, Baptista MJ, Ringe D, Petsko GA, Cookson MR. 2004. The parkinson's disease protein dj-1 is neuroprotective due to cysteine-sulfinic acid-driven mitochondrial localization. Proceedings of the National Academy of Sciences of the United States of America 101.24:9103–9108.

Capriotti E, Calabrese R, Casadio R. 2006. Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information. Bioinformatics 22.22:2729–2734.

Care MA, Needham CJ, Bulpitt AJ, Westhead DR. 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 CR, Lim EP, Kalyanaraman N, Nemesh J, Ziaugra L, Friedland L, Rolfe A, Warrington J, Lipshutz R, Daley GQ, Lander ES. 1999. Characterization of single-nucleotide polymorphisms in coding regions of human genes. Nature Genetics 22:231–238.

Carrier L, Bonne G, Bahrend E, Yu B, Richard P, Niel F, Hainque B, Cruaud C, Gary F, Labeit S, Bouhour JB, Dubourg O, Desnos M, Hagege AA, Trent RJ, Komajda M, Fiszman M, Schwartz K. 1997. Organization and sequence of human cardiac myosin binding protein c gene (mybpc3) and identification of mutations predicted to produce truncated proteins in familial hypertrophic cardiomyopathy. Circulation Research 80.3:427–434.

Caspi R, Altman T, Dale JM, Dreher K, Fulcher CA, Gilham F, Kaipa P, Karthikeyan AS, Kothari A, Krummenacker M, Latendresse M, Mueller LA, Paley S, Popescu L, Pujar A, Shearer AG, Zhang P, Karp PD. January 2010. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic acids research 38.Database issue:D473–D479.

Castellana S, Mazza T. 2013. Congruency in the prediction of pathogenic missense mutations: state-of-the-art web-based tools. Briefings in Bioinformatics 4.

Cavallo A, Martin ACR. 2005. Mapping SNPs to protein sequence and structure data. Bioinformatics 21:1443–1450.

Chaganti S, Ma CS, Bell AI, Croom-Carter D, Hislop AD, Tangye SG, Rickinson AB. 2008. Epstein-Barr virus persistence in the absence of conventional memory B cells: IgM+IgD+CD27+ B cells harbor the virus in X-linked lymphoproliferative disease patients. Blood 112:672–679.

Chen J, Stites W. 2001. Packing is a key selection factor in the evolution of protein hydrophobic cores. Biochemistry 40:15280–15289.

Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC, Zhang Q, Locke DP, Shi X, Fulton RS, Ley TJ, Wilson RK, Ding L, Mardis ER. September 2009. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. Nature methods 6.9:677–681.

Cheng TMK, Lu YE, Vendruscolo M, Lio' P, Blundell TL. 07 2008. Prediction by graph theoretic measures of structural effects in proteins arising from non-synonymous single nucleotide polymorphisms. PLoS Comput Biol 4.7:e1000135.

Cho Y, Gorina S, Jeffrey P, Pavletich N. 1994. Crystal structure of a p53 tumor suppressor-dna complex: understanding tumorigenic mutations. Science 265.5170:346–355.

Claustres M, Horaitis O, Vanevski M, Cotton RGH. 2002. Time for a unified system of mutation description and reporting: a review of locus-specific mutation databases. Genome Research 12:680–688.

Cleland JL, Craik C, 1996. *Protein engineering: principles and practice*. Wiley-Liss.

Clifford RJ, Edmonson MN, Nguyen C, Buetow KH. 2004. Large-scale analysis of non-synonymous coding region single nucleotide polymorphisms. Bioinformatics 20:1006–1014.

Cline E, 2009. Snpwatch: Genetic variants near tumor suppressor genes may increase risk for brain and skin cancer.

Codd EF. June 1970. A relational model of data for large shared data banks. Communications of the ACM 13:377–387.

Collins FS, Brooks LD, Chakravarti A. 1998. A DNA polymorphism discovery resource for research on human genetic variation. Genome Research 8:1229–1231.

Consortium TGO. January 2010. The Gene Ontology in 2010: extensions and refinements. Nucleic Acids Research 38.suppl 1:D331–D335.

Consortium TIH. 2005. A haplotype map of the human genome. Nature 437.

Consortium TU. January 2011. Ongoing and future developments at the Universal Protein Resource. Nucleic Acids Research 39.suppl 1:D214–D219.

Corsaro A, Thellung S, Bucciarelli T, Scotti L, Chiovitti K, Villa V, D'Arrigo C, Aceto A, Florio T. 2011. High hydrophobic amino acid exposure is responsible of the neurotoxic effects induced by E200K or D202N disease-related mutations of the human prion protein. The International Journal of Biochemistry Cell Biology 43:372–382.

Cuff AL, Janes RW, Martin ACR. June 2006. Analysing the ability to retain sidechain hydrogen-bonds in mutant proteins. Bioinformatics 22.12:1464–1470.

Cuff AL, Martin ACR. 2004. Analysis of void volumes in proteins and application to stability of the p53 tumour suppressor protein. Journal of Molecular Biology 344:1199–1209.

Cuff AL, Sillitoe I, Lewis T, Clegg AB, Rentzsch R, Furnham N, Pellegrini-Calace M, Jones D, Thornton J, Orengo CA. January 2011. Extending CATH: increasing coverage of the protein structure universe and linking structure with function. Nucleic Acids Research 39.suppl 1:D420–D426.

David AH. 2005. Whole-genome patterns of common dna variation in three human populations. Science 307:1072–1079.

Dixit A, Yi L, Gowthaman R, Torkamani A, Schork NJ, Verkhivker GM. 10 2009. Sequence and structure signatures of cancer mutation hotspots in protein kinases. PLoS ONE 4.10:e7485.

Durbin R, 1998. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, illustrated, reprint edition.

Durbin RM, Abecasis GR, Altshuler DL, Auton A, Brooks LD, Durbin RM, Gibbs RA, Hurles ME, McVean GA. 2010. A map of human genome variation from population-scale sequencing. Nature 467.7319:1061–1073.

Edgar RC. 2004. MUSCLE: a multiple sequence alignment method with reduced time and

space complexity. BMC Bioinformatics 5:113–113.

Eriksson A, Baase W, Zhang X, Heinz D, Blaber M, Baldwin E, Matthews B. 1992. Response of a protein structure to cavity-creating mutations and its relation to the hydrophobic effect. Science 255:178–183.

Eswar N, Ramakrishnan C. 2000. Deterministic features of side-chain main-chain hydrogen bonds in globular protein structures. Protein Engineering 13:227–238.

Feng Z, Chen L, Maddula H, Akcan O, Berman HM, Westbrook J, July 2003. *ACA Program and Abstract Book*, volume 30 of 2. Northern Kentucky Convention Center. ISSN 0569-4221.

Ferrer-Costa C, Orozco M, de la Cruz X. 2005. Use of bioinformatics tools for the annotation of disease-associated mutations in animal models. Proteins 61:878–887.

Feyfant E, Sali A, Fiser A. 2007. Modeling mutations in protein structures. Protein Scince 16:2030–2041.

Finn RD, Mistry J, Tate J, Coggill P, Heger A, Pollington JE, Gavin OL, Gunasekaran P, Ceric G, Forslund K, Holm L, Sonnhammer ELL, Eddy SR, Bateman A. January 2010. The Pfam protein families database. Nucleic Acids Research 38.suppl 1:D211–D222.

Fisher RA. 1935. The logic of inductive inference. Journal of the Royal Statistical Society Series A 98:39–54.

Flicek P, Amode MR, Barrell D, Beal K, Brent S, Chen Y, Clapham P, Coates G, Fairley S, Fitzgerald S, Gordon L, Hendrix M, Hourlier T, Johnson N, Kähäri A, Keefe D, Keenan S, Kinsella R, Kokocinski F, Kulesha E, Larsson P, Longden I, McLaren W, Overduin B, Pritchard B, Riat HSS, Rios D, Ritchie GR, Ruffier M, Schuster M, Sobral D, Spudich G, Tang YA, Trevanion S, Vandrovcova J, Vilella AJ, White S, Wilder SP, Zadissa A, Zamora J, Aken BL, Birney E, Cunningham F, Dunham I, Durbin R, Fernández-Suarez XM, Herrero J, Hubbard TJ, Parker A, Proctor G, Vogel J, Searle SM. January 2011. Ensembl 2011. Nucleic acids research 39.Database issue.

Forbes SA, Bindal N, Bamford S, Cole C, Kok CY, Beare D, Jia M, Shepherd R, Leung K, Menzies A, Teague JW, Campbell PJ, Stratton MR, Futreal PA. 2011. Cosmic: mining complete cancer genomes in the catalogue of somatic mutations in cancer. Nucleic Acids Research 39.suppl 1:D945–D950.

Frank DA. 2007. Stat3 as a central mediator of neoplastic cellular transformation. Cancer letters 251.2.

Frasconi P, Shamir R, Division NATOSA, 2003. *Artificial Intelligence and Heuristic Methods in Bioinformatics*. NATO science series: Computer and systems sciences. IOS Press.

