UNIVERSITY COLLEGE LONDON

UCL Research Department of Structural and Molecular Biology

Structural analysis of single amino acid polymorphisms

Anja Barešić

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

Declaration

I, Anja Barešić, 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.

Anja Barešić

September, 2011

Abstract

Understanding genetic variation is the basis for prevention and diagnosis of inherited disease. In the 'next generation sequencing' era with rapidly accumulating variation data, the focus has shifted from population-level analyses to individuals. This thesis is centred on the problem of gathering, storing and analysing mutation data to understand and predict the effects single amino acid mutations will have on protein structure and function. I present analysis of a subset of mutations and a new predictive method implemented to expand the coverage of the structural effects by our pipeline.

I characterised a subset of pathogenic mutations: 'compensated pathogenic deviations'. These are mutations which cause disease in humans, but the mutant residues are found as native residues in other species. During evolution, they are presumed to spread through populations by coevolving with another, neutralising mutation. When compared with uncompensated mutations, they often cause milder structural disruptions, prefer less conserved structural environments and are often found on the protein surface.

I describe the development of a new analysis to test the effects of mutations by predicting residues involved in protein-protein interfaces where the structure of the complex is unknown. Two machine learning methods (multilayer perceptrons and, in particular, random forests) show an improvement over previously published protein-protein interface predictors. This new method further increases the ability of the

SAAPdb analysis pipeline to show the effects of mutations on protein structure and function. Furthermore, it is a template for building prediction-based structural analysis methods for the pipeline, where available structural data are insufficient.

In summary this thesis examines mutations from both an evolutionary and a disease perspective. In addition, a novel method for predicting protein interaction regions is developed thus expanding the existing pipeline and furthering our ability to understand mutations and use them in a predictive context.

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
TN	:	True Negative
TP	:	True Positive
TPR	:	True Positive Rate
UniProt	:	UNIversal PROTein resource
UniProtKB	:	UniProt KnowledgeBase

Acknowledgements

I consider myself very fortunate to have been surrounded by numerous amazing people. As the last four years were a much bigger change to me than just furthering my education, this section is on the long side, attempting to address everyone I feel gratitude towards.

First and foremost, I would like to thank my supervisor Andrew Martin, for believing in me, offering support and challenges, and knowing exactly when I needed which. Next, big thanks go to Dr. Irina Tsaneva and Prof. David Jones for providing lots of useful ideas and feedback during my committee meetings. Finally to Dr. Miljenko Huzak and Filippo Leda for advice on statistics used in work on CPDs, and help setting up my first multilayer perceptrons, respectively.

Further, thanks are due to all people from the 636 lab, past and present. You have advised me, listened to me, taught me so much about science, UK and broader stuff. But most of all, thank you for being there on Fridays and making weeks go by with ease and many laughs. And for every PhD student there is that one special friendship, made when you least expect it. Nouf, if I knew how to pray, God would have sent you to me when it was the toughest. This way, I will call it fate.

Over numerous years in education, I have been taught by many. However, only several are remembered for more than their lectures: I have to thank (in chronological order) Renata, Zdravko, and Pavle who had contagious curiosity, love for science, and guided me to where I am today. Without you I would have given up long time ago.

My PhD included leaving home, people I love, and would have been a nightmare if it weren't for the most amazing men of London. Marko Ivin, you know I will never be able to return your kindness in showing me how to feel welcome in this big scary city. Stathis and Lucas, I chose to share a flat with you, but I had no idea you would become my second family. You made sure I didn't fail my extracurricular experience as a postgraduate student, and I admire you for your efforts.

Coming back after years in another scientific community was not easy, and my sanity during the last year of working from Croatia is credited to my morale officers. Mihaela, Hana, Mario and Donatella, your kindness in offering a chair to a homeless scientist will never be forgotten.

The biggest thanks go to my Mum, the most amazing woman ever. I will probably never be able to comprehend the sacrifices you made to make my dreams and ambitions come true, but I am grateful for them more than I can put in words.

And the last on this long list is the one and only (no matter how many similar-named seem to surround me) Marko. I will never forget the comfort, time and money you have given up to support me. I will owe you forever ♡

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Mutations

Mutation (lat. *mutare* = change) in biology denotes any change in the genetic material of an individual, ranging from single nucleotide changes to the gain or loss of entire chromosomes. Mutations are what makes each of us unique in terms of our genetic information. However, what we observe in everyday life, rather than the genetic code itself, is the effect each mutation has on the phenotype of an individual: deleterious mutations cause disease or, in extreme cases, may be lethal; neutral mutations will not display any obvious changes at the level of the phenotype; beneficial ones improve the fitness of an individual[1].

Traditionally, mutations in humans are described using their phenotypic effect and, where available, linking it to the location in the genome. However, in order properly to understand how a genetic modification causes the observed novel phenotype, we need to understand which changes occurred at both the DNA level (change in nucleotide sequence, mRNA stability, gene expression regulation, etc.) and the protein level, i.e. how that mutation affects protein sequence, structure and function. The effect of mutations on the protein structure has been addressed by the SAAPdb project (Hurst *et al.*, 2008): an attempt to gather publicly available mutation data on a regular basis, and automatically assign likely effects of these mutations to the protein structures (SAAPdb is introduced in Section 2.1.7). Before going into more specific details on my contribution to the SAAPdb project, this section introduces the basic biological background on various types of mutations, concluding with a short survey of available mutation effect prediction tools.

[1]this improvement is always measured with respect to present environmental conditions

1.1.1 Deoxyribonucleic acid as the carrier of the genetic information

For the majority of living organisms, the molecule carrying the genetic code is deoxyribonucleic acid, DNA, often termed the 'blueprint of life' (Berg *et al.*, 2006, p. 3). The building block of DNA is a nucleotide, consisting of a monosaccharide (2-deoxyribose), a phosphate and a nucleobase: adenosine (A), cytosine (C), guanine(G) or thymine (T); the structure is presented in Figure 1.1. DNA is structured as a double helix stabilised by the ladder-like stacked hydrogen bonds between adenosine and thymine (creating two hydrogen bonds) or guanine and cytosine (creating three hydrogen bonds). These pairs are referred to as complimentary base pairs.

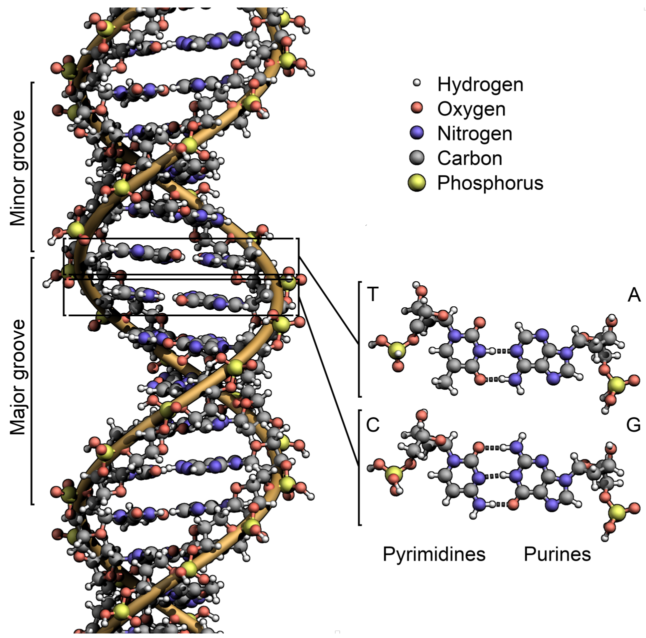

Figure 1.1: Structure of the double-stranded DNA and base pairing schema. *Figure obtained from* `http://en.wikipedia.org/wiki/DNA` *under Creative Commons license.*

1.1.2 Variability in the human genome

The human genome has approximately 3.16 billion base pairs, coding for $20000 - 25000$ proteins (International Human Genome Sequencing Consortium, 2004). Variations among individuals can occur on various scales: from single base pair differences (also termed **point mutations**), insertions/deletions of several-nucleotide-long fragments of DNA, loss of entire genes, to large-scale modifications, such as the loss or rearrangement of entire chromosomes. On average, two humans differ in ~ 3.5 million base pairs and ~ 61000 small insertions/deletions, as well as ~ 6000 copy number variations[2] (Pelak *et al.*, 2010; Moore *et al.*, 2011).

Considering where mutations can appear in terms of cell types, somatic and germline mutations are differentiated. **Germline** mutations occur in the germ cells, a cell lineage ultimately producing mature gametes (i.e. sperm and egg cells) in animals. A mutation anywhere in that lineage will result in the transfer of the mutated genotype to the offspring. At the same time, this genetic change will not affect the parent's phenotype. However, in the next generation it will become incorporated in all cells of the offspring. In contrast, **somatic** mutations occur in any cells that do not belong to the germ cell lineage. These mutations are not passed down to the offspring, however the mutation can spread through the organism in which it has appeared (displaying selection on the level of cells within a single individual), provided the mutation-containing cell is undergoing duplication. Thus the somatic cell and all cells originating from it, will display the altered phenotype, provided there is one.

There is a whole range of effects these genetic aberrations can have on the phenotype of an individual. Ideally, fitness effects should be measured on a continuous scale. However in practice mutations are usually classified into three groups, based on the fitness change: (i) **beneficial** mutations increase the fitness of an individual, (ii) **neutral** ones lack a visible effect on the fitness,

[2] the latter was claimed by the authors to be an overestimate, but no similar studies are currently available with a more reliable estimate

and (iii) **deleterious** mutations (also termed pathogenic) lower the overall fitness.

Not all changes are inherited in a Mendelian manner. So-called 'high-penetrance' mutations will cause a visible phenotypic change (irrespective of the magnitude of this change), and these variations can be discussed in terms of Mendelian inheritance. On the other hand, sometimes the same change in the genomic sequence will cause altered phenotype in some individuals, and no visible change in others – these are termed 'low-penetrance' mutations, and they form a continuum between phenotypically silent and high penetrance disease-associated mutations. Often low-penetrance phenotypes are expressed as a result of interactions between two or more mutations, or as a result of interactions with environmental factors.

Biomedical research (and incentives originating from the pharmaceutical industry) predominantly revolves around human disease and how to improve quality of life. Reflecting this, most of the studies available on human genetic variations present data on pathogenic and neutral mutations (rarely presenting beneficial mutations); the most relevant ones will be introduced in Section 1.1.4. This thesis builds on publicly available mutation data sources, and as such will mostly discuss changes in humans resulting in disease-associated phenotypes, or not affecting the phenotype at all.

1.1.2.1 A single nucleotide polymorphism: the simplest mutation

Strictly a single nucleotide polymorphism (SNP) is defined as an allelic variant where the least frequent allele occurs in at least 1% of a 'normal' population (The International Hapmap Consortium, 2005). In other words it is a beneficial, neutral, or low-penetrance mutation. However, the term is widely used simply to mean any single-base nucleotide substitution. Indeed, the main repository of point mutations in human and other genomes (dbSNP) refers to SNPs in this context.

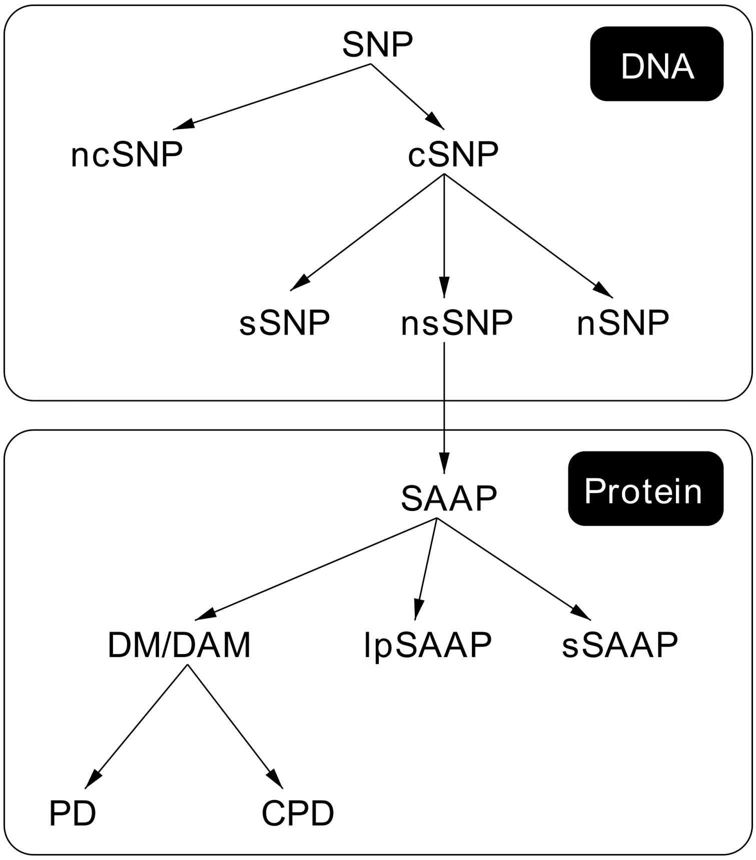

Figure 1.2: Hierarchy of SNP mutations and their effects.
SNP can be non-coding (ncSNP) or coding (cSNP). cSNPs can be synonymous (sSNPs), nonsense (nSNPs), or non-synonymous (nsSNPs). nsSNPs result in a single amino acid polymorphism (SAAP) at the protein level. These can be phenotypically silent (sSAAP), low penetrance (lpSAAP), or high penetrance deleterious mutations (DMs) also known as disease-associated mutations (DAMs). A DAM can be compensated in another species (a compensated pathogenic deviation, CPD) or un-compensated (a pathogenic deviation, PD). *Figure and (adjusted) caption text obtained from* Barešić and Martin (2011).

Various outcomes of this simplest mutation event are presented in Figure 1.2. When a SNP occurs in non-coding regions of the DNA (**ncSNP**), although it will not directly affect the sequence of protein during translation, it can still affect regulatory regions (i.e. transcription factor binding sites) thus affecting expression, or may change mRNA splice sites. A SNP within the coding exons (**cSNP**), can have three outcomes regarding the protein sequence: (i) a synonymous SNP (**sSNP**) is a mutation where the modified codon in which the mutated nucleotide occurs encodes for the same amino acid as the native codon, e.g. TTG→CTG will not change the protein sequence as both codons code for leucine (see Figure 1.3). While an sSNP will not change the encoded protein sequence, it may still affect expression or splicing. (ii) A nonsense SNP (**nSNP**) is a change from an amino-acid-producing codon to a stop codon. This results in premature termination of translation yielding a truncated, often non-functional protein product. (iii) A non-synonymous SNP (**nsSNP**) is a change where the mutated codon is translated to a different amino acid type compared with the native codon. This results in a single amino acid change (also termed a 'single amino acid polymorphism', SAAP) in the protein. For example, TTG→TTC will change a native leucine to a mutant phenylalanine (Figure 1.3), potentially affecting protein folding, stability, or function.

The term 'SNP' is also often used to refer to a SAAP with no phenotypic effect (or with low penetrance), in contrast to a Mendelianly inherited deleterious 'Disease Associated Mutation' (DAM). Types of SAAPs are introduced in Section 1.1.3. Unless otherwise stated, the term SNP will hereafter correspond to a missense mutation (at the DNA or protein level) lacking a documented pathogenic effect, thus presumed to be phenotypically neutral, or having very low penetrance.

1.1.3 Point mutations at the protein level

This thesis analyses only nsSNPs; to be more precise, the focus is on SAAPs as the project revolves around protein structures and the effects of mutations at the

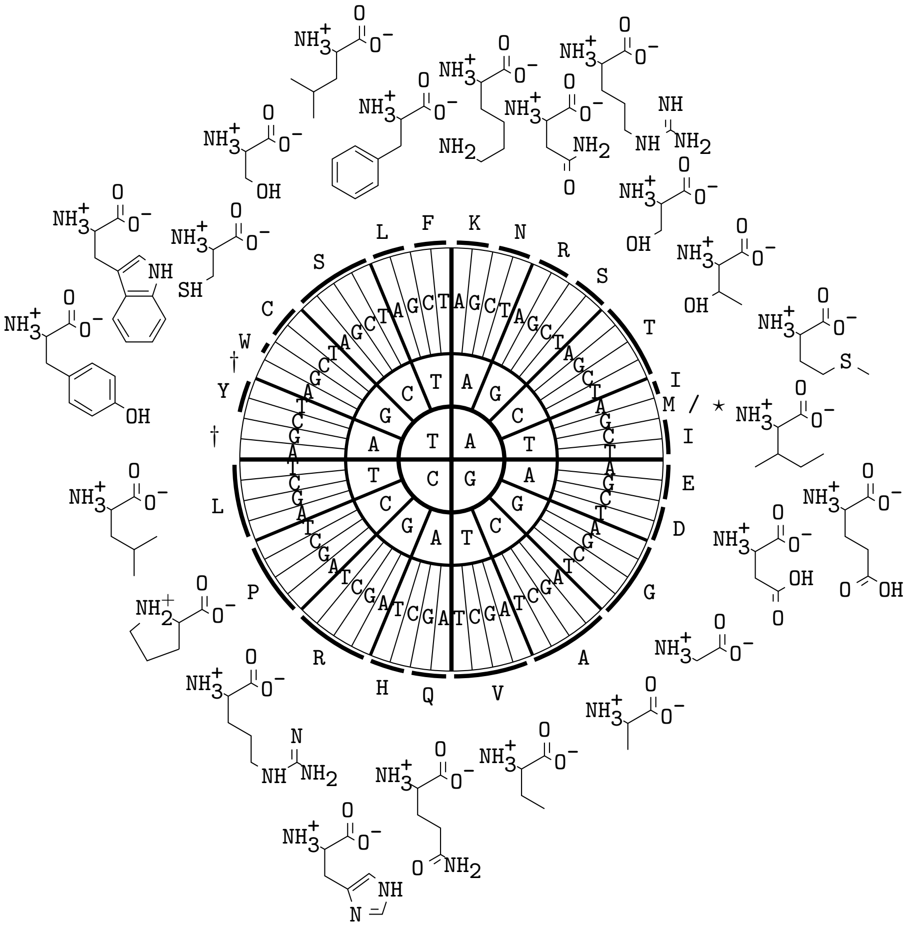

Figure 1.3: DNA codon table (read from the center outwards).
∗ stands for the start codon, and † denote three stop codons. *Figure (originally with RNA codons) created by Florian Hollandt and obtained from* `http://www.texample.net/tikz/examples/rna-codons-table/`.

structural level, presented in the lower rectangle of Figure 1.2. As mentioned above a SAAP is the protein manifestation of an nsSNP, also termed a missense mutation. Hereafter, a SAAP will be defined as any single amino acid change using a unique combination of four values as shown in Figure 1.4. For example, consider the OMIM:107300.0031 entry[3]: a mutation of wild-type serine, the 323^{rd} residue in the sequence of the human antithrombin-III protein (UniProtKB/Swiss-Prot accession number `P01008`), to proline caused by a single base change of TCN→CCN (where N stands for any nucleotide type). This is defined as (P01008:S:323:P).

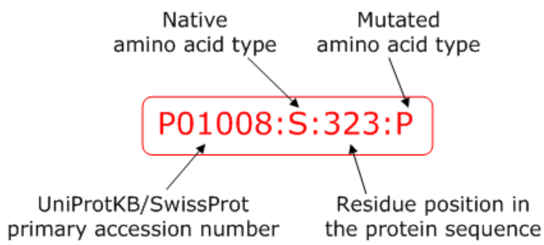

Figure 1.4: An example of a single amino acid polymorphism annotation. Each mutation is a unique combination of: (i) UniProtKB/Swiss-Prot primary accession number of a protein in which the mutation occurs (for definition, see Section 2.1.3), (ii) the amino acid type found native in disease-unaffected 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.

In practice, publicly available data consist of (i) neutral mutations (**sSAAPs** from Figure 1.2, henceforth termed SNPs), deposited for example in dbSNP (Sherry *et al.*, 2001); mildly pathogenic and beneficial ones (**lpSAAPs** found also in dbSNP, often first listed as neutral and then corrected), and disease-associated SAAPs (high-penetrance deleterious mutations, termed **DAMs**, or **DMs**), stored in OMIM (Amberger *et al.*, 2009) and locus-specific mutation databases (LSMDBs). For a list of LSMDBs used in SAAPdb, see Table 2.1.

In simplified terms, for a protein to keep its fitness constant, it needs to maintain functional and structural integrity. A function-affecting SAAP may cause (i) a loss-of-function, or (ii) a gain-of-function. Loss-of-function mutations are often inherited in recessive way: if the individual is heterozygous, he/she will display reduced function,

[3]`http://omim.org/entry/107300`

while the homozygous individuals carrying two copies of the mutation fully lack the functionality performed by that protein. Three examples of loss-of-function SAAPs will be introduced in more detail in Section 3.1.2. In contrast, a gain-of-function mutation is inherited in a dominant manner: the mutated individual displays a novel (often detrimental) phenotype. In case of the Lys183→Arg mutation found in human thyrotropin receptor results in its hypersensitivity to human chorionic gonadotropin, resulting in hereditary gestational hyperthyroidism (Rodien *et al.*, 1998).

Proteins consist of 20 standard amino acids. When compared with the choice of only 4 nucleotides, it is obvious that the protein world offers a significantly greater variety of possible structural combinations. In that respect, identifying the exact structural change caused by a SAAP, and linking it to the observed phenotype is a challenge, addressed by many groups (for more details on available pathogenicity-predicting tools, see Section 1.2.3). Structural features of a protein can be summarised in its ability to **fold** properly into an active form, to stay in that form (its **stability**) and its ability to perform some **function**. Several folding- and stability-impairing SAAP types will be introduced in Section 2.1.7.

In short, SAAPs preventing correct folding, result in an unstructured protein susceptible to protein degradation. Furthermore, most proteins have a surprisingly narrow range of thermodynamic stability between -3 and -10 kcal/mol (DePristo *et al.*, 2005). Should a SAAP cause a drop in stability below these values, the protein will unfold; while an increase in stability usually reduces the responsiveness of the protein to cell signalling and the loss of activity (DePristo *et al.*, 2005). Additionally, often the change of stability results in increased propensity for protein aggregation. Considering that each SAAP leads to a $\Delta\Delta G$ of $0.5 - 5$ kcal/mol, it is not surprising that the majority of SAAPs destabilise the protein and are thus pathogenic.

1.1.4 Cataloguing human mutation data

With the rapid expansion of known genomic sequence space as a result of improved sequencing technologies, the last decade has seen as emergence of numerous databases containing information on human variations, accumulating various types of (mostly) single nucleotide or amino acid variations, and providing additional structural, evolutionary or functional context.

The first major attempt to gather human SNP data (defined as genetic variation occurring in at least 1% of human population) and identify patterns of commonly co-inherited mutations (termed haplotypes) was started in 2002 by research groups in six countries forming the International HapMap Consortium. The **International HapMap Project**[4] (HapMap stands for 'haplotype mapping') focused on identifying novel SNPs, their frequencies and correlations, grouping correlated mutations into haplotypes, and identifying which SNPs could be used for haplotype identification (The International Hapmap Consortium, 2003). Ultimately these haplotypes would be tested for correlation with the common diseases, with the aim of identifying combinations of low-penetrance SNPs (in combination with environmental factors), causative of and/or indicative of complex diseases in humans. Phase three of this project, finished in 2010, has surveyed 1.6 million common SNPs (deposited in dbSNP, a resource to be introduced in Section 2.1.6.2) in genomes obtained from 1184 individuals from 11 different populations (The International Hapmap 3 Consortium, 2010).

The central project developing methodology, establishing strategies and standards for the comparison of full genomic sequences from various humans and finally, providing an exhaustive list of human variations to the public, is the **1000 Genomes Project**[5]. Taking advantage of a drop in prices (and time requirement) for sequencing after the introduction of next-generation sequencers in 2005 (Metzker, 2010), this project aims

[4]`http://hapmap.ncbi.nlm.nih.gov/`
[5]`http://www.1000genomes.org/`

to provide 95% of SNPs occurring in humans (defined as single nucleotide variations with a minimum of 1% frequency in normal population).

In the pilot phase of the project (The 1000 Genomes Project Consortium, 2010), three sub-projects were performed: (i) low-coverage[6] whole-genome sequencing of 179 individuals from four populations (assessing how to compare genomes of different individuals) identified 14.4 million SNPs, 1.3 million short indels and 20000 structural variants, (ii) deep-sequencing of two mother-father-daughter trios using different research facilities and sequencing platforms (tests comparability of different genomic sequence sources) yielded 5.9 million reported SNPs, 650000 short indels and 14000 structural variants, and (iii) exon-sequencing of 697 individuals from seven locations, covering exons of 907 randomly chosen genes. Finally the goal of this project is to provide an extensive dataset of human SNPs by sequencing 2500 human genomes from 25 populations in total at 4x coverage, aiming to have finished by the end of 2011. For more details on databases of mutations, see Sections 1.2.3 and 2.1.6.

1.2 Protein structure

1.2.1 The four levels of protein structure

Protein structure is defined on four levels (Berg *et al.*, 2006), presented in Figure 1.5.

Primary protein structure will hereafter be termed protein sequence, and will be treated as a string with each letter coming from an alphabet of 20 standard amino acids. The secondary structure will be considered as either one of its two major structured components (the α-helix and the β-strand), or as a third option a non-α, non-β alternative termed ‘coil’. The key features of tertiary structure – its

[6]coverage refers to the average number of times each sequence position gets sequenced, e.g. 20x coverage means each residue is present in approximately 20 sequenced segments

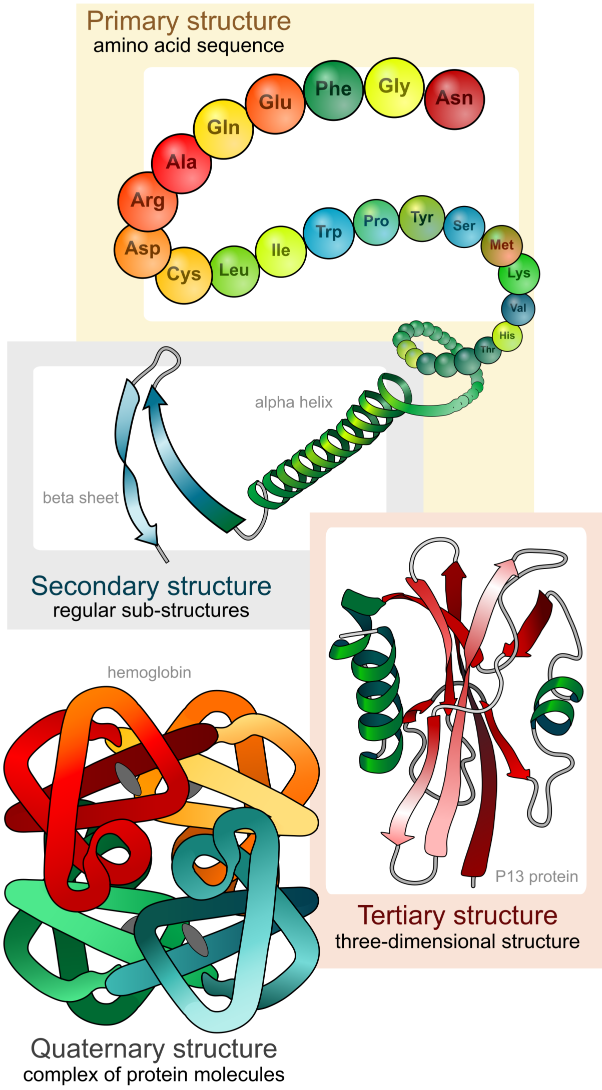

Figure 1.5: The protein structure hierarchy.
Figure obtained from `http://en.wikipedia.org/wiki/File:Main_protein_structure_levels_en.svg` *under Creative Commons license.*

stabilising elements (hydrogen bonds, pairing of charged residues, electrostatic interactions and disulphide bonds and its core-and-surface topology) will be introduced in Section 2.1.7. Most attention will be allocated to the quaternary structure: in particular the difference between the minimal repeating unit in the crystallised protein termed the 'asymmetric unit', and the molecular assembly observed *in vivo*, called the 'biological unit'.

Quaternary structure refers to the number and orientation of chains in the biological unit. If there is only one chain in the biological unit, the protein is a monomer, otherwise we refer to it as a protein complex. If a complex contains all identical chains, it is termed a homomeric complex, whereas if it consists of different protein chains it is referred to as a heteromeric complex. Further, based on the number of chains in the biological unit, a complex can be a dimer, a trimer, a tetramer, etc., up to multimeric structures with several dozens of chains (usually viral capsids). Figure 1.6 presents examples of several types of quaternary structures.

1.2.2 Obtaining three-dimensional structures of proteins

1.2.2.1 X-ray crystallography

X-ray crystallography is the predominant method for obtaining high quality three-dimensional structures of proteins. The idea of purifying proteins into crystals in order to observe diffraction patterns of X-rays after passing through these crystals originated at the beginning of the last century (Bragg, 1913). The principle behind this molecular imaging technique, shown in Figure 1.7, is simple: electromagnetic waves, once emitted onto the sample, get diffracted in a specific manner dependent on the positioning of electrons in a molecule[7]. The results of diffraction are then recorded and after applying Fourier transformations to these data, the electron

[7]only proteins are considered here as targets for structure elucidation, although this method is successfully used for smaller compounds, and other macromolecules (fragments of DNA, whole viruses, etc.)

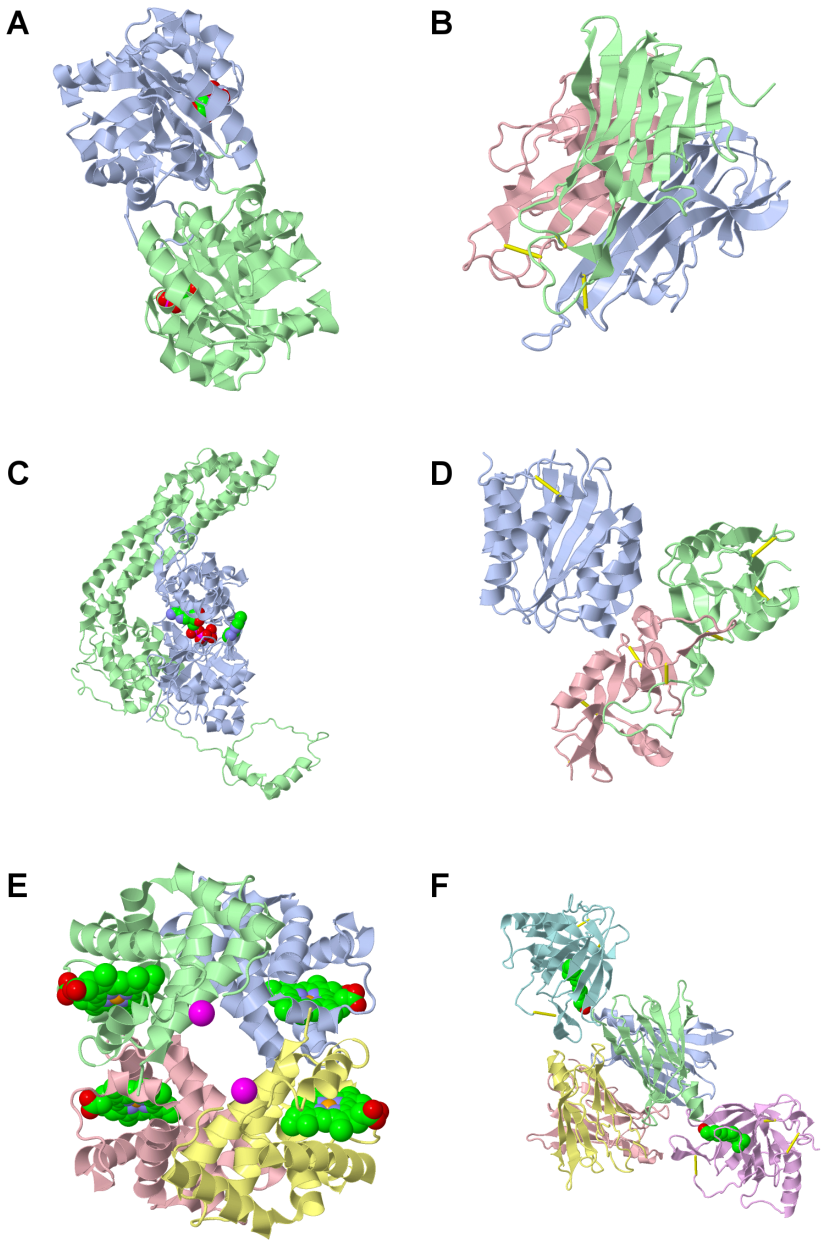

Figure 1.6: Examples of biological assemblies of proteins, one colour used for each chain. (**A**) A homodimer: triosephosphate isomerase from *S. cerevisiae* (PDB ID: 2YPI). (**B**) A homotrimer: human tumour necrosis factor-alpha (PDB ID: 1TNF). (**C**) A heterodimer: Bni1p Formin Homology 2 Domain from *S. cerevisiae* in a complex with ATP-actin from *O. cuniculus* (PDB ID: 1Y64). (**D**) A heterotrimer: human von Willebrand factor in a complex with botrocetin from *B. jararaca* (PDB ID: 1IJK). (**E**) A tetramer: human deoxyhaemoglobin (2 α and 2 β chains) (PDB ID: 2HHB). (**F**) A hexamer: human transthyretin and retinol-binding protein from *G. gallus* (PDB ID: 1RLB). *All images were obtained from* `http://proteopedia.org/wiki/index.php/Main_Page`.

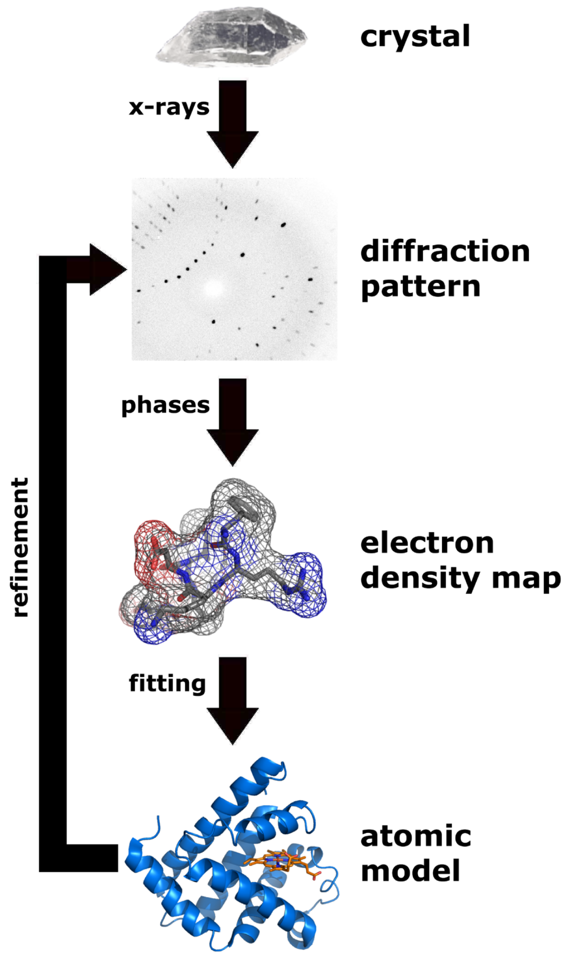

Figure 1.7: X-ray crystallography methodology schema.
Figure obtained from `http://en.wikipedia.org/wiki/File:X_ray_diffraction.png` *under Creative Commons licence.*

density maps of the crystallised molecule are obtained. However, phase information is lost in the collection of the diffraction pattern as only intensities are recorded, creating the so-called ‘phase problem’, which is addressed below. Modelling using chemical restrictions (the experimenter has some prior knowledge about the investigated molecule, e.g. the primary structure of the protein) will then result in obtaining the three-dimensional layout of the atoms from the electron density map.

The procedure for obtaining the protein structure using X-ray crystallography can be divided into three steps:

Obtaining the crystal is necessary because the diffraction signal of a single molecule is very weak: periodically repeating structures will yield signal amplification by wave interactions sufficient for detection above the noise level. This step is still considered to be the main limiting factor in the success of crystallography. The general rule is that the required sample crystal has to have a minimum size of 0.1 mm in all three dimensions, with a minimal amount of chemical impurities[8] and structural anomalies. For more details on theoretical minimal crystal size, and discussion in the context of other technical parameters see Holton and Frankel (2010) and references therein.

Recording diffraction outputs while varying the angle between the X-ray and the sample (typically over the range of slightly more than 180°) results in a two-dimensional image of dots representing diffraction maxima for each measured angle. Each diffraction maximum is a result of interacting waves in the same phase, so the intensity of the dot is a function of the wave’s amplitude. Waves of opposite phase cancel each other out, and this signal is lost.