Frazer KA, Ballinger DG, Cox DR, Hinds DA, Stuve LL, Gibbs RA, Belmont JW, Boudreau A, Hardenbol P, Leal SM, et al. . 2007. A second generation human haplotype map of over 3.1 million snps. Nature 449.7164:851–861.

Friedler A, Hansson LO, Veprintsev DB, Freund SMV, Rippin TM, Nikolova PV, Proctor MR, Rüdiger S, Fersht AR. 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 USA 99:937–942.

Gabanyi M, Adams P, Arnold K, Bordoli L, Carter L, Flippen-Andersen J, Gifford L, Haas J, Kouranov A, McLaughlin W, Micallef D, Minor W, Shah R, Schwede T, Tao Y, Westbrook J, Zimmerman M, Berman H. 2011. The structural biology knowledgebase: a portal to protein structures, sequences, functions, and methods. Journal of Structural and Functional Genomics 2.

George RA, Smith TD, Callaghan S, Hardman L, Pierides C, Horaitis O, Wouters MA, Cot-

ton RGH. 2008. General mutation databases: Analysis and review. Journal of Medical Genetics 45:65–70.

Gilbert-Dussardier B, Segues B, Rozet JM, Rabier D, Calvas P, de Lumley L, Bonnefond JP, 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.

Gilissen C, Hoischen A, Brunner HG, Veltman JA. 2012. Disease gene identification strategies for exome sequencing. European Journal of Human Genetics 20:490–497.

González-Pérez A, López-Bigas N. April 2011. Improving the assessment of the outcome of nonsynonymous SNVs with a consensus deleteriousness score, Condel. American journal of human genetics 88.4:440–449.

Gorham RD, Kieslich CA, Morikis D. 2011. Electrostatic clustering and free energy calculations provide a foundation for protein design and optimization. Annals of Biomedical Engineering 39:1252–1263.

Greenblatt MS, Bennett WP, Hollstein M, Harris CC. 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 PA, Stratton MR, Easton DF. 2006. Statistical analysis of pathogenicity of somatic mutations in cancer. Genetics 173:2187–2198.

Gregory RW, Ben B, Lodewijk B, Robert G, Wolfgang, Huber Andy L, Thomas L, Martin M, Arni M, Steffen M, Marc S, Bill V, 2010. gplots: Various r programming tools for plotting data. R package version 2.8.0.

Hall M, Franke E, Holmes G, Pfahringer B, Reutemann P, H WI. 2009. The WEKA data mining software: An update. SIGKDD Explorations 11:10–18.

Hand DJ. 2009. Measuring classifier performance: A coherent alternative to the area under the ROC curve. Machine Learning 77:103–123.

Hazes B, Dijkstra BW. 1988. Model building of disulfide bonds in proteins with known three-dimensional structure. Protein Engineering 2:119–125.

Henrick K, Thornton JM. 1998. PQS: a protein quaternary structure file server. Trends in Biochemical Sciences 23:358–361.

Heremans L, Heremans K. 1989. Raman spectroscopic study of the changes in secondary structure of chymotrypsin: Effect of pH and pressure on the salt bridge. Biochim Biophys Acta 999:192–197.

Hicks S, Wheeler DA, Plon SE, Kimmel M. 2011. Prediction of missense mutation functionality depends on both the algorithm and sequence alignment employed. Human Mutation 32.6:661–668.

Hughes SE, McKenna WJ. 2005. New insights into the pathology of inherited cardiomyopathy. Heart 91.2:257–264.

Hunter S, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, Bork P, Das U, Daugherty L, Duquenne L, Finn RD, Gough J, Haft D, Hulo N, Kahn D, Kelly E, Laugraud A, Letunic I, Lonsdale D, Lopez R, Madera M, Maslen J, McAnulla C, McDowall J, Mistry J, Mitchell A, Mulder N, Natale D, Orengo C, Quinn AF, Selengut JD, Sigrist CJA, Thimma M, Thomas PD, Valentin F, Wilson D, Wu CH, Yeats C. January 2009. InterPro: the integrative protein signature database. Nucleic Acids Research 37.suppl 1:D211–D215.

Hurst JM, McMillan LEM, Porter CT, Allen J, Fakorede A, Martin ACR. 2009. The SAAPdb web resource: a large-scale structural analysis of mutant proteins. Human Mutation 30:616–

624.

International Human Genome Sequencing Consortium . 2004. Finishing the euchromatic sequence of the human genome. Nature 431:931–945.

Izarzugaza J, Hopcroft L, Baresic A, Orengo C, Martin A, Valencia A. 2011. Characterization of pathogenic germline mutations in human protein kinases. BMC Bioinformatics 12:S1.

Izarzugaza JMG, Redfern OC, Orengo CA, Valencia A. 2009. Cancer-associated mutations are preferentially distributed in protein kinase functional sites. Proteins: Structure, Function, and Bioinformatics 77.4:892–903.

Jabs A, Weiss MS, 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.

Jelesarov I, Karshikoff A. 2009. Defining the role of salt bridges in protein stability. Methods in Molecular Biology 490:227–260.

Jerez JM, Molina I, Garcia-Laencina PJ, Alba E, Ribelles N, Martín M, Franco L. 2010. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. Artificial Intelligence in Medicine 50:105–115.

Jim L, Ben B, Sander O, Eduardo K, Barry R, Hadley W, Anupam T, Olivier E, Gabor G, Toews M, John K, Mike C, Rolf T, Carl W, Julian S, Thomas P, Remko D, Elisa B, Ofir L, 2009. plotrix: Various plotting functions. R package version 2.7.

Kabari LG, Nwachukwu EO, 2012. *Neural Networks and Decision Trees For Eye Diseases Diagnosis, Advances in Expert Systems*. InTech.

Kabsch W, Sander C. 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22:2577–2637.

Kaminker JS, Zhang Y, Watanabe C, Zhang Z. 2007. Canpredict: a computational tool for predicting cancer-associated missense mutations. Nucleic Acids Research 35.suppl 2:W595–W598.

Kanehisa M, Araki M, Goto S, Hattori M, Hirakawa M, Itoh M, Katayama T, Kawashima S, Okuda S, Tokimatsu T, Yamanishi Y. 2008. KEGG for linking genomes to life and the environment. Nucleic Acids Research 36:D480–D484.

Kanehisa M, Goto S, Furumichi M, Tanabe M, Hirakawa M. January 2010. KEGG for representation and analysis of molecular networks involving diseases and drugs. Nucleic acids research 38.Database issue:D355–D360.

Karchin R, Diekhans M, Kelly L, Thomas DJ, Pieper U, Eswar N, Haussler D, Sali A. 2005. Ls-snp: large-scale annotation of coding non-synonymous snps based on multiple information sources. Bioinformatics 21.12:2814–2820.

Karkkainen MJ, Ferrell RE, Lawrence EC, Kimak MA, Levinson KL, McTigue MA, Alitalo K, Finegold DN. 2000. Missense mutations interfere with VEGFR-3 signalling in primary lymphoedema. Nature Genetics 25:153–159.

Katoh K, Misawa K, Kuma K, Miyata T. 2002. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. Nucleic Acids Research 30.14:3059–3066.

Kemball-Cook G, Tuddenham EGD, Wacey AI. 1998. The factor viii structure and mutation resource site: Hamsters version 4. Nucleic Acids Research 26.1:216–219.

Khan S, Vihinen M. 2007. Spectrum of disease-causing mutations in protein secondary

structures. BMC Structural Biology 7.1:56.

Kitatani T, Nakamura Y, Wada K, Kinoshita T, Tamoi M, Shigeoka S, Tada T. Aug 2006. Structure of apo-glyceraldehyde-3-phosphate dehydrogenase from *Synechococcus* PCC7942. Acta Crystallographica Section F 62.8:727–730.

Knox C, Law V, Jewison T, Liu P, Ly S, Frolkis A, Pon A, Banco K, Mak C, Neveu V, Djoumbou Y, Eisner R, Guo ACC, Wishart DS. January 2011. DrugBank 3.0: a comprehensive resource for 'omics' research on drugs. Nucleic acids research 39.Database issue:D1035–D1041.

Krissinel E, Henrick K. 2007. Inference of macromolecular assemblies from crystalline state. Journal of Molecular Biology 372:774–797.

Kumar P, Henikoff S, Ng PC. June 2009. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. Nature protocols 4.7:1073–1081.

Kwok CJ, Martin ACR, Au SWN, Lam VMS. 2002. G6PDdb, an integrated database of glucose-6-phosphate dehydrogenase (G6PD) mutations. Human Mutation 19:217–224.

Lahiry P, Torkamani A, Schork NJ, Hegele RA. 2010. Kinase mutations in human disease: interpreting genotype–phenotype relationships. Nature Reviews Genetics 11.1:60–74.

Lander ES, Linton LM, Birren B, *et al.* . 2001. Initial sequencing and analysis of the human genome. Nature 409:860–921.

Lane DP, Fischer PM. 2004. Turning the key on p53. Nature 427:789–790.

Lee B, Richards FM. 1971. The interpretation of protein structures: Estimation of static accessibility. Journal of Molecular Biology 55:379–400.

Lee J, Lee K, Shin S. 2000. Theoretical studies of the response of a protein structure to cavity-creating mutations. Biophys J 78:1665–1671.

Lee TY, Huang HD, Hung JH, Huang HY, Yang YS, Wang TH. 2006. dbPTM: an information repository of protein post-translational modification. Nucleic Acids Research 34.suppl$_1$: $D622 - -627$.

Leigh SEA, Foster AH, Whittall RA, Hubbart CS, Humphries SE. 2008. Update and analysis of the University College London low density lipoprotein receptor familial hypercholesterolemia database. Annual Human Genetics 72:485–498.

Leinonen R, Akhtar R, Birney E, Bower L, Cerdeno-Tárraga A, Cheng Y, Cleland I, Faruque N, Goodgame N, Gibson R, Hoad G, Jang M, Pakseresht N, Plaister S, Radhakrishnan R, Reddy K, Sobhany S, Ten Hoopen P, Vaughan R, Zalunin V, Cochrane G. January 2011. The European Nucleotide Archive. Nucleic Acids Research 39.suppl 1:D28–D31.

Leiros HK, Willassen NP, Smalls AO. 2000. Structural comparison of psychrophilic and mesophilic trypsins. Elucidating the molecular basis of cold-adaptation. European Journal of Biochemistry 267:1039–1049.

Lesk AM, 2005. *Introduction to protein scienceâĂŤarchitecture, function, and genomics: Lesk, Arthur M.* John Wiley Sons Inc.

Levitt M, Gerstein M, Huang E, Subbiah S, Tsai J. 1997. Protein folding: the endgame. Annu Rev Biochem 66:549–579.

Li B, Krishnan VG, Mort ME, Xin F, Kamati KK, Cooper DN, Mooney SD, Radivojac P. 2009. Automated inference of molecular mechanisms of disease from amino acid substitutions. Bioinformatics 25.21:2744–2750.

Li C, Iosef C, Jia CYH, Gkourasas T, Han VKM, Shun-Cheng Li S. 2003. Disease-causing sap mutants are defective in ligand binding and protein folding. Biochemistry 42.50:14885–

14892. PMID: 14674764.

Liao SM. 2009. Genetics. Journal of Medical Ethics pages 306–309.

Liu T, Lin Y, Wen X, Jorissen RN, Gilson MK. January 2007. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. Nucleic Acids Research 35.Database issue:198–201.

Logan T, Clark L, Ray SS. 2010. Engineered disulfide bonds restore chaperone-like function of DJ-1 mutants linked to familial Parkinson's disease. Biochemistry 49:5624–5633.

Ma X, Wright J, Dou S, Olsen P, Teeter L, Adams G, Graviss E. 2002. Ethnic divergence and linkage disequilibrium of novel snps in the human nli-if gene: evidence of human origin and lack of association with tuberculosis susceptibility. Journal of Human Genetics 47.3:140–145. http://dx.doi.org/10.1007/s100380200016.

Mande SC, Hol WG, Mainfroid V, Goraj K, Martial JA, Kalk KH. 1994. Crystal structure of recombinant human triosephosphate isomerase at 2.8 ÅĚ resolution. triosephosphate isomerase-related human genetic disorders and comparison with the trypanosomal enzyme. Protein Science 3.5:810–821.

Manning G, Whyte DB, Martinez R, Hunter T, Sudarsanam S. December 2002. The Protein Kinase Complement of the Human Genome. Science 298.5600:1912–1934.

Manson AL, 2002. *Cell biology and genetics*. Elsevier Health Sciences, 2002, 2, illustrated edition.

Marian A. 2002. Modifier genes for hypertrophic cardiomyopathy. Current opinion in cardiology 17.3:242252.

Markowski CA, Markowski EP. 1990. Conditions for the effectiveness of a preliminary test of variance. The American Statistician 44:322–326.

Martin ACR, Facchiano AM, Cuff AL, Hernandez-Boussard T, Olivier M, Hainaut P, Thornton JM. 2002. Integrating mutation data and structural analysis of the TP53 tumor-suppressor protein. Human Mutation 19:149–164.

Martin ACR. 2005. Mapping PDB chains to UniProtKB entries. Bioinformatics 21.23:4297–4301.

Martin ACR, Facchiano AM, Cuff AL, Hernandez-Boussard T, Olivier M, Hainaut P, Thornton JM. 2002. Integrating mutation data and structural analysis of the TP53 tumor-suppressor protein. Human Mutation 19:149–164.

Mathe E, Olivier M, Kato S, Ishioka C, Hainaut P, Tavtigian SV. 2006. Computational approaches for predicting the biological effect of p53 missense mutations: a comparison of three sequence analysis based methods. Nucleic Acids Research 34.5:1317–1325.

Matys V, Kel-Margoulis OV, Fricke E, Liebich I, Land S, Barre-Dirrie A, Reuter I, Chekmenev D, Krull M, Hornischer K, Voss N, Stegmaier P, Lewicki-Potapov B, Saxel H, Kel AE, Wingender E. January 2006. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. Nucleic acids research 34.Database issue:D108–D110.

McDonald IK, Thornton JM. 1994. Satisfying hydrogen bonding potential in proteins. Journal of Molecular Biology 238.5:777–793.

McKusick V, December 1998. *Mendelian Inheritance in Man: A Catalog of Human Genes and Genetic Disorders*. Johns Hopkins University Press, Baltimore, 12th edition.

McKusick V, Francomano C, Antonarakis S, 1992. *Mendelian inheritance in man: catalogs of autosomal dominant, autosomal recessive, and X-linked phenotypes*. Number v. 1. Johns Hopkins University Press.

McKusick VA, 2000. Online Mendelian Inheritance in Man (OMIM)(TM). McKusick-Nathans Institute for Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD).

McMillan LEM, Martin ACR. 2008. Automatically extracting functionally equivalent proteins from SwissProt. BMC Bioinformatics 9:418.

McMillan LEM, 2009. *Post-genomic structural analysis of single amino acid polymorphisms*. PhD thesis, UNIVERSITY COLLEGE LONDON, UCL Research Department of Structural and Molecular Biology.

Meyer D, Leisch F, Hornik K. 2003. The support vector machine under test. Neurocomputing 55:169–186.

Mierswa I, Wurst M, Klinkenberg R, Scholz M, Euler T, 2006. Yale: rapid prototyping for complex data mining tasks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA. ACM.

Mitchell JB, Thornton JM, Singh J, Price SL. 1992. Towards an understanding of the arginine-aspartate interaction. Journal of Molecular Biology 226.1:251 – 262.

Mitchell T, 1997. *Machine learning*. McGraw Hill, 1st edition.

Moo-Penn WF, Schmidt RM, jue DL, Bechtel KC, Wright JM, Horne MK, Haycraft GL, Roth EF, Nagel RL. 1977. Hemoglobin s travis: a sickling hemoglobin with two amino acid substitutions [b6(a3)glutamic acid to valine and b 142(h20) alanine to valine]. European Journal of Biochemistry 77.3:561–566.

Mood A, Graybill FA, Boes DC 1974. *Introduction to the Theory of Statistics*, pages 241–246. McGraw-Hill, 3rd edition edition.

Morris JR, Keep NH, Solomon E. 2002. Identification of residues required for the interaction of bard1 with brca1. Journal of Biological Chemistry 277.11:9382–9386.

Mouillet E. 2008. PubMed 2009. Santé (Montrouge, France) 18.4:234–240.

Mount DW, 2004. *Bioinformatics: sequence and genome analysis*. CSHL Press.

Murray R, Harper H, 2000. *Harper's Biochemistry*. Appleton & Lange.

Nelson DL, Terhorst C. 2000. X-linked lymphoproliferative syndrome. Clinical Experimental Immunology 122.3:291–295.

Ng PC, Henikoff S. 2001. Predicting deleterious amino acid substitutions. Genome Research 11.5:863–74.

Ng PC, Henikoff S. March 2002. Accounting for human polymorphisms predicted to affect protein function. Genome research 12.3:436–446.

Ng PC, Henikoff S. July 2003. SIFT: predicting amino acid changes that affect protein function. Nucleic Acids Research 31.13:3812–3814.

Nickerson DA, Taylor SL, Weiss KM, Clark AG, Hutchinson RG, StengÃërd J, Salomaa V, Vartiainen E, Boerwinkle E, Sing CF. 1998. DNA sequence diversity in a 9.7-kb region of the human lipoprotein lipase gene. Nature Genetics 19:233–240.

Northey JGB, Di Nardo AA, Davidson AR. 2002. Hydrophobic core packing in the SH3 domain folding transition state. Nature Structural Biology 9:126–130.

Oetting WS. 2011. Exploring the functional consequences of genomic variation: The 2010 human genome variation society scientific meeting. Human Mutation 32.4:486–490.