Processing diffraction patterns includes Fourier transformation and solving the phase problem, followed by fitting. Fourier transformations are used to obtain electron density maps from diffraction patterns, provided the amplitude and the

[8] desired sample purity is 90%, ideally over 98%

phase of the wave are known. While the amplitude is obtained from the dot intensity, calculating the phase is not a trivial procedure. The 'phase problem' is solved by methods such as multiple isomorphous replacement, molecular replacement, or multi-wavelength anomalous dispersion to recover the phases (Hendrickson and Ogata, 1997, and references therein). The vast majority of proteins will not be fully rigid in the crystal: there will be flexible regions resulting in small differences between repeating units within the crystal. These inconsistencies ultimately result in noise in the diffraction patterns, and the experimenter usually fits a model by adding information on the protein sequence or other structural restraints in order to obtain the missing structural data.

One limitation of diffraction is resolution: the minimal separation observable from the diffraction pattern is dependent on (and approximately equal to) the wavelength of the electromagnetic source used for the diffraction. X-rays are used because their wavelengths are in the range of $0.5 - 4$Å, similar to the distances observed when studying covalent bonds and atomic radii. Another indication of the quality of the structure is the R-factor. It measures how different the theoretical diffraction pattern of the modelled structure is from the pattern obtained experimentally. The R-factor can range from 0.00 (perfect match) to 0.63 (set of random atoms). The general threshold for a reliable structure is $R < 0.20$ (Morris *et al.*, 1992). Evaluating modelled structures by the R-factor may result in overfitting, so a crystal structure has another value associated with it, R_{free} (Brünger, 1992); in each refinement step, 90% of the atomic model is used for the improvement, and the remaining 10% are used to evaluate the improvement in this step. The advantages of R_{free} over R have been discussed in detail by Kleywegt and Jones (1997).

Once a novel protein structure (or an improvement over a previously submitted structure in terms of resolution or R-factor) is obtained, it is submitted to a central repository: the Protein Data Bank (Berman *et al.*, 2000), which will be introduced in Section 2.1.1. Typically, an entry refers to a single crystallographic experiment and it contains details of the experimental conditions, modelling procedures, information about the protein sequence, chains and their length, ligands and finally, the coordinates of the atoms.

It is worth noting that crystal packing is not a spontaneous event for most proteins (under physiological conditions), and the assemblies reported in the PDB do not always present chain number, orientation and interchain contacts that reflect *in vivo* structures. Therefore, the so called **asymmetric unit**, observed in the crystal prepared for the structure-solving experiment does not necessarily reflect the **biological unit** (the biologically-active form of the protein also termed the quaternary structure, see Section 1.2.1). This distinction should be considered whenever PDB data are used as models for the biologically-active conformations, as further discussed in the context of protein-protein interfaces in Section 4.2.1.1.

1.2.2.2 Nuclear magnetic resonance spectroscopy

The main alternative method to X-ray crystallography used to determine the structural details of proteins is **nuclear magnetic resonance** spectroscopy, NMR. In short, NMR relies on the different response of atoms in a sample (based on the spin properties of atomic nuclei) to exposure to a magnetic field (Rabi *et al.*, 1938). The sample in a solution is exposed to radio-frequency pulses: magnetic nuclei absorb some of the energy from the pulse and start resonating at a specific frequency. The radio waves which they give off as they relax reveal information about the chemical environment surrounding that nucleus.

These two methods of elucidating protein structures have significantly different approaches, methodological advantages and pitfalls, and thus are often used in combination for the full picture of the three-dimensional structure. NMR gives a better picture of the molecular dynamics, as the crystallographic sample is immobilized in the crystal lattice, occasionally not in the native state of the examined protein. On the other hand, NMR can only process relatively small, water-soluble compounds, whereas X-ray crystallography can solve any structure, provided a crystal of the appropriate size and purity can be obtained. Discussion of the advantages of one method over the other (mostly in terms of validity as templates for computational protein design) is further covered by Schneider *et al.* (2009), and references therein. On average, available crystal structures provide structural information in higher resolution, a single structure per file, and because of the inferred information in NMR structure files, this experimental method is unsuitable for determination of hydrogen bonds and other detailed information. Therefore, only structures obtained by X-ray crystallography are considered hereafter in this thesis.

1.2.3 Effects of mutations on protein structure

After mutation data became abundant in public databases, several groups turned to developing tools that would automatically calculate properties of these mutations, in particular, trying to predict the effect the mutation has on the phenotype. Databases of pathogenic and neutral mutations will be described in Section 2.1.6.1 and Section 2.1.6.2, respectively. If information on the phenotypic effect is unknown (owing to the lack of experimental or clinical data) or not specified, several tools set out to predict whether a mutation is detrimental or not; these predictors are briefly introduced in Section 1.2.3.1. However, in order for the prediction to be successful, information had to be accumulated on the features distinguishing deleterious from neutral mutations; several tools and algorithms that provide large-scale analyses of SNPs (nsSNPs or SAAPs, in particular) are introduced in Table 1.1.

TopoSNP (Stitziel *et al.*, 2004) and ModSNP (Yip *et al.*, 2004) focused primarily on positioning mutations onto the, usually native, protein structure, where one was available (in case of TopoSNP), or on a modelled structure (ModSNP). It is worth mentioning here that any properties obtained from modelled protein structures are not experimentally-verified, and should be avoided when structural properties of SAAPs are used for pathogenicity prediction.

Table 1.1: Structural features of single amino acid polymorphisms – a literature survey.
Web-based resources providing added structural information for datasets of SAAPs. Mutation types refer to the categories of mutations, as provided by the given resource. The last column indicates whether the structure with (usually modelled) mutated amino acid type is provided by the resource.

Method	URL	Mutation types	Modelled[a]	Features analysed	Visualisation
TopoSNP (Stitziel *et al.*, 2004)	`http://gila.bioengr.uic.edu/snp/toposnp`	SNPs, DAMs	No	Sequence conservation, geometry (core or surface), relative entropy of substitution	Yes
ModSNP (Yip *et al.*, 2004)	through Swiss-Prot web pages[b]	UniProtKB/Swiss-Prot variants	Yes	BLOSUM score, cross-references to sequence and structural information	No
SNPeffect (Reumers *et al.*, 2006)	`http://snpeffect.vib.be`	SNPs, DAMs, unclassified	Yes	Aggregation and amyloid-forming zones, phosphorylation, glycosylation, cellular localisation, transmembrane regions, change in free energy, functional sites, secondary structure, solvent accessibility	Yes
stSNP (Uzun *et al.*, 2007)	`http://glinka.bio.neu.edu/StSNP/`	SAAPs	Yes	Metabolic pathway from KEGG (Kanehisa, 1997)	Yes
SAAPdb (Hurst *et al.*, 2008)	`http://www.bioinf.org.uk/saap/db/`	SNPs, DAMs	No	Protein-protein interfaces, binding sites, sequence conservation, function-, folding- and stability-impairing positions	Yes

[a] if the structure of the protein is not available, is the structure obtained through homology modelling?
[b] lists information on pathogenicity, provided it exists in the Swiss-Prot entry

StSNP (Uzun *et al.*, 2007) and SNPeffect (Reumers *et al.*, 2006) use structural information to calculate various properties of mutation's position in the structure (e.g. transmembrane regions, phosphorylation/glycosylation site, putative aggregation sites, etc.), or to characterise the nature of the substitution (e.g. secondary structure-disrupting, introducing changes in solvent-accessibility, etc.).

Finally, a web-based tool currently providing the widest range of structural information for a nsSNP (both pathogenic and neutral) is **SAAPdb** (Hurst *et al.*, 2008), described in detail in Section 2.1.7. In short, it maps a mutation to PDB structure(s) where available, and describes sequence conservation and a range of folding-, function-, stability- and interface-impairing effects that the mutation may have on the protein structure. On a dataset of 4319 DAMs and 2022 SNPs successfully mapped to a position in at least one protein structure, Figure 1.8 displays the distribution of pathogenic versus neutral SAAPs in terms of their effects on the protein structure. In this study, Hurst *et al.* (2008) showed that disease-associated SAAPs display more radical changes (in terms of volume change, native and mutated amino acid similarity), more often affect conserved residues, and are preferentially found in the protein core when compared with the neutral SAAPs.

1.2.3.1 Mutation pathogenicity prediction: currently available tools

Clearly, the distinction between neutral and disease-related variations is interesting as a diagnostic and preventive tool, and costly wet-lab studies can slowly be replaced by *in silico* predictive tools.

In general, the predictive modelling process starts with gathering a list of mutations with known pathogenicity level, i.e. a training dataset, and a list of features, presumably correlated with the level of pathogenicity. Some machine learning method is then applied to build a classifier identifying the combination of feature values

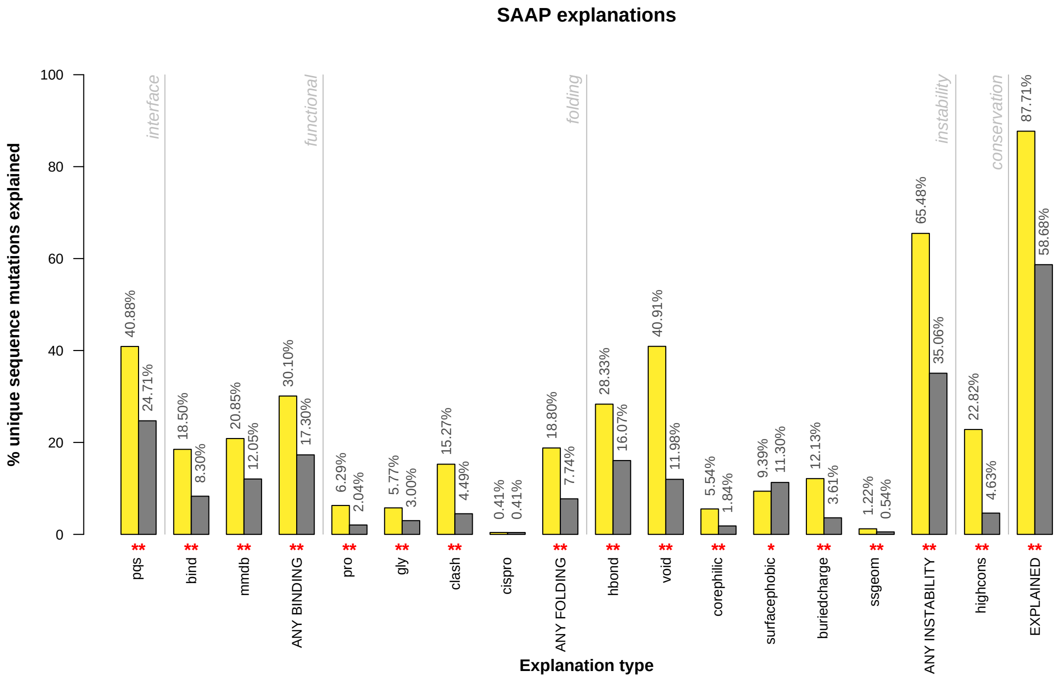

Figure 1.8: Structural effects of neutral and pathogenic SAAPs.
The results for DAMs and SNPs are shown in yellow and grey, respectively. * denotes $p < 0.05$ and ** denotes $p < 0.01$, when χ^2 test with Yates correction is used to test different frequencies per category, in two mutation datasets. For the category definitions, check Table 2.2, or the original publication. *Figure obtained from* Hurst *et al.* (2008)*, with permission from the authors.*

which indicate 'neutral' or 'disease-associated' outcome for a mutation. Choosing the appropriate training dataset is tricky: nsSNPs are abundant, but defined (in the case of the dbSNP) only as a less frequent variant of the wild-type allele. In other words, there is no guarantee that every nsSNP is phenotypically neutral: some can be mildly detrimental (or beneficial) or of low penetrance, but this effect has not yet been documented. In fact, some SNPs are initially reported as neutral variations, only to appear later in OMIM; for an example see Hurst *et al.* (2008). In addition to OMIM, DAMs are stored in various small LSMDBs in different formats, and while these mutations are clearly pathogenic, getting sufficiently large (and representative) datasets requires a lot of data mining and processing.

A list of the most successful publicly-available pathogenicity predictors is presented in Table 1.2. A wide range of machine learning methods have been utilised: SVMs (LS-SNP, SNPs3D, HybridMeth), neural networks (PMUT, SNAP), profile-based scoring function (SIFT), random forest (nsSNP Analyzer), naive Bayesian classifier (PolyPhen); it seems all these methods are equally appropriate for the task of SNP pathogenicity prediction.

In terms of predictors, it is tempting to use only sequence-based information since protein sequences are more abundant than structures and functional annotations. Indeed several quite successful predictors have been built exclusively based on sequence data: e.g. PolyPhen (Ramensky *et al.*, 2002), SIFT (Ng and Henikoff, 2003), one implementation of PMUT (Ferrer-Costa *et al.*, 2005) and SNPs3D (Yue *et al.*, 2006), followed by HybridMeth (Capriotti *et al.*, 2006) and SNAP (Bromberg and Rost, 2007). However, improved performance is obtained by adding structural information such as solvent-accessibility and secondary structure elements (nsSNP Analyzer (Bao *et al.*, 2005)), thermodynamic stability (SNPs3D (Yue *et al.*, 2006)), and functional information, e.g. the pathways in which a protein is involved (LS-SNP (Karchin *et al.*, 2005)). Both Hybrid and SNAP showed improved accuracy and robustness once structural information was added. Thusberg *et al.* (2010) have shown that although the above mentioned methods all show reasonable accuracy, the outputs of different

Table 1.2: Tools predicting pathogenicity of SAAPs – a literature survey.
Web-based resources predicting whether mutation is neutral or pathogenic. For definition of performance measures, see Section 5.1.1.4. Performance is reported as in the original publications, owing to different evaluation methodology.

Method	URL	Model type	Predictors	Performance
SIFT (Ng and Henikoff, 2003)	`http://sift.bii.a-star.edu.sg/`	Scoring function	Sequence conservation	69% accuracy for DAMs, 20% accuracy for SNPs
SNPs3D (Yue *et al.*, 2006)	`http://www.snps3d.org/`	Support vector machines	Sequence conservation	FPR 10%, FNR 20%
			Sequence conservation, solvent accessibility, hydrophobicity, electrostatic interactions, atom packing	FPR 15%, FNR 26%
PMUT (Ferrer-Costa *et al.*, 2005)	`http://mmb2.pcb.ub.es:8080/PMut/`	Neural network	Sequence-based	84% success rate (accuracy), 67% improvement over random model
			Sequence-based, solvent-accessibility, predicted secondary structure	87% success rate, 73% improvement over random model
nsSNP Analyzer (Bao *et al.*, 2005)	`http://snpanalyzer.utmem.edu/`	Random forest	Solvent accessibility, environmental polarity, secondary structure, probability of substitution, native-mutant amino acid similarity	FPR 38%, FNR 21%
MAPP (Stone and Sidow, 2005)	`http://mendel.stanford.edu/SidowLab/downloads/MAPP/`	Standard likelihood analysis	Alignment-based deviation from profile hydropathy, polarity, charge, free energy in α and β conformations and volume	Accuracy 64-80% on 4 mutagenesis datasets, 63-77% accuracy for DAMs only
PolyPhen-2 (Adzhubei *et al.*, 2010)	`http://genetics.bwh.harvard.edu/pph2/`	Naive Bayes	8 sequence-based, 3 structure-based properties	At FPR 20%, TPR 92% and 73% for HumDiv and HumVar datasets, respectively
HybridMeth (Capriotti *et al.*, 2006)	`http://gpcr.biocomp.unibo.it/cgi/predictors/PhD-SNP/PhD-SNP.cgi`	Support vector machines	Sequence-based, or profile-based if available	Overall accuracy 74%, 80% for DAMs and 65% for SNPs
SNAP (Bromberg and Rost, 2007)	`http://www.rostlab.org/services/SNAP`	Neural network	Sequence-based, structural and functional features	78% overall accuracy, 77% for DAMs and 80% for SNPs

methods are not highly correlated, indicating that the attribute space has not been sufficiently exploited, and there is still room for significant improvement in the field of automated SNP pathogenicity prediction.

Furthermore, the aforementioned methods have focused so far on the prediction of DAMs; the more interesting (and less trivial) problem to solve would be developing a successful predictor of low penetrance mutations as many of the diseases unexplained at present are caused by complex interactions between several low penetrance mutations. Finally, these advances (in both DAM and lpSAAP prediction) would result in classifiers precise enough to be used in clinical practice. At the moment, this practice is not recommended (Kumar *et al.*, 2009), rather, these tools should be considered an experimental aid, i.e. to narrow down a list of potential disease-associated mutation candidates. Nonetheless, with PolyPhen, SIFT is regularly used in clinical diagnostics laboratories (Nick Lench, personal communication).

1.3 A list of aims

This thesis sets out to broaden the knowledge on single amino acid polymorphisms and their structural features. In general, the work performed by Hurst *et al.* (2008) is expanded in two directions: an analysis of a specific subset of SAAPs, followed by the introduction of `predicted interface` structural effect, a novel category to be added to the SAAPdb pipeline. More precisely, the work on **compensated pathogenic deviations** in Chapter 3 analyses an evolutionary interesting subset of DAMs, comparing them to ‘background’ uncompensated DAMs. Next, analysis of features in Chapter 4 and consequently prediction of the **protein-protein interface residues** in Chapter 5 is a methodological refinement for SAAPdb: it will expand the space of likely structural effects one can detect for a SAAP to include protein-protein interfaces in multichain complexes, when the structure of the complex is not solved.

Chapter 2

An Introduction to Tools and Resources

This chapter presents resources, tools and algorithms used in the following chapters. Finally it provides an overview of statistical tests used throughout this thesis.

2.1 Data resources

This section covers basic databases and web-resources used in this thesis. It starts with resources of protein structural data: the PDB and PQS (and an alternative format for these data – XMAS), followed by the main database of sequences: UniProt, and the mapping between sequence and structure: PDBSWS. Next is a database providing a carefully created dataset of functionally-equivalent homologous proteins (FOSTA) and databases containing various mutation data: dbSNP, OMIM and LSMDBs. Finally, SAAPdb is introduced, the centre point of all projects presented in this thesis.

2.1.1 PDB

The Protein Data Bank[1], **PDB**, is the central repository for structural data on proteins and, contrary to its name, other macromolecules (Berman *et al.*, 2000). The PDB is a data bank, i.e. a set of individual files, each containing plain-text information about a single experiment. Each entry can be divided into a header and the atomic coordinates section: the header provides experimental details (authors, methodology, experimental conditions, number of chains in the asymmetric unit, ligand types and numbers, modelling procedure, etc.), and the body of the file has coordinates for each observed atom (`ATOM` and `HETATM` entries for the macromolecule and non-protein/nucleic acid atoms, respectively).

As of July 2011, the PDB contains 74428 structures, 93% of which are structures of proteins, the remainder comprising nucleic acids and other molecules of interest. In terms of experimental techniques used to obtain the structures, the majority of PDB data (87%) was obtained by X-ray crystallography, 12% by NMR spectroscopy (for introduction of these methods, see Section 1.2.2), and the remaining $\sim$ 1% of structures by electron microscopy and other methods.

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

The PDB supplies pointers to the protein sequence data, providing cross-links to UniProtKB/Swiss-Prot sequences in headers of some files. In order to expand this sequence-structure mapping to all proteins for which sequence and structure are available, Andrew Martin has developed a tool called PDBSWS with improved coverage over PDB's sequence mapping (Martin, 2005). PDBSWS will be introduced in more detail in Section 2.1.4.

2.1.1.1 PDB remediation

Processing PDB data is never a trivial task, owing to the data format either being poorly defined in some aspects, or misused by authors submitting the structures. Considering this is one of the most frequently accessed biochemical data resources[2], the issue of the PDB being inconsistent and on occasions erroneous, deserves to be addressed here as the vast majority of the results presented in this thesis rely on structural data originating from the PDB.

Since its creation, this resource has grown 10000 times, and recently it was in definite need of a systematic update, both in terms of data format and deposited data clean-up. Most of the issues were addressed in the 2007 PDB remediation project; to list a few: (i) adjusting terminology to adhere to IUPAC nomenclature, (ii) giving names to non-standard amino acid types, rather than having standard amino acid, and `HETATM` entries with the same residue numbering, (iii) giving each `ATOM` a chain identifier (all single chain entries now have chain 'A' rather than an empty string); for the full list of changes, check Henrick *et al.* (2008). These recent major improvements (another two smaller remediations were published in 2008 and 2011), along with constant effort by the PDB's curators, improved overall accuracy of structural data, consequently reducing the fraction of data lost while analysing them using automated pipelines.

[2]in 2009 on average, more than 7 files were downloaded from it every second, and in 2011 over 2000 other web sites linked to PDB pages (Bluhm *et al.*, 2011)

However, a number of inconsistencies and exceptions still exist among PDB data: multiple atoms or and chains with the same coordinates (1C0I, 1GTV, respectively), overlapping atoms causing steric clashes (8CHO), atoms listed to interact with themselves (1EH9), non-existing atom names ('H2' for hydrogen in 966C), residues numbered in the C′ to N′ direction (3SGA), to name but a few.

2.1.1.2 PDB data in XML-like format

While PDB files contain an abundance of structural data, consistency of which is being constantly improved, there are several weaknesses to its flat file format. The most important issue is the inability to add extra information at the level of protein atoms. Some software utilise the columns used to store B-values or occupancies for this purpose, but such programs cannot be run sequentially to add the results of different analyses. There is a lot of higher-level information that can be calculated from the raw experimental data stored in the PDB format, e.g. solvent-accessibility, hydrogen bonds, secondary structure, interacting residues, etc. In large-scale analyses like the SAAPdb pipeline and building a training set of protein-protein interfaces[3], it is computationally infeasible to re-calculate this information from scratch every time it is required.

Transforming PDB data into an XML-like format would enable creating additional data layers, where the additional information would be stored and easily accessed. Exactly to that end a hybrid XML/ASN.1 format, **XMAS**, was developed by Andrew Martin (Martin, personal communication). All PDB files are automatically processed and data are added on hydrogen bonds (calculated according to the Baker and Hubbard (1984) method), secondary structure elements (Kabsch and Sander (1983)) and solvent-accessibility (according to Lee and Richards (1971)).

[3] to name just a few using thousands of PDB files in one run

In 1999 when XMAS was created, the PDB team had already started introducing their own XML-format for PDB files: PDBML (Westbrook *et al.*, 2005). However, even at present XMAS is considered superior since it structures the data in chains and residues, whereas PDBML recognises only `ATOM` level entries (providing residues and chain information for every atom), thus complicating data parsing. Finally, XMAS was designed for leaf-heavy data (special attention was payed to minimisation of markup for the leaves), enabling it to hugely reduce storage space.

2.1.2 PQS

While the PDB provides the tertiary structures of proteins, it is often misleading in terms of quaternary structure information. As previously mentioned, the asymmetric unit is the minimal unique unit in the protein crystal, however, it is not indicative of the biological unit - the quaternary structure the protein as found in *in vivo*. Although PDB files sometimes specify biological units provided by the experimentalists (in headers), that information is scarce and often experimentally unverified.

Henrick and Thornton (1998) developed the Protein Quaternary Structure[4] (**PQS**) database, an automated system that builds biological units (BUs) from asymmetric units (ASUs) provided in PDB files. In brief, every structure in the PDB obtained by X-ray crystallography is separated into individual chains, thus removing crystal contacts, and then the quaternary structure of the macromolecule is rebuilt. The output PQS file is conveniently in PDB format, enabling easy parsing of PQS files utilising various tools and scripts prepared for PDB data handling. When compared to biological units provided in PDB files (for PDB files containing that information), PQS corrects biological units for 18% of structures (Xu *et al.*, 2006).

In spite of being the most widely used automated tool for quaternary structure prediction, PQS is outperformed by **PISA**, Protein Interfaces, Surfaces and Assemblies[5],

[4] `http://www.ebi.ac.uk/pdbe/pqs/`
[5] `http://www.ebi.ac.uk/msd-srv/prot_int/pistart.html`

developed by Krissinel and Henrick (2007). Based on thermodynamic stability calculations, PISA predicts different biological units from PQS for 23% of structures, often resulting in a smaller assembly than PQS. Although PISA shows more promising performance on the data on which it was trained, (90% of correctly predicted quaternary structures compared with 77% for PQS), PQS was used throughout this thesis for two reasons. First, PISA currently does not provide a bulk download of predicted BUs for the whole PDB. Second, PISA is not appropriate for protein-ligand interface studies, because it fixes positions of ligands as surface modifiers. While this simplification does not affect the work presented here, it might become an issue if the protein-protein interface prediction is expanded to protein-ligand interfaces.

The most reliable approach to quaternary structure prediction is to consider only modelled biological units which were further verified through manual quaternary structure curation, as is done by **PiQSi**[6] (Levy, 2007). This community-based tool confirms or dismisses PQS-determined quaternary structures, after consulting literature for evidence on the number of subunits, SwissProt annotation and/or PISA annotation of close homologues. As of 2010, this initiative has tested $\sim$ 15000 PDB structures, correcting 15% of sampled biological units. Furthermore, on a benchmark set of 187 PQS entries, Levy (2007) showed that PiQSi detects and corrects biological units in 34% of cases. Although these results display an obvious need to improve and/or manually verify current automated methods, the gain in accuracy does not justify the low coverage of only $\sim$ 25% of currently available PDB structures when extensive coverage of protein quaternary structure space is required.

Consequently, the work in Chapters 4 and 5 resorts to PQS as the main source of biological units. This method seemed to be a good compromise between reliability of a model, and the coverage of structures in different protein families. However, future updates of the work will have to switch to PISA, as PQS has been discontinued in 2010, and a switch to PISA is suggested.

[6] `http://supfam.mrc-lmb.cam.ac.uk/elevy/piqsi/piqsi_home.cgi`

2.1.3 UniProtKB/Swiss-Prot

The Universal Protein Resource[7] (**UniProt**, (The UniProt Consortium, 2009)) is the most comprehensive, publicly accessible repository of protein sequence data. It is currently divided into four databases: UniParc (archives of protein sequences), UniRef (clustered sequences for faster searches), UniMES (database of metagenomic data) and UniProtKB (core database of annotated protein sequences). The UniProt Knowledgebase, **UniProtKB**, consists of automatically retrieved, unprocessed sequences in **UniProtKB/TrEMBL** and a smaller database of manually-curated, non-redundant sequences termed **UniProtKB/Swiss-Prot**. The current version of UniProtKB/Swiss-Prot (Release 2011-07) contains 530264 sequence entries: 188 million amino acids, averaging 354 amino acids per entry, from 199650 references. UniProtKB/TrEMBL currently contains 16 million sequences: 5 billion amino acids, and an average of 323 amino acids per sequence (Release 2011-07).

Every protein sequence has a unique identifier – the primary **accession number** (in the past termed primary accession code, hence the AC abbreviation), an entry name and optional secondary accession numbers. The primary accession number (primary AC) is unique for a sequence and is stable over time. In case several sequences are merged owing to redundancy, all but one will become secondary accession numbers and will be kept for future reference. Similarly, if a sequence is revised or split into several entries, each entry will get a new primary AC, and the old one will become a secondary AC. Accession numbers are a combination of six symbols, currently in two acceptable formats: [A-N,R-Z][0-9][A-Z][A-Z, 0-9][A-Z, 0-9][0-9] or [O,P,Q][0-9][A-Z, 0-9][A-Z, 0-9][A-Z, 0-9][0-9]. For example, the full name of the human p53 protein entry presented in Figure 2.1[8] is 'cellular tumor antigen p53'. It has a primary AC `P04637` (green) and several secondary ACs (for example `Q15086` and `Q9UQ61`) (orange). Henceforth unless otherwise stated, when an accession number is mentioned, it refers to the primary accession number of the protein in question.

[7] `http://www.uniprot.org/`
[8] `http://www.uniprot.org/uniprot/P04637`

```
ID   P53_HUMAN                 Reviewed;          393 AA.
AC   P04637; Q15086; Q15087; Q15088; Q16535; Q16807; Q16808; Q16809;
AC   Q16810; Q16811; Q16848; Q2XN98; Q3LRW1; Q3LRW2; Q3LRW3; Q3LRW4;
AC   Q3LRW5; Q86UG1; Q8J016; Q99659; Q9BTM4; Q9HAQ8; Q9NP68; Q9NPJ2;
AC   Q9NZD0; Q9UBI2; Q9UQ61;
DT   13-AUG-1987, integrated into UniProtKB/Swiss-Prot.
DT   24-NOV-2009, sequence version 4.
DT   28-JUN-2011, entry version 187.
DE   RecName: Full=Cellular tumor antigen p53;
DE   AltName: Full=Antigen NY-CO-13;
DE   AltName: Full=Phosphoprotein p53;
DE   AltName: Full=Tumor suppressor p53;
GN   Name=TP53; Synonyms=P53;
OS   Homo sapiens (Human).
```

Figure 2.1: A section of the UniProtKB/Swiss-Prot human p53 entry.
The entry name is shown in blue, the primary accession number in green, and secondary accession numbers in orange.

The **entry name**[9] is a more intuitive label implying the biological role of the sequence, but UniProt states that it should not be used to refer to Swiss-Prot sequences in the literature. The entry name contains up to five alphanumeric symbols constituting a protein identifier, followed by an underscore symbol and up to five alphanumeric symbols corresponding to the species in which the protein was found. For the `P04637` example listed above, the entry name is `P53_HUMAN`, shown in blue in Figure 2.1.

2.1.4 PDBSWS

PDBSWS[10] provides mapping from a PDB residue to a UniProtKB residue, either Swiss-Prot or TrEMBL, in the form of a relational database, accessible through a RESTful web service (Martin, 2005). It uses cross-references to UniProtKB entries available in PDB files, then adds cross-references to PDB entries found in UniProtKB files (created by SSMap (David and Yip, 2008)), and finally, it attempts to map the remaining PDB chains to UniProtKB entries by 'brute-force scanning' (sequence-level mapping). Finally, the UniProtKB sequence is aligned to the ATOM sequence from the PDB file and an alignment-based (residue-level) mapping is stored in the database.

[9] formerly protein identifier, ID
[10] `http://www.bioinf.org.uk/pdbsws/`

It is worth noting here that, while this database was originally built *uniquely* to map from the PDB to UniProtKB, it can be used to provide PDB residue(s) for a given UniProtKB entry. However in this direction, since several PDB entries could have mapped to the same UniProtKB entry, the mapping is not unique and does not give any indication of the 'best' PDB file for a given UniProtKB entry. It may also be incomplete as UniProtKB entries having just one or two mutations compared with a PDB file will not be mapped if there is also an exact (or better) match.

PDBSWS is at the moment, the most appropriate method for sequence to structure mapping available. Being an in-house tool, it is easily accessible and regularly updated. It outperforms other methods in coverage and/or level of automation (SIFTS and MSD (Velankar *et al.*, 2005)), and is preferred over methods lacking residue-level mapping (Seq2Struct (Via *et al.*, 2005), not updated since 2006). Therefore, it has been the method of choice in the SAAPdb project, and throughout this thesis, whenever sequence-structure pairs were required at residue- or protein-levels.

2.1.5 FOSTA

Functional Orthologues from Swiss-prot Text Analysis[11], **FOSTA**, is a relational database of families of automatically annotated functionally-equivalent proteins (McMillan and Martin, 2008). The schema of the algorithm extracting families of functionally-equivalent proteins is shown in Figure 2.2.

For a given human protein, FOSTA identifies a list of homologues using a BLAST search (introduced in Section 2.2.3) against the UniProtKB/Swiss-Prot database. It then uses a series of text analyses of the UniProtKB/Swiss-Prot annotations, initially looking for a match in the protein identifier part of the UniProtKB/Swiss-Prot entry name (see Section 2.1.3), followed by the EC number, and finally by matching synonyms at multiple levels of specificity from the UniProtKB/Swiss-Prot description

[11] `http://www.bioinf.org.uk/fosta/`

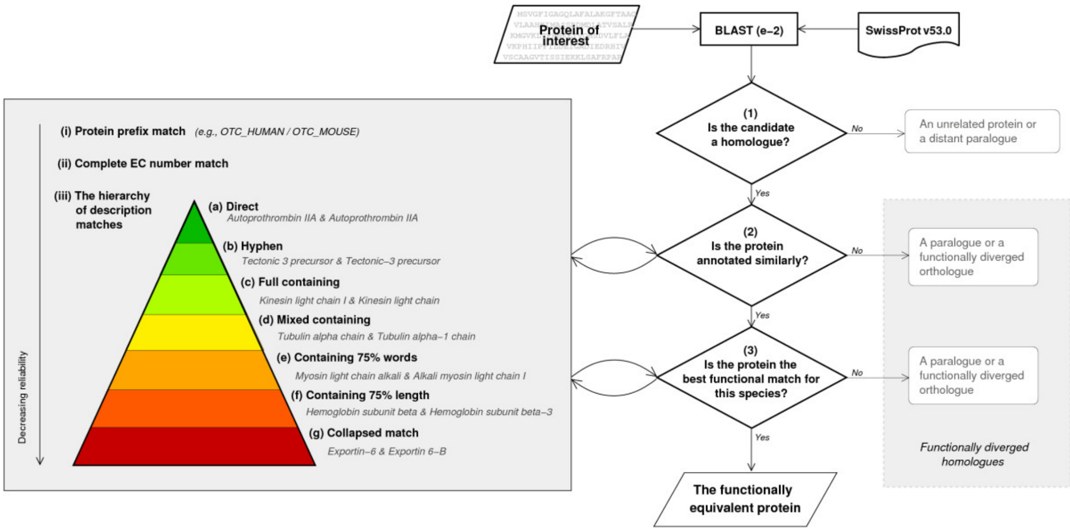

Figure 2.2: A schema of the FOSTA method.
The FOSTA filtering process: homologues are identified by BLAST-ing against the UniProtKB/Swiss-Prot database (i); these are then filtered to retain only those with similar function (ii); finally one protein per species (the FEP, or functionally equivalent protein) is chosen using a hierarchy of functional matches to eliminate functionally diverged homologues (FDHs) (iii). *Figure and caption obtained from* McMillan and Martin (2008)*, with permission from the authors.*

(`DE`) field. Provided a non-human homologue passes any of the three filters, it is marked as a **functionally-equivalent protein**, FEP, and added to the same FEP family with the human protein.

Although McMillan and Martin (2008) demonstrated the high quality of Swiss-Prot functional annotation, they identified examples where FOSTA correctly assigns FEPs with otherwise questionable functional annotation, and several families sharing the same entry name protein identifier. For examples of these inconsistencies, see the `HOX` proteins and the `PROC_HUMAN` example in McMillan and Martin (2008). Thus, FOSTA is preferred over standard lists of orthologues when highly reliable data are required, because when gathering very distant orthologues using traditional methods, they may diverge in function (for example, owing to mutations in functional residues).

2.1.6 Databases of single amino acid polymorphisms

As discussed in Section 1.1, there are many ways of dividing mutations into types and subtypes. This thesis focuses on protein-level variations and considers only SAAPs: substitutions of one amino acid in the protein sequence at a time. SAAPs can, based on their effect on the phenotype, be divided into neutral and pathogenic. The main data sources of deleterious mutations will be presented in Section 2.1.6.1, and a resource providing data on neutral or low-penetrance SAAPs is presented in Section 2.1.6.2.

2.1.6.1 Databases of disease-associated mutations: OMIM and LSMDBs

There is an ever growing number of annotated deleterious single amino acid polymorphisms (SAAPs), mainly because these mutations are simple to link to a disease state, and are consequently interesting for diagnostic and disease-prevention purposes. The most wide-ranging resource of disease-associated SAAPs is the Online Mendelian Inheritance in Man, **OMIM** (McKusick, 2000), at present date containing 20706 mutations obtained from peer-reviewed literature[12]. Although this is the largest source of missense mutations, it is not meant to be an exhaustive list of all Mendelian disorders; instead it provides a list of relevant examples (Amberger *et al.*, 2009). If not otherwise stated, when OMIM mutations are mentioned in this thesis, this refers to the missense mutations found in OMIM, with silent SNPs, frameshifts, larger insertions and deletions and nonsense mutations removed from consideration.

More elaborate lists of mutations for a given gene or disorder can be found in numerous locus-specific mutation databases (**LSMDBs**), created and maintained by research groups interested in a particular condition or a group of diseases. At the moment, some $\sim$ 1500 publicly-available LSMDBs listed by the Human Genome

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

Variation Society (HGVS) can be found at `http://www.hgvs.org/dblist/glsdb.html`, with various other sources scattered all over the Internet and published literature. Many of these LSMDBs are stored in LOVD format: the Leiden Open-source Variation Database is a 'LSMDB-in-a-Box' tool, an attempt to impose and provide a clear and unified format for human variation data (Fokkema *et al.*, 2011). Unfortunately however, LOVD does not support bulk download of the data at the moment.