Olivier M, Eeles R, Hollstein M, Khan MA, Harris CC, Hainaut P. 2002. The IARC TP53

database: new online mutation analysis and recommendations to users. Human Mutation 19:607–614.

Olson M, Hood L, Cantor C, Botstein D. September 1989. A common language for physical mapping of the human genome. Science (New York, N.Y.) 245.4925:1434–1435.

Olzmann JA, Brown K, Wilkinson KD, Rees HD, Huai Q, Ke H, Levey AI, Li L, Chin LS. 2004. Familial parkinson's disease-associated l166p mutation disrupts dj-1 protein folding and function. Journal of Biological Chemistry 279.9:8506–8515.

Pace CN, Grimsley GR, Thomson JA, Barnett BJ. 1988. Conformational stability and activity of ribonuclease T1 with zero, one, and two intact disulfide bonds. Journal of Biological Chemistry 263:11820–11825.

Pavlov Y, 2000. *Random Forests*. VSP.

Pearson WR. Nov 1991. Searching protein sequence libraries: comparison of the sensitivity and selectivity of the smith-waterman and fasta algorithms. Genomics 11.3:635–650.

Pearson WR, Lipman DJ. 1988. Improved tools for biological sequence comparison. Proceedings of the National Academy of Sciences of the USA 85:2444–2448.

Petitjean A, Mathe E, Kato S, Ishioka C, Tavtigian SV, Hainaut P, 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.

Pey AL, Stricher F, Serrano L, Martinez A. November 2007. Predicted effects of missense mutations on native-state stability account for phenotypic outcome in phenylketonuria, a paradigm of misfolding diseases. American journal of human genetics 81.5:1006–1024.

Piirilä H, Väliaho J, Vihinen M. 2006. Immunodeficiency mutation databases (IDbases). Human Mutation 27:1200–1208.

Ponder J, Case D. 2003. Force fields for protein simulations. Adv Protein Chem 66:27–85.

Porter CT, Bartlett GJ, Thornton JM. 2004. The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. Nucleic Acids Research 32:D129–D133.

Pruitt KD, Tatusova T, Klimke W, Maglott DR. January 2009. NCBI Reference Sequences: current status, policy and new initiatives. Nucleic acids research 37.Database issue:D32–36.

Ptitsyn OB. 1998. Protein folding and protein evolution: Common folding nucleus in different subfamilies of c-type cytochromes? Journal of Molecular Biology 278:655–666.

Ptitsyn OB, Ting KL. 1999. Non-functional conserved residues in globins and their possible role as a folding nucleus. Journal of Molecular Biology 291:671–682.

R Development Core Team , 2008. R: A language and environment for statistical computing. ISBN 3-900051-07-0.

Ramensky V, Bork P, Sunyaev S. 2002. Human non-synonymous SNPs: server and survey. Nucleic Acids Research 30:3894–3900.

Reva B, Antipin Y, Sander C. September 2011. Predicting the functional impact of protein mutations: application to cancer genomics. Nucleic acids research 39.17:e118.

Richard P, Charron P, Carrier L, Ledeuil C, Cheav T, Pichereau C, Benaiche A, Isnard R, Dubourg O, Burban M, Gueffet JP, Millaire A, Desnos M, Schwartz K, Hainque B, Komajda M, for the EUROGENE Heart Failure Project . 2003. Hypertrophic cardiomyopathy: Distribution of disease genes, spectrum of mutations, and implications for a molecular diagnosis strategy. Circulation 107.17:2227–2232.

Richards FM. 1997. Protein stability: Still an unsolved problem. Cell Mol Life Sci 53:790–

802.

Rose PW, Beran B, Bi C, Bluhm WF, Dimitropoulos D, Goodsell DS, Prlic A, Quesada M, Quinn GB, Westbrook JD, Young J, Yukich B, Zardecki C, Berman HM, Bourne PE. January 2011. The RCSB Protein Data Bank: redesigned web site and web services. Nucleic acids research 39.Database issue.

Rost B, Sander C. 1993. Prediction of protein secondary structure at better than 70% accuracy. Journal of Molecular Biology 232.2:584 – 599.

Rothenberg MA, Chapman CF, 1989. *Dictionary of Medical Terms for the Nonmedical Person*. Barron's Educational Series, Inc., 2nd edition.

Rumelhart DE, Hinton GE, Williams RJ. 1986. Learning representations by back-propagating errors. Nature 323:533–536.

Ruti Parvari AL. 2012. The mutations associated with dilated cardiomyopathy. Biochemistry Research International 2012.

Ryan M, Diekhans M, Lien S, Liu Y, Karchin R. 2009. Ls-snp/pdb: annotated non-synonymous snps mapped to protein data bank structures. Bioinformatics 25.11:1431–1432.

Saar-Tsechansky M, Provost F. 2007. Handling missing values when applying classification models. Journal of Machine Learning Research 8:1625–1657.

Salama JJ, Donaldson I, Hogue CW. 2001. Automatic annotation of BIND molecular interactions from three-dimensional structures. Biopolymers 61:111–120.

Sayers EW, Barrett T, Benson DA, Bolton E, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Federhen S, Feolo M, Fingerman IM, Geer LY, Helmberg W, Kapustin Y, Landsman D, Lipman DJ, Lu Z, Madden TL, Madej T, Maglott DR, Marchler-Bauer A, Miller V, Mizrachi I, Ostell J, Panchenko A, Phan L, Pruitt KD, Schuler GD, Sequeira E, Sherry ST, Shumway M, Sirotkin K, Slotta D, Souvorov A, Starchenko G, Tatusova TA, Wagner L, Wang Y, Wilbur WJ, Yaschenko E, Ye J. January 2011. Database resources of the National Center for Biotechnology Information. Nucleic acids research 39.Database issue.

Schmid CD, Perier R, Praz V, Bucher P. January 2006. EPD in its twentieth year: towards complete promoter coverage of selected model organisms. Nucleic Acids Research 34.suppl 1:D82–D85.

Schrödinger, LLC , August 2010. The PyMOL molecular graphics system, version 1.3r1. PyMOL, The PyMOL Molecular Graphics System, Version 1.3, Schrödinger, LLC.

Schuler G. 1997. Pieces of the puzzle: expressed sequence tags and the catalog of human genes. Journal of Molecular Medicine 75.10:694–8.

Schwarz JM, Rodelsperger C, Schuelke M, Seelow D. August 2010. MutationTaster evaluates disease-causing potential of sequence alterations. Nature Methods 7.8:575–576.

Scriver CR, Hurtubise M, Konecki D, Phommarinh M, Prevost L, Erlandsen H, Stevens R, Waters PJ, Ryan S, McDonald D, et al. . 2003. Pahdb 2003: what a locus-specific knowledge-base can do. Human Mutation 21.4:333–344.

Segal MR. 2004. Machine Learning Benchmarks and Random Forest Regression.

Seidman CE, Seidman J. 2011. Identifying sarcomere gene mutations in hypertrophic cardiomyopathy: A personal history. Circulation Research 108.6:743–750.

Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, Sirotkin K. January 2001. dbSNP: the NCBI database of genetic variation. Nucleic acids research 29.1:308–311.

Shi Z, Krantz BA, Kallenbach N, Sosnick TR. 2002. Contribution of hydrogen bonding to protein stability estimated from isotope effects. Biochemistry 41:2120–2129.

Shih HH, Brady J, Karplus M. 1985. Structure of proteins with single-site mutations: a minimum perturbation approach. Proceedings of the National Academy of Sciences of the USA 82:1697–1700.

Shihab HA, Gough J, Cooper DN, Stenson PD, Barker GLA, Edwards KJ, Day INM, Gaunt TR. 2013. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. Human Mutation 34:57–65.

Sidransky D, Hollstein M. 1996. Clinical implications of the p53 gene. Annual Review of Medicine 47:285–301.

Sim NL, Kumar P, Hu J, Henikoff S, Schneider G, Ng PC. 2012. SIFT web server: Predicting effects of amino acid substitutions on proteins. Nucleic Acids Research 40:W452–W457.

Snow ME, Amzel LM. 1986. Calculating three-dimensional changes in protein structure due to amino-acid substitutions: the variable region of immunoglobulins. Proteins 1:267–279.

Speicher M, Antonarakis S, Motulsky A, Vogel F, 2010. *Vogel and Motulsky's Human Genetics: Problems and Approaches*. SpringerLink: Springer e-Books. Springer Berlin Heidelberg.

Stabler SP, Jones RT, Head C, Shih DT, Fairbanks VF. 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 c 69:237–243.

Stead LF, Wood IC, Westhead DR. 2011. Kvsnp: accurately predicting the effect of genetic variants in voltage-gated potassium channels. Bioinformatics 27.16:2181–2186.

Stenson PD, Ball EV, Howells K, Phillips AD, Mort M, Cooper DN. December 2009. The Human Gene Mutation Database: providing a comprehensive central mutation database for molecular diagnostics and personalized genomics. Human genomics 4.2:69–72.