The main challenges with this wealth of data are (i) parsing them into the same format, and (ii) extracting unique mutations, by mapping to the SwissProt sequence and providing the protein identifier, position, and native and mutant amino acid types. The first issue is addressed in the Martin group by developing parsers, one per LSMDB format, which generate a common XML format. This is obviously a slow and tedious process and any unification in data formats is welcomed. The second issue has been addressed in the Martin group by an OMIM-to-UniProtKB/Swiss-Prot sequence position mapper, see Section 2.2.2.

2.1.6.2 dbSNP

dbSNP[13] is the central database of single nucleotide polymorphism data, also including short insertions and deletions (Sherry *et al.*, 2001). There are two kinds of SNPs in this database, with different level of validation: *submitted* have 'ss' preceding the `SNPid` number, and *validated* start with 'rs': validated SNPs have been confirmed by non-computational methods, i.e. frequency studies. The current version of dbSNP (build 132) contains 4.4 million validated human SNPs and mutation data for another 83 organisms. The current SAAPdb release is populated by non-synonymous SNPs from build 129 and also uses mappings to protein sequences, provided in dbSNP.

[13]`http://www.ncbi.nlm.nih.gov/projects/SNP/`

2.1.7 SAAPdb

The **S**ingle **A**mino **A**cid **P**olymorphisms **d**ata**b**ase, **SAAPdb**[14], is a PostgreSQL relational database of single amino acid polymorphisms, with both neutral and pathogenic phenotypes (Hurst *et al.*, 2008). It provides a range of likely structural effects of SAAPs on structures of human proteins based on mappings of the mutations to the structural data. This project originated as a limited analysis of seven structural effects of mutations in p53 (Martin *et al.*, 2002) and G6PD (Kwok *et al.*, 2002) in the Martin group. Eventually several sources of mutation data were integrated and other structural analyses were added to the pipeline. To keep terminology used here in accordance with mutation types introduced in Section 1.1.3, deleterious mutations termed PDs in the SAAPdb paper (Hurst *et al.*, 2008) are here termed DAMs, while PDs represent a subset of DAMs which have not been observed compensated in a functionally-equivalent protein.

SAAPdb can be divided into two parts: (i) tables with formatted mutation data mapped to protein sequence and protein structure (where available), presented in Section 2.1.7.1, and (ii) the results of automated analyses (a pipeline) assigning likely structural effects to all mutations successfully mapped to structure described in Section 2.1.7.2. The schema of SAAPdb is shown in Figure 2.3.

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

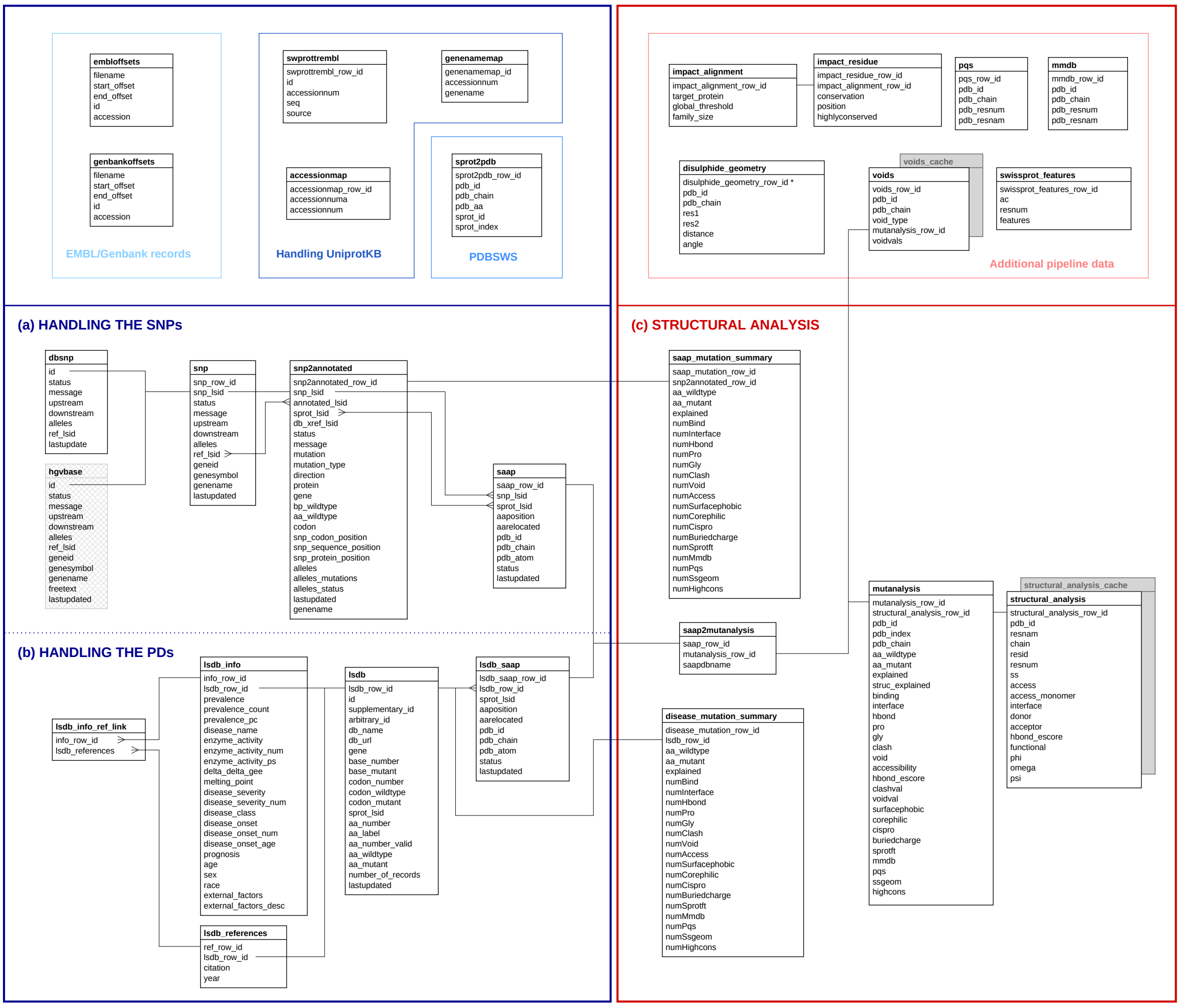

Figure 2.3: SAAPdb schema.
Mutation data and mappings to protein sequence and structures are coloured blue, and results of structural analyses in red. Foreign keys are connected with black lines. *Adapted from Lisa McMillan's PhD thesis.* (McMillan, 2009)

2.1.7.1 Mutation data in SAAPdb

Mutations are divided into separate tables based on pathogenicity: DAMs have reported pathogenic effects and SNPs cover silent and low-penetrance SNPs (for definitions see Section 1.1.2.1). Some SNPs may have low-penetrance pathogenicity, without their effect being observed or correlated to the mutation in a Mendelian sense. In work by Hurst *et al.* (2008), six mutations which were found in both datasets were removed from the SNP dataset: three were found in dbSNP and OMIM and another three in dbSNP and IARC p53 simultaneously. Each SAAP is stored with mappings to sequence data and to structural data, where structures are available.

Databases used to populate the latest version of SAAPdb (released on August 28th, 2008) are shown in Table 2.1: one SNP source, OMIM and ten LSMDBs. The Martin group has been working on expanding this list in terms of coverage of deleterious mutations: for example, since a parser for the LOVD format introduced in Section 2.1.6.1 has been developed for the ZAP70Base and this format is becoming the *de facto* standard for submission of DAM data, several more LOVD-based LSMDBs listed on the HGVS web site could be included in the next SAAPdb build.

The SNPs extracted from dbSNP are all validated non-synonymous SNPs in coding regions of the human genome. Mapping to sequence data for SNPs is provided by dbSNP. DAM mappings are retrieved where available, and then added, verified and/or corrected by a OMIM-to-UniProtKB/Swiss-Prot mapping algorithm ((Martin, in preparation), for details of the algorithm, see Section 2.2.2). Mapping to structural data for both SNPs and DAMs is achieved using PDBSWS as described in Section 2.1.4.

Table 2.1: Mutation data deposited in SAAPdb

	Database	Mutations mapped to sequence	Mutations mapped to structure	Data source
SNPs	dbSNP	8190	811	`http://www.ncbi.nlm.nih.gov/projects/SNP/`
DAMs	ADABase (adenosine deaminase deficiency)	38	0	`http://bioinf.uta.fi/ADAbase/`
	G6PDdb (glucose 6-phosphate dehydrogenase)	103	103	`http://www.bioinf.org.uk/g6pd/`
	HAMSTeRS (haemophilia A)	526	228	`http://europium.csc.mrc.ac.uk/WebPages/Main/main.htm`
	IARC p53 Database (p53)	1490	1490	`http://www-p53.iarc.fr/`
	KinbaseDriver (protein kinase domain)	66	26	(Izarzugaza *et al.*, 2011)
	KinbasePassenger (protein kinase domain)	66	14	(Izarzugaza *et al.*, 2011)
	LDLR (low density lipoprotein receptor)	504	471	`http://www.ucl.ac.uk/ldlr/Current/index.php?select_db=LDLR`
	OMIM	7119	2704	`http://www.ncbi.nlm.nih.gov/omim/`
	OTC (ornithine transcarbamylase)	148	145	(Tuchman *et al.*, 2002)
	SOD1db (SOD1-related ALS1)	96	96	`http://alsod.iop.kcl.ac.uk/Als/index.aspx`
	ZAP70Base (ZAP70 deficiency)	5	5	`http://bioinf.uta.fi/ZAP70base/index2.html`

2.1.7.2 Likely structural effects in the SAAPdb pipeline

The SAAPdb pipeline currently contains 15 structural analyses and one sequence-based analysis[15], shown in Table 2.2, all aiming to 'explain' how SAAPs 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 `Perl` script or `C` program and will output a positive ('explained') or negative ('not explained') result for every SAAP in every category. Therefore the SAAPdb result for a mutation can be viewed as a vector of binary values (1 for $\surd$ and 0 for $\times$), as shown in Figure 2.4.

Annotated Database to PDB Mapping						Explained by															
PDB ID	PDB Chain	PDB Residue	AA Wildtype	AA Mutant	Mutation explained	Binding	Interface	HBond	Pro	Gly	Clash	Void	Surface phobic	Core philic	Cispro	Buried charge	UP features	MMDBBIND	SSgeom	PQS	Highly conserved
1ek5 A details	A	34	N	S	✓	✓	✓	×	×	×	×	×	×	×	×	×	✓	×	×	✓	×
1ek6 A details	A	34	N	S	✓	✓	✓	×	×	×	×	×	×	×	×	×	✓	×	×	✓	×
1ek6 B details	B	34	N	S	✓	✓	✓	×	×	×	×	×	×	×	×	×	✓	×	×	✓	×

Figure 2.4: Structural effects assigned by SAAPdb to a DAM in human UDP-galactose 4-epimerase.
SAAPdb (http://bioinf.org.uk/saap/db/) was queried for UniProtKB/Swiss-Prot accession number Q14376 and Asn43→Ser was chosen as a representative pathogenic mutation (OMIM: 230350). 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 PQS, binding and interface site residue, and it carries a UniProtKB/Swiss-Prot functional identifier.

[15] introduced elsewhere, check Martin*et al.* (2002), Cuff (2004) and McMillan (2009)

Table 2.2: SAAPdb categories.
The horizontal line separates structural categories from the sequence-based one.

Category	Effect of mutation
Interface[a]	Affecting residues in contact with a different protein chain or ligand based on biological units reported in the PDB.
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.
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 $\geq 275\text{Å}^3$ to open in the protein owing to the substitution with a smaller residue.
corephilic[d]	Introducing a hydrophillic 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.

The structural explanation categories are described in detail by Hurst *et al.* (2008).
[a]Interface-damaging; [b]Functionally-impairing; [c]Folding (fold-preventing); [d]Instability (destabilizing); [e]Sequence conservation.

2.2 Algorithms and tools

This section introduces methodology used in the work chapters. It starts with two algorithms used for data pre-processing tasks: solvent-accessibility calculation enables us to divide residues in the protein structure into buried and surface ones, and OMIM sequence mapping verification sorts out inconsistencies in OMIM residue numbering. Next, two key algorithms are introduced: BLAST identifies the likely homologues, and CLUSTALW and MUSCLE align them. Finally, PISCES provides a list of non-redundant structures.

2.2.1 Solvent-accessible surface calculation

The molecular surface area of a residue or atom (sometimes also termed 'Connoly surface area' (Connoly, 1983)), is defined as the as the area of that residue or atom, in contact with the solvent molecule. More often used term is the **A**ccessible **S**urface **A**rea (**ASA**), where instead of measuring the area of the atom's or residue's surface a solvent molecule can reach, the area the *centre* of solvent molecule covers is measured.

The algorithm to obtain ASA measures the area of the surface obtained by rolling a sphere along the surface of the molecule (in this case a protein), obtained by merging van der Waals radii of atoms in the protein. The sphere's radius is usually set to 1.4Å, the van der Waals radius of a water molecule. The area the centre of the solvent probe covers while rolling along the protein surface is the ASA, defined by Lee and Richards (1971). Further in the same work, Lee and Richards presented average solvent-accessibilities for the 20 standard amino acid types $ASA_{av}(X)$ in Ala-X-Ala tripeptides thus introducing the relative solvent-accessibility of a residue, **rASA**:

$$rASA(X) = ASA(X)/ASA_{av}(X) \tag{2.1}$$

where $ASA(X)$ is the observed accessible surface area for a residue of type X, hereafter termed *absolute* solvent-accessibility value, and referred to as ASA. $rASA$ values > 1 indicate above average absolute ASA, common for the first or last residues in a chain, or residues with unusual (and often erroneously measured or modelled) bond angles or lengths.

In this thesis an in-house implementation `solv` (Martin, 1999) was used to calculate relative and absolute solvent-accessibility, based on Lee and Richards (1971) solvent-accessibility calculation method, and using the default sphere radius of 1.4Å. `solv` calculates a total of four solvent-accessibility values: relative and absolute solvent-accessibility of a residue in the structure as provided by the PDB file termed $rASA^c$ and ASA^c respectively ('c' indicating this value was measured in the whole PDB complex). The second pair of values are relative and absolute solvent-accessibility in the monomeric chain, obtained by simply separating the PDB entry into individual chains and then applying the Lee-Richards algorithm, termed $rASA^m$ and ASA^m, respectively ('m' stands for 'monomeric'). Obviously, for residues in PDB entries consisting of a single chain, solvent-accessibilities in the monomer and in the complex will be identical. `solv` is also used on PQS files to identify interacting residues by looking at the difference in relative solvent-accessibility between complex and monomeric chains (for more details, see Section 4.2.1.3).

2.2.2 OMIM-to-UniProtKB/Swiss-Prot mapping

Both OMIM and LSMDBs provide information of various age and from many sources. While these SAAPs are predominantly reported in the [AC, position, native, mutated] format presented in Section 1.1.3, the residue position needs to be verified, and often corrected. To address this issue, Andrew Martin has developed an OMIM-to-UniProtKB/Swiss-Prot mapping algorithm (manuscript in preparation)[16], the algorithm for which is shown in Figure 2.5.

In short, a partial sequence is constructed from all the native residues found in OMIM for a single protein. This sequence fragment is matched to the complete (Swiss-Prot) sequence, in order to identify an offset which will yield the most successful mapping (in terms of correctly aligned 'native' residues) between the two.

2.2.3 BLAST

Basic Local Alignment Search Tool, **BLAST**, finds similar sequences to the provided (query) sequence, in a database of sequences (Altschul *et al.*, 1990). While there are many BLAST implementations aimed at different tasks and sequence types, only the basic protein-sequence-targeted version will be presented here: `blastp`. The user provides a query protein sequence, and the database against which to search against for 'similar' sequences. The level of similarity, and various aspects of the search can be finely tuned through a variety of user-adjustable parameters.

The speed of the search is achieved by partitioning the queried sequence, and all sequences in the target database into all possible substrings of a given length. The default length of these substrings (also termed words) for `blastp` is three. The next step is finding the words in the database with similarity above the threshold with the words in the query sequence – these will be the seeds for building local alignments. The similarity between two words is typically measured using one of the substitution matrices, see for example Dayhoff *et al.* (1978), Henikoff and Henikoff (1992), Altschul (1993) and Altschul *et al.* (2005).

Once a word in a database entry is identified with sufficient similarity to the word in the query sequence, creating a high-scoring segment pair, HSP, these two words are aligned. Then the algorithm expands the alignment in both directions as long as the

[16] mappings available from `http://www.bioinf.org.uk/omim/`

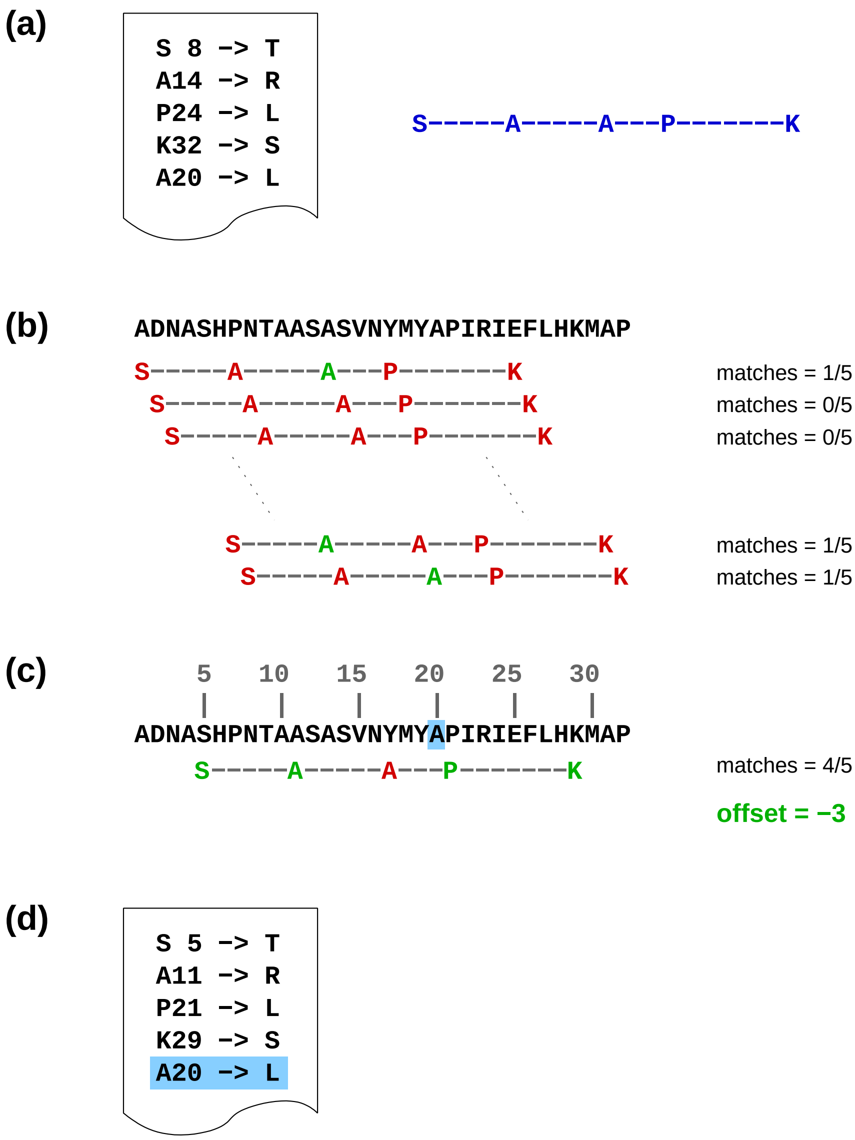

Figure 2.5: (a): a partial sequence is reconstructed from the native residues described in the OMIM record; (b): this partial sequence is slid along the UniProtKB/Swiss-Prot sequence to which it is mapped in OMIM and the number of matches for each position is recorded (matches are shown in green, mismatches are shown in red); (c): the best matching position is used to calculate the offset (note that the A20 record (shown in blue) could be correct with an offset of 0 (i.e., the OMIM annotation is correct) as an alanine does exist at position 20); (d): the offset is applied to the matched original mutations (i.e., the residues found to match in (c)) to generate a corrected numbering and all probably correct mutations (those matched using an offset of 0) are also included in the dataset (again, the probably correct A20 example is highlighted in blue). *Image and caption obtained from Lisa McMillan's PhD thesis* (McMillan, 2009).

similarity is above certain threshold, outputting the aligned pair (the query sequence and a 'hit' from the database) along with several performance measures. The two main measures of similarity are P-value and E-value.

The P-value is the probability of observing a certain similarity by chance: i.e. the similarity is the result of random chance, and thus not indicating an evolutionary connection of the queried sequence and the hit. The expectation measure, E-value, denotes the number of times one expects to observe a given similarity score (or better) by chance in a database of a given size – essentially it is P multiplied by the size of the database searched to find potential hits. Like the P-value, the good hits are the ones with the low E-values: usually $E < 0.01$ is used as a threshold for a significant hit (indicating a potential homologue of the queried sequence).

2.2.4 Aligning protein sequences

Almost every bioinformatics project requires aligning protein or nucleotide sequences at some stage. The predominant tool for this purpose is CLUSTALW (Thompson *et al.*, 1994), although novel, and more efficient algorithms have emerged in the last decade (listed in Section 2.2.4.2). CLUSTALW- and MUSCLE-created alignments are used in Chapters 3 and 4, respectively.

2.2.4.1 CLUSTALW

CLUSTALW (Thompson *et al.*, 1994) is a dynamic programming multiple alignment method, based on pairwise alignments of all sequences. Briefly, CLUSTALW first creates a distance matrix for each pair of sequences, and builds a tree from all the provided sequences based on the neighbour joining method (Saitou and Nei, 1987). The multiple sequence alignment is built by aligning two sequences at a time, starting from the terminal nodes on the tree, using sequence weighting to reduce the importance of very similar (thus containing lots of duplicate information) sequences. In addition to sequence weighting, improved alignment sensitivity is obtained by variable, residue-specific gap penalties, aiming to restrict gaps in secondary structure elements more than gaps in unstructured sequence segments, and preferring creation of larger gaps over opening many short ones.

2.2.4.2 Choice of MUSCLE over other tools

While it is still the most popular and widely used tool, CLUSTALW is outperformed by several recently developed methods (Edgar and Batzoglou, 2006): T-coffee (Notredame *et al.*, 2000), PROBCONS (Do *et al.*, 2005), MUSCLE (Edgar, 2004b) and MAFFT (Katoh *et al.*, 2005). Despite being highly accurate, T-coffee and PROBCONS become very time-consuming when large sequence datasets (> 100 sequences) are aligned (Edgar, 2004b; Edgar and Batzoglou, 2006). MUSCLE and MAFFT have lower time complexity, however MUSCLE achieves marginally better accuracy than MAFFT (Edgar, 2004c). For this reason, MUSCLE was chosen as the preferred MSA tool, appropriate for production of comparable and reproducible alignments with an optimal accuracy-to-speed trade-off.

2.2.4.3 The MUSCLE algorithm

MUSCLE[17] (Multiple Sequence Comparison by Log-Expectation) will be briefly presented here, for more details see Edgar (2004b). MUSCLE produces an optimal multiple sequence alignment in three steps: draft alignment, progressive alignment and refinement steps. Depending on the settings, the refinement step can be omitted[18], increasing the speed of the method, at the expense of reduced accuracy.

[17] download from `http://www.drive5.com/muscle/download3.6.html`
[18] implemented as MUSCLE-prog option

A progressive alignment is created at the end of each step, in a similar way:

- a similarity measure is applied to every pair of sequences yielding a distance matrix
- a binary tree is calculated from the matrix
- moving from the leaves towards the root, pairs of sequences (in the case of two leaves) or pairs of profiles (in the case of non-leaf nodes) are aligned until the root is reached, producing a progressive multiple alignment of all sequences

In the first step, a *draft alignment*, M_1, is created from unaligned sequences, focusing on speed rather than on the alignment quality. The *kmer* distance of every sequence pair is calculated; more related sequences will share more common subsequences (words) of length k and their similarity score will be higher. All pairwise similarity scores are stored in the similarity matrix D_1. After UPGMA clustering (Sneath and Sokal, 1973) of the distance matrix, a binary tree, T_1, is obtained. The preliminary (draft) alignment is built, aligning nodes, starting with leaf nodes towards the root.

In the second step, the draft alignment is further optimised. The errors produced by *kmer* distance calculations are corrected by using the Kimura distance[19] (Kimura, 1980), resulting in a new distance matrix, D_2. Again, after clustering D_2 with UPGMA, a tree, T_2, is produced and an improved progressive alignment, M_2, is created.

The final step is based on 'tree-dependent restricted partitioning' (Hirosawa *et al.*, 1995). T_2 is bipartitioned (divided into two sub-trees) by deleting an edge[20]. The profile alignment is calculated for every sub-tree and these two profiles are aligned,

[19]Kimura distance, $-log_e(1 - D - D^2/5)$, unlike *kmer* distance proportional to D (where D is fractional identity of two compared sequences, for more details, see Edgar (2004a)), corrects for multiple mutation events at the same position

[20]starting from the leaves, edges progressively closer to the root are deleted, one at a time

creating a multiple alignment, M_3. If the refinement step was successful and M_3 has improved over M_2, M_2 is discarded for the refined alignment which becomes a base for the next refinement iteration. Both second and third steps are iterated for the user-specified number of cycles, or until the alignments converge.

2.2.5 Creating non-redundant protein datasets with PISCES

PISCES[21] (Wang and Dunbrack, 2005) is a server providing clusters of protein entries in the PDB format, grouped by sequence identities, i.e. once a sequence identity threshold is set to X, all entries within a cluster will have maximum sequence identity of X. Several user-adjustable filters have been implemented in order to set the acceptable resolution range, R factor threshold and range of chain lengths from the PDB files, and to remove NMR entries and entries lacking all-atom coordinates (C_α-only entries). Entries within a cluster are ordered by method (X-ray crystallography then NMR); then by ascending resolution within same-method entries; then by ascending R-factor if the resolution is the same (Wang and Dunbrack, 2003).

If a set of PDB entries is culled using PISCES, choosing the first protein from every cluster[22] ensures an even coverage of all the PDB: it provides a dataset of proteins in which no two have more than X pairwise sequence similarity. In Chapter 4, this method was used to obtain a list of non-redundant protein chains, in order to provide an unbiased set of protein-protein interfaces.

[21] `http://dunbrack.fccc.edu/Guoli/PISCES_OptionPage.php`
[22] by definition, this is the highest-quality structure in that cluster

2.3 Statistical methods

This section presents the basic statistical concepts and tests used throughout this thesis. The χ^2 test and Fisher's exact test are used when categorical data are tested for difference in frequency distributions (usually two datasets for the presence or absence of a single feature). Student's t-test is used for two populations surveyed for a feature measured on a continuous scale: it tests the significance of the difference in the means of the two samples. Finally linear regression models approximate the behaviour of two-dimensional data with a single line and as such, provide the line of best fit through the observed data.

2.3.1 χ^2 test

The Chi-squared test (χ^2 test) (Mood *et al.*, 1974) is a nonparametric test used on nominal, categorical data to compare a frequency distribution of a sample to a theoretical frequency distribution (i.e. a *goodness of fit test*). Alternatively, two samples are compared, the null-hypothesis being that they are drawn from the same frequency distribution (*test of independence*). Data are divided into n datasets, and k categories of outcomes. Outcome categories have to be mutually exclusive and frequency probabilities for a given dataset, over all categories have to sum to 1. When defined in this way, the test has $(n-1)(k-1)$ degrees of freedom and the test statistic is calculated as follows:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{2.2}$$

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

The χ^2 test assumes sampled data conform to the χ^2 distribution, which is a special case of the gamma distribution. When expected counts of 5 or less appear in the

2×2 contingency table, this assumption leads to significant errors: it increases the χ^2 value and thus erroneously decreases the p-value. This issue is somewhat corrected by introduction of the **Yates correction** for continuity (Yates, 1934): subtracting an additional 0.5 from the difference between the observed and expected value increases the p-value, but this procedure is known to over-correct. The only way to completely avoid using this assumption about the distribution of the tested data, is to use an exact test introduced in Section 2.3.2.

In this thesis the χ^2 test, implemented as a `Perl` script, was used when comparing frequencies of disulphide bonds, hydrogen bonds and secondary structure elements in interface and non-interface surface patches in Section 4.3.2. Two datasets were tested for one property at a time, thus having a 2×2 contingency table and performing a test of independence with one degree of freedom. All data points were labelled 1 (property exists) or 0 (property not observed), and raw counts were tested without normalisation.

2.3.2 Fisher's exact test

The Fisher's exact test (Fisher, 1935) is used instead of a χ^2 test when counts ≤ 10 or empty fields occur in 2×2 contingency table. It provides an *exact* p-value, thus removing discrepancies between the sampling and theoretical χ^2 distribution for small datasets. `fisher.test` implemented in `R` was used, using default parameters.

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

For the example shown in Table 2.3, the p-value is:

$$p = \frac{(x_A + x_B)!(y_A + y_B)!(x_A + y_A)!(x_B + y_B)!}{N!x_A!x_B!y_A!y_B!} \tag{2.3}$$

where X and Y are datasets, A and B are categories, and N is the total count $x_A + x_B + y_A + y_B$.

The only limitation of the Fisher's exact test is its calculation complexity: if the dataset is very large it soon becomes unfeasible to calculate the p-value. Indeed, this is the reason why the χ^2 test was used instead in Chapter 4, as mentioned in the section above.

2.3.3 Bonferonni correction for multiple testing

Multiple testing using the same test on the same dataset, increases the chance of observing a relevant score by chance, i.e. a false positive. In order to eliminate this multiple testing bias, the Bonferonni correction needs to be employed: for every performed test, the α value should be reduced N times for a result to be assessed as significant, where N is the number of times a tests was repeated on the same data. Only p-values smaller than α/N are then considered statistically significant.

2.3.4 T-test

The t-test is a non-parametric statistical test measuring the significance of the difference in means of two normally-distributed populations. Student's t-test is often used as a synonym, although strictly speaking, Student's t-test assumes that the variances of the two populations are equal. Further, Markowski and Markowski (1990) have shown that, in the case where two samples have roughly the same size, Student's t-test can still be successfully used, irrespective of variance differences between samples. Finally, if the two populations differ in both variance and dataset sizes, Welch's t-test is used (Welch, 1947), calculating the t-statistic (for the null hypothesis that the means of the two samples are of equal values) as follows:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \tag{2.4}$$

where $\bar{X}$ is the mean of the sample, s^2 is the sample variance and N is the number of data points in that sample. In this case, the degrees of freedom cannot be calculated, rather, they are approximated using the Welch-Satterthwaite equation (Equation 28 in Welch (1947)). The approximation of the total degrees of freedom is based on the linear combination of degrees of freedom from each of the sample's variances, a value not directly linkable to the sample size. `t.test` implemented in the `R` language was used, defaulting to the two-sided Welch's t-test.

2.3.5 Linear regression models

A linear regression model outputs a line, defined by the slope and the y-axis intercept with the best fit for data consisting of two variables (Wilkinson and Rogers, 1973). There are several uses for this line equation: sometimes it represents a biologically interesting parameter, sometimes it is used to infer an X value for an unknown Y value or *vice-versa*. Using `lm` in `R`, linear regression was inferred by the least squares method where the line

$$y = \alpha x + \beta \tag{2.5}$$

is found in order to minimise the sum of squared residuals (SSE)

$$SSE = \sum_{i=1}^{n} e_i^2 \tag{2.6}$$

where $e_i = y_i - \hat{y}_i$ is the residual, y_i is the observed value, $\hat{y}_i$ is the expected value (value on the line for x_i) and n is the number of data points. Given the average of x values $\bar{x}$ and the average of y values $\bar{y}$, the best fit line slope and intercept are, respectively:

$$\alpha = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \beta = \bar{y} - \alpha\bar{x} \tag{2.7}$$

If the experimentalist has some prior knowledge about the data, and consequently the line parameters, one can 'force' the line to pass through a certain data point. Such a model is termed a 'restrained linear regression model', and should be used with caution (only for very obvious restraints) as it adds bias to the model, and in turn, reduces the effect data have on the line formula. An example of sensible restraints imposed on the linear model is presented in the CPD sphere conservation analysis, in Section 3.2.3.1.

It should be noted that this method does not return an indication of how sensible it is to approximate a data relationship with a straight line, it simply provides the most appropriate line equation given the data. Calculating a correlation coefficient will show whether, and to what extent, the two variables are dependent.

Chapter 3

Compensated Pathogenic Deviations

The work presented in this chapter was a continuation of preliminary work in an undergraduate project by Hubert Rogers. While I reused some of his methodologies (the dataset preparation and mapping to the sequence and structure), I had to re-implement most of the code used in Sections 3.2.1 and 3.2.5, with the help of Lisa McMillan. Additionally I implemented analyses presented in Sections 3.2.3, 3.2.4 and 3.2.5.1. Some of the work presented in this chapter has been published in Barešić *et al.* (2010) and Barešić and Martin (2011).

3.1 Introduction

3.1.1 Protein evolution: an overview

Molecular evolution is mostly investigated at the protein level, since proteins are traditionally viewed as effector molecules, thus enabling a more straightforward link from their function/structure to the observed changes in phenotype. This section defines several basic evolutionary mechanisms and terms using examples from the protein world, to be used throughout the following chapters.

At the molecular level, the evolutionary path of a protein is affected by other molecules, i.e. interaction partners and environmental factors. When two molecules display a similar evolutionary path, they are considered to be **coevolving**, usually resulting from a shared cellular pathway, localisation, expression pattern or co-adaptation (Pazos and Valencia, 2008). In co-adaptation, two molecules affect each other's evolution. A similar scenario is termed **epistasis**: the total change in fitness cannot be obtained by adding the fitness contributions of individual alleles, owing to the inter-dependence of these alleles. In classical population genetics terminology, epistasis refers to interdependency between different gene products or proteins; for a review see Cordell (2002). It is intuitive to expand this principle to two SAAPs within the same protein having a combined effect on fitness, which could not be predicted from each SAAP occurring alone in that protein: for example, when these two SAAPs form a new hydrogen bond, thus affecting that protein's stability and aggregation rate.

Let's consider a special case: **sign epistasis**, when a detrimental effect of a pathogenic SAAP (P) is turned into a mildly positive (or at least a neutral) one through epistatic interactions with another SAAP (C). In other words, C is neutralising whatever the negative effect of P is (thus the symbol 'C' as a compensatory mutation), while on its own, C would not display a beneficial effect on the fitness. This phenomenon is also known as 'fitness reversal' and it is an important mechanism for organisms to sample protein space through various SAAPs, traversing along ridges of the fitness landscape (Cowperthwaite *et al.*, 2006). Employing the same annotation as above: if P is found as a disease-associated SAAP in one organism and at the same time, it is found neutralised through compensation with C (or several variations) in a homologue, this special case of a DAM is termed a **compensated pathogenic deviation**, a CPD.

Understanding the evolution of disease-associated mutations (DAMs) is facilitated by exploration of the structural context which allows for these mutations to appear, and propagate through generations. Thus CPDs provide valuable insights in the evolution of disease-associated mutations and proteins in general, through epistatic selection.

3.1.2 Compensated pathogenic deviations

Compensated pathogenic deviations are disease-associated mutations in a protein of one species (usually human), which occur as the wild-type in a functionally-equivalent protein, FEP, of another species. Functionally-equivalent proteins have previously been defined in Section 2.1.5. This phenomenon was first discussed by Kimura (1985) who initially called them 'compensatory neutral mutations'; the term 'compensated pathogenic deviations' was introduced later by Kondrashov (2002). Throughout this chapter, disease-associated mutations have been divided into two datasets based on the presence or absence of the observed compensation: where compensation has occurred mutations are called compensated pathogenic deviations, CPDs, and where no compensation was found mutations are simply termed pathogenic deviations, PDs.

Classification of a human missense mutation as a CPD is based on a multiple sequence alignment of the mutation-containing human protein and its functionally-equivalent homologues. Consider the multiple sequence alignment of the antithrombin-III protein shown in Figure 3.1. Mutations Ala416→Pro[1] and Ala416→Ser[2], both cause susceptibility to trombophilia owing to antithrombin-III deficiency in the human antithrombin-III protein (UniProtKB/Swiss-Prot:P01008). Since position 416 is fully conserved and neither proline nor serine appear as wild-type residues among functionally-equivalent homologues of ANT3_HUMAN, these two mutations are classified as PDs. On the contrary, Ala419→Val[3] is a CPD, because native valine occurs in bovine and sheep sequences in the same sequence position. Positions 416 and 419 in Figure 3.1 refer to the residue numbering in the UniProtKB/Swiss-Prot sequence; according to the OMIM residue numbering these are positions 384 and 387, respectively. For the details of the OMIM-to-UniProtKB/Swiss-Prot mapping, see Figure 2.5.

```
                                                                           A416S
                                                                           A416P  A419V
                                                                               |  |
P41361|ANT3_BOVIN      SFSVKEQLQDMGLEDLFSPEKSRLPGIVAEGRSDLYVSDAFHKAFLEVNEEGSEAAASTV
P32262|ANT3_SHEEP      SFSVKEQLQDMGLEDLFSPEKSRLPGIVAEGRNDLYVSDAFHKAFLEVNEEGSEAAASTV
P01008|ANT3_HUMAN      GFSLKEQLQDMGLVDLFSPEKSKLPGIVAEGRDDLYVSDAFHKAFLEVNEEGSEAAASTA
Q5R5A3|ANT3_PONAB      GFSLKEQLQDMGLVDLFSPEKSKLPGIVAEGRDDLYVSDAFHKAFLEVNEEGSEAAASTA
P32261|ANT3_MOUSE      GFSLKEQLQDMGLIDLFSPEKSQLPGIVAGGRDDLYVSDAFHKAFLEVNEEGSEAAASTS
Q03352|ANT3_CHICK      ------------------------------------------------------------

P41361|ANT3_BOVIN      ISIAGRSLNSDRVTFKANRPILVLIREVALNTIIFMGRVANPCVD
P32262|ANT3_SHEEP      ISIAGRSLNLNRVTFQANRPFLVLIREVALNTIIFMGRVANPCVN
P01008|ANT3_HUMAN      VVIAGRSLNPNRVTFKANRPFLVFIREVPLNTIIFMGRVANPCVK
Q5R5A3|ANT3_PONAB      VVIAGRSLNPNRVTFKANRPFLVFIREVPLNTIIFMGRVANPCVK
P32261|ANT3_MOUSE      VVITGRSLNPNRVTFKANRPFLVLIREVALNTIIFMGRVANPCVN
Q03352|ANT3_CHICK      ---------------------------------------------
```

Figure 3.1: A CPD example in the human antithrombin-III protein aligned to its non-human FEPs (obtained from FOSTA).
Ala416→Pro and Ala416→Ser are PDs, Ala419→Val is a CPD; all three cause antithrombin-III deficiency. The CPD is shown in green, the disease-associated amino acid type found as native in non-human FEP in grey, the PD in blue and the putative compensatory mutations in purple.

[1] OMIM:107300.0007 (Barbui *et al.*, 1983)
[2] OMIM:107300.0027 (Harper *et al.*, 1991)
[3] OMIM:107300.0042 (White *et al.*, 1992)

3.1.3 Compensatory mutations

The pathogenic effect of a CPD is assumed to be neutralized in the non-human FEP through epistatic interaction(s) with other mutation(s), occurring within the same protein (intragenic compensation), or in an interacting partner (intergenic compensation) (Poon *et al.*, 2005). This neutralising mutation is hereafter termed a 'compensatory mutation', since its epistatic effect with a CPD results in a loss of the deleterious phenotype, and ultimately simultaneous fixation of the mutation pair.

The simplest (and easiest to identify) form of compensation is 'one-on-one' compensation, where a SAAP fully neutralises the negative effect a DAM has on the fitness. The other, more likely scenario is 'sphere compensation' where the effects of a series of small variations, usually among residues surrounding the DAM in the protein structure (thus the name), add up to fully neutralise the pathogenicity of a DAM. When modelling the average number of suppressor mutations[4], Poon *et al.* (2005) achieved the best fit using an L-shaped gamma distribution, with an average of 11.8 compensatory mutations for a compensated mutation. Furthermore, they estimated 78% of compensatory events were intragenic, with somewhat more intergenic compensatory events in viruses than in prokaryotes and eukaryotes. This trend is not surprising: viruses have a significantly smaller pool of proteins owing to the small genome, providing fewer positions for the intragenic compensation to emerge.

While sphere compensation is more widespread, the one-on-one compensation is easier to identify, and is not uncommon (Kondrashov *et al.*, 2002; Ferrer-Costa *et al.*, 2002). Four examples of one-on-one compensation are presented in Sections 3.3.4.3–3.3.4.6. In the case of the model CPD from Figure 3.1, Ala419→Val, neighbouring residues which have diverged in sequence have been shown in purple in Figure 3.1: these are the likely compensatory sites provided the compensation is local for this mutation. There is no experimental evidence[5] whether this mutation is neutralised

[4] which are, in fact, a synonym for compensatory mutations
[5] usually obtained through mutagenesis screening

by the additive effect of all of these nearby substitutions, or only one of them is sufficient for the full fitness reversal.

3.1.4 Evolution of CPDs

Evolutionary processes are often explained in terms of fitness landscapes (Wright, 1932), where peaks represent genotypes with high fitness, and valleys represent less fit genotypes. Data on fitness landscapes are limited by the availability of genetic sequences: the wild-type sequences and low-penetrance SNP data (for example, from dbSNP) correspond to landscape peaks, while disease-associated mutations from OMIM and LSMDBs[6] represent valleys. Almost all possible genetic sequences are unfit, so for a protein to evolve over time, only a discrete series of rare, fit sequences may be used as steps in the evolutionary journey (Kondrashov *et al.*, 2002). One of the ways to traverse between adjacent peaks in the fitness landscape is through CPDs: individually pathogenic mutations become fixed in the population through epistatic selection with compensatory mutations.

The unusually high fixation rate of what were predicted to be pathogenic phenotypes was first observed in *in silico* modelled RNA evolution by Cowperthwaite *et al.* (2006). They also found that the overall fitness did not decrease as fast as expected from the accumulation of detrimental changes. Rather, more than half of the originally pathogenic mutations encountered fitness reversals as a result of accumulation of compensatory mutations, often occurring after the initial pathogenic mutation.

Povolotskaya and Kondrashov (2010) presented a model of unidirectional evolution of proteins in sequence space in which CPDs play the main role. They found that proteins have not reached their limit in divergence from one another, and they are still sampling protein sequence space. Further, they suggest that for a protein sequence, 2% of possible missense mutations are not forbidden and are thus available

[6] some of which have been incorporated into SAAPdb, see Table 2.1

for travelling along the fitness landscape. Let's consider a protein consisting of 100 amino acids, where each position can adopt 19 different missense changes: only 38 of these combinations (2% of 19×100) are allowed, at any given moment. After the first mutation, the sequence and the structural context for epistasis change, but again 38 missense mutations are 'allowed', one of which is the reverse mutation. In other words, there is 1:38 chance for that protein to revert to the original form, and a 97.4% chance for the protein to increase the distance (in terms of sequence similarity) from the original, indicating a strong tendency for previously unseen protein sequence combinations to be sampled.

3.1.4.1 Timeline of occurrence of the compensatory and compensated mutations

As previously mentioned, a CPD (P) is deleterious, and without the existence of a compensatory mutation (C), it will not be fixed in a wild-type non-human sequence. In one of their models, DePristo *et al.* (2005) proposed a simultaneous occurrence of a P–C pair in the same organism. That scenario is highly improbable as it would require a high mutation rate. It is more likely that the compensatory mutation evolved slightly earlier and is present in a population in low frequency, enabling the CPD to occur without causing detrimental consequences. Another option is, provided the mutation rate is sufficiently high, for the compensatory mutation to appear soon after CPD emergence (similar to the scenario Cowperthwaite *et al.* observed in RNA *in silico* evolution mentioned above), presuming P exists in the population in low copy numbers. At that point, the P–C pair undergoes epistatic selection and is ultimately fixed in the population.

3.1.4.2 Effect of CPDs on the organismal fitness

Focusing on the co-occurrence of CPDs and compensatory mutations, DePristo *et al.* (2005) proposed two hypotheses of CPD evolution based on models of biophysical properties. In the first scenario, a compensatory mutation C is phenotypically neutral and stable, thus fixing itself quickly in the population. A pathogenic mutation P is unstable, and can become fixed only if it occurs *after* the compensatory mutation C, resulting in a CPD (the P–C pair) which has higher fitness owing to epistasis.

In the second model, both P and C are individually deleterious, but together have a neutral effect, giving rise to a fitness peak. It is known that small frequencies of low-fitness mutations exist in large populations, an effect termed 'population delocalisation' (DePristo *et al.*, 2005). Consequently, if P occurs in the same individual before detrimental C has been eliminated, it is possible for the P–C genotype to become fixed within the population, while neither of the deleterious intermediates would be fixed on their own. Again, a less likely but possible scenario is that both C and P occur simultaneously.

3.1.4.3 Frequency of compensation among deleterious mutations

Initial studies of CPDs by Kondrashov *et al.* (2002) in the human genome and Kulathinal *et al.* (2004) in the *Drosophila* genome indicated a fairly constant ratio of compensation among the disease-associated mutations. More recently, as more analyses appeared, a correlation seems to have emerged between the frequency of compensation and the minimum sequence identity threshold used to filter out distantly related homologues: 0.14%, 0.4%, 12.5% and 17.8%[7] for human/chimpanzee/neanderthal (Zhang *et al.*, 2010), dipteran-only proteins (Kulathinal *et al.*, 2004), proteins of all species with > 50% sequence identity (Kondrashov *et al.*, 2002) and > 10% sequence identity (Ferrer-Costa *et al.*, 2007),

[7]exact counts can be found in Table 3.1

respectively. Although the authors above used very different methodology making it unreasonable to calculate a correlation coefficient from these scores, there is a noticeable increase in CPD frequency with the decrease in evolutionary distance. The dependency of the CPDs on the method used to obtain the mutation datasets will be discussed further in Section 3.3.1.5.

3.1.5 Structural features of CPDs

In a recent study, Ferrer-Costa *et al.* (2007) demonstrated that both the structural environment and the nature of the substitution play an important role for the development of compensatory mutations facilitating a CPD. Their results show statistically significant differences in the solvent accessibility of CPD residues as well as intrinsic properties of the mutation (change in amino acid volume, hydrophobicity and BLOSUM62 scores (Henikoff and Henikoff, 1992)) when compared with the control dataset - pathogenic deviations. They suggested that (i) CPDs are more often found on or close to the protein surface, (ii) mutations to residues making a large number of contacts are more difficult to compensate than those making few contacts, and (iii) CPDs are, on average, more conservative substitutions than PDs.

3.2 Methods

In order to gather a comprehensive dataset of well-annotated compensated mutations, missense mutations were gathered from OMIM (McKusick, 2000; Amberger *et al.*, 2009) (April 2008 release), mapped to sequence data, and then mapped to available protein structures. Successfully mapped mutations were then divided into two datasets, each mutation being sorted either as a 'PD' or a 'CPD'. Three aspects of the CPDs were examined: (i) preferences for amino acid types in Section 3.2.2, (ii) properties of the residues surrounding a CPD/PD in the protein structure, particularly solvent-accessibility and sequence conservation, described in Section 3.2.3, and

(iii) types of local structural effects, using 14 structural explanations implemented in SAAPdb (Hurst *et al.*, 2008), presented in Section 3.2.5.

3.2.1 Obtaining the dataset

All mutations analysed in this chapter were missense mutations (from one standard amino acid type to another) - neither native nor mutated codon was allowed to be a stop-codon. A distinct mutation (a CPD or a PD) was defined as a unique combination of four parameters, as shown in Figure 1.4 where the information about the protein's accession number, the native and the mutated amino acid type originated from OMIM, and the residue position was obtained from OMIM-to-UniProtKB/Swiss-Prot mapping (to match UniProtKB/Swiss-Prot residue numbering), as described below.

3.2.1.1 Mapping OMIM mutations to sequence

As previously mentioned, OMIM provides mapping of SAAPs to protein sequence which does not necessarily correspond to the latest sequence data in the UniProtKB/Swiss-Prot. A method to verify or correct the sequence position of every missense mutation provided by the OMIM database was described in Section 2.2.2. Hereafter mutations are reported with respect to the UniProtKB/Swiss-Prot residue numbering, and where it differs from OMIM numbering, both sequence positions are stated for easier reference.

3.2.1.2 Mapping OMIM mutations to structure

After being correctly mapped to a residue in an up-to-date UniProtKB/Swiss-Prot sequence, every mutation was mapped to a residue in a PDB structure using PDBSWS (introduced in Section 2.1.4). A given sequence may have been mapped to multiple

PDB crystal structures, the optimal structure was chosen on the basis first of sequence identity with the UniProtKB/Swiss-Prot sequence, second of resolution and third of R-factor.

3.2.1.3 Multiple sequence alignments of mutation-containing human proteins

In order to identify which OMIM mutations were found to be compensated, multiple sequence alignments were built, one for every mutation-containing human protein. Every alignment contained the human UniProtKB/Swiss-Prot sequence, and the UniProtKB/Swiss-Prot sequences of all reliable (not containing `'hypothetical'`, `'probable'`, `'putative'`, `'-like'` or `'homolog'` in UniProtKB/Swiss-Prot `description` field) functionally-equivalent proteins identified by the FOSTA method (for more details, see Section 2.1.5 and McMillan and Martin (2008)). Once a list of FEPs was obtained, the alignment was created using `ClustalW`[8] (Thompson *et al.*, 1994), with default parameters.

3.2.1.4 Classification into compensated and uncompensated mutations

Upon successful mapping to at least one residue in a protein structure and aligning the mutation-containing sequence to at least one functionally-equivalent protein, each mutation was assigned as a compensated mutation (CPD) or an uncompensated mutation (PD). This algorithm is presented in Figure 3.2.

Columns from the multiple sequence alignment containing disease-associated mutations in the human protein were identified. If any of the non-human residues in these columns (aligned to the human pathogenic mutation) matched the amino acid causing

[8] introduced in Section 2.2.4.1

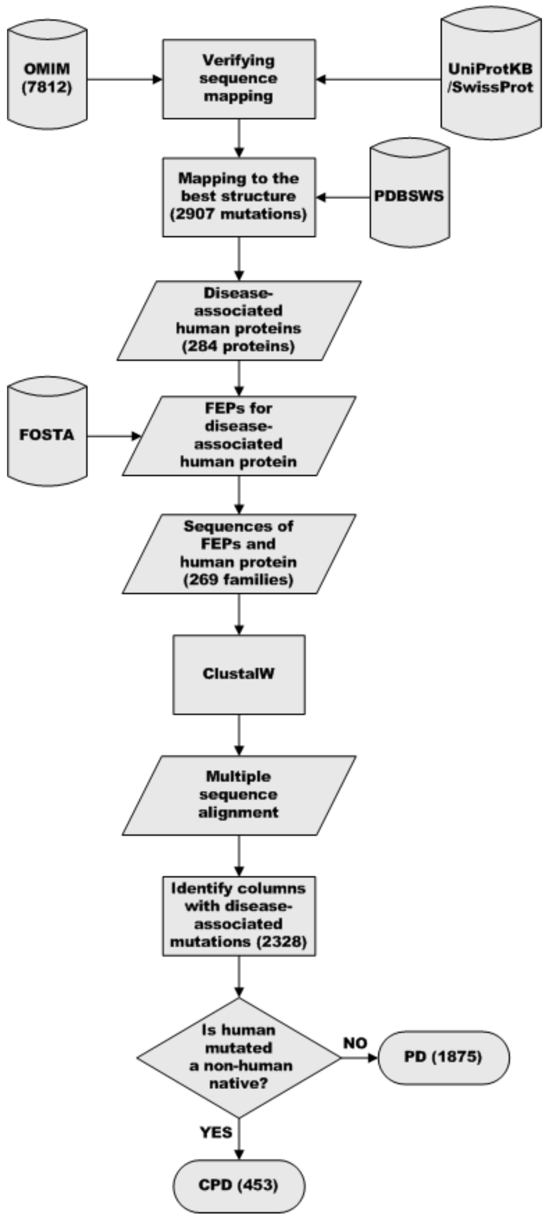

Figure 3.2: Creating CPD and PD datasets from the mutation data, the structural and sequence data and data about functionally-equivalent proteins (FEPs). Where not stated otherwise, numbers in parentheses denote counts of mutations.

the disease in humans, that mutation was sorted into the CPD dataset. An example of a CPD defined in this way was introduced in Section 3.1.2 and Figure 3.1.

3.2.2 Amino acid content of CPDs

The tendency for certain amino acid types to occur preferentially as the native or mutated amino acid among CPDs was based on the frequency of that amino acid type in the CPD dataset. The frequency of every amino acid type among CPDs as the native residue was calculated as follows:

$$F_{native}(X) = \frac{N_{native}(X)}{N_{total}} \tag{3.1}$$

where X is one of the 20 standard amino acid types, $N_{native}(X)$ is the number of CPDs having that amino acid type as the native residue, and N_{total} is the total count of native residues among CPDs (i.e. the number of CPDs processed). Similarly, the frequency of amino acid types among mutant residues in the CPD dataset was defined as:

$$F_{mutant}(X) = \frac{N_{mutant}(X)}{N_{total}} \tag{3.2}$$

Now two sets of propensities have been calculated: 'CPDs-native' and 'CPDs-mutant':

$$Pr_{native}(X) = ln(\frac{F_{native}(X)}{F_{background}(X)}); Pr_{mutant}(X) = ln(\frac{F_{mutant}(X)}{F_{background}(X)}) \tag{3.3}$$

where $F_{background}(X)$ is the frequency of amino acid type X in the background dataset of sequences. The background dataset comprised the full sequences of 245 human proteins containing all OMIM mutations successfully mapped to a protein structure in the Section 3.2.1.2.

The same procedure was repeated for the PDs, resulting in another two sets of propensity scores.

3.2.3 Conservation within an 8Å sphere around mutations

Residues spatially surrounding a compensated mutation were investigated for potential compensatory mutations. The method used for identification of likely compensatory mutation sites based on sequence conservation of spatially neighbouring residues is described below, using the sequence alignment shown in Figure 3.3.

```
                                                                              A416S
                                                                              A416P  A419V
                                                                                  |  |
P41361|ANT3_BOVIN         SFSVKEQLQDMGLEDLFSPEKSRLPGIVAEGRSDLYVSDAFHKAFLEVNEEGSEAAASTV
P32262|ANT3_SHEEP         SFSVKEQLQDMGLEDLFSPEKSRLPGIVAEGRNDLYVSDAFHKAFLEVNEEGSEAAASTV
P01008|ANT3_HUMAN         GFSLKEQLQDMGLVDLFSPEKSKLPGIVAEGRDDLYVSDAFHKAFLEVNEEGSEAAASTA
Q5R5A3|ANT3_PONAB         GFSLKEQLQDMGLVDLFSPEKSKLPGIVAEGRDDLYVSDAFHKAFLEVNEEGSEAAASTA
P32261|ANT3_MOUSE         GFSLKEQLQDMGLIDLFSPEKSQLPGIVAGGRDDLYVSDAFHKAFLEVNEEGSEAAASTS
Q03352|ANT3_CHICK         ------------------------------------------------------------
```

Figure 3.3: A CPD example in the human antithrombin-III protein aligned to its non-human FEPs.
The Ala419Val CPD mutation is assigned two C/T ratios shown in grey, one for each CPD-containing sequence (ANT3_BOVIN and ANT3_SHEEP), while for the Ala416Pro PD mutation, four C/T ratios were calculated, one for every FEP sequence (also shown in grey).

After identifying the Ala419→Val mutation in human sequence P01008 as a CPD because P41361 and P32262 (cow and sheep FEPs, respectively) contained a native valine at position 419, the best-quality crystal structure (PDB ID: 2B5T) mapped to the human protein (P01008) was examined for all residues having at least one atom within 8Å of Ala419 (the CPD residue). These 'in range' residues were then mapped back onto the alignment and 'in range' alignment columns were checked for sequence divergence from the human sequence in the P41361 and P32262 sequences (the two sequences which showed a native Val419). This algorithm is summarised in Figure 3.4. Here an approximation was made that, having identified residues within 8Å of the mutation in the human structure, the equivalent residues in non-human sequences are assumed also to be within 8Å of the CPD residue in their respective structures. Thus all 'in range' differences in both of the FEP sequences compared with the human sequence were considered to be potential local compensatory mutations.

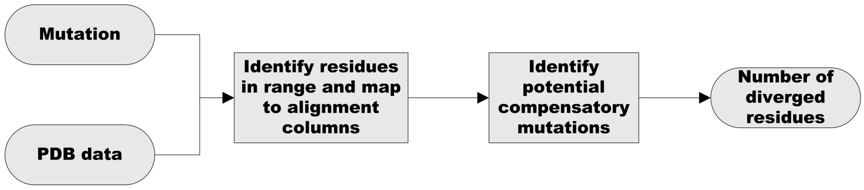

Figure 3.4: The number of mutated residues within 8Å of a CPD/PD mutation was counted.

A C/T ratio was calculated for each of the CPD-containing FEP sequences (P41361 and P32262), where C was the number of local (potential compensatory) mutations and T was the total number of 'in range' columns checked for that sequence. In other words, the C/T ratio was the fraction of spatially neighbouring residues which were mutated. The ratio was recorded together with the overall pairwise sequence identity: pairwise sequence identity indicated the average expected C/T ratio for that protein pair (human and CPD-containing FEP).

Figure 3.3 also contains a pathogenic deviation (PD). At column 416, a mutation to proline was shown to be deleterious, and no proline was identified at this location in the FEPs from other species. For PDs, the C/T ratios were calculated for every non-human sequence aligned to the PD-containing human sequence and recorded with the pairwise sequence identity. This counting procedure was repeated for every alignment of sequences, examining both CPDs and PDs.

3.2.3.1 Linear model of ‘in sphere’ sequence conservation

C/T ratios were tested to see whether they could be approximated with a linear model. Regression showed that data could indeed be appropriately described by a linear equation (Pearson correlation coefficient was -0.79 for CPDs and -0.82 for PDs). Constrained linear regression was performed on each of the datasets, using the `lm` test implemented in `R`. The constraint used was to force lines of best fit to pass through the $(100, 0)$ point, as further discussed in Section 3.3.3.1. Lines of best fit describing the CPD and PD datasets were tested: t-test was used on the line slope estimators using the null-hypothesis that the slopes of these two lines are not significantly different.

3.2.4 Division into buried and surface mutations

Every structurally-mapped mutation was evaluated in terms of relative solvent accessibility, using a local implementation of the Lee and Richards algorithm (Lee and Richards, 1971) on ATOM records in PDB files (for more details of the algorithm, see Section 2.2.1). According to criteria by Miller *et al.* (1987), all residues with relative solvent-accessibility greater than 5% were considered *surface* residues, whereas residues with $rASA^c \leq 5\%$ were termed *buried* residues.

3.2.5 SAAPdb analysis of CPDs

The disease-associated mutation was labeled by the SAAPdb pipeline as having or not having each of the likely structural effects presented in Table 2.2. It is worth noting that one mutation can be assigned multiple likely structural effects, and that the sequence conservation category was not used in this chapter, as the focus was on the structural effects of compensated mutations.

The fraction of CPDs (or PDs) determined to cause a structural effect, F_{cat}, was calculated as:

$$F_{cat} = N_{cat}/T_{cat} \tag{3.4}$$

where *cat* is one of the SAAPdb categories of structural effects, N_{cat} is the number of CPDs (or PDs) annotated by the SAAPdb as resulting in that structural effect, and T_{cat} is the total number of mutations in the CPD (or PD) dataset. The difference between calculated fractions of the two datasets was tested by a two-tailed Fisher's exact test[9] for statistical significance in each structural category. Two significance thresholds were used to identify categories with significantly different frequencies in CPD and PD datasets: $p < 0.01$ and $p < 0.05$.

Next, each SAAPdb category was assessed for significance after the correction for multiple testing was applied. The method was introduced in Section 2.3.3; in brief, the obtained p-values above had to be multiplied by 14 in order to be greater than 0.01 and 0.05 and therefore significant, since Fisher's exact test was repeated 14 times on the same dataset[10].

3.2.5.1 Monte Carlo simulations

Because of the division of data into PDs and CPDs via a negative observation, a Monte Carlo simulation (Kroese *et al.*, 2011) was used to test how likely it is to obtain the same significance values by chance.

[9] introduced in Section 2.3.2

[10] the 15^{th} category, 'explained', is calculated as a combination of all 14 structural categories and is therefore not an independent analysis

The dataset contained 447 CPDs and 1753 PDs found in SAAPdb[11]; these data points were merged and 447 mutations were chosen at random to create set A, the remaining 1753 being set B. For each of the structural explanation categories, a p-value was calculated (as before, using the Fisher's exact test) based on this random division of the data. The random division and calculation of p-values was repeated 10000 times and, for each structural explanation, the fraction of 'random p-values' that were lower than the observed p-value was recorded.

3.2.6 Potential compensatory mutation examples

The four compensation examples presented in Figures 3.12–3.15 were created using RasMol (Sayle and Milner-White, 1995). Simple modelled structures shown in subfigures **C** and **D** of each figure were obtained using `mutmodel` (Martin *et al.*, 2002), each time replacing a single sidechain using the minimum perturbation protocol (Shih *et al.*, 1985), where the sidechain's torsion angles are rotated to find the optimum orientation.

3.3 Results and discussion

This project expands previously mentioned studies by carefully selecting the largest presently known CPD dataset, described in Section 3.3.1. Section 3.3.1.1 compares and contrasts properties of the dataset introduced here, CPD_{AB}, and other publicly available datasets of CPDs. In order to verify the quality of the dataset, Sections 3.3.1.2–3.3.1.4 summarise the spread of CPDs over human protein space, protein family space and within protein families, respectively. Section 3.3.1.5 discusses the frequency of compensated mutations among deleterious mutations, followed by an

[11]the remaining 9 CPDs and 97 PDs were lost owing to different OMIM versions between the version used in this work and the one used previously when SAAPdb was built

analysis of preferences for amino acid types among both CPDs and PDs in Section 3.3.2.

The main goal of this project was to examine the location of compensatory mutations within protein structure, and the nature of pathogenic mutations which can be compensated. More precisely, it provides a structural context to the CPDs by answering the following questions: (i) in Section 3.3.3.1, how conserved are their structural surroundings, (ii) in Section 3.3.3.2, where in the protein structures are CPDs prevalently found, and finally (iii) in Section 3.3.4, which effects these mutations have on protein structures.

3.3.1 The CPD dataset

2328 disease-associated mutations from OMIM (McKusick, 2000; Amberger *et al.*, 2009) occurring in 245 human proteins were successfully mapped both to a residue in a UniProtKB/Swiss-Prot (Boeckmann *et al.*, 2003) sequence and to a structure in the Protein Databank (PDB) (Berman *et al.*, 2000). Of these, 453 mutations (19.46%) were found as a native residue in at least one non-human aligned functionally-equivalent protein sequence and thus annotated as CPDs. The remaining 1875 mutations (80.54%) were labelled as PDs. This dataset of CPDs and PDs will hereafter be referred to as CPD_{AB}. Considering that the PD dataset was based on negative observation, it is possible that, with expansion of known sequence and structural space, some PDs will become CPDs. On a similar note, according to Dawson *et al.* (2010), sequence quality must be considered, thus some of the CPDs might have been false positives owing to experimental errors.

In the last decade, there have been several analyses of CPDs (Kondrashov *et al.*, 2002; Kulathinal *et al.*, 2004; Poon *et al.*, 2005; Ferrer-Costa *et al.*, 2007; Zhang *et al.*, 2010; Barešić *et al.*, 2010). All methods obtain a dataset of DAMs, decide on the homologues to be searched for compensated mutations, align these homologues to the

Table 3.1: Datasets of compensated pathogenic deviations described in the literature. *Table obtained from* Barešić and Martin (2011).

Dataset	Species	Identity cut-off	Alignment method	Human° proteins	# DAMs	#CPDs
Kondrashov	Any		CLUSTALW	32	4880^{+}	608
	mammals†	$> 50\%$		3		20
Kulathinal	Diptera			475°	1527	6
Ferrer-Costa*	Any	$\geq 10\%$ ($> 60\%$)	Pfam	287 (24)	9334	1658 (52)
	mammals			184		847
Barešić	Any	None *	MUSCLE	245	2328	453
Zhang-missense	Human,$^{\diamond}$ neanderthal, chimpanzee		ANFO	2628	44348	62
Poon Set-A	Any			$43^{\ddagger}$	115	88
Poon Set-B	Any			$17^{\ddagger}$	59	49

The Poon Set-A includes mutations brought about by mutagenic agents while Set-B does not.

$^{+}$ Precise numbers are somewhat unclear. They report 608 CPDs and that this is approximately 10% of DAMs. In table 1 of their paper (Kondrashov *et al.*, 2002), there are 4272 'known missense' mutations which are most likely PDs since the last row of the table has more CPDs than 'known missense' mutations. This makes a total of 4880 (4272+608) DAMs.

† Kondrashov tested all found orthologues (with no sequence identity threshold) for CPDs and then switched to mammalian-only orthologues to identify compensatory mutations

° In the Kulathinal dataset (Kulathinal *et al.*, 2004), the reference species is *D. melanogaster*

* Numbers in parentheses refer to the CPDs used for structural analysis (Ferrer-Costa *et al.*, 2007)

* Functional-equivalence among homologues used instead of a sequence identity threshold (Barešić *et al.*, 2010)

$^{\diamond}$ Dataset originates from Zhang *et al.* (2010)

‡ There is no reference species in the work of Poon *et al.* (2005)

DAM-containing sequence and finally, identify compensated mutations. As already mentioned in Section 3.1.4.3, despite all the authors using a very similar definition of CPDs, the methodology implementing the steps listed above differs significantly. Therefore the review by Barešić and Martin (2011) attempts to compare and contrast all published CPD datasets, and Table 3.1, reproduced from that review, is presented here and referred to throughout this chapter.

3.3.1.1 The use of functionally-equivalent proteins instead of close homologues

The most important difference in building CPD_{AB} is the use of functionally-equivalent proteins (FEPs) rather than orthologues derived from Pfam (Finn *et al.*, 2006) (as used by Ferrer-Costa), or from BLAST (as used by Kondrashov). In both papers, orthologues had to satisfy sequence identity thresholds to ensure that diverged homologues were not used for the CPD identification: Ferrer-Costa used protein families defined in Pfam and removed homologues with $< 10\%$ sequence identity with the human sequence, and Kondrashov used only human proteins that can be aligned to a minimum of three sequences with $> 50\%$ sequence identity each. Some of these orthologues could have diverged in function and, where they have, key functional residues were, by definition, subjected to mutation (McMillan and Martin, 2008). While the broader sets of sequences used in other work have lead to identification of additional CPDs, using more restricted sets of FEPs obtained from the FOSTA database (McMillan and Martin, 2008) ensured that this situation will not arise. Finally, this project set out to compare compensated an uncompensated SAAPs in the broadest sense possible, and for that the accuracy of the PD subset was crucial: limiting ourselves to CPDs restricted to recently evolved homologues would hinder this attempt.

Ferrer-Costa *et al.* (2007) identified a significantly larger set of 811 human proteins containing mutations (compared with 245 in CPD_{AB}). Many of these mapped only to homologues of high sequence identity (Table 3.1), whereas the CPD_{AB} dataset included only mutations mapped to structure. In addition, they extracted mutation data from UniProtKB/Swiss-Prot annotations, resulting in a different set of mutations from those identified from OMIM here. 35% of the larger (sequence-based) Ferrer-Costa disease-associated protein dataset contained at least one CPD location, but the relative accessibility analysis was based on only 24 proteins with available protein structures.

All other CPD datasets listed in Table 3.1 either focused on the recently diverged homologues when searching for CPDs (Kulathinal and Zhang datasets), or provided no new structural features of gathered CPDs (Kondrashov). Moreover, CPD detection by Kondrashov and colleagues was based on a small number of proteins reported to have large numbers of pathogenic deviations (at least 50 per protein). As a result, the percentage of human proteins containing a CPD in the Kondrashov dataset was significantly higher than in the CPD_{AB} and Ferrer-Costa datasets, at the same time adding bias towards protein families with many reported DAMs. The Poon (2005) dataset is removed from further consideration as a significant fraction of this dataset belongs to artificially induced mutation, and thus is not suitable for discussion of natural emergence of compensated mutations. To conclude, so far the CPD_{AB} seemed the best compromise among publicly available datasets of CPDs, and as such was suitable for comparing with uncompensated deleterious mutations.

3.3.1.2 Redundancy of CPD-containing human proteins

2328 OMIM mutations, successfully mapped to a protein structure, were found in 245 human proteins. 85 of these proteins contained at least one CPD. The distribution of these 85 proteins over the human sequence space was tested, in order to identify whether CPD_{AB} dataset is biased towards a small number of redundant human proteins.

The sequence identity over the whole pairwise alignment length was calculated for every pair of CPD-containing human proteins by an in-house implementation of the Needleman & Wunsch algorithm (1970), and was used as a measure of sequence redundancy. The distribution of these $85 \times 84/2 = 2570$ sequence identity values is shown in Figure 3.5. With the mean sequence identity of 6.79% between two CPD-containing human proteins, CPD_{AB} is not only an extensive dataset, but it also shows very little redundancy.

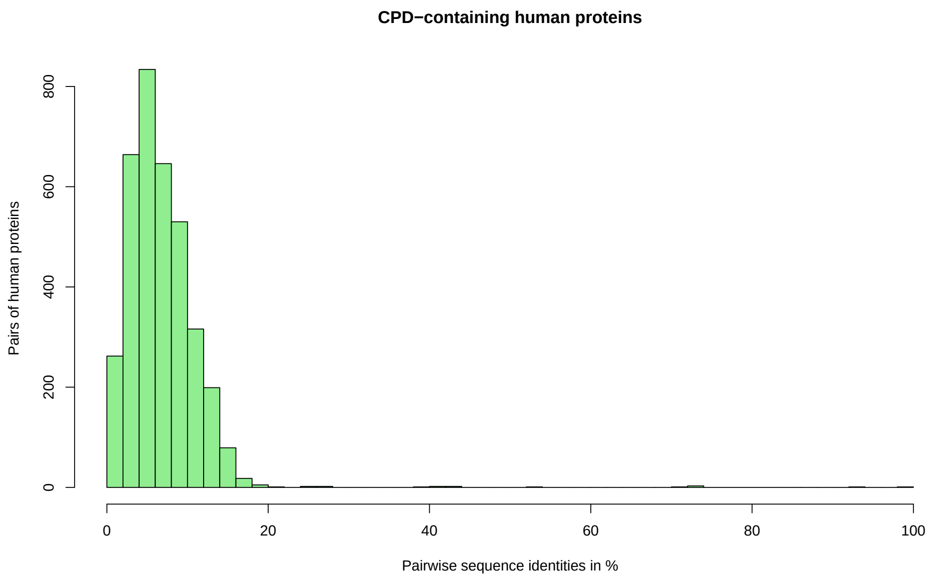

Figure 3.5: Diversity of CPD-containing human proteins.
Diversity was calculated as the mean pairwise sequence identity of each pair of human protein sequences. Mean standard sequence identity (average of 2570 data points) was 6.79%, with standard deviation of 4.92%.

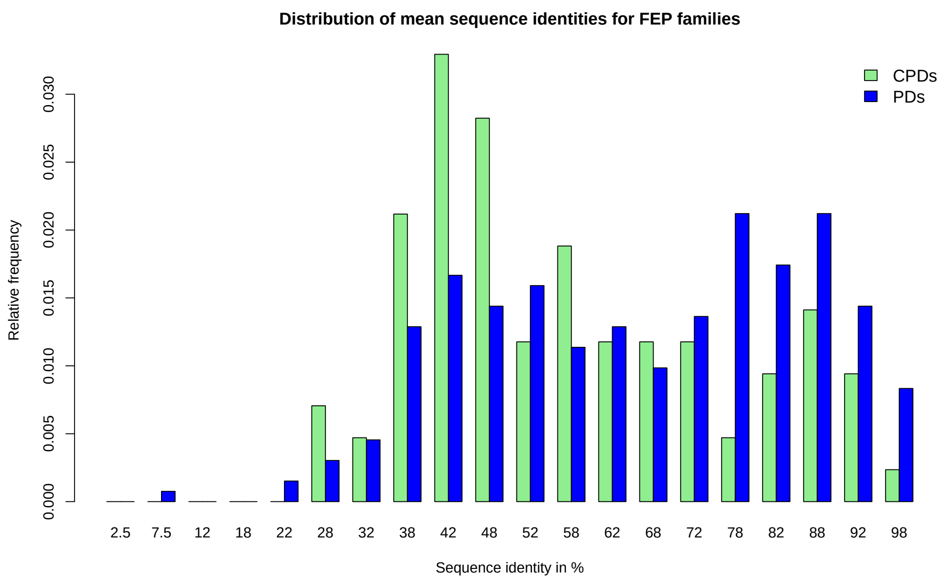

Figure 3.6: Diversity of FEP families containing PDs (264) and CPDs (85).
Some families may occur in both datasets. The histogram is normalized such that the total height of the bars is the same between the two sets. Diversity was calculated as the mean pairwise sequence identity within the family.

3.3.1.3 Redundancy within FOSTA families

In order to identify whether there was any bias in alignments used to identify CPDs, all FOSTA families were examined in terms of their sequence redundancy. Again, redundancy was measured as an averaged pairwise sequence identity, but this time for all the sequence pairs within one FOSTA family (a mutation-containing human sequence and all functionally-equivalent sequences).