Stenson PD, Ball EV, Mort M, Phillips AD, Shaw K, Cooper DN 2002. *The Human Gene Mutation Database (HGMD) and Its Exploitation in the Fields of Personalized Genomics and Molecular Evolution*, chapter 1. John Wiley Sons, Inc.

Stewart DE, Sarkar A, Wampler JE. 1990. Occurrence and role of cis peptide bonds in protein structures. Journal of Molecular Biology 214:253–260.

Stickle DF, Presta LG, Dill KA, Rose GD. 1992. Hydrogen bonding in globular proteins. Journal of Molecular Biology 226:1143–1159.

Stitziel NO, Binkowski TA, Tseng YY, Kasif S, Liang J. 2004. toposnp: a topographic database of nonâĂŘsynonymous single nucleotide polymorphisms with and without known disease association. Nucleic Acids Research 32.suppl 1:D520–D522.

Stone EA, Sidow A. 2005. Physicochemical constraint violation by missense substitutions mediates impairment of protein function and disease severity. Genome Research 15:978–986.

Strachan T, Read A, 2011. *Human Molecular Genetics 4*. Garland Science/Taylor & Francis Group.

Strange RW, Antonyuk SV, Hough MA, Doucette PA, Valentine JS, Hasnain SS. 2006. Variable metallation of human superoxide dismutase: Atomic resolution crystal structures of cu-zn, zn-zn and as-isolated wild-type enzymes. Journal of Molecular Biology 356.5:1152 – 1162.

Studer RA, Dessailly BH, Orengo CA. 2013. Residue mutations and their impact on protein structure and function: detecting beneficial and pathogenic changes. Biochemical Journal 449.3:581–594.

Sugawara H, Ogasawara O, Okubo K, Gojobori T, Tateno Y. 2008. Ddbj with new system

and face. Nucleic Acids Research 36.suppl 1:D22–D24.

Sunyaev SR, Lathe WC, Ramensky VE, Bork P. 2000. SNP frequencies in human genes an excess of rare alleles and differing modes of selection. Trends in Genetics 16:335–337.

Sunyaev SR, Eisenhaber F, Rodchenkov IV, Eisenhaber B, Tumanyan VG, Kuznetsov EN. May 1999. PSIC: profile extraction from sequence alignments with position-specific counts of independent observations. Protein Engineering 12.5:387–394.

Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. 2003. Random Forest: A classification and regression tool for compound classification and QSAR modeling. Journal of chemical information and computer sciences 43:1947–1958.

Taillon-Miller P, Gu Z, Li Q, Hillier L, Kwok PY. 1998. Overlapping genomic sequences: A treasure trove of single-nucleotide polymorphisms. Genome Research 8:748–754.

Tanford C, 1980. *The Hydrophobic Effect: Formation of Micelles and Biological Membranes 2d Ed*. J. Wiley.

Tavtigian SV, Deffenbaugh AM, Yin L, Judkins T, Scholl T, Samollow PB, de Silva D, Zharkikh A, Thomas A. 2006. Comprehensive statistical study of 452 brca1 missense substitutions with classification of eight recurrent substitutions as neutral. Journal of Medical Genetics 43.4:295–305.

Tavtigian SV, Greenblatt MS, Lesueur F, Byrnes GB. 2008. In silico analysis of missense substitutions using sequence-alignment based methods. Human Mutation 29.11:1327–1336.

Thangudu R, Manoharan M, Srinivasan N, Cadet F, Sowdhamini R, Offmann B. 2008. Analysis on conservation of disulphide bonds and their structural features in homologous protein domain families. BMC Structural Biology 8.1:55.

Thomas PD, Campbell MJ, Kejariwal A, Mi H, Karlak B, Daverman R, Diemer K, Muruganujan A, Narechania A. September 2003. PANTHER: A Library of Protein Families and Subfamilies Indexed by Function. Genome Research 13.9:2129–2141.

Thompson JD, Prigent V, Poch O. 2004. Leon: multiple alignment evaluation of neighbours. Nucleic Acids Research 32.4:1298–1307.

Thusberg J, Olatubosun A, Vihinen M. 2011. Performance of mutation pathogenicity prediction methods on missense variants. Human Mutation 32:358–368.

Tian J, Wu N, Guo X, Guo J, Zhang J, Fan Y. 2007. Predicting the phenotypic effects of nonsynonymous single nucleotide polymorphisms based on support vector machines. BMC Bioinformatics 8:450–450.

Ting KLH, Jernigan RL. 2002. Identifying a folding nucleus for the lysozyme/alpha-lactalbumin family from sequence conservation clusters. J Mol Evol 54:425–436.

Tong W, Hong H, Fang H, Xie Q, Perkins R. 2003. Decision Forest: Combining the predictions of multiple independent decision tree models. Journal of Chemical Information and Computer Sciences 43:525–531.

Torshin IY, Harrison RW. 2001. Charge centers and formation of the protein folding core. Proteins 43:353–364.

Tuchman M, Jaleel N, Morizono H, Sheehy L, Lynch MG. 2002. Mutations and polymorphisms in the human ornithine transcarbamylase gene. Human Mutation 19:93–107.

Tweedie S, Ashburner M, Falls K, Leyland P, Mcquilton P, Marygold S, Millburn G, Osumi-Sutherland D, Schroeder A, Seal R, Zhang H, Consortium TF. January 2009. FlyBase: enhancing Drosophila Gene Ontology annotations. Nucleic Acids Research 37.suppl_1:D555–559.

Venselaar H, Te Beek TAH, Kuipers RKP, Hekkelman ML, Vriend G. 2010. Protein structure analysis of mutations causing inheritable diseases. An e-Science approach with life scientist friendly interfaces. BMC Bioinformatics 11:548–548.

Wang DG, Fan JB, Siao CJ, Berno A, Young P, Sapolsky R, Ghandour G, Perkins N, Winchester E, Spencer J, others . 1998. Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome. Science 280.5366:1077–1082.

Wang G, Dunbrack RL. 2005. PISCES: Recent improvements to a PDB sequence culling server. Nucleic Acids Research 33:W94–W98.

Wang P, Dai M, Xuan W, McEachin RC, Jackson AU, Scott LJ, Athey B, Watson SJ, Meng F. July 2006. SNP Function Portal: a web database for exploring the function implication of SNP alleles. Bioinformatics 22.14:e523–e529.

Wang Q, Buckle AM, Fersht AR. 2000. Stabilization of GroEL minichaperones by core and surface mutations. Journal of Molecular Biology 298:917–926.

Wang Y, Addess KJ, Chen J, Geer LY, He J, He S, Lu S, Madej T, Marchler-Bauer A, Thiessen PA, Zhang N, Bryant SH. 2007. MMDB: annotating protein sequences with Entrez's 3D-structure database. Nucleic Acids Research 35:D298–D300.

Warburton E, April 2008. Low penetrance genes vs. high penetrance genes. www.geneticsandhealth.com.

Wedemeyer WJ, Welker E, Scheraga HA. 2002. Proline cis-trans isomerization and protein folding. Biochemistry 41:14637–14644.

Welch BL. 1947. The generalization of "Student's" problem when several different population variances are involved. Biometrika 34:28–35.

Wheeler DL, Barrett T, Benson DA, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Edgar R, Federhen S, Geer LY, Kapustin Y, Khovayko O, Landsman D, Lipman DJ, Madden TL, Maglott DR, Ostell J, Miller V, Pruitt KD, Schuler GD, Sequeira E, Sherry ST, Sirotkin K, Souvorov A, Starchenko G, Tatusov RL, Tatusova TA, Wagner L, Yaschenko E. 2007. Database resources of the national center for biotechnology information. Nucleic Acids Research 35.suppl 1:D5–D12.

Wilcock D, Pisabarro MT, López-Hernandez E, Serrano L, Coll M. May 1998. Structure Analysis of Two CheY Mutants: Importance of the Hydrogen-Bond Contribution to Protein Stability. Acta Crystallographica Section D 54.3:378–385.

Wilmot CM, Thornton JM. 1988. Analysis and prediction of the different types of beta-turn in proteins. Journal of Molecular Biology 203:221–232.

Witten IH, Frank E, 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition.

Wroe R, Wai-Ling Butler A, Andersen PM, Powell JF, Al-Chalabi A. August 2008. ALSOD: the Amyotrophic Lateral Sclerosis Online Database. Amyotrophic lateral sclerosis : official publication of the World Federation of Neurology Research Group on Motor Neuron Diseases 9.4:249–250.

Yamaguchi S, Brailey LL, Morizono H, Bale AE, Tuchman M. 2006. Mutations and polymorphisms in the human ornithine transcarbamylase (otc) gene. Human Mutation 27.7:626–632.

Yates F. 1934. Contingency table involving small numbers and the χ^2 test. Supplement of the Journal of the Royal Statistical Society 1:217–235.

Yue P, Melamud E, Moult J. 2006. Snps3d: Candidate gene and snp selection for association studies. BMC Bioinformatics 7.1:166.

Zhang Y, Rajapakse J, 2009. *Machine Learning in Bioinformatics (Wiley Series in Bioinformatics)*. Wiley Series in Bioinformatics. Wiley.