Mean family sequence identities are shown in Figure 3.6. First, both CPD and PD datasets presented in this study were evenly spread across families with different levels of diversity. Second, while compensatory events were more common in more diverse families (i.e. those which, on average, contain more distantly related members), they occurred even in families which show very low diversity. In summary, FOSTA alignments sufficiently cover families of FEPs with both ancient and recent common ancestors.

3.3.1.4 Sequence identity distribution over human-FEP pairs

As previously mentioned, earlier work on CPDs mainly focused on mutations found compensated in relatively close homologues (Kondrashov *et al.*, 2002; Kulathinal *et al.*, 2004; Zhang *et al.*, 2010). Since there was no sequence similarity threshold imposed during the CPD_{AB} dataset build, the question arose of what the distances were between the human sequence containing disease-associated mutations, and the functionally-equivalent homologue in which the mutated residue type was found as native. A distance from the human sequence, in terms of pairwise sequence identity, was measured for every CPD-containing FEP sequence, e.g. ANT3_BOVIN and ANT3_SHEEP in Figure 3.1, and pairwise sequence identity was recorded, yielding 3218 human-FEP pairs for 453 CPDs, shown in Figure 3.7.

This graph shows that there is a significant number of CPDs within moderately and highly sequence-diverged homologues, which have probably been excluded from the previous studies. In addition, three peaks in the human-FEP distribution (Figure 3.7) are suggested ($\sim 85\%$, $\sim 70\%$ and $\sim 45\%$ possibly corresponding to mammalian, other-eukaryotic and prokaryotic homologues (Kondrashov, personal communication)).

3.3.1.5 Prevalence of compensation among disease-associated mutations

To summarise the comparison of the datasets presented above, there are several advantages of the algorithm utilised in creation of the CPD_{AB} dataset. First, there were no assumptions made about the distance of the homologue from the human sequence, and the sequence features of the homologue; all restrictions were based on functional annotation. Second, there were no restrictions imposed on the number of sequences in the multiple sequence alignment, or the number of compensated mutations observed in the alignment. Third, by choosing mutations from a widely used resource like OMIM, it was ensured that mutations occurred in proteins that were often sufficiently interesting for protein structure to have been solved. In turn, this guaranteed a high level of cross-referencing between sequence and structure, thus enabling structural characterisation of the large fraction of mutations.

Similar to the Ferrer-Costa dataset of CPDs[12], CPD_{AB} had 19.5% compensated mutations. While some of the CPDs presented in this work may be false positives, detected in distant homologues and possibly resulting from poorly built alignments in highly variable regions, it will be interesting to follow the change in compensation frequency as more sequences become available from high-throughput sequencing, and as FOSTA grows in terms of family number and sizes. The current projection is that,

[12] where 18% (1658/9334) of the larger, sequence-based dataset of mutations were found to be compensated

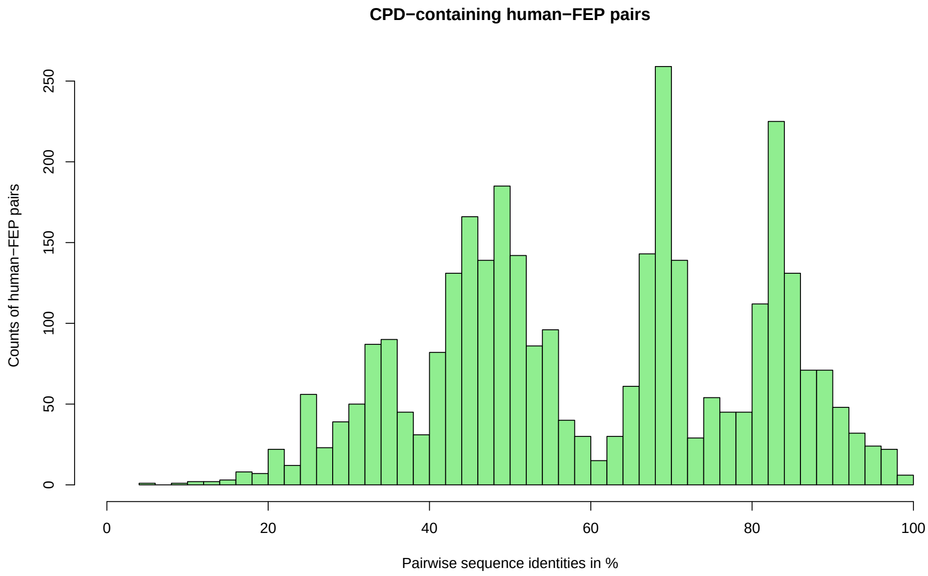

Figure 3.7: Distance between the human sequence and FEP sequence containing mutated residue type as native.
It should be noted here that several homologues could have been identified for a unique disease-associated mutation.

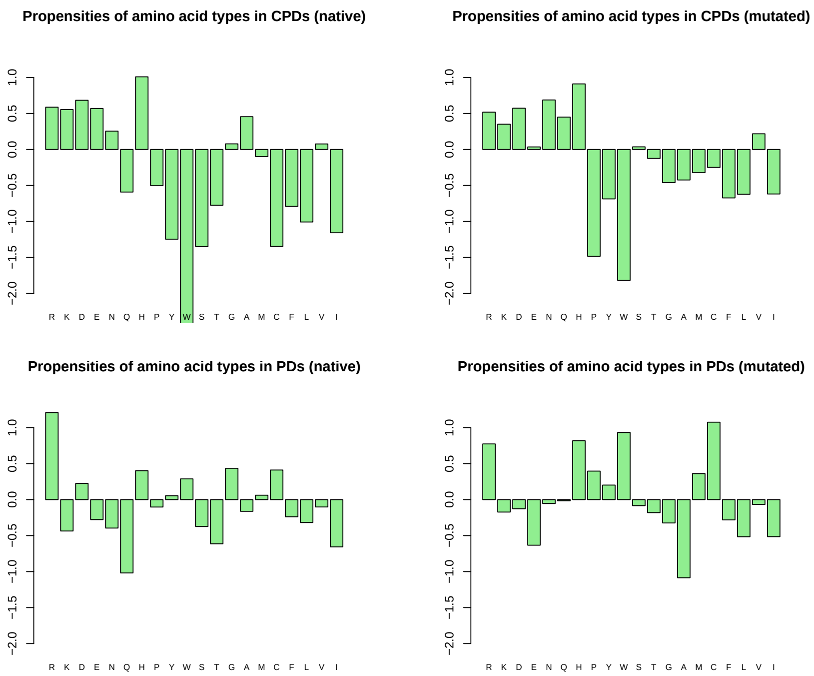

Figure 3.8: Amino acid propensities in PDs and CPDs.
Amino acid types have been ordered by ascending hydrophobicity on the Kyte & Doolittle scale (introduced in Section 4.1), from left to right.

when no sequence similarity cut-offs are imposed on the orthologues, compensation occurs once for every five DAMs.

3.3.2 Distribution of amino acid types among mutations

Both CPDs and PDs were considered in terms of their native and mutated amino acid type preferences, as shown in Figure 3.8. A positive value, e.g. H in 'CPDs-native', denotes that the histidine was overrepresented as a native residue among CPDs, when compared with the frequency of histidines among human proteins from which the mutations were extracted from. In other words, mutations from histidine to another amino acid were more common than expected among CPDs.

These propensities are interesting to analyse in the light of amino acid 'age'. According to Jordan *et al.* (2005), amino acid types can be divided into ancient ones (P, A, G and E) and the more recently emerged ones (F, C, M, H and S). Furthermore, since the recent ones had less time to evolve and sample their roles in different positions in the proteins, it is to be expected that protein sequences get depleted in the ancient amino acid types, at the account of more recent ones[13]. Indeed, both PDs and CPDs mostly follow this trend, with some exceptions. There are fewer methionines and histidines in 'CPDs-mutant' than in 'CPDs-native': presumably these amino acids are involved in functions or interactions for which it is hard to compensate using another amino acid. Additionally, proline should be less common among mutant than among native residues; among PDs the opposite was observed, indicating that the introduction of proline is often pathogenic, and probably requires complex and/or multiple changes fully to compensate for its effect on protein fitness.

While no clear trends emerged in this survey, it will be interesting to repeat this analysis when SAAPdb has been updated, and both the PD and CPD datasets increase.

[13]where this substitution is not detrimental to the protein

3.3.3 CPD localisation in the protein structure

3.3.3.1 Sequence conservation within the sphere

This analysis compared frequencies of mutations occurring in the residues surrounding a CPD or a PD in the structure, i.e. it is set to search for the 'sphere compensation'. In common with Kondrashov *et al.* (2002), it hypothesizes that compensatory mutations, neutralizing a CPD's pathogenicity, are likely to be physically close to a CPD and, more precisely, participate in short-range interactions with the compensated mutation. To define residues making a one-residue-wide sphere around the mutated residue in the protein structure, all residues with any atoms within 8Å distance from any atoms in the CPD residue, based on PDB atom coordinates, were considered 'in sphere' residues.

Figure 3.9 shows the distribution of sequence variability in residues surrounding a CPD, compared with PDs. C/T ratios (where C was the number of local potentially compensatory mutations and T was the total number of 'in range' columns in the alignment checked for that sequence, see Section 3.2.3) represent the fraction of in-range residues that are mutated. Ratios were also taken for PDs in order to control for sequence variability. Owing to the great number of points on the graph, and in order to see if there is any major difference between the two datasets, C/T ratios were averaged for each dataset in 1% sequence identity bins, as shown in Figure 3.10. Indeed averaged C/T ratios indicated an increased sequence variability around CPDs, when compared with spheres around the PDs.

Restrained linear regression was performed on the full datasets to obtain lines of best fit, the restraint being the biologically obvious condition that both lines have to pass through 0 mutations when the sequence identity is 100%. The line equations show a significant increase in the slope for the CPD dataset (Z-statistic=7.860, with $p < 0.05$). When the restraint was removed and the linear model was built again, the Z-statistic increased, indicating still significantly different 'in sphere' conservation

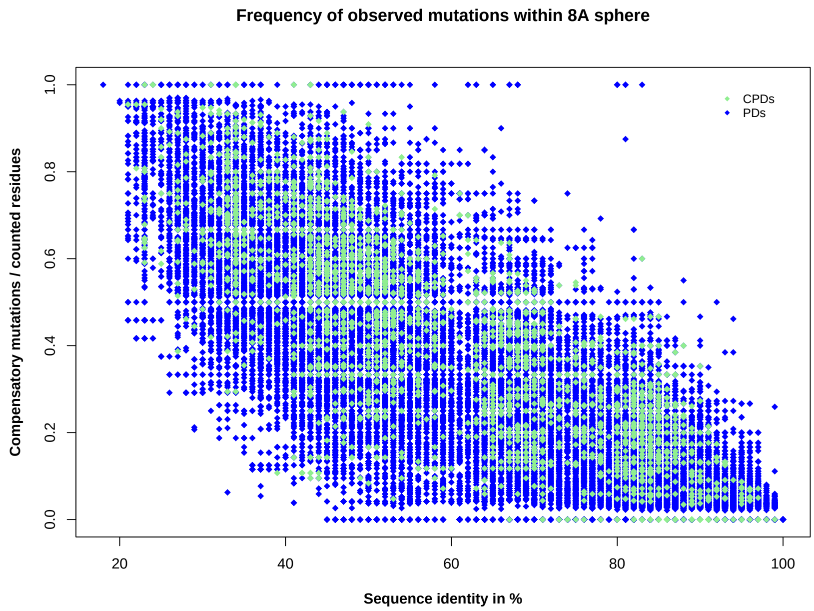

Figure 3.9: Dependency of the local mutation ratio on sequence identity.
The C/T ratio for residues within an 8Å sphere of each mutation is plotted against pairwise sequence identity for both CPDs and PDs.

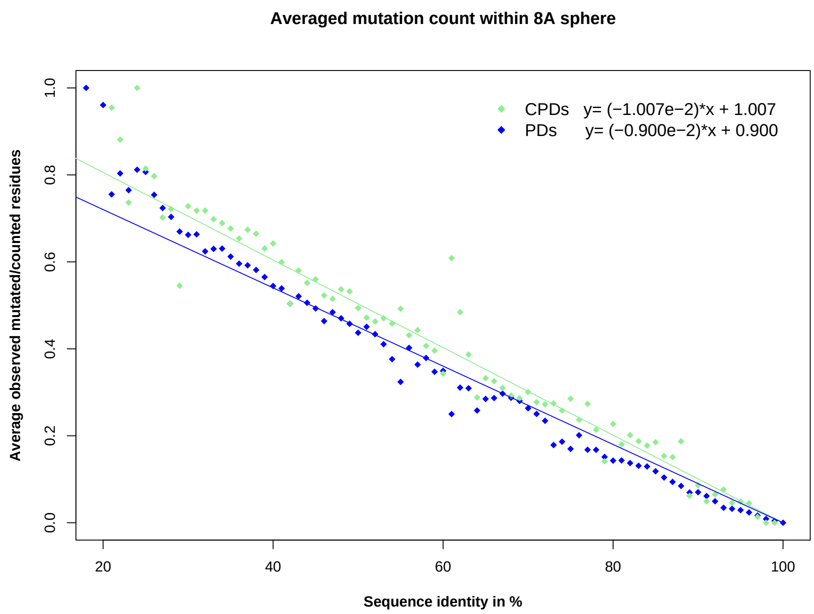

Figure 3.10: Dependency of the local mutation ratio on sequence identity.
The line of best fit, obtained by linear regression with a (100, 0) constraint for both complete datasets (i.e. the data shown in a): 3138 data points for CPDs and 74429 data points for PDs) is shown together with the average C/T ratio for each 1% sequence identity bin to illustrate the trends in the data.

trends between CPDs and PDs. This increase in the number of diverged residues in the structural neighbourhood of CPDs strongly supports the hypothesis that compensation is commonly a local effect, as previously suggested by Kondrashov *et al.* (2002).

Now let's consider the values of the slopes obtained by linear modelling. For the CPDs in Figure 3.10, the best-fit line had a slope of -1.007. Considering that the -1.00 slope corresponds to random mutation events happening at the rate proportional to the average sequence identity for the two sequences in question, it is obvious that CPDs reflect a set of random mutational events, i.e. they are the direct result of the random genetic drift. In contrast, the lower slope of the PD linear model (-0.9) points to spheres with higher conservation than expected by random drift (for structural or functional reasons), meaning that compensation is less likely to occur within these spheres. In other words, if the environment of a residue is conserved through evolution owing to some functional or structural constraints, a mutation is less likely to be easily compensatable. This decreased likelihood of compensation in more conserved regions corresponds to previous conclusions made by Ferrer-Costa *et al.* (2007) about CPDs occurring in less structurally constrained locations, provided the residues are assumed to be conserved owing to structural, and not functional constraints.

3.3.3.2 Buried vs. surface mutations

An average relative solvent accessibility was calculated for all compensated and uncompensated mutations in CPD_{AB}. With the mean $rASA = 43.4 \pm 28.0\%$ for the CPDs and mean $rASA = 26.9 \pm 27.2\%$ for the PDs, the CPDs presented higher propensity for solvent-exposed residues in the protein structure. Again, this analysis confirmed, on a significantly larger dataset, previous results by Ferrer-Costa *et al.* (2007) indicating that CPDs prefer more solvent-accessible positions. In the case of a surface residue, it is reasonable to assume that compensation may appear through interaction with other mutated molecules, although Poon *et al.* (2005) claim intra-chain compensation is a lot more common.

In order to test whether the average 'in sphere' conservation differs between buried and surface residues, CPDs and PDs were divided based on solvent accessibility, as described in Section 3.2.4. Lines of best fit were then modelled for buried/surface CPDs and buried/surface PDs. The same procedure was repeated with two buried/surface thresholds: $rASA = 5\%$ and $rASA = 10\%$ [14].

Table 3.2: Linear models of conservation in buried and surface CPDs and PDs.

	Best-fit line slope	Best-fit line r^2 [b]	Correlation coeff.[c]
CPDs all	-1.007×10^{-2}	90.5%	-0.79
CPDs buried (≤5%)[a]	-1.040×10^{-2}	96.0%	-0.81
CPDs buried (≤10%)	-1.025×10^{-2}	95.7%	-0.79
CPDs surface (>5%)	-0.999×10^{-2}	89.1%	-0.78
CPDs surface (>10%)	-1.001×10^{-2}	88.9%	-0.78
PDs all	-0.900×10^{-2}	84.5%	-0.82
PDs buried (≤5%)	-0.894×10^{-2}	89.7%	-0.86
PDs buried (≤10%)	-0.892×10^{-2}	89.4%	-0.86
PDs surface (>5%)	-0.906×10^{-2}	85.1%	-0.82
PDs surface (>10%)	-0.909×10^{-2}	84.5%	-0.82

[a]Numbers in brackets indicate which rASA values were used as a cut-off value
[b]adjusted r^2 from `lm` test implemented in `R`
[c]Pearson correlation coefficient

The Pearson correlation coefficient between the linear model and data was calculated for every combination; high correlation coefficients obtained justified the approximation of C/T ratios with a linear correlation, see Table 3.2, fourth column. As shown in the second column of Table 3.2, the slopes of the lines of best fit for all buried/surface cases were almost indistinguishable from the equivalent lines in the full datasets, with CPD slopes close to -1.00 and somewhat lower PD slopes. The only notable difference was that CPDs showed a slight increase in slope being -1.025 or -1.040 when only buried CPDs were taken into account. These values suggest that, when CPDs occur in the protein core, they

[14] $rASA \leq 5\%$ or $rASA \leq 10\%$ classified as buried, and $rASA > 5\%$ or $rASA > 10\%$ classified as surface

are accompanied by a somewhat higher than random local mutation rate. The changes in PD slopes when only buried or surface PDs were considered were negligible.

3.3.4 Structural analysis of the effects of CPDs

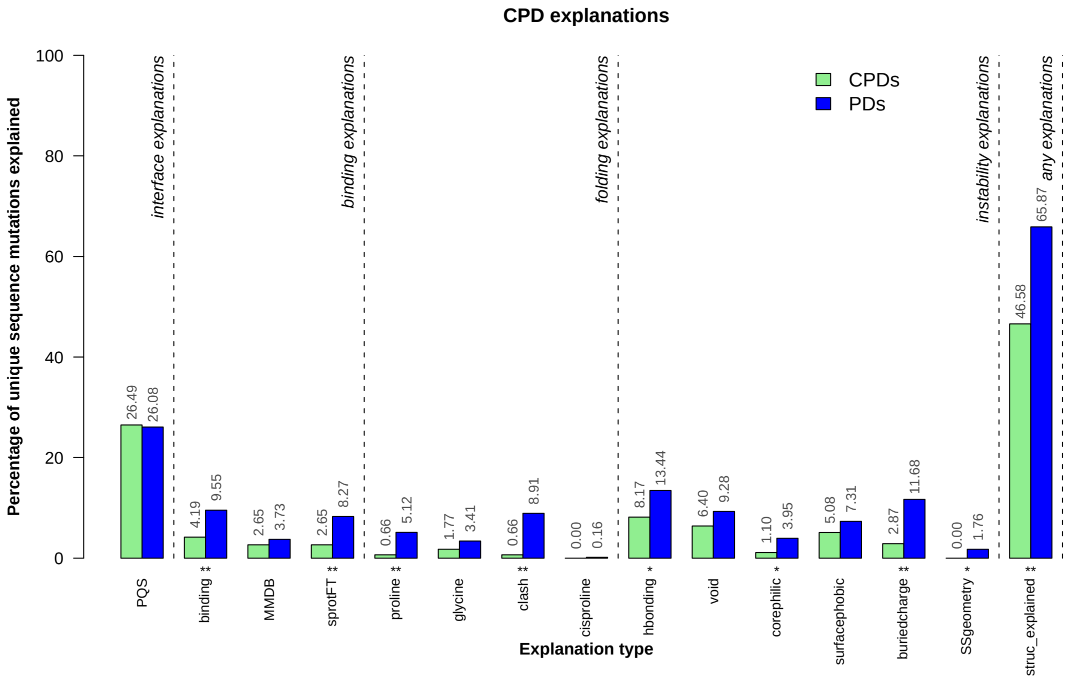

Figure 3.11: Relative frequencies of predicted structural effects for CPDs and PDs. Values are indicated at the top of each bar. Significantly different bars (Fisher's exact test, see Table 3.2) after application of the Bonferroni correction for multiple testing are indicated (* $p < 0.05$, ** $p < 0.01$). Each bar represents F_{cat}, as defined in Section 3.2.5. The 'struc_explained' bar is a summary representing explanation by any of the other structural tests shown in this figure. In the case of this category, no correction was applied giving $p = 6.71 \times 10^{-14}$.

Fractions of PDs and CPDs for which structural effects have been identified in SAAPdb are shown in Figure 3.11, grouped by classes and divided into categories of likely structural effects. Analysis of relative frequencies in fourteen categories covered four classes of disrupting effects: protein interface, binding properties, protein folding and stability. These categories were briefly described in Table 2.2 and have been explained in detail by Hurst *et al.* (2008). Differences between the two datasets give an insight into which types of structural disruptions are more likely to be compensated, showing that the compensation of pathogenic mutations is highly dependent on the nature of the mutation's effect on the structure.

3.3.4.1 Testing each SAAPdb category for CPD-PD difference

After mapping each OMIM mutation to the most reliable protein structure, as described in Section 3.2.1.2, SAAPdb was queried for each mutation, and structural effects likely to be caused by that mutation were extracted. 447 CPDs and 1753 PDs were found processed in the latest version of SAAPdb. For each structural category presented in Table 2.2, the fraction of the CPDs and PDs observed causing that structural effect has been listed in Figure 3.11, and the Bonferroni-corrected p-values from a Fisher's exact test have been listed in Table 3.3.

In order to compensate for the multiple testing on the same dataset which increases the probability of observing significant PD-to-CPD difference by chance, observed p-values were adjusted using rigorous Bonferonni correction. It should be noted here that, instead of evaluating which p-values are smaller than corrected α values, 0.01/14 and 0.05/14 respectively, all p-values were multiplied with 14. This is to facilitate data interpretation, as a person is likely automatically to evaluate presented p-values by comparing them with $\alpha = 0.01$ and $\alpha = 0.05$. Even after correcting for full dependence between SAAPdb categories, although these are at best only partially dependent, the dataset presented here clearly shows different trends in CPDs and PDs in categories: binding, UniProtKB/Swiss-Prot features, proline, clash and buried charge, as shown in Table 3.3. These results are further discussed for each class in Sections 3.3.4.3–3.3.4.6.

Table 3.3: Difference in frequencies in structural effect categories observed between CPDs and PDs, raw frequencies, and after correcting for multiple testing. p-values refer to a two-tailed Fisher's exact test ($d.f. = 1$); Bonferroni-corrected p-values are uncorrected p-values multiplied by 14, to allow them to be compared with conventional α values of 0.05 and 0.01. Significance levels are marked as: * $p < 0.05$, ** $p < 0.01$. The mc-value shows the result of a Monte Carlo simulation and is the fraction of random divisions of the data which obtain the observed uncorrected p-value or better (see text).

Structural category	Uncorrected p-value	Uncorrected significance	Corrected p-value	Corrected significance	mc-value
PQS[a]	0.86		$1^{\dagger}$		0.81
binding[b]	1.09×10^{-4}	**	1.5×10^{-3}	**	0.00
MMDB[b]	0.32		$1^{\dagger}$		0.31
sprotFT[b]	6.46×10^{-6}	**	9.04×10^{-5}	**	0.00
proline[c]	1.57×10^{-6}	**	2.20×10^{-5}	**	0.00
glycine[c]	7.05×10^{-2}		9.87×10^{-1}		0.07
clash[c]	5.68×10^{-13}	**	7.95×10^{-12}	**	0.00
cisproline[c]	1		$1^{\dagger}$		0.36
hbonding[d]	1.99×10^{-3}	**	2.79×10^{-2}	*	0.00
void[d]	5.16×10^{-2}		7.22×10^{-1}		0.05
corephilic[d]	1.32×10^{-3}	**	1.85×10^{-2}	*	0.08
surfacephobic[d]	0.10		$1^{\dagger}$		0.09
buriedcharge[d]	4.62×10^{-10}	**	6.47×10^{-9}	**	0.00
SSgeometry[d]	1.31×10^{-3}	**	1.83×10^{-2}	*	0.00

[a]Interface explanations
[b]Functional explanations
[c]Folding (fold-preventing) explanations
[d]Instability (destabilizing) explanations
$1^{\dagger}$ indicates corrected p-value was greater than 1

3.3.4.2 Confirming results with Monte Carlo simulations

Assignment of mutations as PDs or CPDs is based on a negative observation, i.e. that this mutation, known to cause disease in humans, has *not* been observed as the native residue in a FEP from another species. Consequently, the number of CPDs may be an under-estimate simply because FEPs have not yet been observed demonstrating that compensation can take place.

In order to test that the significance of the results observed above was not a result of random partitioning of the data, a 10000-iteration Monte Carlo simulation was run as described in Section 3.2.5.1. The results shown in Table 3.3, indicate that where the observed (Bonferroni-corrected) p-value was < 0.01, the probability of seeing this p-value by chance was zero (i.e. mc-value $= 0.00$ when $p < 0.01$). Where $p < 0.05$, there was a $> 91.7\%$ probability that the results were not obtained by chance (i.e. mc-value ≤ 0.083 when $p < 0.05$).

To conclude, even when using very stringent Bonferonni correction, some statistically significant differences between CPDs and PDs in terms of SAAPdb structural categories have been identified. The same trends have also been observed as significant during Monte Carlo simulations. In other words, random partitioning of the mutation data failed to replicate observed differences in frequencies between CPDs and PDs for some SAAPdb categories: these features are indeed specific for compensated missense mutations. Therefore, these results show compelling evidence that compensated and uncompensated mutations have different effects on the protein structure. The following four sections will cover the main classes of structural effects represented in SAAPdb, providing one example for every effect class: interface, binding, folding and stability.

3.3.4.3 Interface disrupting effects

Interface residues have been defined as surface residues in the monomer which undergo a change in relative accessibility of ≥10% on complex formation. 26.5% of CPDs and 26.1% of PDs occurred in interface residues found in PQS files (Henrick and Thornton, 1998). This is the only structural category for which the frequency of CPDs is the same, or greater than, the frequency of PDs. This finding was in agreement with the recent observations that CPDs are often found in residues having fewer intra-protein interactions (Ferrer-Costa *et al.*, 2007) (and hence have fewer structural constraints), indicating that it might be relatively easy to compensate for the detrimental effects of interface residues. An example of a compensated mutation in a protein interface is shown in Figure 3.12.

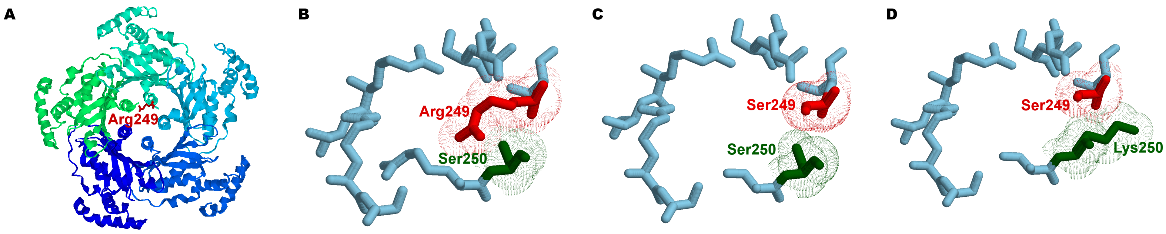

Figure 3.12: Potential compensation of a mutation affecting an interface residue. **a)** The position of Arg249→Ser is shown on the human GTP cyclohydrolase pentamer structure, PDB ID: 1FB1. This CPD occurs at an *interface* in the pentamer and causes dopa-responsive dystonia. **b)** Detail of Arg249 and its interaction with Ser250 from a neighbouring monomer. Multiple non-bond interactions between Arg249 and Ser250 contribute to pentamer stability. **c)** The Arg249→Ser mutation causes the loss of function in GCH1_HUMAN by losing multiple non-bonded interactions (modelled structure shown) and hence destabilizing its structure. **d)** The *Rickettsia bellii* FEP has compensated for the Ser249 lost contacts by introducing Lys250 (modelled structure).

3.3.4.4 Mutations affecting binding

A significantly greater fraction of PDs than CPDs was assigned as making specific binding interactions (hydrogen bonds defined according to the rules of Baker and Hubbard (1984), or non-bonded contacts) to a ligand or another protein chain (Figure 3.11, category 'binding'). Using data from the MMDBBIND database (Bader *et* *al.*, 2001) to identify binding residues rather than the PDB data, also showed a greater fraction of PDs than CPDs, but the difference was not statistically significant.

It is not surprising that, owing to the specific properties required for H-bonds or interactions at interfaces, these results showed compensating for a mutation at a specific binding residue is usually difficult. An example of a potential compensated mutation at a binding residue is shown in Figure 3.13.

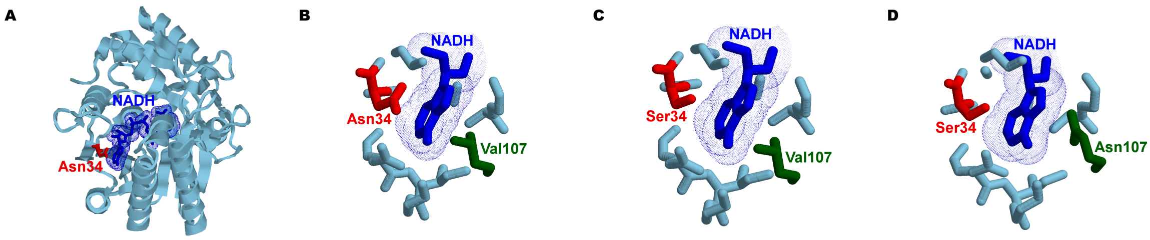

Figure 3.13: Potential compensation of a mutation affecting a binding residue. **a)** Asn34→Ser position is shown on the human UDP-glucose 4-epimerase structure, PDB ID: 1EK6. This CPD occurs in a *binding* site and in a *PQS* interface and causes epimerase-deficiency galactosemia. **b)** Detail of Asn34 and its interaction with NAD^+. **c)** The Asn34→Ser mutation causes the loss of hydrogen bond with the exogenous NAD^+, needed for the normal function of the human protein (modelled structure). **d)** The *Streptococcus thermophilus* and *Streptococcus mutans* FEPs have compensated for the Ser34 by introducing Asn107, which in turn stabilizes protein-ligand interaction, shown on the modelled structure.

3.3.4.5 Folding disruption effects

This class of structural effects describes cases where the mutation is likely to prevent correct folding of the protein and is represented by (i) mutations from cis-proline, to proline and from glycine (where backbone torsion angles are unfavourable for the replacement residue), and (ii) introduction of a bulkier, clash-causing residue. Mutations from cis-proline were very rare in CPD_{AB}, and were not considered further.

Mutations from another amino acid to proline are expected to be damaging to protein structure when the native residue has a backbone conformation disallowed by proline's cyclic sidechain. Previously shown results on amino acid propensities additionally confirmed that prolines were rare among CPD-mutant residues, obviously the complexity of the required compensation is large enough to overturn the trend of accumulation of proline in recent protein sequences, for more details see Section 3.3.2.

Substitution to a clash-causing residue was extremely rare among CPDs compared with PDs. This is not surprising as compensating for a clashing residue would probably need several, chronologically earlier, cascading compensatory mutation events to create a void large enough to accommodate the clashing residue; such a void would itself be destabilizing. A rare example of a clash compensation is observed in human triosephosphate isomerase FEPs, as shown in Figure 3.14.

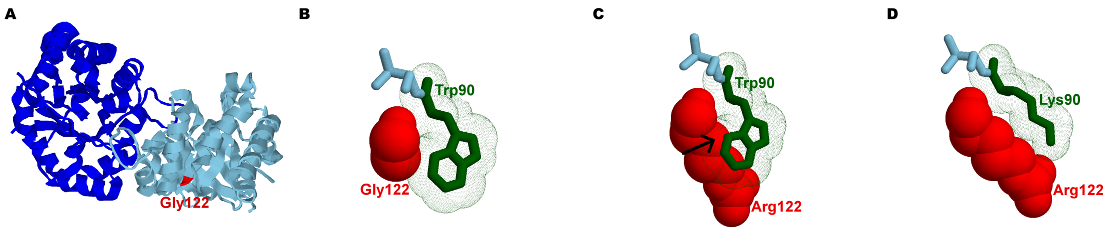

Figure 3.14: Potential compensation of a mutation affecting a folding residue. **a)** The position of Gly122→Arg is shown on the human triosephosphate isomerase dimer structure, PDB ID:1WYI. This CPD causes a *clash* and a *buried charge*, and increases thermo-sensitivity of the human protein. **b)** Detailed position of Gly122 and Trp90. **c)** The Gly122→Arg mutation causes atom clash, indicated by the arrow, between larger sidechain of Arg122 and native Trp90 (modelled structure). **d)** Substituting Trp90 with a smaller Lys compensates for the introduction of the Arg122 in several bacterial FEPs (*Aquifex aeolicus*, *Coxiella burnetii*, *Mycoplasma gallisepticum*, *Treponema pallidum*, *Xylella fastidiosa*, *Chromohalobacter salexigens*), shown on the modelled structure.

3.3.4.6 Mutations affecting protein stability

Mutations affecting protein stability introduce no physical barriers to *prevent* correct folding, but reduce the stability of the correctly folded form below that of unfolded or misfolded states (Hurst *et al.*, 2008). Disruption of hydrogen bonding, creation of voids, misplaced charges, hydrophillics, or hydrophobics, and disruption of disulphides all fall into this category. Such mutations may be temperature-sensitive (such as the Val143→Ala mutation in p53 (Martin *et al.*, 2002)) and are the main category of interest in 'rescuing' protein function (Bullock and Fersht, 2001; Friedler *et al.*, 2003; Friedler *et al.*, 2002).

Very few cases of disruption of disulphides were observed, and this category was not considered further.

Mutations that affect hydrogen-bonding were identified in SAAPdb according to the method of Cuff *et al.* (2006). Considering the fact that hydrogen bonds have a strong effect on protein stability (Cuff *et al.*, 2006) and that precise geometries are involved, it is not surprising that mutations affecting hydrogen-bonding were found very commonly in both datasets. The high frequencies in both datasets, 8.17% of CPDs and 13.44% of PDs, indicated a common occurrence of both mutation types in hydrogen bonding residues, although there are significantly fewer hydrogen-bond disrupting CPDs than PDs. This suggests that it is difficult to make compensatory mutations which counteract the disruption of the intricate hydrogen-bonding network in the protein core.

The creation of voids of volume $> 275\text{Å}^3$ did not show a significant difference between the CPD and PD datasets. The Cuff void calculation method (Cuff and Martin, 2004) calculates the volume of voids assuming that no movements occur in the protein structure. In reality it is likely that several small movements of sidechains and backbone will occur to fill the void (at least partially). Only if these movements are too great will the stability and function of the protein be disrupted. It appears that in the CPDs, voids can be compensated for by replacing one or more local sidechains with a larger residue. A number of small changes can compensate as effectively as a single larger change and these may be accommodated more easily if, in evolution, they occur before the CPD. Figure 3.15 shows an example of a compensated void mutation in glucose-6-phosphate dehydrogenase.

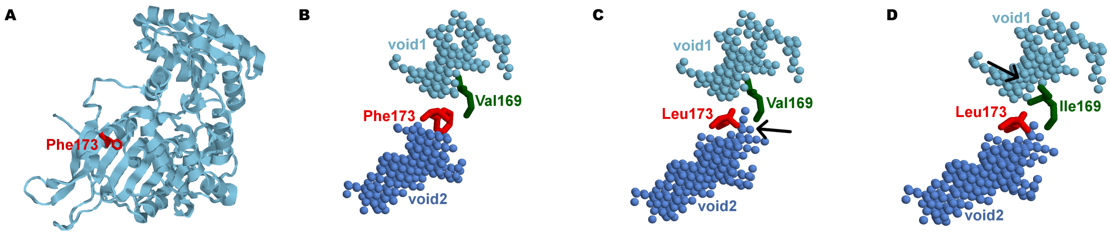

Figure 3.15: Potential compensation of a stability-reducing mutation. **a)** The Phe173→Leu is shown on the human glucose 6 phosphate dehydrogenase structure, PDB ID:2BH9. This CPD creates a *void* and causes neonatal jaundice. **b)** Detail of Phe173 and its relative position to Val169. **c)** Substitution of aromatic Phe173 with a smaller leucine creates an enlarged 'void2' in the protein core, indicated by the arrow (modelled structure). **d)** Several bacterial FEPs have compensated for the void creation by substituting Val169 with a larger residue: Leu, Ile or Met. The compensatory effect of Val169→Ile in *Buchnera aphidicola subsp. Schizaphis graminum* and *subsp. Baizongia pistacia* FEPs, shown here on a modelled structure. Introducing a larger isoleucine reduces the 'void1' size, increases the distance between the two voids, and in turn stabilizes the structure (indicated by an arrow). In **b)**–**d)**, only the two residues of interest are shown. The small spheres fill buried voids surrounding the residues and bounded by the rest of the protein structure.

Introducing a hydrophilic residue or an unsatisfied charge in the protein core (Hurst *et al.*, 2008) were significantly less likely to be compensated for, again, showing the great complexity of interactions among tightly packed buried residues. Compensating for a buried hydrophilic or charge would require introduction of a compensatory hydrophilic or charged residue (which, by itself, would be destabilizing) in a precise orientation in the core. The observation that such events are rare argues for the first DePristo hypothesis described in Section 3.1.4.2, in which phenotypically neutral compensatory mutations are introduced before the compensated mutation. Introducing a hydrophobic residue on the surface seems to be easier to compensate for, although a detailed analysis of multi-chain proteins and complexes with ligands would be required in order to explain these mechanisms fully.

In summary, frequencies of structural effects in both datasets presented here were quite similar to PD frequencies presented by Hurst *et al.* (2008). The differences in frequencies between overall counts per category (CPDs+PDs), and PD counts in that earlier work are a result of that PD dataset including other mutation sources in addition to OMIM. However, some categories typical for buried residues (such as introducing a hydrophilic residue, buried charge, clash and SS-geometry) show a striking difference between PDs and CPDs, indicating these effects are less likely to be compensated for.

3.4 Conclusions

The results presented here have three main novel aspects: (i) the orthologous proteins have been chosen on the basis of functional equivalence rather than sequence identity thresholds, (ii) CPDs have been surveyed in a structural context on a much larger scale than previous work and (iii) the range of surveyed effects of CPDs on protein structure is greater than in previous work. The SAAPdb database (Hurst *et al.*, 2008) was utilised to analyze the specific structural effects of CPDs in a range of structural categories, comparing them with PDs. The reliability of the analyses was increased by using data on functionally-equivalent proteins for the multiple sequence alignments, because even relatively similar sequences can diverge in function (McMillan and Martin, 2008). The large size of the dataset and its wide spread across different protein families appears sufficient for a broad structural analysis of human disease-associated single amino acid mutations and cases where these have been compensated in other species.

Compensation of disease-associated mutations is fairly common and should not be neglected when protein evolution and/or disease-associated mutations are researched. Compensation through epistatic interactions with compensatory mutation(s) is mostly intragenic (Poon *et al.*, 2005). The complexity of compensatory events ranges from the simple one-on-one scenario (which is less common, although several examples have been introduced throughout this chapter) to a series of compensatory events, usually in close spatial vicinity to the pathogenic mutation.

The ratio of the pathogenic mutations for which compensation has been observed is very methodology-dependent. In particular, it seems to increase when the conditions for homologues included in the alignments are relaxed, and based on the CPD_{AB} dataset and other datasets presented in Table 3.1, it is estimated that up to 20% of DAMs encounter full fitness reversal.

Obviously the quality of the dataset presented in this chapter could be further improved: an update of the presented set of analyses is planned once SAAPdb is updated with the recent mutation data and protein sequences (yielding more accurate FOSTA annotations and sequence-to-structure mappings). In addition, in an age when there is mass production of genomic data owing to the recent advances in sequencing technologies, one should exercise caution when analysing any mutation data and make sure falsely annotated mutations, which are actually the results of errors in the sequencing technology, are identified, and filtered out as false positive data.

In terms of their structural features, CPDs obtained in this project prefer protein surfaces, and in general, less conserved and structurally constrained areas. The analysis of structural surroundings of compensated mutations indicated that the variation and potential compensatory mutations occur mostly through random genetic drift, while uncompensatable pathogenic mutations tend to occur in more conserved protein structure segments. This preference of PDs for the more conserved environments is a novel finding, however, it is in agreement with preferential localisation of PDs in more buried residues: PDs often have more intraprotein interactions, and thus have more complicated structural restrictions. Moreover, this analysis confirmed that compensation is predominantly a local effect.

Structural analysis by the SAAPdb pipeline, which indicates the likely local structural effects of a mutation, showed important features of the CPD dataset. First, CPDs in humans were less often assigned any likely structural effect, suggesting again that they cause less significant disruption of local structure. This confirmed results by Ferrer-Costa *et al.* (2007), suggesting that CPDs cause 'milder' changes than PDs in physico-chemical properties. Second, CPDs often occur in interfaces. According to the first evolutionary model proposed by DePristo *et al.* (2005), introduction of phenotypically neutral mutations (which are then able to compensate for a CPD) is a necessary first step before a CPD mutation can occur. Previously a high occurrence of neutral mutations in interface residues was shown (Hurst *et al.*, 2008), and this may create an amenable environment for CPD occurrence. Thus it was not surprising to find the PQS-interface category being the only structural category having a slightly higher frequency of assigned CPDs than PDs (Figure 3.11). In contrast, disease-associated mutations were less likely to be compensated for when the residue had more complex intra-protein interactions (i.e. in the protein core), which would often require multiple compensatory events. Furthermore, based on structural categories as defined by SAAPdb, CPDs are more likely to be found among surface residues, with the exception of specific binding residues which make key hydrogen-bonding or van der Waals interactions across an interface. It is also possible that other factors may result in compensation such as changes in expression levels.

In conclusion, this chapter presents a detailed structural comparison of the occurrence of compensated pathogenic deviations. This analysis set out to confirm and expand the work done by Ferrer-Costa and colleagues, in which they have found CPDs to be preferentially located on the surface, on average in positions with less severe structural changes (in terms of amino acid volume and hydrophobicity change, and BLOSUM62 scores), and on average fewer structural restrictions than PDs. The analysis presented in this chapter, using an order of magnitude larger dataset of CPDs (for more details on datasets, see Table 3.1), confirmed aforementioned structural trends, and expanded them on a larger set of SAAPdb features than three utilised by Ferrer-Costa *et al.* (2007). Through a large-scale structural analysis, this analysis further confirmed the hypothesis that compensation tends to be a local effect, since local sequence variation around a CPD was greater than around sites of PDs in functionally-equivalent proteins of the same sequence identity. Thus we have begun to differentiate compensated and uncompensated mutations on the basis of their effects on protein structure. This gives us insights into evolutionary mechanisms and may shed light on pathogenicity in humans.

In the future, research on the coevolution of compensated and compensatory mutations will most likely focus on two main areas: (i) elucidating the exact mechanisms used by individuals to travel along the ridges of the fitness landscape and (ii) development of compensation prediction tools, mainly to be used for pharmaceutical purposes. The former task has so far been limited by the availability of high-quality protein sequence and structure data. With the recent popularisation of genome-wide association studies, much attention has turned to identification of so-called ‘cluster-reference triplets’ and identifying compensatory trends from only three, recently diverged sequences, e.g. Zhang *et al.* (2010) and Povolotskaya and Kondrashov (2010). The latter task of compensation prediction is interesting from the aspect of targeted reversal of known DAMs. Recently, the first Critical Assessment of Genome Interpretation (CAGI) challenge was held (Callaway, 2010). This series of challenges sets out to evaluate state-of-the-art mutation phenotype prediction tools in a transparent way. One of the datasets containing p53 mutations[15] evaluates potential function-rescue compensatory mutations for a list of known mutations deactivating this tumour-suppressor protein, adding a more practical and clinical aspect to this phenomenon.

[15] `http://genomeinterpretation.org.content/p53/`

Chapter 4

Characteristics of Protein Interfaces

The underlying motivation behind the project presented in this and the next chapter is to improve the coverage of single amino acid polymorphisms in SAAPdb with protein-protein interface information, currently limited by the availability of data on protein-protein interface structures.

In short, this chapter gathers protein-protein interface data and surveys a range of chemical, structural and family-specific features, comparing them to the rest of the protein surface. In other words, a dataset of surface segments and a list of features were prepared, to be used in the next chapter to build a predictor of putative interfaces on protein surfaces. This predictor will enhance the annotation of neutral and pathogenic SAAPs located in protein-protein interfaces, when added to the SAAPdb pipeline.

4.1 Introduction

Protein-protein interactions are fundamental for a range of cellular functions, e.g. cell cycle regulation, apoptosis, cellular motors, pathogen recognition, communication among cells, etc. Proteins vary greatly in the numbers of interactions they make with other proteins, from 'loners' with one interacting partner to 'hubs' interacting with dozens of other proteins, sometimes reusing the same interface for several binding partners over time (Keskin *et al.*, 2008). Comparative analysis of human interaction databases shows that the number of complexes greatly exceeds the number of interacting proteins, in humans (Futschik *et al.*, 2007) as well as in other species (Missiuro *et al.*, 2009); Bork *et al.* (2004) estimated an average of 3–10 interacting partners per yeast protein. Typically, the more advanced the species is on the evolutionary scale, the more connected the protein network is, indicating advancement in regulation of processes (Keskin *et al.*, 2008).

Unfortunately, the number of protein complexes deposited in the PDB (Berman *et al.*, 2000) is not representative of this great diversity, mainly owing to various experimental complications in co-crystallization of multichain protein complexes. Protein-protein complexes constitute only 50% of protein structures in the PDB[1], the remainder are monomers and proteins in complexes with nucleotide chains, small peptide chains and ligand molecules.

At the moment, the part of the SAAPdb pipeline identifying mutations in protein-protein interfaces consists of three analyses, all relying on the structural data available in the Protein Data Bank; SAAPdb structural effects of mutations have been defined in Section 2.1.7.2. 30% of mutations deposited in the current version of SAAPdb[2] are recognised by the `pqs` category. The `interface` category of SAAPdb is considered unreliable: it relies on PDB annotation of interacting residues in complexes, many of which are non-biological crystal contacts. Finally the `binding` category recognises

[1] as of October 2010

[2] 96954 mutations distributed over 2042 protein structures

residues involved in specific binding interactions with another protein chain or ligand. For more details on definitions of SAAPdb structural categories see Hurst *et al.* (2008).

In this chapter, sections 4.1.1–4.1.3 define an interface, and review approaches to interface identification, interface types, size and topology. Next, methods to build and analyse a novel dataset of protein-protein interfaces are presented, followed by Section 4.3.1 presenting the dataset of interfaces gathered and filtered for non-redundancy and high-quality structural data. Last, Section 4.3.2 reviews trends obtained for eight features on the aforementioned dataset of interfaces, comparing them with previously published results based on other datasets. This chapter concludes with a list of features validated as interface predictors, to be used by the machine learning tools to build a predictor in Chapter 5.

4.1.1 What is an interface?

An interface in general, defines the area of contact between two molecules in a complex. In the case of protein-protein interfaces, it is a subset of residues or atoms on the surfaces of both chains that participate in hydrogen bonds, van der Waals or electrostatic interactions with the interacting protein chain. Protein complexes display a wide range of binding affinities, from micromolar to nanomolar, thus corresponding to a change of free energy in the range of -6 to -19 kcal/mol upon complexation (Keskin *et al.*, 2008), and life spans from seconds to days (Janin *et al.*, 2008).

4.1.2 Identifying protein-protein interfaces

There are two prevailing approaches to identification of interface residues (or atoms) on the protein surface, both based on structural information available for the complex.

(1) Distance-based methods employ a distance threshold, e.g. $0.5 - 2$Å (Janin *et al.*, 2008) or $vdW + 1$Å (Negi and Braun, 2007), where vdW denotes the sum of the van der Waals radii of the two atoms being examined. Any atom within the defined distance of any atom in the interacting partner is labelled as an interface atom. In the case of residue-based analyses, this annotation is extended to complete residues containing at least one interface atom.

(2) Solvent-accessibility-based interface detection defines a cutoff for the decrease in relative solvent-accessibility upon complexation e.g. Jones and Thornton (1997).

The two methods are considered to be equally reliable, since most of the applied thresholds manage to detect the same interfaces. However, even a small change in distance or ASA cutoff can change the average size of interfaces detected, their boundaries, and the ratio to non-interface surface. This is one of the main causes of data inconsistency among interface analyses, and seriously complicates comparison of different, previously published methods (de Vries and Bonvin, 2008). Furthermore, some of the features analysed, like sequence conservation, are particularly sensitive to variations in interface definition (de Vries and Bonvin, 2008). The work presented here is based on interface residues defined using relative solvent-accessibility decrease upon binding, for details see Section 4.2.1.3.

4.1.3 The main properties of interfaces

4.1.3.1 Types of protein-protein interfaces

Protein-protein interfaces can be characterised in several ways. The simplest case is where two or more protein chains constitute a complex fold: if they fold cooperatively and are never found as stable monomers, these complexes are considered obligate complexes[3]. If constituents of a complex fold independently and can be isolated as functional proteins in both bound and unbound states, they are termed non-obligate or three-state folders[4]. Based on the reversibility of complexation, non-obligate complexes can be both transient and permanent in nature (Nooren and Thornton, 2003), while obligate complexes are by definition permanent. Furthermore, transient complexes can be divided based on the strength of binding, as shown in Figure 4.1. Finally, a distinction can be made based on the similarity among chains constituting a protein complex (usually presented in terms of sequence identity). For a complex of, for example two chains, a homodimer will consist of two identical chains, while a heterodimer will consist of chains of different sequence. For examples of homo/hetero-dimers and -trimers, see Figure 1.6 in introduction on mutations.

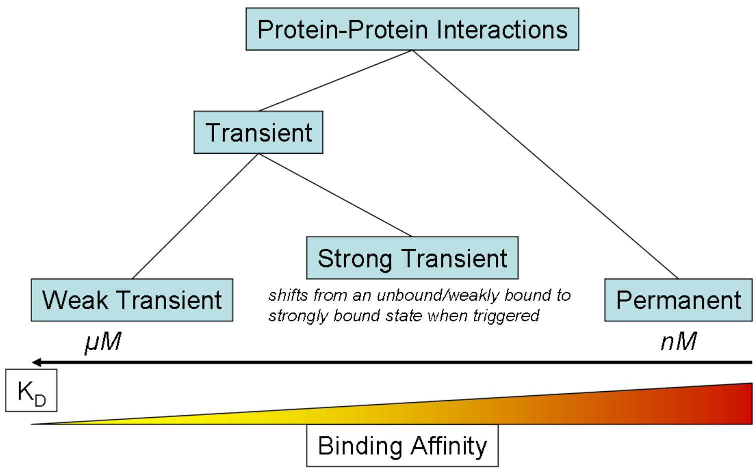

Figure 4.1: Interface types shown overlapping on a scale of binding affinities and dissociation constants.
Figure obtained from Perkins *et al.* (2010)*, with permission from the authors.*

[3]also termed two-state complexes, observed either in unbound and unfolded, or folded and bound form

[4]three states correspond to unfolded, folded unbound and folded bound

Figure 4.1 presents a summary of interface types, showing that all previously mentioned interface types actually form a continuum over a wide range of binding affinities (Perkins *et al.*, 2010).

4.1.3.2 Interface size

A typical protein-protein interface is a continuous patch of surface residues with an average area of $1600 \pm 400\text{Å}^2$, $\sim 800\text{Å}^2$ on each subunit surface (Lo Conte *et al.*, 1999). Homodimeric interfaces are larger (Bahadur *et al.*, 2003), with an interface surface of $\sim 1940\text{Å}^2$ (Yan *et al.*, 2008). When less stringent criteria for interface residues were employed, some authors obtained an average transient interface area of $\sim 2100 \pm 1250\text{Å}^2$, e.g. Headd *et al.* (2007). Most complexes have one continuous interface surface, or less often, one standard-sized patch and several adjacent smaller patches (Chakrabarti and Janin, 2002).

4.1.3.3 Solvent-accessibility of an interface

Interface residues, when unbound, are on average more solvent-accessible than the rest of the protein surface (Chen and Zhou, 2005). Porollo and Meller presented substantial differences between predicted relative solvent-accessibility values (as calculated by Lee & Richards (1971)) for interface residues and observed rASA in high-quality structural data (Porollo and Meller, 2007). Surprisingly, they further showed on several datasets using both support vector machines and neural networks, that the *difference* between predicted and observed values outperformed rASA values as predictors for interface residues. While this evidence clearly indicated a correlation between interface residues and solvent accessibility, this feature was left out of the list of predictors, under suspicion that it might not be an independent predictor of interfaces[5]. Whether this is really true, or adding rASA as a predictor improves

[5] classification of residues into surface or interface, introduced in detail in Section 4.2.1.3, is based on the difference between rASA in complex and monomeric form

accuracy of interface prediction is to be tested in the next chapter, when ASA-based interface predictor is benchmarked against the dataset of interfaces defined using the drop in solvent-accessibility criterion (for more details see Section 5.3.5.2).

4.1.3.4 Topology: core and rim model

In terms of topology, an interface in a complex is viewed as a *core* of solvent-inaccessible residues, surrounded by the *rim* of residues somewhat accessible to solvent (Chakrabarti and Janin, 2002). This exclusion of water molecules from the space of an interface core facilitates van der Waals contacts across the contact surface of two interacting chains. In order for the core to occur, an interface is required to have a minimum size of 600Å^2 (Bogan and Thorn, 1998). Distribution of amino acid types in the interface core is similar to the composition of protein core residues, while rim residues resemble the rest of the protein surface (Chakrabarti and Janin, 2002). Note that the ratio of core to rim surface, and consequently the size of the average interface, often depends on the choice of solvent-accessibility threshold to separate rim residues from non-interface residues.

Alanine-scanning mutagenesis has shown an uneven distribution of free energy across the core: only certain core residues, termed interface *hot spots*, significantly contribute to the binding free energy change, while mutations in the other core residues have less of an impact on binding affinity (Bogan and Thorn, 1998). Usually, a minimum of 2.0 kcal/mol increase in binding free energy upon mutation to alanine is used to define hot spots (Bogan and Thorn, 1998; Moreira *et al.*, 2007); these residues have distinctive amino acid composition with high propensities for (often highly conserved) polar amino acids (Hu *et al.*, 2000; Porollo and Meller, 2007). Li *et al.* (2004) showed that hot spots often occur in pockets on the protein surface, visible even in the protein's unbound state, hence providing a recognition pattern for the interacting partner. Thus it is now widely accepted that these electrostatic interactions are crucial for recognition between interacting partners (i.e. specificity), whereas the hydrophobic effect contributes to the stability of the bound complex (Moreira *et al.*, 2007).

4.1.3.5 Previously identified interface-specific characteristics

Throughout the last two decades, there have been numerous attempts to characterise a typical protein-protein interface. All approaches have been based on a similar idea: if an interface displayed a property uncommon on the protein surface, filtering segments of protein surfaces for that property could yield previously unidentified interface regions, even when the interacting partner and/or the orientation of molecules during interaction were unknown. The first analyses were performed on a limited number of protein structures available at that time in the Protein Data Bank (Jones and Thornton, 1997; Lo Conte *et al.*, 1999; Chakrabarti and Janin, 2002; Bahadur *et al.*, 2003). However, when sufficient datasets of interface-containing structures became available, common physico-chemical features of interfaces started to emerge (Neuvirth *et al.*, 2004; Bradford and Westhead, 2005; Yan *et al.*, 2008). No property has strong enough predictive power to be used as a successful predictor of interfaces on its own (Bradford and Westhead, 2005). The following sections review the most commonly cited interface properties, provided they have been mentioned (and similarly defined) in more than one protein-protein interface analysis.

All surveyed interface features can be grouped into three types:

Sequence-based features can be calculated from amino acid sequence alone. This includes *amino acid propensities* and *hydrophobicity*.

Structure-based features require structural data for that complex. Coordinates of atoms are used to calculate *planarity*, *protrusion*, *hydrogen bonding*, *disulphide bridges* and *secondary structure elements*. By definition, classification into an interface or non-interface surface atom or residue is a structure-based feature; for different approaches to interface definition see Section 4.2.1.3.

Profile-based features require information on homologues or functionally-equivalent proteins (FEPs (McMillan and Martin, 2008)). These homologues or FEPs are aligned to the interface-containing protein, providing family-specific interface properties, here: two *sequence conservation scores*, based on all homologues identified by a BLAST search and FEPs, respectively.

4.2 Methods

While there is a decent amount of accumulated knowledge on the topic of protein-protein interfaces, several common pitfalls were identified from the literature, some of which have been mentioned in Section 4.1.3.5. Without repeating these, the aim was:

- to obtain a thoroughly curated up-to-date dataset of protein chains and their interface segments (Section 4.2.1)

- to explore a range of sequence- and structure-based protein properties, to find interface-determining features (Sections 4.2.2–4.2.4)

Technical aspects of these procedures, and tools and resources used in the process are described hereafter.

4.2.1 Obtaining the dataset

4.2.1.1 Obtaining interface-containing structures

The first task when analysing interfaces was to obtain an extensive, well-annotated and up-to-date dataset of protein structures containing only biologically-relevant protein-protein contacts. In order to avoid inclusion of crystal contacts among interfaces, rather than using raw structural data, biological units from the Protein Quaternary Structure (PQS) resource (Henrick and Thornton, 1998) were used to identify and analyse protein-protein interfaces. All 58397 protein structures stored in PQS as of March 2009 in the form of their biological units were extracted from `ftp://ftp.ebi.ac.uk/pub/databases/msd/pqs/BIOLIST`, and termed the PQS_{all} dataset.

4.2.1.2 Filtering for high-quality multichain structures

PQS_{all} was filtered to eliminate viral capsids, NMR entries, low resolution, high R-factor and monomeric entries, resulting in the $PQS_{filtered}$ dataset of proteins, as shown in Figure 4.2. Structures of viral capsids were eliminated because these complexes are considered exceptions in terms of interface properties: each subunit has contact surfaces with multiple other subunits; there is limited knowledge on the number of chains in biological units; and the mechanism of the capsid assembly is still poorly understood (Zlotnick, 2005).

For a structure to have been classified as a protein-protein complex, the final requirement was that the complex consisted of at least two amino acid chains, both with a minimum length of 30 amino acids. Chains with fewer than 30 residues were labelled as peptide chains and were discarded during filtering.

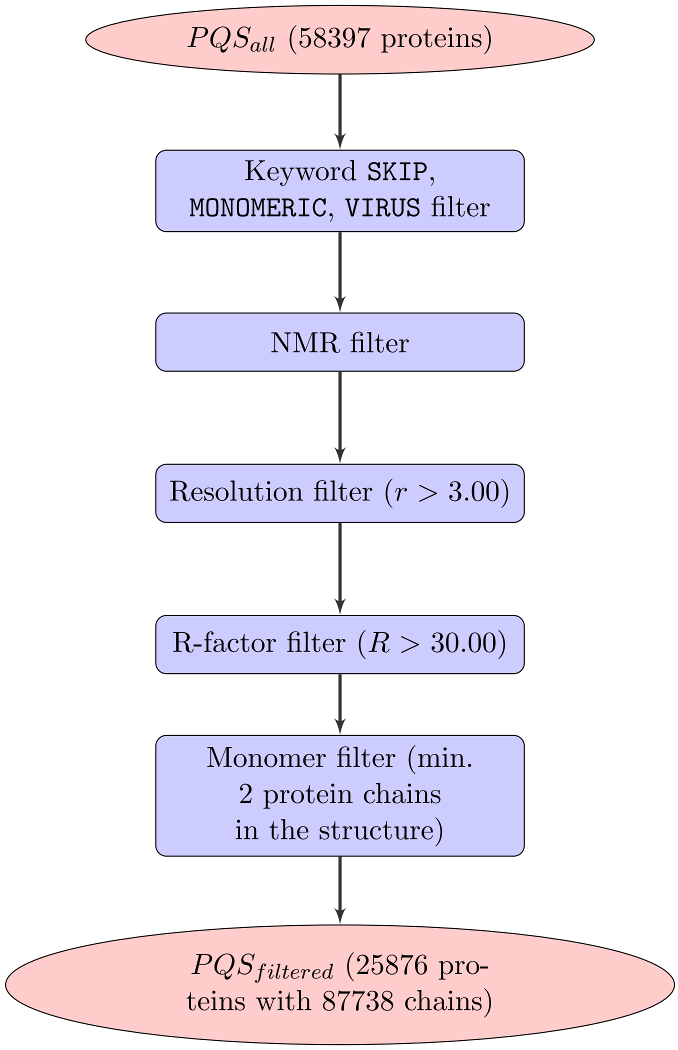

Figure 4.2: Preparation of protein-protein interfaces dataset.
PQS_{all} contains a list of biological units for all PQS files (March 2009). The fiters shown in blue specify what they *eliminated* from the dataset: monomers, viral capsids, NMR structures, high resolution, low model quality structures and peptide chains, respectively.

When the coverage of total protein space is considered, the PDB is biased towards proteins which are easier to crystallize, and towards protein families which are current hot-topics in structural biology, e.g. cancer-related proteins and proteins related to widely-spread severe diseases. To remove this bias, all $PQS_{filtered}$ chains were clustered based on sequence similarity and one chain was chosen from each cluster, resulting in a set of non-redundant chains, PQS_{nr}. Clustering was performed using the PISCES[6] (Wang and Dunbrack, 2005) server with a 25% sequence similarity threshold, culling 'by chain' and with all other parameters set to default values. The representative chain was selected from each of the 4345 clusters based on the highest structure resolution and then if tied, the lowest R-values.

4.2.1.3 Identifying buried, surface and interface residues

All residues in PQS_{nr} were tested for localisation in the three-dimensional structure of the complex, based on solvent-accessibility. Relative solvent-accessibility of each residue in the complex ($rASA^c$) and in monomeric chain ($rASA^m$) was precalculated and stored in XMAS format, as explained in Section 2.1.1.2. If a residue had $rASA^m > 5$ it was classified as *surface*, otherwise it was a *buried* residue[7]. Surface residues were further divided into *interface* and *non-interface*, based on the difference in relative solvent-accessibility between monomeric and complex form. More precisely, interface residues needed to satisfy:

$$rASA^m - rASA^c \geq 10 \tag{4.1}$$

[6] `http://dunbrack.fccc.edu/Guoli/PISCES_InputB.php`
[7] a criterion introduced by Miller *et al.* (1987)

4.2.2 Sequence-based properties of interfaces

Hydrophobicity and amino acid propensities do not need anything but sequence information on protein chains to be calculated, hence these are termed sequence-based interface properties.

4.2.2.1 Amino acid propensities

20 standard amino acid types were examined for preferential occurrence in interfaces, or on non-interface surface using the method of Liang *et al.* (2006). To start with, background fractions of monomeric solvent-accessible surface area (ASA^m) occupied by each amino acid type among observed interface residues were calculated as follows:

$$F_{intf}(X) = \frac{ASA_{total}(X)}{ASA_{total}(intf)} \tag{4.2}$$

where X was one of the 20 amino acid types, $F_{intf}(X)$ was the ASA-based percentage of residues of type X among the interface residues, $ASA_{total}(X)$ was the sum of ASA^m for all residues of type X in the interface dataset, and $ASA_{total}(intf)$ was the sum of ASA^m for all residues in the interface dataset. By analogy,

$$F_{surf}(X) = \frac{ASA_{total}(X)}{ASA_{total}(surf)} \tag{4.3}$$

was the percentage of residues of type X in the non-interface surface dataset. Additionally, $\overline{ASA_{surf}}(X)$ was the average ASA^m value for all residues of type X found in the surface dataset. Propensity for a residue of type X to be in an interface, $Pr(X)$, was defined as:

$$Pr(X) = \left(ln \frac{F_{intf}(X)}{F_{surf}(X)} \right) \times \frac{ASA^m(X)}{\overline{ASA_{surf}}(X)} \tag{4.4}$$

where $ASA^m(X)$ was the empirically obtained monomeric ASA for that residue. The method by Liang *et al.* (2006) was chosen as the most appropriate for two reasons. First, it is ASA-based, taking into account the contribution of sidechain sizes: larger amino acids will, on average, account for larger fractions of interface surface than residues with small sidechains. Additionally, $ASA(X)$ incorporates empirical information into the propensity calculation; an improvement over previously used formulae, e.g. by Dong *et al.* (2007) where every residue of a given type was considered to have the same, average contribution:

$$Pr(X) = \ln \frac{\frac{C_{intf}(X)}{C_{intf}}}{\frac{C_{surf}(X)}{C_{surf}}} \tag{4.5}$$

where C_{intf} and C_{surf} are total counts (rather than ASA-contributions) of interface and surface residues, respectively, and $C_{intf}(X)$ and $C_{surf}(X)$ are counts of interface and surface residues or residue type X, respectively. This generalisation has two dangerous consequences: each residue of a type X is assumed to contribute equally to the interface/surface, and all residues (irrespective of their size) are treated equally likely to be observed in an interface, resulting in an overestimation of bulky residues in both interface and surface datasets.

Since the scale of $Pr(X)$ is logarithmic, a positive $Pr(X)$ value means residue type X is more often found in the interface than the non-interface dataset, while a negative $Pr(X)$ indicates a residue type underrepresented in interfaces.

4.2.2.2 Hydrophobicity

Numerous hydrophobicity scales have been developed tailored for various experimental problems, for a review see Cornette *et al.* (1987). Here, Kyte and Doolittle scale was chosen (Kyte and Doolittle, 1982): it combines several previously developed methods in a dimensionless range from -4.5 for a hydrophillic arginine to 4.5 for hydrophobic isoleucine:

Table 4.1: Kyte & Doolittle hydrophobicity scale.

Amino acid	R	K	D	E	N	Q	H	P	Y	W
Hydrophobicity	-4.5	-3.9	-3.5	-3.5	-3.5	-3.5	-3.2	-1.6	-1.3	-0.9
Amino acid (cont.)	S	T	G	A	M	C	F	L	V	I
Hydrophobicity	-0.8	-0.7	-0.4	1.8	1.9	2.5	2.8	3.8	4.2	4.5

4.2.3 Structural properties of interfaces

Properties such as shape identifiers (planarity and protrusion), interactions among spatially close residues or atoms (hydrogen and disulphide bonds) and organisation of residues in secondary structure elements require information on the coordinates of atoms in a protein chain in the folded state.

4.2.3.1 Planarity

Planarity of each residue was calculated using `PRINCIP` (included in the `SURFNET` package (Laskowski, 1995)). This algorithm calculates the best-fitting plane through a set of residues, and then provides the root mean squared error[8] from that plane as a measure of planarity[9]. In order to define a plane for a residue, its coordinates were used along with the coordinates of the 7 closest residues on the protein surface (based on the distance between C_α atoms).

[8]defined in Section 5.1.1.4

[9]the lower the error rate, more planar the set of residues are

4.2.3.2 Preparing the Benchmark 4.0 dataset for protrusion analysis

The protrusion of chain residues into the solvent should not be calculated on the PQS chains just separated into monomers, without prior application of some kind of surface smoothing algorithm, for more details see Section 4.3.2.4. Therefore, a set of protein-protein complexes, with solved structures in both bound and unbound form were obtained from the Benchmark 4.0 dataset[10] (Hwang *et al.*, 2010). First NMR structures and chains which did not exist in PQS were eliminated. Then each bound chain was annotated with interface and non-interface surface residues, using the same method as in Section 4.2.1.3. Mapping these residues to equivalent residues on the unbound chains was performed using PDBSWS (Martin, 2005); for more details see Section 2.1.4. A residue on an unbound chain was mapped to a residue on a bound chain, via the equivalent residue in the UniProtKB/Swiss-Prot protein sequence. As a result, interface and surface residues from bound chains were mapped across to surface residues on 199 unbound chains, and prepared for further protrusion analysis.

4.2.3.3 Protrusion

Average residue protrusion was calculated by `PROTRUDER` (Simon Hubbard, 1994, unpublished), as a numerical score in the range of $0.0 - 9.0$, as originally used by Jones and Thornton (1997). Protrusion was calculated on 199 unbound chains from the Benchmark 4.0 dataset (Hwang *et al.*, 2010), prepared as described above.

[10] `http://zlab.umassmed.edu/benchmark/`

4.2.3.4 Secondary structure elements

Secondary structure was calculated per-residue by an in-house implementation of the `SSTRUC` program (Smith and Thornton, 1989) and stored in XMAS format. `SSTRUC` is an improvement of well known `DSSP` algorithm (Kabsch and Sander, 1983). Secondary structure elements, according to Kabsch and Sander, are presented in Table 4.2. `SSTRUC` uses the same annotation, adding lower-case categories for residues at the ends of secondary structure elements, relaxing hydrogen bonding requirements for these residues.

Table 4.2: Kabsch & Sander secondary structure elements.

Category	Description
H	α-helix
B	A single bridge displaying β-structure-characteristic hydrogen bonds
E	Multiple consecutive bridges (i.e. β-strand)
G	3_{10}-helix
I	π-helix
T	Hydrogen-bonded turn
S	Bend (based on backbone angle of segments upstream and downstream of residue in question)

4.2.3.5 Disulphide bonds

Residues were searched for intrachain disulphide bonds based on the distance between $S\gamma$ atoms of neighbouring cysteines. An in-house script checks for pairs of cysteines in the same protein chain with $S\gamma$ atoms less than 2.25Å apart, labels them as participating in disulphide bonding, and stores this information in the XMAS file. The 2.25Å threshold is chosen based on the average distance between disulphide sulphurs of 2.03Å determined by Hazes and Dijkstra (1988), adding $\sim 10\%$ for inconsistencies in structural data.

4.2.3.6 Hydrogen bonds

Similarly to disulphide bonds, hydrogen donors and acceptors were identified as defined by Baker and Hubbard (1984) and stored during XMAS file preprocessing. Briefly, if the hydrogen atoms have known positions (i.e. their coordinates can be calculated), the distance between the hydrogen and the hydrogen-acceptor must be not greater than 2.5Å with the donor-hydrogen-acceptor angle between 90° and 180°, for the hydrogen bond to be assigned. In case the hydrogen atom is not defined, a hydrogen bond is assigned when the donor-acceptor distance is not greater than 3.35Å and the angle is again between 90° and 180°. These rules correspond to the definition of hydrogen bond introduced by Cuff *et al.* (2006) in their Figure 1, with the exception of reducing donor-acceptor distance from 3.5Å to 3.35Å, as the former condition was found to be too liberal.

4.2.4 Profile-based properties of interfaces

Profile-based amino acid propensities and sequence conservation scores both require sequence information from the homologues of the interface-containing chain, in the form of multiple sequence alignments. To that end, two types of alignments were created for every chain in PQS_{nr}: FOSTA alignments contained fewer homologues, while having restrictive criterion of functional-equivalence for each homologue, whereas BLAST alignments contained more homologues at the same time increasing the chance of functionally-diverged homologues being included in the alignments. In both cases, for the alignment to be created, the minimum size was the query sequence plus four homologues, and the maximum size was 200 homologues, creating alignments with 5-201 sequences aligned, respectively.

4.2.4.1 Multiple sequence alignments

First, families of functionally-equivalent homologues were extracted from FOSTA[11] (McMillan and Martin, 2008). All non-redundant PQS chains were mapped to UniProtKB/Swiss-Prot chains using PDBSWS (Martin, 2005). Then families containing the mapped UniProtKB/Swiss-Prot chain, and at least four functionally-equivalent homologous chains were aligned using `Muscle3.7`[12] (Edgar, 2004c) with default parameters.

The second alignment dataset was extracted by using BLAST[13] (Altschul *et al.*, 1990) to identify homologues of the PQS chain sequence in the UniProtKB/Swiss-Prot database, using all default parameters, filtering out low-complexity regions and all homologues with $E > 0.01$. All remaining homologues containing terms `‘putative’`, `‘predicted’`, or `‘hypothetical’` in their descriptions were removed as unreliable. Finally if the list of homologues had more than 200 sequences, it was reduced to the 200 sequences containing the lowest E-values. Families passing the minimum alignment size criterion were aligned with the homologues using `Muscle3.7` with default parameters.

4.2.4.2 Sequence conservation

The conservation score of each residue was calculated using the `Valdar01` method implemented in `scorecons` (Valdar, 2002) with the substitution matrix normalised so all matches have the highest score of 1 (Karlin and Brocchieri, 1996). Thus each residue was assigned two scorecons scores, provided both alignments were successfully created for that protein chain.

[11]introduced in Section 2.1.5
[12]introduced in Section 2.2.4.3
[13]see Section 2.2.3

4.3 Results and discussion

4.3.1 Dataset of interfaces in protein-protein complexes

Contrary to the work published in the 1990s when the number of structures in the PDB was too low to obtain a comprehensive dataset, structural data nowadays sufficiently cover various interface types and offer enough data for solid statistical analysis. There have been numerous sets of interfaces used in recent years (reviewed for example by de Vries and Bonvin (2008)), none of which successfully passed the following tests: (i) crystal contacts were removed, (ii) complexes were culled using all filters from Figure 4.2, (iii) peptide chains and protein-ligand interfaces were removed, and finally (iv) redundant interfaces (based on sequence homology) were removed. Therefore, to the best of the author's knowledge, PQS_{nr} is, at the time of writing, the most extensive dataset of non-redundant, high-quality structures of protein-protein complexes, including both obligate and non-obligate interfaces.