Appendices

[A] The UniProtKB/Swiss-Prot file format description

Entry information (ID, AC and DT)

The ID line provides the entry name. The *primary* AC, followed by the *secondary* ACs are indicated in the AC line, and DT line provides the entry date. See line #1-8 in Figure 2.1).

Name and origin (DE, GN AND OS)

The description (DE) line contains the protein name, synonyms and abbreviations. Proteins may be described using any number of synonyms. Also included in this line is an indication of whether or not the protein is a 'Fragment', and the EC number if relevant. The GN line gives gene and locus names. The species (OS) line indicates the species and taxonomy information. In the example, there are four synonyms: 'Cellular tumor antigen p53'; 'Antigen NY-CO-13'; 'Phosphoprotein p53' and 'Tumor suppressor p53' (line #9-14 in Figure 2.1).

Cross references (DR)

The DR line UniProtKB/Swiss-Prot provides cross-references between databases (this data is used to construct datasets with which to benchmark FOSTA (McMillan and Martin, 2008) against another method Inparanoid). P53_HUMAN is cross referenced to ENA records X02469 and CAA26306.1; PIR records A25224 and DNHU53; RefSeq records NP_000537.3 and NM_000546.4; PDB records 1A1U and other databases entries (lines #15-18 in Figure 2.1).

Features (FT)

UniProtKB/Swiss-Prot provides more than 30 feature keys (FT) that include sequence, structural and function annotations found in the protein. These may be transferred by homology, or there may be experimental evidence or non-experimental qualifiers ('Potential', 'Probable' and 'By similarity') which indicate the status of the annotation (lines #19-33 in Figure 2.1).

Sequence (SQ)

The sequence (SQ) line provides the sequence; total amino acid count; molecular weight and a cyclic redundancy check (CRC) value. This is followed by a terminating line ('//'), which designates the end of an entry, (lines #34-35 in Figure 2.1).

[B] Improved MutModel Program

```
Usage: mutmodel [-m resspec newres] [-e clashMethod] [-c chitab]
                [-r refcoor] [-v] [-f conffile] [-d] [-o] [-x]
                [-s stepsize] [-t tolerance] [-p paramfile]
                [-l Lennard-Jones-Cutoff] [infile.pdb [outfile.pdb]

       -m Specify a mutation where resspec is of the form [c]nnn[i]
          (where [c] is an optional chain name, nnn is a residue
          number and [i] is an optional insert code).
          Multiple mutations may be specified with multiple
          -m options. newres may be 1-letter or 3-letter code.
       -e Specify the clash evaluation method: 1: Boolean;
          2: Linear clash; 3: VdW (Lennard-Jones);
          4: VdW/Torsion. [Default: 1]
       -p Specify energy parameter file. [Default: eparams.dat]
       -c Specify the Chi equivalents table (Default: chitab.dat
          in the current directory or $DATADIR)
       -r Specify the reference coordinates file
          (Default: coor in the current directory or $DATADIR)
       -v Verbose mode; reports whether the side-chain replacement
          was acceptable.
       -f Write each conformation to a PDB format file
       -d Debugging: Print clash table and choice
       -o Only do MOP, not MPP
       -s Specify search step size (degrees). [Default: 30.00]
       -x Generate a random model for the side-chain
       -t Specify tolerance in energy for accepting the parent
          conformation or a standard rotamer position.
          [Default: 1.0]
       -l Specify distance cutoff for van der Waals (Lennard-Jones)
          energy calculations. Atom pairs with greater separation
          are ignored. [Default: 8.0]
```

```
        If input and output files are not specified, standard
        input/output will be used.

MutModel performs a very simple side-chain replacement using the
minimum perturbation protocol (MPP). The side-chain is replaced
and then spun around its Chi1 and Chi2 torsion angles to find a
position which makes minimal bad contacts as evaluated using one
of three clash evaluation methods (see -e).

Note that using methods 3 and 4, if a conformation is written
from this program and then the energy is calculated, it will
differ somewhat from the energy calculated when the conformation
was generated. This is because the PDB format rounds the atom
coordinates to 3 decimal places. Each VDW energy can then be
out in the 5th decimal place, but this can accumulate to a
surprisingly large difference in final energy.

Note that the ideal value for -l is 14.5A but this increases
the run  time by ~10x. The default of 8.0A gives a good
tradeoff between accuracy an speed - smaller values will
speed it up more at the expense of accuracy.
```

[C] Predicting Damaging Mutations – JSON file


```
{"SAAFS":   { "uniprotac": "P12883",
              "resnum": 869,
              "native": "R",
              "mutant": "G",
              "pdbs"  : [
{"SAAF":
   { "file": "/acrm/data/pdb/pdb2fxm.ent",
     "pdbcode": "2fxm",
     "residue": "A869",
     "mutation": "G",
     "structuretype": "crystal",
     "resolution": "2.70A",
     "rfactor": "24.20%",
     "results": {
     "Binding": {"Binding-BOOL": "OK"},
     "BuriedCharge": {"BuriedCharge-NATIVE-CHARGE": "1", "BuriedCharge-MUTANT-CHARGE": "0",
                      "BuriedCharge-BOOL": "OK", "BuriedCharge-RELACCESS": "51.582"},
     "CisPro": {"CisPro-BOOL": "OK", "CisPro-NATIVE": "ARG", "CisPro-OMEGA": "178.996"},
     "Clash": {"Clash-ENERGY": "0.00", "Clash-BOOL": "OK"},
     "CorePhilic": {"CorePhilic-MUTANT-HPHOB": "0.16", "CorePhilic-BOOL": "OK",
                    "CorePhilic-RELACCESS": "77.011", "CorePhilic-NATIVE-HPHOB": "-1.8"},
     "Glycine": {"Glycine-MUTANT-THRESHOLD": "", "Glycine-BOOL": "OK",
                 "Glycine-NATIVE-THRESHOLD": "0.35","Glycine-NATIVE-ENERGY": "",
                 "Glycine-PSI": "-46.473", "Glycine-MUTANT-ENERGY": "",
                 "Glycine-PHI": "-59.091", "Glycine-NATIVE-BOOL": "OK",
                 "Glycine-MUTANT": "GLY", "Glycine-NATIVE": "ARG"},
     "HBonds": {"HBonds-ENERGY": "NULL", "HBonds-BOOL": "OK", "HBonds-ZVAL": "NULL",
           "HBonds-ATOM": "NULL", "HBonds-PARTNER-RES": "NULL",
"HBonds-PARTNER-ATOM": "NULL"},
     "Impact": {"Impact-BOOL": "BAD", "Impact-NSEQ": "9", "Impact-THRESHOLD": "0.67",
                "Impact-CONSSCORE": "1"], "Interface": {"Interface-BOOL": "BAD",
                "Interface-RELACCESS": "51.582", "Interface-RELACCESS-MOL": "77.011"},
     "Proline": {"Proline-MUTANT-THRESHOLD": "0.53", "Proline-BOOL": "OK",
                 "Proline-NATIVE-THRESHOLD": "", "Proline-NATIVE-ENERGY": "",
                 "Proline-PSI": "-46.473", "Proline-MUTANT-ENERGY": "", "Proline-PHI": "-59.091",
                 "Proline-NATIVE-BOOL": "OK", "Proline-MUTANT": "GLY", "Proline-NATIVE": "ARG"],
     "SFrotFT": {"SFrotFT-BOOL": "OK", "SFrotFT-FEATURES": "00000000000000", "SFrotFT-NAMES": ""},
     "SSGeom": {"SSGeom-BOOL": "OK"},
     "SurfacePhobic": {"SurfacePhobic-MUTANT-HPHOB": "0.16", "SurfacePhobic-BOOL": "OK",
                       "SurfacePhobic-RELACCESS": "77.011", "SurfacePhobic-NATIVE-HPHOB": "-1.8"},
     "Voids": {"Voids-MUTANT": [11.799,10.381,10.000,7.106,6.356,4.966,4.745,2.792,2.351,0],
               "Voids-MUTANT-LARGEST": "11.799000", "Voids-BOOL": "OK", "Voids-NATIVE": [11.799,
               10.381,10.000,7.106,6.356,4.966,4.745,2.792,2.351],
   "Voids-NATIVE-LARGEST": "11.799000"}}}}
,
{"SAAF":
   { "file": "/acrm/data/pdb/pdb2fxm.ent",
     "pdbcode": "2fxm",
     "residue": "B869",  ...    }}
                         ]
           }
}
```


Figure 1: An example of a JSON file.
UniProt: P12883, an Arginine amino acid in postion 869 mutated to Glycine.

[C.i] JSON file explanation

Table 1: JSON file category explanation.

Category		Explanation
"Binding":	"Binding-BOOL": "OK"	Is this residue making an H-Bond or VDW contact with a ligand or another protein chain?
"BuriedCharge":	"BuriedCharge-NATIVE-CHARGE": "1"	Charge on the native amino acid
	"BuriedCharge-MUTANT-CHARGE": "0"	Charge on the mutant amino acid
	"BuriedCharge-BOOL": "OK"	Is this a buried residue (<25% accessibility) where charge has changed
	"BuriedCharge-RELACCESS": "51.582"	accessibility (out of 100%)
"CisPro":	"CisPro-BOOL": "OK"	Was this a proline with a cis peptide bond
	"CisPro-NATIVE": "ARG"	The native amino acid
	"CisPro-OMEGA": "178.996"	The peptide bond dihedral angle
"Clash":	"Clash-ENERGY": "0.00"	Energy calculation for any clash - can be anything about -1000 to +100000
	"Clash-BOOL": "OK"	We define a bad clash as >34.33 (99% of clashes are less than this in native proteins)
"CorePhilic":	"CorePhilic-MUTANT-HPHOB": "0.16"	Mutant residue hydrophobicity
	"CorePhilic-BOOL": "OK"	Is this introducing a hydrophilic in the core where there was a hydrophobic: -0.1 is used as a threshold for hydrophilic/hydrophobic CorePhilic-RELACCESS: "-1"
	"CorePhilic-RELACCESS": "77.011"	Relative accessibility - only calculated when the native is hydrophobic and mutant is hydrophilic
	"CorePhilic-NATIVE-HPHOB": "-1.8"	Native residue hydrophobicity
"Glycine":	"Glycine-MUTANT-THRESHOLD": ""	
	"Glycine-BOOL": "OK"	Was this a native Gly with unusual backbone phi/psi angles being mutated to something else
	"Glycine-NATIVE-THRESHOLD": "0.35"	
	"Glycine-NATIVE-ENERGY": ""	

Continued on next page

Table 1 – *Continued from previous page*

Category		Explanation
	"Glycine-PSI": "-46.473"	Backbone Calpha-C dihedral angle
	"Glycine-MUTANT-ENERGY": ""	
	"Glycine-PHI": "-59.091"	Backbone N-Calpha dihedral angle
	"Glycine-NATIVE-BOOL": "OK"	Was this a native Gly with unusual backbone phi/psi angles being mutated to something else
	"Glycine-MUTANT": "GLY"	
	"Glycine-NATIVE": "ARG"	
"HBonds":	"HBonds-ENERGY": "NULL"	Energy for the HBond with the mutant âĂŞ NULL if not formed
	"HBonds-BOOL": "OK"	Was this a residue involved in an HBond and the mutant can't retain the HBond?
	"HBonds-ZVAL": "NULL"	Also energy for the HBond with the mutant âĂŞ NULL if not formed
	"HBonds-ATOM": "NULL"	Atom involved in an HBond
	"HBonds-PARTNER-RES": "NULL"	Partner residue if in an HBond
	"HBonds-PARTNER-ATOM": "NULL"	Partner atom if in an HBond
"Impact":	"Impact-BOOL": "BAD"	Was this residue significantly conserved
	"Impact-NSEQ": "9"	Number of sequences in the alignment In this case treat results with caution as we really want >=10 sequences
	"Impact-THRESHOLD": "0.67"	Threshold (0-1) for significant conservation
	"Impact-CONSSCORE": "1"	Conservation at this position in the alignment
"Interface":	"Interface-BOOL": "BAD"	Was this residue in an interface? Relative accessibility changed by >=10
	"Interface-RELACCESS": "51.582"	Relative accessibility (0-100) of this residue
	"Interface-RELACCESS-MOL": "77.011"	Relative accessibility (0-100) of this residue in a monomer
"Proline":	"Proline-MUTANT-THRESHOLD": "0.53"	
	"Proline-BOOL": "OK"	Was this a mutation to a proline where the phi/psi angles can't accommodate proline
	"Proline-NATIVE-THRESHOLD": ""	
	"Proline-NATIVE-ENERGY": ""	

Continued on next page

Table 1 – *Continued from previous page*

Category		Explanation
	"Proline-PSI": "-46.473"	Backbone Calpha-C angle
	"Proline-MUTANT-ENERGY": ""	
	"Proline-PHI": "-59.091"	Backbone N-Calpha angle
	"Proline-NATIVE-BOOL": "OK"	
	"Proline-MUTANT": "GLY"	
	"Proline-NATIVE": "ARG"	
"SProtFT":	"SProtFT-BOOL": "OK"	Was this residue described in SwissProt as a 'feature' (e.g. active site, PTM, etc)
	"SProtFT-FEATURES": "00000000000000"	Which features were affected
	"SProtFT-NAMES": ""	Which features were affected - as text
"SSGeom":	"SSGeom-BOOL": "OK"	Was this a Cys in a disulphide (from PDB file)
"SurfacePhobic":	"SurfacePhobic-MUTANT-HPHOB": "0.16"	Hydrophobicity of mutant residue
	"SurfacePhobic-BOOL": "OK"	Are we replacing a hydrophilic with a hydrophobic on the surface?
	"SurfacePhobic-RELACCESS": "77.011"	Relative accessibility (0-100)
	"SurfacePhobic-NATIVE-HPHOB": "-1.8"	Hydrophobicity of native residue
"Voids":	"Voids-MUTANT": [11.799,10.381, ...]	Top 10 voids in the mutant
	"Voids-MUTANT-LARGEST": "11.799000"	Largest void in the mutant
	"Voids-BOOL": "OK"	Are we introducing a void > 275Å^3 when there wasn't
	"Voids-NATIVE": [11.799,10.381, ...]	Top 10 voids in the native
	"Voids-NATIVE-LARGEST": "11.799000"	Largest void in the native

[D] MYH7

Table 2: SAAPpred performance 10 Model on (All MYH7 mutation) using one PDB per variance and using multiple PBD

	ALL MY7H [onePDB]							ALL MY7H [MultiPDB]						
	TP	FN	TN	FP	SENS	F1	ACC	TP	FN	TN	FP	SENS	F1	ACC
Run without -norm option for normalization and including PDB ID 3dpt														
SAAPpred M1	216	19	0	0	0.919	0.958	0.919	655	150	0	0	0.814	0.897	0.814
SAAPpred M2	222	13	0	0	0.945	0.972	0.945	589	216	0	0	0.732	0.845	0.732
SAAPpred M3	235	0	0	0	1.000	1.000	1.000	683	122	0	0	0.848	0.918	0.848
SAAPpred M4	230	5	0	0	0.979	0.989	0.979	728	77	0	0	0.904	0.950	0.904
SAAPpred M5	234	1	0	0	0.996	0.998	0.996	626	179	0	0	0.778	0.875	0.778
SAAPpred M6	208	27	0	0	0.885	0.939	0.885	720	85	0	0	0.894	0.944	0.894
SAAPpred M7	231	4	0	0	0.983	0.991	0.983	677	128	0	0	0.841	0.914	0.841
SAAPpred M8	235	0	0	0	1.000	1.000	1.000	731	74	0	0	0.908	0.952	0.908
SAAPpred M9	235	0	0	0	1.000	1.000	1.000	598	207	0	0	0.743	0.852	0.743
SAAPpred M10	235	0	0	0	1.000	1.000	1.000	743	62	0	0	0.923	0.960	0.923
Average					0.971	0.985	0.971	675	130	0	0	0.839	0.911	0.839
Run with -norm option for normalization and excluding PDB ID 3dpt														
SAAPpred M1	212	23	0	0	0.902	0.949	0.902	585	107	0	0	0.845	0.916	0.845
SAAPpred M2	228	7	0	0	0.970	0.985	0.970	591	101	0	0	0.854	0.921	0.854
SAAPpred M3	219	16	0	0	0.932	0.965	0.932	488	204	0	0	0.705	0.827	0.705
SAAPpred M4	221	14	0	0	0.940	0.969	0.940	645	47	0	0	0.932	0.965	0.932
SAAPpred M5	222	13	0	0	0.945	0.972	0.945	598	94	0	0	0.864	0.927	0.864
SAAPpred M6	218	17	0	0	0.928	0.962	0.928	486	206	0	0	0.702	0.825	0.702
SAAPpred M7	229	6	0	0	0.974	0.987	0.974	598	94	0	0	0.864	0.927	0.864
SAAPpred M8	201	34	0	0	0.855	0.922	0.855	459	233	0	0	0.663	0.798	0.663
SAAPpred M9	210	25	0	0	0.894	0.944	0.894	569	123	0	0	0.822	0.902	0.822
SAAPpred M10	219	16	0	0	0.932	0.965	0.932	471	221	0	0	0.681	0.810	0.681
Average					0.927	0.962	0.927	549	143	0	0	0.794	0.882	0.794

Table 3: SAAPpred performance 10 Model on (HCM-MYH7 mutations) using one PDB per variance and using multiple PBD