The PQS_{nr} dataset consisted of 4345 non-redundant protein chains from 4014 protein complexes. These chains were spread across 37/40 (93%) of CATH architectures, and 860/1233 (70%) of CATH folds (topology level of the hierarchy)[14]. The broad coverage of CATH entries indicates not only that PQS_{nr} contains exclusively high-quality structural data, it provides a representative sample of protein chain space (and, by analogy, of protein-protein interfaces space).

Each chain was interacting with at least one other protein chain within the same PQS complex thus providing a protein-protein interface site, and on average had 23.5% buried residues and 76.5% surface residues. 20.4% of residues found on the protein surface were labelled as interface, the rest comprised non-interface surface control cases – hereafter referred to as the 'surface dataset'. These chains were further employed for parameter space analysis and, in the following chapter, patch building, classifier training and testing.

[14]based on the CATH v3.4, for more details on the methodology see Cuff *et al.* (2011)

As shown in Figure 4.3, both interface and surface residues spread over the whole range of solvent-accessibility values, with a slight enrichment in interface residues among higher ASA values.

4.3.2 Identifying interface-specific features

Each of the following sections (4.3.2.1–4.3.2.7) analyses a feature presumed to have some predictive power in interface classification. Each feature (a property) is analysed on the PQS_{nr} dataset, compared to previously identified trends, and finally, a conclusion is given whether it is likely to be useful in the next chapter when a predicting model is built. For the summary of trends observed in this extensive analysis, see Table 4.3.

4.3.2.1 Amino acid propensities

The simplest way to differentiate a dataset of interfaces from the rest of the protein surface is to compare the amino acid composition. All previous surveys found that interfaces differed from the surface amino acid type distribution (Jones and Thornton, 1996; Chakrabarti and Janin, 2002; Liang *et al.*, 2006; Yan *et al.*, 2008), sometimes reporting different patterns in amino acid preferences, probably owing to variations in methodology. The results presented below are ASA-based rather than residue-count-based, and corrected for the ASA of that residue position, as defined in Section 4.2.2.

Initial analysis on a limited dataset of 70 proteins showed an abundance of aromatic and aliphatic residues in interfaces, and a lower frequency of all charged residues, except arginine (Chakrabarti and Janin, 2002). On a larger dataset, Yan *et al.* (2008) observed that aliphatic and polar residues (except arginine) were underrepresented in interfaces, and confirmed a high frequency of aromatic and hydrophobic amino acid types. Arginine was found overrepresented in some datasets (Zhou and Shan,

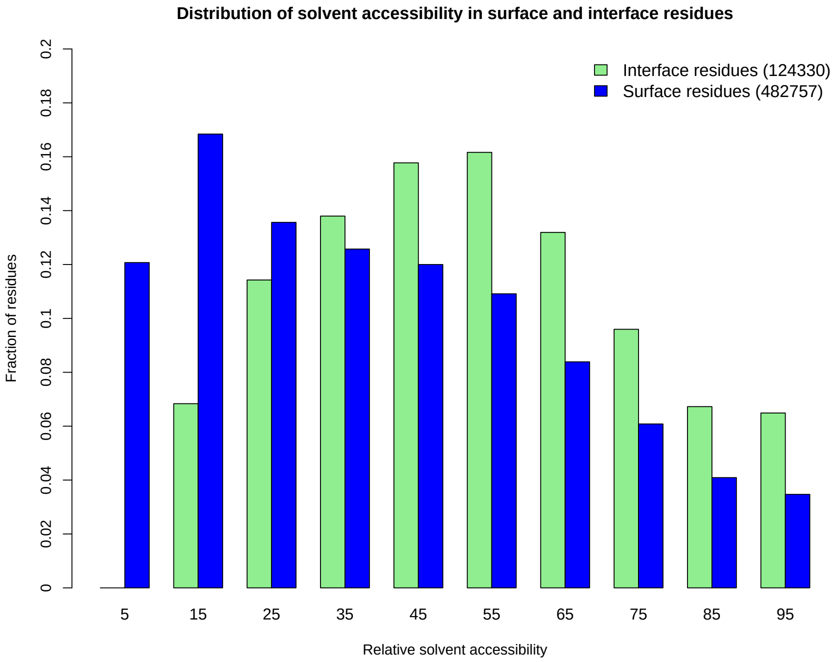

Figure 4.3: Relative solvent accessibility of surface and interface residues. There was a significant difference in rASA values between interface ($M = 52.82$, $SD = 23.26$) and surface residues ($M = 39.49$, $SD = 25.43$), $t(df = 207473.6) = 176.7002, p$-value $< 2.2e-16$, when two-tailed Welch two-sample t-test was performed. Total counts of residues per category are shown in the legend in brackets. All 187471 buried residues would, by definition, fit into the first bin, not shown here. Residues with $rASA > 100$ values are shown in the last bin. For core-surface sorting criterion, see Section 4.2.1.3.

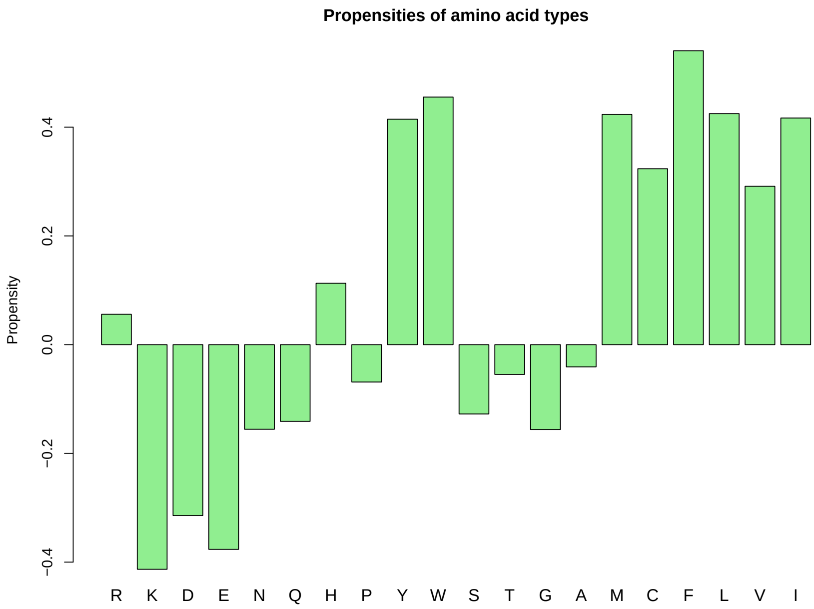

Figure 4.4: Propensities of amino acid types in interface residues. Propensity value was calculated using Equation (4.4). Residue types are ordered by ascending hydrophobicity value according to Kyte & Doolittle scale, see Table 4.1.

2001; Chakrabarti and Janin, 2002; Ofran and Rost, 2003; Bahadur *et al.*, 2004; Liang *et al.*, 2006; Yan *et al.*, 2008), although another recent analysis questioned this trend, showing no difference from the rest of the surface (Neuvirth *et al.*, 2004).

Contributions of each amino acid type in terms of solvent-accessible surface for PQS_{nr} are shown in Figure 4.4, revealing clear trends for some amino acid types to prefer or avoid interfaces, when compared with the rest of the protein surface. Similarly to previously identified propensities mentioned above, interfaces are depleted in polar and charged amino acid types (except arginine and histidine), and enriched in aromatic and aliphatic residues (except glycine and alanine). It is worth noting that arginine was again found to be overrepresented in interfaces – a feature first shown on smaller interface datasets and then dismissed by Neuvirth *et al.* (2004), using amino acid counts rather than fractions of solvent-accessible surface. Overall, patterns in amino acid propensities are likely to contribute strongly to the predictive power when an interface prediction classifier is trained.

4.3.2.2 Hydrophobicity

As expected from the difference in amino acid composition, interfaces, especially their cores, have been found to be more hydrophobic than the rest of the protein surface (Chakrabarti and Janin, 2002; Ofran and Rost, 2003; Yan *et al.*, 2008). Furthermore, the comparison of different types of complexes revealed that homomeric complexes have higher average interface hydrophobicity than heterocomplexes, owing to larger interface size and thus, higher core-to-rim ratio (Headd *et al.*, 2007). Homomeric complexes are mostly obligate (Ofran and Rost, 2003) and, as discussed in Section 4.1.3.1, complexation occurs at the same time as protein folding so these residues can be considered as the protein core.

Figure 4.4 shows residue types ordered by hydrophobicity values, from hydrophillic ones on the left, towards hydrophobic amino acids on the right. Excluding arginine, histidine and aromatic residues (these have been discussed above), there is a notable preference for hydrophobic residues in the interfaces, and less hydrophobic ones on the rest of the protein surface. It can therefore be concluded hydrophobicity is a predictor of interfaces, and it is not completely overlapping with amino acid propensities, thus justifying its use in addition to propensities while training the model.

4.3.2.3 Planarity

The interfaces of homomeric complexes have better shape complementarity and tighter packing with fewer water molecules than heteromeric complexes. There have been several attempts to describe the shape of an interface formally, most commonly through assigning a planarity value to the interface residue and its neighbours, or to a patch. Planarity is quantified as the RMSE for the best-fitting plane for a patch (see Jones and Thornton (1997) and references therein). It has been suggested that interfaces are more planar than the average surface patch (Jones and Thornton, 1997), the effect being more distinctive when only heterodimers were considered (Chakrabarti and Janin, 2002). In accordance, a significantly lower planarity score was observed for interfaces when compared to surface residues, shown in Figure 4.5.

4.3.2.4 Protrusion

Several other shape quantifiers were introduced besides planarity, i.e. protrusion (Jones and Thornton, 1997), shape index and curvedness (Bradford *et al.*, 2006), concavity and "rufness" [sic] (Pettit *et al.*, 2007). These properties varied in definition, and with the exception of protrusion, none was used again after the initial introduction in a single publication. It was therefore decided that protrusion would be tested as the second shape attribute, potentially adding new structural patterns to the learning process.

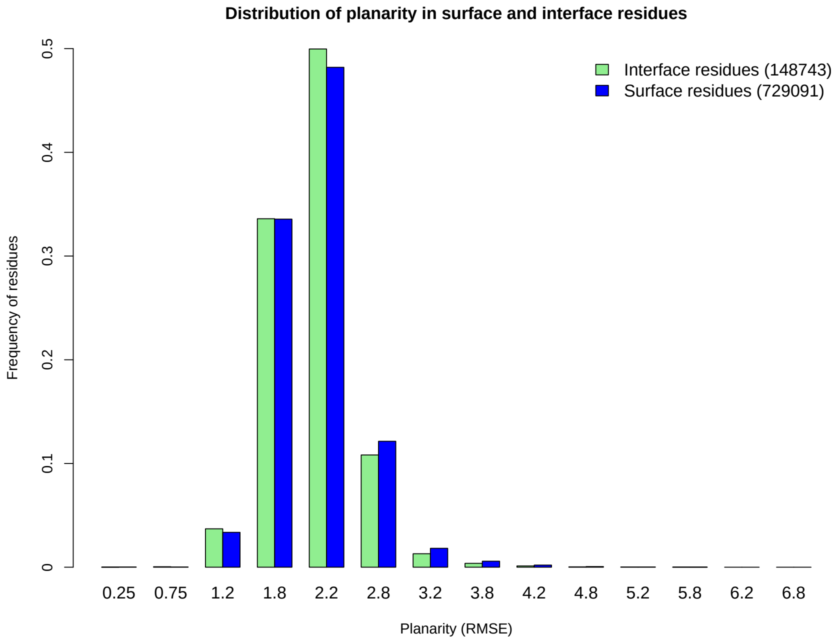

Figure 4.5: Planarity values for interface and surface residues. There was a significant difference in planarity values between interface ($M = 2.12$, $SD = 0.38$) and surface residues ($M = 2.14$, $SD = 0.40$), $t(df = 223562.5) = -19.5522, p$-value $< 2.2e - 16$, when two-tailed Welch two-sample t-test was performed. Total counts of residues per category are shown in legend in brackets.

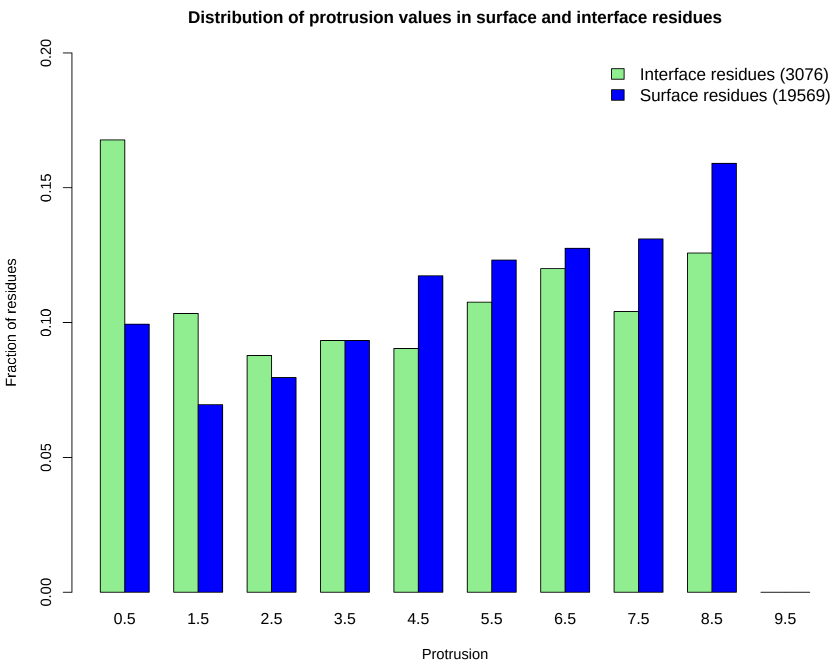

Figure 4.6: Average protrusion indexes, for interface and surface residues in Benchmark 4.0 There was a significant difference in protrusion values between 3076 interface ($M = 4.45$, $SD = 2.86$) and 19569 surface residues ($M = 5.12$, $SD = 2.66$), $t(df = 3959.994) = -12.2799, p$-value $< 2.2e - 16$, when a two-tailed Welch two-sample t-test was performed.

Protrusion index was calculated by PROTRUDER, reproducing the protrusion index from Jones and Thornton (1997). It has been argued that monomeric structures obtained by merely separating chains from a complex can have misleading positions of interface atoms[15], thus artificially increasing local protrusion indexes. These monomeric structures will show interfacial atoms in energetically unfavourable conformations sticking out unnaturally into the solvent; these interactions are stabilised via interactions with the other chain in the complex state. In order to obtain a more realistic interface surface, one should run energy minimisation on each monomer after separation, taking into account interactions of interface atoms with solvent molecules. Implementing this step requires too much computer time for calculation of protrusion as a predictor of interfaces.

Alternatively, one can utilise structures where both single chains and the complex form have been solved in separate experiments, and deposited in the PDB. Such triplets of PDB entries (one bound and two unbound structures) have been periodically published in a gold-standard dataset created for protein docking simulations (Hwang *et al.*, 2010). The latest version, Benchmark 4.0, contains 176 triplets[16]. After filtering chains with different chain identifiers in PDB and PQS files, NMR structures, and chains for which PDBSWS does not map the same UniProtKB/Swiss-Prot accession number to both bound and unbound chain, 199 chains comprised the Benchmark 4.0 dataset of interface and surface residues.

Preliminary analysis of average protrusion index for interface and surface residues on the Benchmark 4.0 data, shown in Figure 4.6 indicates that interfaces adopt less protruding positions. However, this dataset is not extensive: when redundant chains were removed it covered only 139 chains, only 3% of the PQS_{nr} dataset used to survey all other physico-chemical predictors. Furthermore, this trend was less extensive than the decrease in planarity score[17], so planarity is considered to be a better indicator of interface shape. Since extending the protrusion calculation to the full PQS_{nr} dataset

[15] personal communication with Prof. David T. Jones, UCL

[16] http://zlab.umassmed.edu/benchmark/

[17] concluded from smaller absolute value of the t-statistic

is a non-trivial computational task, protrusion index was excluded from the list of features used in the prediction step and was given a low predictive power value in Table 4.3. However elimination of protrusion might have been unjust: protrusion index should be revisited once more structures become available in both bound and unbound forms.

4.3.2.5 Secondary structure elements

Initially, Neuvirth *et al.* (2004) showed a slight enrichment in β-strands and long loops in interfaces. In a more recent study, same authors argued that there was no clear preference for secondary structure in interfaces (Neuvirth *et al.*, 2007); the trends changed depending on which methodology was used to define secondary structure.

All residues were tested for secondary structure, using a method introduced in Section 4.2.3.4. Residues labelled with 'h' or 'H' by `SSTRUC` were considered to be helix residues. Likewise, residues labelled with 'e' or 'E' belonged to β-strands. Other `SSTRUC` categories were sorted as C (for coil), effectively covering all non-α, non-β residues.

Contrary to the previous work by Neuvirth *et al.* (2004), PQS_{nr} displays significant depletion in β-strands and significant enrichment in α-helices in the interfaces (see Figure 4.7). While findings on the PQS_{nr} dataset do not confirm previous analyses, secondary structure was still employed as a predictor, to be potentially removed by a classifier if it should display dependencies on other interface features. Consequently, it has been labelled as a medium predictor in Table 4.3.

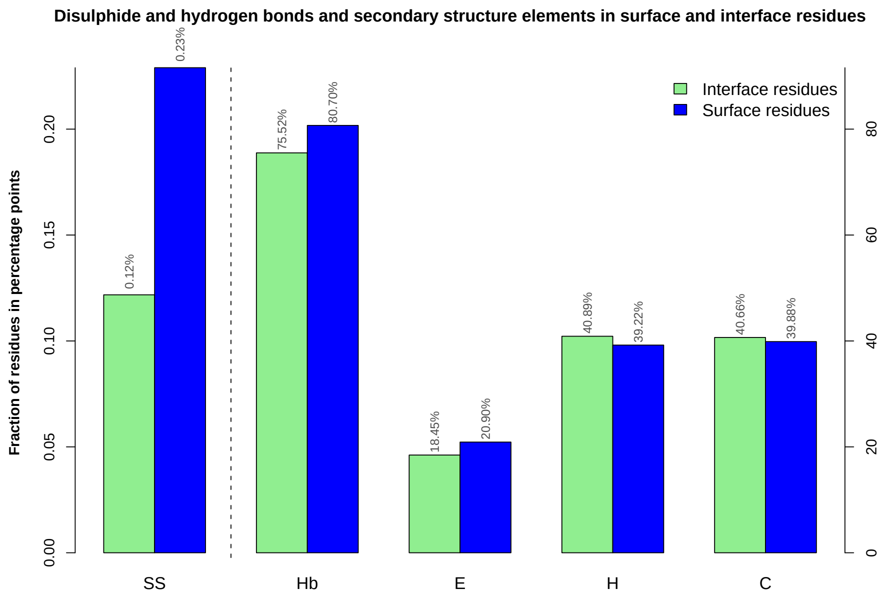

Figure 4.7: Disulphide bonds, hydrogen bonds and secondary structure elements in interface and surface residues.
Secondary structure elements are β, α or else, labelled with E, H and C, respectively. Frequency of disulphide bonds among residues is shown on a scale shown on the left, while the scale for hydrogen bonds and secondary structure fractions is shown on the right side of the graph. All categories show statistically significant difference between interface and surface categories (χ^2 test, $p < 0.01$), even after Bonferroni correction for multiple testing was applied.

4.3.2.6 Disulphide bonds and hydrogen bonding

The frequencies of hydrogen bonds, disulphide bonds and electrostatic interactions per interface and per area unit are highly dependent on the definition of the interface surface and the quality of structural data. Charged residues were found to be depleted in the analysis of interfaces in the PQS_{nr} dataset, and it was shown that electrostatic interactions are crucial for specificity rather than being often found as energetic hot spots (Moreira *et al.*, 2007) (see Section 4.1.3.4). Therefore it made no sense to use electrostatic interactions as a predictor, and the focus was shifted to intrachain hydrogen and intrachain disulphide bonds.

On average, hydrogen bonds were found to occur once every $\sim 190\text{Å}^2$ in Chakrabarti and Janin's analysis of transient heterocomplexes (Chakrabarti and Janin, 2002), one hydrogen bond was observed every 230Å^2 in transient homocomplexes and one every 210Å^2 in obligate homodimers (for review see Table 2 in Janin *et al.* (2008)). In PQS_{nr}, both disulphide bonds and hydrogen bonds were more likely to occur among surface residues and depleted among interface residues, as presented in Figure 4.7. Disulphide bridges were two orders of magnitude less common than hydrogen bonds in PQS_{nr}, but unlike charged residues, cysteines were overrepresented among interface residues (as previously shown in Figure 4.4), making a decreased frequency of disulphide and/or hydrogen bonds potentially useful when building a predictor.

4.3.2.7 Sequence conservation

Using sequence conservation scores to identify protein-protein interfaces is somewhat controversial. Several studies indicate that interfaces are more conserved in terms of protein sequence than the non-interface surface residues (Valdar and Thornton, 2001; Zhou and Shan, 2001; Ofran and Rost, 2003; Neuvirth *et al.*, 2004), with heteromeric complexes exhibiting higher conservation scores than homomeric complexes (Yan *et al.*, 2008). Additionally, Guharoy and Chakrabarti (2005) showed that biologically relevant interfaces have more conserved residues in the core than in the rim, while nonspecific crystal contacts do not present that trend. Therefore, the topology of conservation in a putative protein-protein interface can be used to discriminate crystal contacts from biological interfaces.

On the other hand, sequence-based classification into interface and non-interface residues (or atoms) has no way of diversifying between buried and surface residues (or atoms) without using input from structural data. In that way the non-interface category has significantly more data points than the interface category, thus inevitably losing specificity (many are false positives) (de Vries and Bonvin, 2008). Therefore sequence-based interface prediction methods still exhibit poorer performance than most of the structural predictors (Kufareva *et al.*, 2007), some recent work even questioning sequence conservation as a relevant factor for protein interface recognition (Caffrey *et al.*, 2004; Neuvirth *et al.*, 2007).

This project revolved around protein-protein interfaces with available structural data. Sequence conservation score is merely an additional attribute, and it will be viewed as a (potentially improving) modifier to the structure-based interface prediction.

Interface sequence conservation analysis was performed twice, using two different types of multiple sequence alignments. First, conservation score was based on FOSTA families of functionally-equivalent proteins (McMillan and Martin, 2008). While FOSTA ensures that every interface-containing protein chain is aligned exclusively with chains having the same function, the availability of such families is low: only 866 chains ($\sim 20\%$) from PQS_{nr} had four or more FEPs available for the alignment. In contrast, the second conservation score obtained homologues using BLAST, with an E-value cutoff of 0.01. There were 3122 chains ($\sim 72\%$) with BLAST-based conservation scores.

Both conservation scores confirmed that interface residues are significantly more conserved than other surface residues. As expected, FOSTA alignments displayed more conserved interfaces, however, it is worth keeping in mind that this dataset of interfaces was severely reduced, potentially affecting classifier training. Additionally, comparison of FOSTA- and BLAST-based conservation in Figures 4.8 and 4.9 respectively, displays both conservation scores have a bimodal distribution, one peak around 0.5 and the other towards a sequence conservation of 1.0. There is a slight difference: BLAST data have a higher 0.5 peak, while FOSTA data have more almost-perfectly-conserved sequence positions, both among interfaces and non-interface surface residues.

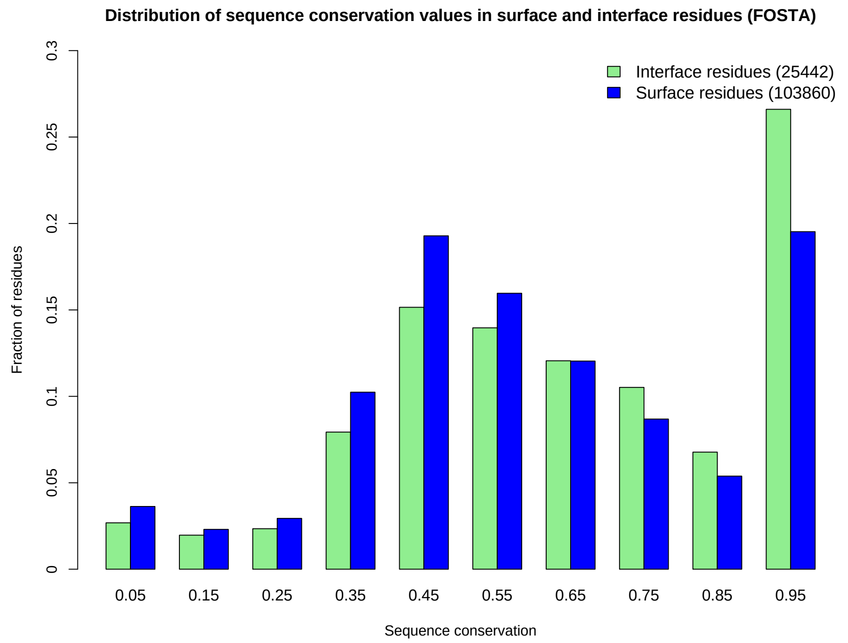

Figure 4.8: FOSTA-based sequence conservation in interface and surface residues. There was a significant difference in conservation values between interface ($M = 0.66$, $SD = 0.6$) and surface residues ($M = 0.61$, $SD = 0.26$), $t(df = 38544.01) = 30.8154, p$-value $< 2.2e - 16$, when two-tailed Welch two-sample t-test was performed.

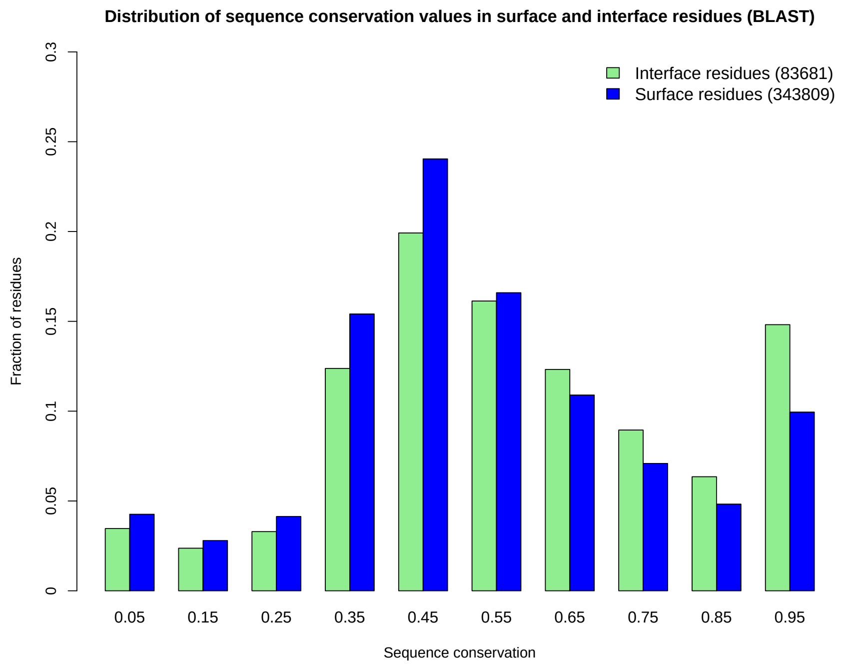

Figure 4.9: BLAST-based sequence conservation in interface and surface residues. There was a significant difference in conservation values between interface ($M = 0.58$, $SD = 0.25$) and surface residues ($M = 0.53$, $SD = 0.23$), $t(df = 123168.8) = 52.7088, p$-value $< 2.2e - 16$, when two-tailed Welch two-sample t-test was performed.

4.3.2.8 An ideal interface

Ideally a search for a protein-protein interface on a surface of a single protein chain should yield a single contiguous surface segment of $800 - 1000\text{Å}^2$, with higher than average planarity (manifested confusingly as lower planarity score) and hydrophobicity, and depleted in charged residues. When amino acid types are examined in spatially neighbouring residues on the same chain, the interface will have fewer than average disulphide and hydrogen bonds, and have overrepresented α-helices and underrepresented β-strands. Furthermore, a typical interface will have an O-ring shaped topology, with more conserved residues in the centre; these highly conserved residues are likely to be polar amino acids (Hu *et al.*, 2000). Finally, this patch will have residues with higher solvent-accessibility than other patches on the surface of the queried protein. The set of consensus features has been shown in Table 4.3, comparing previously listed trends with patterns observed within PQS_{nr} dataset, thus providing guidelines for the prediction step introduced in the next chapter.

Table 4.3: Physico-chemical and structural features of interface predictors – literature trends compared to PQS_{nr} results. Expected predictive power of each interface feature was suggested based on the comparison of previous findings with results presented in Section 4.3.2.1 and the statistical significance of trends shown on PQS_{nr}.

Property	Literature trends	Dataset	$PQS^{\star}_{nr}$	Predictive power
Amino acid propensities	$\Uparrow$ aromatic, aliphatic, Arg	Chakrabarti[†]	$\Uparrow$ aromatic, aliphatic (except Glu, Ala), Arg	High
	$\Downarrow$ polar (except Arg)	Yan[‡]	$\Downarrow$ polar (except Arg, His)	
Hydrophobicity	$\Uparrow$	Chakrabarti	$\Uparrow$	High
	$\Downarrow$	Yan		
Planarity	$\Uparrow$	Jones[°]	$\Uparrow$	Medium
Protrusion	$\Uparrow$	Jones	$\Uparrow$	Low
Hydrogen bonds	None	n/a	$\Downarrow$	High
Disulphide bonds	n/a	n/a	$\Downarrow$	High
Secondary structure	$\Uparrow \beta$, $\Downarrow \alpha$	Neuvirth[*]	$\Uparrow \alpha$, $\Downarrow \beta$	Medium
Sequence conservation (FOSTA)	n/a	n/a	$\Uparrow$	High
Sequence conservation (BLAST)	$\Uparrow$	Neuvirth	$\Uparrow$	Medium

⋆ 4345 non-redundant protein chains of all interface types
† 70 complexes (Chakrabarti and Janin, 2002)
‡ 2310 protein chains (Yan *et al.*, 2008)
° 64 complexes (Jones and Thornton, 1997)
* 57 heteromeric, non-redundant, transient complexes (Neuvirth *et al.*, 2004)

4.4 Conclusions

This chapter examines protein-protein interfaces in terms of their chemical, structural and evolutionary properties, using a carefully selected dataset of high-quality structures of protein complexes. Interfaces presented features consistent with previous findings on mostly smaller datasets, and some new methodological issues were identified in the process.

First, it is crucial to ensure that only biological interfaces are used, devoid of crystal contacts which are an inevitable consequence of solving protein structures using X-ray crystallography. The work presented here, similar to many recently developed methods, uses Protein Quaternary Structure (Henrick and Thornton, 1998) as a source of modelled biological units. PQS was recently discontinued, and substituted with more reliable automated PISA-modelled (Krissinel and Henrick, 2007) biological units. However, it is unclear whether one should use PISA-modelled quaternary structures (provided for all PDB structures) or manually curated, inevitably more correct, PiQSi (Levy, 2007) biological units (covering $\sim 25\%$ of currently available structures).

Second, some complexes are less appropriate for this type of analysis (for example, viral capsid complexes, discussed in Section 4.2.1.2), as their nature could hamper the detection of interface-specific features. In particular, antigen-antibody complexes have a specific evolutionary mechanism (Liang *et al.*, 2006) (a high frequency of somatic mutations to optimise binding to the antigen), and could have therefore resulted in misleading multiple sequence alignments in Section 4.2.4.1. Moreover, transmembrane proteins display structural features which impose strict restrictions on amino acid composition and structural features, adding unnecessary bias to the dataset (Nugent and Jones, 2009). Removing both these types of interfaces from the PQS_{all} dataset might in the future provide more general interface features.

The classification of residues on the surface of a protein into interface or non-interface is based on incomplete information. Since the PDB provides structural information for a fraction of complexes, it is likely that many of the proteins used here have alternative, previously unobserved interfaces, besides the one used to define an interface in this chapter. Indeed, the goal of this project is to build a model predicting these previously (experimentally) unproven interfaces. Thus defining all residues on the surface of a protein not participating in interaction with an observed binding partner as non-interface is based on negative information, and is far from ideal. However, as the coverage of the total space of biological interfaces is increased by adding structures into the PDB, the models should have more information from which to learn, and ultimately result in more reliable prediction of protein-protein interfaces. Bearing this in mind, special care was dedicated to obtaining and carefully filtering interface data to be provided to the classifiers.

This survey started with a minimal set of attributes: there was no ambition to provide an extensive set of interface-specific characteristics. The aim was to cover previously identified chemical and structural patterns in protein-protein interfaces, and test whether sequence data on homologues provides any additional information when identifying likely interfaces. Sequence data raised special interest owing to conflicting reports ranging from full uselessness to claims that sequence information alone is sufficient for successful interface prediction. Amino acid propensities, hydrophobicity, planarity, hydrogen and disulphide bonds are all expected to perform well in the next chapter as strong predictors of protein-protein interfaces. Secondary structure elements might aid the prediction, but have lower predictive power, and protrusion as a surface shape indicator has been eliminated owing to the lack of appropriate interface data (see Section 4.3.2.4). Sequence conservation is expected to improve model's performance: less the BLAST-based, and more when adding the FOSTA-based conservation. These two conservation scores should be tested separately, and as a combination of two predictors.

When used to train various types of classifiers, this range of parameters should be sufficient to provide a reasonable prediction of putative interface patches on the surface of a protein chain. Once the appropriate type of machine learning model is identified for this classification problem and this dataset, the range of interface attributes could be revisited to be optimised (e.g. by combining potentially dependent attributes into a single score) and/or expanded in order to enhance the model's performance. Potentially at that stage, protrusion could be re-tested, hopefully on a larger dataset of bound structures, for which unbound monomeric structures are also available.

Throughout this chapter, the χ^2-test and t-test have been used to test whether interface residues displayed significantly different trends from other surface residues. Both of these tests might overemphasize the difference in frequencies or means because they were calculated using very large datasets – both categories consisted of thousands of points. While all interface properties displayed a significant difference in frequencies or means, this was taken as a starting guideline for the final choice of attributes; once the models are built they will provide true measures of attribute importance, and the list of final properties used for prediction can be adjusted accordingly.

Chapter 5

Protein-Protein Interface Prediction

The extensive interface data analysis and feature detection in the last chapter was performed with the aim of building a protein-protein interface prediction tool, to complement interface detection algorithms already developed within the SAAPdb pipeline. SAAPdb currently provides analysis of mutations in interfaces in complexes deposited in the PDB; an obvious enhancement was to expand the range of SAAPdb structural effects to *predicted* interfaces.

5.1 Introduction

There is a sizeable gap between the number of available monomeric and complex structures in the Protein Data Bank. In order to accommodate this gap, and facilitated by the constant decrease in costs of computation time, numerous methods predicting likely interface regions in monomeric structures have emerged over the last decade. Typically, prediction methods identify a dataset of known interfaces, analyse this dataset to obtain a list of features discriminating interfaces from the rest of the protein surface, and then separate the known complexes into monomers to be used as training and testing examples. Training is used to obtain the optimal set of features and to learn a model based on the selected features, ultimately producing the best prediction performance on a remaining test set.

This introductory section will start with a short review of machine learning in Section 5.1.1, covering basic principles of data handling and referring to two methods utilised in this chapter. Next, an overview of the commonly used publicly-available machine-learning-based interface prediction tools is provided in Section 5.1.2, focusing on the two methods chosen to be developed here, and compared to previously published work in Section 5.3.4. There are numerous published protein-protein interaction analyses as this is one of the central problems of structural bioinformatics. The literature review below does not aim to present all of them; rather it focuses on the methods suitable for the specific modelling task at hand: predictions to be incorporated into the SAAPdb structural analysis[1], and the efficacy of the surveyed methods. Additionally, only methods where the training datasets were publicly available, performance measures were clearly stated, and predictors were available for benchmarking purposes were considered.

[1]assigning SAAPs to likely interface sites, where the structure of the complex is unknown

5.1.1 Introduction to machine learning

We live in an age when data accumulates faster than ever. While data storage is becoming cheaper by the day, thus encouraging tracking of all kinds of naturally-occurring processes and results of human activity, one has to remember that, as Albert Einstein is supposed to have said "information is not knowledge". We have to make sense of these data and learn from them, rather than just gather them in their raw form. Unfortunately, our potential to analyse large amounts of information does not keep up with data accumulation (Witten and Frank, 2005).