	MY7H-HCM [onePDB]							MY7H-HCM [MultiPDB]						
	TP	FN	TN	FP	SENS	F1	ACC	TP	FN	TN	FP	SENS	F1	ACC
Run without -norm option for normalization and including PDB ID 3dpt														
SAAPpred M1	170	18	0	0	0.904	0.950	0.904	541	128	0	0	0.809	0.894	0.809
SAAPpred M2	177	11	0	0	0.941	0.970	0.941	487	182	0	0	0.728	0.843	0.728
SAAPpred M3	188	0	0	0	1.000	1.000	1.000	569	100	0	0	0.851	0.919	0.851
SAAPpred M4	183	5	0	0	0.973	0.987	0.973	604	65	0	0	0.903	0.949	0.903
SAAPpred M5	187	1	0	0	0.995	0.997	0.995	522	147	0	0	0.780	0.877	0.780
SAAPpred M6	162	26	0	0	0.862	0.926	0.862	590	79	0	0	0.882	0.937	0.882
SAAPpred M7	185	3	0	0	0.984	0.992	0.984	554	115	0	0	0.828	0.906	0.828
SAAPpred M8	188	0	0	0	1.000	1.000	1.000	610	59	0	0	0.912	0.954	0.912
SAAPpred M9	188	0	0	0	1.000	1.000	1.000	499	170	0	0	0.746	0.854	0.746
SAAPpred M10	188	0	0	0	1.000	1.000	1.000	613	56	0	0	0.916	0.956	0.916
Average					0.966	0.983	0.966	558.9	110.1	0	0	0.836	0.909	0.836
Run with -norm option for normalization and excluding PDB ID 3dpt														
SAAPpred M2	182	6	0	0	0.963	0.981	0.963	488	82	0	0	0.856	0.922	0.856
SAAPpred M2	181	7	0	0	0.963	0.981	0.963	488	82	0	0	0.856	0.922	0.856
SAAPpred M3	173	15	0	0	0.920	0.958	0.920	406	164	0	0	0.712	0.832	0.712
SAAPpred M4	176	12	0	0	0.936	0.967	0.936	530	40	0	0	0.930	0.964	0.930
SAAPpred M5	175	13	0	0	0.931	0.964	0.931	491	79	0	0	0.861	0.926	0.861
SAAPpred M6	171	17	0	0	0.910	0.953	0.910	402	168	0	0	0.705	0.827	0.705
SAAPpred M7	182	6	0	0	0.968	0.984	0.968	493	77	0	0	0.865	0.928	0.865
SAAPpred M8	156	32	0	0	0.830	0.907	0.830	378	192	0	0	0.663	0.797	0.663
SAAPpred M9	165	23	0	0	0.878	0.935	0.878	471	99	0	0	0.826	0.905	0.826
SAAPpred M10	173	15	0	0	0.920	0.958	0.920	391	179	0	0	0.686	0.814	0.686
Average					0.914	0.955	0.914	453.3	116.7	0	0	0.795	0.883	0.795

Table 4: SAAPpred performance 10 Model on (DCM-MYH7 mutations using one PDB per variance and using multiple PBD).

	MY7H-DCM [onePDB]							MY7H-DCM [MultiPDB]						
	TP	FN	TN	FP	SENS	F1	ACC	TP	FN	TN	FP	SENS	F1	ACC
Run without -norm option for normalization and including PDB ID 3dpt														
SAAPpred M1	20	1	0	0	0.952	0.976	0.952	28	32	0	0	0.467	0.636	0.467
SAAPpred M2	20	1	0	0	0.952	0.976	0.952	23	37	0	0	0.383	0.554	0.383
SAAPpred M3	21	0	0	0	1.000	1.000	1.000	44	16	0	0	0.733	0.846	0.733
SAAPpred M4	20	1	0	0	0.952	0.976	0.952	44	16	0	0	0.733	0.846	0.733
SAAPpred M5	21	0	0	0	1.000	1.000	1.000	33	27	0	0	0.550	0.710	0.550
SAAPpred M6	20	1	0	0	0.952	0.976	0.952	34	26	0	0	0.567	0.723	0.567
SAAPpred M7	20	1	0	0	0.952	0.976	0.952	43	17	0	0	0.717	0.835	0.717
SAAPpred M8	20	1	0	0	0.952	0.976	0.952	39	21	0	0	0.650	0.788	0.650
SAAPpred M9	21	0	0	0	1.000	1.000	1.000	22	38	0	0	0.367	0.537	0.367
SAAPpred M10	21	0	0	0	1.000	1.000	1.000	47	13	0	0	0.783	0.879	0.783
Average	20.4	0.6	0	0	0.972	0.986	0.972	35.7	24.3	0	0	0.595	0.736	0.595
Run with -norm option for normalization and excluding PDB ID 3dpt														
SAAPpred M1	20	1	0	0	0.952	0.976	0.952	45	9	0	0	0.833	0.909	0.833
SAAPpred M2	21	0	0	0	1.000	1.000	1.000	46	8	0	0	0.852	0.920	0.852
SAAPpred M3	21	0	0	0	1.000	1.000	1.000	37	17	0	0	0.685	0.813	0.685
SAAPpred M4	21	0	0	0	1.000	1.000	1.000	54	0	0	0	1.000	1.000	1.000
SAAPpred M5	21	0	0	0	1.000	1.000	1.000	47	7	0	0	0.870	0.931	0.870
SAAPpred M6	21	0	0	0	1.000	1.000	1.000	37	17	0	0	0.685	0.813	0.685
SAAPpred M7	21	0	0	0	1.000	1.000	1.000	44	10	0	0	0.815	0.898	0.815
SAAPpred M8	21	0	0	0	1.000	1.000	1.000	37	17	0	0	0.685	0.813	0.685
SAAPpred M9	20	1	0	0	0.952	0.976	0.952	43	11	0	0	0.796	0.887	0.796
SAAPpred M10	21	0	0	0	1.000	1.000	1.000	36	18	0	0	0.667	0.800	0.667
Average					0.991	0.995	0.991	42.6	11.4	0	0	0.789	0.878	0.789

Table 5: SAAPpred performance 10 Model on (Others-MYH7 mutations) using one PDB per variance and using multiple PBD.Others:LVNC, ASD, Endocardial Fibroelastosis, RCM, Myopathycentral Core, Miopatoa Distal De Laing, Ebsein, Mysin Strong Miopathy and Distal Myopathy.

	MY7H-Others [onePDB]							MY7H-Others [MultiPDB]						
	TP	FN	TN	FP	SENS	F1	ACC	TP	FN	TN	FP	SENS	F1	ACC
Run without -norm option for normalization and including PDB ID 3dpt														
SAAPpred M1	26	1	0	0	0.963	0.981	0.963	39	45	0	0	0.464	0.634	0.464
SAAPpred M2	25	2	0	0	0.926	0.962	0.926	41	43	0	0	0.488	0.656	0.488
SAAPpred M3	27	0	0	0	1.000	1.000	1.000	54	30	0	0	0.643	0.783	0.643
SAAPpred M4	27	0	0	0	1.000	1.000	1.000	51	33	0	0	0.607	0.756	0.607
SAAPpred M5	27	0	0	0	1.000	1.000	1.000	45	39	0	0	0.536	0.698	0.536
SAAPpred M6	26	1	0	0	0.963	0.981	0.963	56	28	0	0	0.667	0.800	0.667
SAAPpred M7	26	1	0	0	0.963	0.981	0.963	55	29	0	0	0.655	0.791	0.655
SAAPpred M8	27	0	0	0	1.000	1.000	1.000	56	28	0	0	0.667	0.800	0.667
SAAPpred M9	27	0	0	0	1.000	1.000	1.000	40	44	0	0	0.476	0.645	0.476
SAAPpred M10	27	0	0	0	1.000	1.000	1.000	64	20	0	0	0.762	0.865	0.762
Average	26.5	0.5	0	0	0.982	0.991	0.982	50.1	33.9	0	0	0.597	0.743	0.597
Run with -norm option for normalization and excluding PDB ID 3dpt														
SAAPpred M1	26	1	0	0	0.963	0.981	0.963	63	11	0	0	0.851	0.920	0.851
SAAPpred M2	27	0	0	0	1.000	1.000	1.000	63	11	0	0	0.851	0.920	0.851
SAAPpred M3	26	1	0	0	0.963	0.981	0.963	51	23	0	0	0.689	0.816	0.689
SAAPpred M4	25	2	0	0	0.926	0.962	0.926	67	7	0	0	0.905	0.950	0.905
SAAPpred M5	27	0	0	0	1.000	1.000	1.000	66	8	0	0	0.892	0.943	0.892
SAAPpred M6	26	1	0	0	0.963	0.981	0.963	52	22	0	0	0.703	0.825	0.703
SAAPpred M7	27	0	0	0	1.000	1.000	1.000	67	7	0	0	0.905	0.950	0.905
SAAPpred M8	25	2	0	0	0.926	0.962	0.926	50	24	0	0	0.676	0.806	0.676
SAAPpred M9	26	1	0	0	0.963	0.981	0.963	61	13	0	0	0.824	0.904	0.824
SAAPpred M10	26	1	0	0	0.963	0.981	0.963	50	24	0	0	0.676	0.806	0.676
Average	2 6.1	0.9	0	0	0.967	0.983	0.967	59	15	0	0	0.797	0.884	0.797