Machine learning is a computer science field addressing exactly this issue - it tries to identify structure in given data, in an automated or semi-automated way through a process called data mining. The purpose of data mining is threefold, according to Witten and Frank (2005): it aims better to understand, explain (in 'human-readable' terms) or predict features of data. More formally, Mitchell called this process learning, and defined it as follows (Mitchell, 1997):

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Using Mitchell's terminology, the class of tasks (i.e. the aim of the learning process) is termed the **concept description**. Experience comes in the form of some examples, also called data points, training examples or **instances** (all three terms will be used interchangeably henceforth). Each instance usually has a unique identifier, followed by a set of measured **attributes** (also termed features). The choice of attributes is based on the experimenter's suspicion that they may contribute new knowledge to the concept description. Each instance will have a value assigned for each attribute. Based on the value type, attributes are **numerical** if they can be expressed on a numerical scale, or **categorical**, if the attribute description is defined as a finite set of mutually exclusive categories. These categories can also be numerical, but not continuous (for example, binary attributes with possible outcomes of 0 or 1), or expressed as some kind of non-numeric description (e.g. times of day: morning, noon, afternoon, evening) and termed **nominal** attributes.

5.1.1.1 Different machine learning approaches

There are two main types of machine learning: supervised and unsupervised learning. In **supervised learning**, the aim of the model is to predict how the combinations of values of input attributes *affect the outcome*, also expressed in the form of an attribute. If the output is categorical, the machine learning process is called **classification** and the attribute predicted by this model is termed the **class attribute** - all models built hereafter are of this type. On the other hand, if the outcome of a prediction can be expressed on a continuous numerical scale, and the model can be formalized as a numerical function of input variables, the model is called a **regression model**.

The learning process, when building a classifier, starts by accessing a set of instances with known attribute and class values (a **training dataset**). Modelling then produces a set of rules[2] the efficiency of which is tested on a dataset of instances with known attribute values, but lacking the class value (a **testing dataset**). Alpaydin (2009) defines classification as learning the mapping function $g(\cdot)$ from the input space X to the output space Y:

$$y = g(x|\Theta) \tag{5.1}$$

by optimising the set of parameters Θ in order to minimise the error on the training set. In this thesis, only binary classification will be considered predicting the presence or absence of a single feature: y in that case can adopt only two values 1 or 0,

[2]the methodology used to obtain the rules and their format differ between different supervised classifiers

corresponding to 'class feature exists' and 'class feature does not exist' for the given instance, respectively.

Unsupervised learning occurs when a learning concept focuses on *relationships between the attributes*, rather than trying to predict an outcome. In this case, there is no testing set, and the model outputs training instances grouped used some similarity measure. The main types of unsupervised methods are **association learning** where any structure among attributes is sought; and **clustering** where generating groups (clusters) of instances is the goal, without necessarily identifying the underlying structure common for instances in the same group.

Several other categories of learning exist, like semi-supervised learning (learning on a mixture of labelled and unlabelled instances) and reinforcement learning (where no training instances are provided) which are outside the scope of this thesis. For more details on these concepts, see for example Mitchell (1997) or Alpaydin (2009).

5.1.1.2 Data sampling

The central problem of machine learning is the quality of the data used for the learning step: mainly its size and how representative it is of the variety of features in the full population that is modelled. The relative size of the sample cannot be affected by the experimentalist: data are always limited or we would not have the need to model a system, we would simply observe it as a whole.

The gathered data (with known attribute values and class value) have to be divided into training data and testing data. If too many data points are used for training, one might build an excellent model, but the testing dataset might not be representative, thus misleadingly presenting low performance during evaluation. In the opposite case, the model will not be robust owing to the lack of training data, but the evaluation step will be very thorough. The optimal balance is achieved by iteratively using all instances for both training and testing: a process called **cross-validation**. N folds are chosen (usually 3, 5, or 10), and data are divided into N non-overlapping subsets of equal size. Then N models are built, each time using a different fold for testing, and all other folds merged for training. Evaluation of cross-validation is reported as averaged scores from each iteration.

An extreme case of cross-validation is the **leave-one-out** validation (also knows as 'jackknifing'), where the number of iterations (folds) equals to the number of data points: in each step all but one instance are used for model building, and tested on one data point. However this procedure is very resource-heavy so it makes sense to use it only when a very limited sample is available.

Both cross-validation and leave-one-out validation are data sampling *without replacement*, i.e. once an instance is sampled from the pool of instances, it is removed, and cannot be sampled again. In contrast, data sampling *with replacement*, also called **bootstrapping**, during sampling always leaves the instance in the original pool, and just copies it to the testing dataset. In this way, each sampled instance is chosen from the original N instances, which allows repeated sampling of the same instance. In fact, if sampling with replacement is performed N times from the dataset of N instances to be included in the testing set, typically 63.2% of instances will be chosen (obviously, some more than once), leaving 36.8% of instances for the **out-of-bag** (OOB) testing set. To use a simple example, sampling without replacement would be dividing a group of children in two football teams, whereas sampling with replacement would be drawing the names of children winning a prize from a hat, and returning the name back to the hat, so the same child can win more than one prize.

The other important issue during classification is the ratio of data points with each of the classes. This ratio has to be maintained throughout all the partitions of training and testing data, in order to avoid creating unbalanced models. For example, if by random data partitioning all instances with one class value were in the test set, and the training set had only values with the other class value, the model will simply predict the latter class value in 100% of the cases. The common method used to avoid this issue is to **stratify** both training and testing dataset, i.e. make sure the ratio of data points with each class value is maintained while randomly sampling data points for each fold in cross-validation, and the testing dataset.

Finally, provided with a limited sample, the model should perform equally well on the entire population being modelled[3], even if some of the patterns present in the population are not present in the training data. Alternatively, if the classifier is overfitted: it will present misleadingly high performance during training, but when tested on slightly different instances, it will prove inadequate. The only true test against overfitting is testing on a completely new set of instances, that have not been included in any phase of modelling (for more details see Section 5.1.1.5).

5.1.1.3 Handling missing data

Often some gathered data points do not have known values for all attributes used in the modelling process. There are various causes for missing values: for example an error could have occurred during the measuring process (e.g. an instrument malfunctioned), or it made no sense to perform 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 aided his treatment). In any case, the experimenter building a model usually does not have a possibility of repeating the measurement, so the modelling has to proceed with missing values.

There are three main strategies for handling missing data: (i) removing a data point, (ii) creating a new category, provided the feature is nominal or, (iii) imputing the value from data points with known values for that attribute. For more details see Witten and Frank (2005) or Saar-Tsechansky and Provost (2007), and references therein. Removing data points makes sense if the remaining dataset is not so small to seriously affect the model's performance. Indeed, training data used throughout

[3]performance is measured during training, but we really want the model to be a good predictor of the future data

this chapter were abundant enough to utilise this procedure (for details on the dataset like its size, see Sections 5.3.1–5.3.2). Next, when the attribute for which the value is missing is nominal, one can create a novel attribute category 'missing value', provided this is not adding severe bias by equating all the instances with the lacking value. Finally, there are several ways of predicting what would be the most likely value for the instance. For the review on missing data imputation, see for example Jerez *et al.* (2010).

5.1.1.4 Model evaluation

The aim of classification is to build a model based on some known data, which will successfully sort new instances into the right class. Consider the possible outcomes of a binary classification for a data point. The test instance, by definition, has a known true class value (positive or negative), and a predicted class value (again, positive or negative). The four combinations are presented in Table 5.1. An instance with a positive class value can correctly be classified as true positive, TP, or erroneously as a false negative, FN. *Vice-versa*, an instance with the negative measured class value can correctly be classified as true negative, TN, or erroneously be labelled as false positive, FP.

Table 5.1: Outcomes of a two-class prediction, also termed confusion matrix.

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

The 'success' or how 'correct' the classification is, is a relative term and depends on the purpose of the model. For example, when used as a diagnostic tool, missing a likely positive should be avoided at all costs, while falsely predicted positives are acceptable. In contrast, in the case of protein-protein interface prediction problem presented here, the predicted interface should always correspond to a true interface; at the same time it is more acceptable that a (small) fraction of true interfaces are missed. Consequently, there are different measures of model performance, and Table 5.2 surveys the ones used for binary classification evaluation, also listing the ranges of values the measures can adopt.

Table 5.2: Binary classification performance measures.

Name	Formula	Range of values
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	$[0,1]$
Precision*	$\frac{TP}{TP+FP}$	$[0,1]$
Sensitivity†	$\frac{TP}{TP+FN}$	$[0,1]$
Specificity$^{\circ}$	$\frac{TN}{TN+FP}$	$[0,1]$
False positive rate	$\frac{FP}{FP+TN}$ = 1−specificity	$[0,1]$
False negative rate	$\frac{FN}{FN+TP}$ = 1−sensitivity	$[0,1]$
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

* positive predictive value (PPV)
† true positive rate (TPR), coverage or recall
$^{\circ}$ true negative rate (TNR)
(*) $\hat{x}_i$ 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

Accuracy, also termed 'overall success rate', measures the fraction of correctly predicted cases, compared to all cases considered. The opposite is error rate, defined as 1−accuracy. **Precision** indicates how many instances predicted to be positive really are so, in other words this indicates how likely the model is falsely to annotate a hit as positive. In the case of interface prediction, this is particularly important. False positives in SAAPdb are erroneously annotated as structurally-damaging, thus failing sensibly to narrow down the list of potentially interesting mutations for the database user testing candidate mutations. **Sensitivity**, on the other hand, indicates how many true positives were missed, a crucial feature to avoid when using models in medicine[4]. **Specificity** indicates for true negatives the same thing sensitivity does for true positives.

The **F-measure** is the harmonic mean between precision and sensitivity: usually used with an equally weighted contribution of these two. While being a more general measure of accuracy than the first four listed here, it neglects the TN. Therefore the more appropriate general performance indicator is the **Matthews correlation coefficient**, MCC: it shows how well the predicted class correlates with the actual class (from -1 for anti-correlation, through 0 meaning no correlation, to 1 for the perfect correlation). Further, the MCC is the only evaluation metric including all four counts from Table 5.1 into a single value.

In general, the model is better the higher the values in the upper part of Table 5.2 are, and the lower the error rates are (provided output is a continuous numerical value, and error measures are applicable). However, during model optimisation, usually one has to trade one performance, to increase the other. This is achieved by trying out various attribute combinations and adjusting the model's parameters until the desired correctness is achieved.

For models with numerical outputs (like neural networks), in contrast to binary classifiers (e.g. random forests), three more performance measures can be applied: root mean squared error (RMSE), mean absolute error (MAE) and area under the curve (AUC), the 'curve' being the receiver operating characteristic (ROC) curve.

Root mean squared error is a square-root of variance of the residual[5]. In other words, the difference between the expected and observed value for each data point is first squared, then averaged, and then a square-root is taken. **Mean absolute error** is the averaged sum of absolute errors, each obtained

[4]missing an existing disease could have fatal consequences, whereas over-predicting diseased states, however traumatising for the patient, can be rectified at a later time

[5]variance of residuals is also called **mean squared error**

as the absolute difference between predicted and observed class value for a data point.

There is one more performance measure, provided a classifier ranks the outcome, or assigns probabilities, or confidence values: the receiver operating characteristic (ROC) curve, and the corresponding **area under the curve** (AUC). Plotting true positive rate against false positive rate, the learned model can be compared with the performance of a random model, i.e. a predictor randomly outputting a class value, irrespective of the input values. As presented in Figure 5.1, a random model has $AUC = 0.50$ (the curve 'A'). A perfect predictor with zero error rate would have $AUC = 1.00$ (curve 'D' is the closest to this ideal scenario), reaching a TPR value of 1 for all FPR values. However, there have been some recent doubts expressed by Hand (2009) about comparing different classifiers by using AUC since each ROC curve is a result of a different misclassification metric. Therefore a mixture of performance characteristics should be used when evaluating a model, paying special attention to which suboptimal behaviour of the model could be tolerated when it is applied in practice.

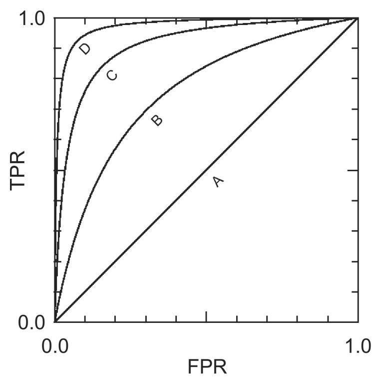

Figure 5.1: Receiver operating characteristic curve.
A is a random model, **B**, **C** and **D** show improvement over random prediction, in ascending order. *Figure obtained from* `http://www5e.biglobe.ne.jp/~tbs-i/psy/tsd/node3.html`, *with minor modifications.*

5.1.1.5 Benchmarking

To sum up data sampling and evaluation strategies, it is important to focus on the future performance of the model, while training and testing on a limited set of present instances. Benchmarking assumes independent and transparent testing (usually several similar models) on data never 'seen' by the model(s). While considered absolutely neccessary, especially to prove that one method outperforms another one without any doubt, it is rarely performed as it requires a great deal of effort, computing time, and (most scarce) data appropriate for the task modelled, that have not yet been utilised for that problem.

Two types of classifiers have been built in this chapter, and a short introduction to their methodology is presented in more detail hereafter.

5.1.1.6 Neural networks

The multilayer perceptron is a type of feedforward[6] neural network building the class-prediction function using backpropagation to minimise the errors during learning by adjusting the weights of the connections between the network's nodes (Rumelhart *et al.*, 1986).

In short, the multilayer perceptron is a model which can be represented minimally as three layers of interconnected nodes (also termed neurons, neurodes or perceptrons), with weights on all connections. As presented in Figure 5.2, the architecture of the model is divided into the input layer (one neuron for every attribute[7]), the (usually single) hidden layer with a user-defined number of hidden nodes, and the output layer with a node for every class category. The term 'hidden' refers to the fact that, unlike the input and the output layer, hidden nodes do not represent an observable property. Indeed, one of the downsides of multilayer perceptrons is that the data structure

[6] nodes connected in a non-circular fashion

[7] in the case of a neural network, these are exclusively numerical

they learn, although highly efficient at prediction, is not always easily transferable to human-readable terms. In other words, it can be used as a 'black box' for prediction, but unlike trees comprising rules it is not trivial to visualise.

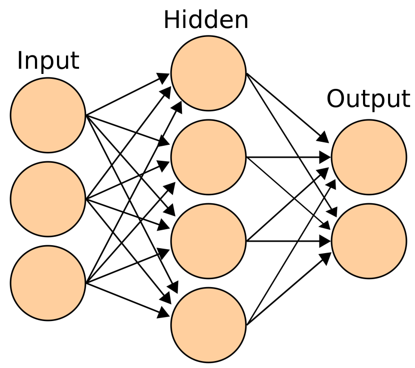

Figure 5.2: Multilayer perceptron schema.
Nodes are organised in three types of layers: input, hidden, and output layer, and the weights on the connections between the nodes are optimised. *Figure obtained from* `http://en.wikipedia.org/wiki/File:Artificial_neural_network.svg` *under Creative Commons license.*

Using mathematical terminology, a multilayer perceptron is a function mapping input values to the class value. Every node transforms the input using a nonlinear activation function, and the final output value is a linear combination of the outputs of weighted hidden layer nodes. The aim of the model is iteratively to learn the weights which will minimise the error rate on the presented instances, i.e. the training set. This error minimisation is often referred to as 'gradient descent', because in every iteration error is reduced in a stepwise fashion, hopefully reaching a global (rather than local) error rate minimum. Another way to avoid finding a local minimum is by introducing 'momentum', i.e. a small amount of random noise introduced into the system in every epoch. Finding the appropriate ratio of learning rate[8] and the momentum is the key to model optimisation to achieve sufficient generalisation and specialisation.

The user specifies the ending conditions for the learning process: either by defining the number of epochs, or by the defining a stopping condition when the learning rate has not changed for the last n epochs. While the number of iterations could be very high, allowing the model to sample error space finely around the achieved minimum,

[8]the 'size' of the error decrease in every step

the learning process is usually stopped soon after the learning rate change plateaus in order to avoid overfitting the model to the training data (especially when the training set is of a limited size), a process also known as 'early stopping'.

5.1.1.7 Random forest

When a classification model is required for a dataset containing many attributes, often a dimensionality-reduction needs to be employed because many commonly used models display suboptimal performance when highly interdependent or irrelevant attributes are mixed with the informative ones. This attribute-optimisation step is resource-demanding, and the standard way to avoid it is to resort to **decision trees**: the preferred model of choice for high-dimension modelling tasks.

A decision tree algorithm in every iteration, surveys all the possible splits of all attribute values, determining the split condition which maximises the information gain (Witten and Frank, 2005). This procedure is recursively repeated for every node created by the split, until no information gain can be achieved, or until the maximum tree size is reached.

Unfortunately, while robust in respect to high-dimensional data, a single decision tree often lacks accuracy when trained on a limited dataset. Several authors recently proposed an obvious improvement of this model by building a set of T trees (conveniently termed a forest), instead of a single tree. In this case, the final predicted classification of an example is a combination[9] of predictions made by every tree, generally outperforming decision trees (Svetnik *et al.*, 2003). The most efficient method for learning a solution to a problem using an ensemble of T trees is the **random forest**, introduced by Breiman (2001).

A random forest builds a user-specified number of trees, each trained and evaluated on a bootstrap sample of the same dataset, with the remaining 1/3 of instances

[9] often simply a majority vote

comprising the 'out-of-bag' test set; for more details on bootstrap sampling see Section 5.1.1.2. The number of trees is limited by the computer power; in practice, the more trees a model produces, the better the performance. Moreover, using the Law of Large Numbers, Breiman (2001) proved that there is an upper bound for the generalisation error, therefore adding trees to the random forest does not lead to overfitting.

While building a tree, a split of a node is calculated based on a randomly chosen subset of m_{try} attributes: this type of tree building is sometimes referred to as the **random tree** algorithm. In this respect the random forest resembles the bagging algorithm (Breiman, 1996), where a split is based on all p available attributes, with an obvious gain in speed when $m_{try} < p$. The tree is then built until there are no more information-gaining splits and no pruning is performed. Finally, the model is applied to out-of-bag instances, and the performance on them is reported as the 'out-of-bag error' (OOB error), which refers to 1-accuracy on the OOB set of instances.

According to Breiman (2001), a low m_{try} means a low correlation between the trees, but at the same time, each individual tree in the model will be less informative (as it covers a narrow range of provided attributes in every split). Increasing m_{try} will yield more similar trees, and each tree will provide more accurate prediction. Consequently, optimal performance of random forests lies within medium m_{try} and T values, and some optimisation of these two parameters should be utilized (Svetnik *et al.*, 2003). In fact, Svetnik claims that as long as m_{try} is not 1 or p, this parameter does not significantly influence the method's performance, at the same time providing great improvement in speed over bagging, and even over some decision trees.

Svetnik *et al.* (2003) have shown that random forests have several advantages over decision trees: (i) in every tree-splitting step they use a subset of parameters significantly reducing the tree-building time, (ii) the time-consuming cross-validation step is avoided by bootstrapping and evaluating the method on the OOB dataset (usually $\sim 1/3$ of the training dataset) and (iii) complexity of the tree-building step is reduced by omitting the pruning step. Moreover, while being very resource-efficient for large datasets with many attributes, random forest shows comparable performance to boosting (Meyer *et al.*, 2003) and decision forests (Tong *et al.*, 2003) and outperforms bagging. Finally it is a straightforward method to use 'off-the shelf' with only two parameters: the number of attributes tested while building a tree m, and the number of trees T which should be set to as high as the computational resources permit.

Another convenient feature of this classifier is that it provides a measure of importance of each attribute used for training. After each tree is built, a misclassification rate for a feature on the OOB set can be calculated when that attribute's values are randomized (Breiman, 2001). The difference between that misclassification rate, and the OOB error is termed the raw importance measure of an attribute. It is worth noting here that, while culling the attributes with the highest importance scores removes the less informative attributes, this parameter optimisation does not guarantee the highest scoring attributes are not mutually highly dependent.

5.1.2 Predicting protein-protein interfaces from structural data

When building interface prediction models, authors have used numerous datasets, reviewed in great detail by de Vries and Bonvin (2008); it is often simply the most recent version of the Protein Data Bank (Berman *et al.*, 2000), filtered to eliminate noise from the structural data (i.e. low-quality structural data and redundant structures). Interestingly, despite being intensively researched in recent years, there is no consensus dataset of protein-protein interfaces, and there is still a fair amount of disagreement over typical interface-distinguishing features.

Additionally, as described in Section 4.1.2, there is no unique definition of an interface: some use a distance threshold from residues on the interacting chain, others use the decrease in solvent-accessibility upon complexation. When sequence conservation is added as a property, one gets to choose from a plethora of conservation scores (e.g. as reviewed by Valdar (2002)). Finally, different authors use the same predictor names for differently defined predictors, i.e. amino acid propensities can be defined at the patch-level (Jones and Thornton, 1997) or residue-level (Dong *et al.*, 2007), yet the same terminology is used in both cases. These inconsistencies complicate comparison of methods and their performance, as will be discussed in Section 5.3.5.

Several recently-developed methods showing promising protein-protein interface prediction performance are reviewed in Table 5.3, with the accuracy and coverage (sensitivity) values as listed in the original papers. Most of these methods are patch-based rather than residue-based, use a combination of structural, evolutionary and sequence-based predictors, and report overall success rates (during cross-validation) up to $\sim 75\%$ (Keskin *et al.*, 2008).

The most successful interface prediction methods are based on both physical and chemical features, with optional evolutionary data; Kufareva *et al.* (2007) argued that while family alignment profiles marginally increase performance for some proteins, when alignments contain small numbers of sequences, or the interface has to adapt easily to several interacting partners, removing evolutionary information might increase the robustness of the interface prediction. Moreover, considering all the methods listed in Table 5.3 and (by extension) others listed in Table S1 in the review by de Vries and Bonvin (2008), it seems that analysing interfaces by sampling patches on the protein surface is superior to methods assigning a score/probability for individual surface residues.

In terms of machine learning methods, this project focused on two: neural networks and random forests. These two were chosen because the former has been ubiquitously used for this purpose, and the latter has recently shown great potential in modelling similar bioinformatics problems, and moreover, these two methods cover the opposite spectra of classification methodology: numerical and tree-based categorical classification.

Table 5.3: Overview of the state-of-the-art protein-protein interface prediction methods. Methods marked with * are patch-based. Performance of each method extracted from the literature, where the method was first introduced. NN=neural network, SVM=support vector machine, PLS=partial least squares regression.

Method	Algorithm	Predictors	Dataset	Accuracy	Sensitivity	Availability
ProMate* (Neuvirth *et al.*, 2004)	Bayesian	Atom distribution, chem. character of atoms, pairs of amino acids, hydrophobic patches rank, evolutionary conservation, sequence distance within the circle, secondary structure, bound water	A[a]	70%	63%	Batch calculations on MultiProMate web page
cons-PPISP (Chen and Zhou, 2005)	NN	sequence-profiles and solvent-accessibility	B[b]	86%	17%	Batch analysis available via provided python submission scripts
PINUP* (Liang *et al.*, 2006)	Linear regression	Energy-based score, conservation and propensities	A	44.5%	42.2%	Discontinued, available as a downloadable package
PPI-Pred* (Bradford and Westhead, 2005)	SVM	Surface shape, conservation, electrostatic potential, hydrophobicity, residue propensity, ASA, curvedness	C[c] A	76%, 51%[d]	N/A	Batch calculations on the web page[e], currently unavailable
SPPIDER (Porollo and Meller, 2007)	NN	Difference between SABLE-predicted and observed solvent-accessibility, conservation and structural features	D[f]	64%	60.3%	Single-chain queries through the web page
PIER* (Kufareva *et al.*, 2007)	PLS regression	propensities of 32 atom groups, based on the ASA	E[g]	60%	50%	Single-chain queries through the web page
meta-PPISP (Qin and Zhou, 2007)	Linear regression	Raw scores from cons-PPISP, ProMate and PINUP	F[h]	51%	50%	Batch analysis available via provided python submission scripts

[a] 57 transient heterocomplexes, both unbound and bound structures present in PDB, antigen-antibody complexes excluded
[b] 1156 homo- and heterodimers
[c] 180 transient and obligate complexes
[d] on the ProMate dataset
[e] `http://bmbpcu36.leeds.ac.uk/ppi_pred/`
[f] 262 heterocomplexes and 173 non-redundant homocomplexes
[g] 748 complexes, obligate and transient, antigen-antibody complexes excluded
[h] 35 non-redundant enzyme-inhibitor complexes, Enz35

5.1.2.1 Neural networks developed for protein-protein interface prediction

Several groups have utilised neural networks for interface prediction in the last decade. Promising performance was achieved by Fariselli *et al.* (2002). Using a sequence profile for an 11-residue patch (the predicted residue and 10 closest spatial neighbours) in three-fold cross-validation, they reported 72% accuracy, 56% sensitivity and a correlation coefficient of 0.43 for the interface class. However, this performance was later claimed to be an overestimate, when tested by Porollo and Meller (2007), owing to the redundancy of the dataset. The neural network built by Ofran and Rost (2003) relies solely on sequence information achieving poor sensitivity (not reported in the original publication, rather estimated by de Vries and Bonvin (2008)) and was consequently removed from further comparison with methods built on structural data. Another neural network, actually a consensus of several models, cons-PPISP (Chen and Zhou, 2005) was trained and tested on interfaces inferred from PDB structures, thus allowing bias towards crystal contacts. Initially displaying high accuracy (86%), but low sensitivity (17-19%), it was further refined to achieve improved performance on their test set. Although that goal was eventually achieved by combining several neural networks, optimisation on the specific (and fairly limited in size) test set, apparently lead to overfitting, as Zhou and Qin (2007) reported significantly lower performance during their benchmarking procedure.

Porollo and Meller (2007) used SABLE, a previously developed predictor of relative solvent-accessibility, as one of the interface-specific features. They noticed that this tool, when predicting solvent-accessibility of a residue, corresponds better to the values in the bound rather than the unbound state. Therefore, when utilised on monomeric chains, some surface residues have unexpectedly low predicted relative solvent-accessibility, and these residues frequently correspond to interface residues. Consequently they built SPPIDER, a multilayer perceptron combining conservation, structural attributes and rASA values, outperforming the Fariselli model mentioned above. They claimed SPPIDER had high sensitivity and specificity (presented in

Table 5.3), however Zhou and Qin (2007), when presenting novel interface instances, achieved a specificity of 47% with 43% sensitivity, with even lower performance on the CAPRI25 dataset.

5.1.2.2 Random forests aimed at interface prediction

Random forest is a binary classifier, based on the majority-voted prediction of several random trees. It was chosen for several favourable features: (i) convenience for datasets with a large number of (potentially overlapping) attributes, (ii) no tendency towards overfitting, (iii) it calculates fewer parameters requiring less CPU time than other models with a similar dataset, (iv) since it is a tree-based method, it has an easy-to-visualize model structure (at least single trees do), and finally, (v) it provides principal-component-like attribute importance, which can be used for dimensionality reduction.

Šikić *et al.* (2009) have used random forests to build both sequence-only based and combined sequence- and structure-based interface predictors with 80% precision and 25% recall and 76% precision and 38% recall during 10-fold cross-validation, respectively. Unfortunately, they do not provide their classifier, nor have they benchmarked it against other interface datasets omitted from the training procedure. However, random forests are becoming increasingly popular (Breiman, 2001): they have shown great potential in prediction of biological problems such as activity prediction from chemical structure (Svetnik *et al.*, 2003), renal tumour classification (Shi *et al.*, 2005) and genome-wide association studies and detection of multiple-sclerosis-linked gene candidates (Goldstein *et al.*, 2010).

The comparative analysis of several datasets using different machine learning methods, summarised in Table 5.4, showed that although all the methods display high accuracies measured during cross-validation, their performance deteriorates severely when new data are used (Zhou and Qin, 2007). There is an obvious potential for improvement, and hopefully, careful data extraction and feature selection will improve the models, regardless of the choice of test set (used during benchmarking).

Table 5.4: The benchmarking of several interface-prediction methods on the two datasets (Zhou and Qin, 2007).
For the dataset of 35 enzyme-inhibitor complexes (Enz35), accuracy was measured at a coverage of 50%, and for 25 CAPRI targets accuracy was measured at 30% coverage.

Datasets	PPI-Pred	SPPIDER	cons-PPISP	ProMate	PINUP	meta-PPISP
Enz35	27%	33%	36%	38%	48%	50%
CAPRI25	23%	25%	26%	26%	28%	31%

To that end, this project set out to develop a novel protein-protein interface classifier, optimised for the residue-level prediction within SAAPdb.

5.2 Methods

This section lists methods used to perform data preparation, utilising and optimising several machine learning methods to build an interface predictor, and finally benchmarking against similar previously published methods on an independent dataset of interfaces, upon which none of the methods was trained. The technical aspects of the above mentioned procedures, and tools and resources used in the process, are described in Sections 5.2.1–5.2.6, and they all refer to the PQS_{nr} dataset prepared and presented in the previous chapter.

5.2.1 Preparing patches of various sizes

In order to sample parts of the protein surface and test for a certain combination of physico-chemical properties indicating a likely interface, the surface had to be fragmented. These fragments, hereafter termed **surface patches**, were built to be continuous with a fixed maximum patch radius, approximately circular in shape, and include at least one highly solvent-accessible residue, called the **patch centre**.

The PQS_{nr} dataset contained 4345 protein chains containing protein-protein interfaces, residues sorted into buried and surface residues, the latter further divided into interface and non-interface residues. When separating surface residues into interface and non-interface, an additional set of markedly solvent-accessible residues ($rASA^m > 25$) was collected, each representing a patch centre, i.e. a starting point when building a patch.

Before presenting the algorithm developed to build surface patches, several terms need to be introduced:

Geometry vector is a Euclidean vector, defined for a residue. Its initial point is the C_α atom of the residue in question, and the terminal point is the centre of geometry of the 10 spatially closest neighbours of that residue (calculated by averaging C_α atom coordinates).

Solvent vector is defined for a residue as the opposite vector to the geometry vector. More precisely, its initial point is also the C_α atom of the residue; it has the same length but the opposite direction to the geometry vector.

Solvent angle, defined for two residues, is an angle between the solvent vectors of the two residues.

Contact radius is defined for a pair of atoms, as the sum of the van der Waals radii of the two atoms, plus a tolerance distance (here set as 0.2Å). Two atoms are *in contact* if the distance between these atom centres is less than the contact radius.

Once a patch radius had been specified, the algorithm builds a patch iteratively from each patch centre in PQS_{nr}. The algorithm was:

i `Determine all surface residues with at least one atom centre within the patch radius from the patch centre. These are candidate residues for that patch, and their atoms are added to the set of unlabelled atoms` U.

ii `Label the highest` ASA^m `atom in the patch centre residue as belonging to that surface patch.`

iii `For each labelled atom` L, `test if any of the unlabelled candidate atoms (`U`) are within the contact radius. If` L `and` U `are in contact, and if the solvent angle between` L `and` U `is less than` $120°$, `label` U.

iv `Repeat step iii until no new candidate atoms are labelled.`

The 120° angle check was used similarly to previous work by Jones and Thornton (1997) and Pettit *et al.* (2007), in order to eliminate opposite sides of a deep pocket being merged into one patch, which would result in discontinuous (and biologically meaningless) patches.

The only tuning parameter when building a patch was the patch radius. Three kinds of input for the classifiers were prepared using three patch sizes from PQS_{nr}, shown in Table 5.5. Each was a set of surface patches on all chains and three patch sizes were chosen: single-residue, patch radius 9Å and 14Å and were named P_{nr_res}, P_{nr_9} and P_{nr_14}, respectively. For more details, see discussion in Section 5.3.2.

Table 5.5: A range of patch sizes.
Patch area is calculated using patch radius and formula for the area of the circle: $r^2\pi$. Counts of residues were obtained by building all patches with certain radius using PQS_{nr} and the aforementioned algorithm, and calculating the mean number of residues in patches.

Radius (Å)	Area (Å^2)	Residues	Corresponds to:
r=9	254	7	Smallest observed biologically relevant interfaces (Section 5.3.2)
r=14	616	20	Minimal interface size to obtain a hydrophobic core (Bogan and Thorn, 1998)
r=16	804	26	Average interface size by Lo Conte *et al.* (1999)

5.2.2 Preprocessing interface attributes

5.2.2.1 Class value

A patch could adopt one of two class values: interface, I or surface (i.e. non-interface surface), S. Let P be the set of residues in the patch under consideration, and I is the set of interface residues for the chain to which this patch belongs, previously identified and stored as explained in Section 4.3.1. Then $rASA_P^m$ denotes the sum of solvent-accessible surface area of all residues in a patch and by analogy, $rASA_I^m$ denotes the solvent-accessible surface area of residues within that patch, which have previously been labelled as interface residues:

$$rASA_P^m = \sum_{i \in P} rASA^m(i), \quad rASA_I^m = \sum_{j \in I} rASA^m(j) \tag{5.2}$$

where $rASA^m(i), rASA^m(j)$ are monomeric rASA value for residues i and j, respectively[10]. Then

$$rASA_o = \frac{rASA_I^m}{rASA_P^m} \tag{5.3}$$

[10] calculated by the `solv` algorithm, as specified in Section 2.2.1

represents the fraction of the surface area of a patch occupied by the atoms in interface residues, i.e. the interface-patch overlap. If a patch overlapped with an interface in at least half of its surface ($rASA_o \geq 0.5$), that patch was assigned class I. If there was no overlap between a patch and the interface on that chain ($rASA_o = 0$), the patch was labelled as class S. All other patches[11] remained unlabelled and were removed from further consideration.

5.2.2.2 Training attributes

In addition to the class value, each patch had eight features assigned to it, identified as predictive of interfaces in the last chapter. For simplicity, attributes were grouped into four combinations (sequence-based attributes are part of the STR category):

STR Structural: propensities, hydrophobicity, planarity, disulphide bonds, hydrogen bonds, secondary structure

STR+F Structural + FOSTA: all structural, FOSTA-based conservation

STR+B Structural + BLAST: all structural, BLAST-based conservation

ALL Structural + FOSTA + BLAST: all structural, FOSTA-based conservation, BLAST-based conservation

For P_{nr_9} and P_{nr_14} the value of a property X for a patch was calculated as the arithmetic mean of values of X for all residues in that patch:

$$X_P = \sum_{i=1}^{N} x_i/N \tag{5.4}$$

[11] in effect, corresponding to the rim of the interface

where $i = 1, ..., N$ are N residues in patch P, and x_i is value of property X for residue i. Since patches in P_{nr_res} each comprised a single residue, patch values were identical to that residue's for all attributes except planarity. Planarity of a single-residue patch was calculated based on the coordinates of that residue and the 7 spatially closest neighbours.

The overall secondary structure type assigned to the whole patch is defined based on the percentage of residues belonging to a secondary structure type, as proposed by Jones and Thornton (1995):

H , if $\alpha > 20\%$ and $\beta < 20\%$,

E , if $\alpha < 20\%$ and $\beta > 20\%$,

EH , if $\alpha > 20\%$ and $\beta > 20\%$,

C , if $\alpha \leq 20\%$ and $\beta \leq 20\%$.

As many machine learning models support only numerical non-class attributes, the secondary structure attribute needed to be transformed. Typically, a nominal attribute H with N possible categories is transformed into N binary attributes. The transformation from nominal secondary structure attribute $SS \in (H, E, EH, C)$, into binary attributes SS' is shown in Equation (5.5), increasing the total number of attributes by three (in the case when EH secondary structure is allowed, i.e. for $r = 9$ and $r = 14$ patches), or two (for single-residue patches) extra secondary structure attributes.

$$SS' = \begin{cases} (1,0,0,0), & \text{if } SS = H \\ (0,1,0,0), & \text{if } SS = E \\ (0,0,1,0), & \text{if } SS = EH \\ (0,0,0,1), & \text{if } SS = C \end{cases} \tag{5.5}$$

Alternative encodings are also possible – see Section 5.4.

5.2.3 Building WEKA classifiers

There are numerous approaches and algorithms in machine learning, and often an experimenter needs to try out several of them to identify the most suitable technique for the problem in question. Built to that end, several resources such as WEKA (Hall *et al.*, 2009) and RapidMiner (Mierswa *et al.*, 2006) implement a wide range of tools and techniques, simultaneously offering a user-friendly interface to create, optimise and evaluate machine learning experiments.

This machine learning started by trying out all available methods implemented within the WEKA platform: an open-source, Java-based collection of machine learning algorithms. All supervised classifiers implemented in `WEKA3.6.3` were trained on P_{nr_9}, with all default parameters. For a full list of classifiers, see Figure 5.4. Two supervised methods that have been tested in more detail over a wider range of parameters and training data setups will be presented below in Sections 5.2.3.1 and 5.2.3.2.

5.2.3.1 Multilayer perceptrons

The WEKA implementation of the test (`weka.classifiers.functions.MultilayerPerceptron`) was used. The model was trained on normalised attribute values[12] using a sigmoid function, using 5 or 50 hidden nodes over 500 epochs, with all other model variables set to WEKA's defaults.

5.2.3.2 Random forests

The `weka.classifiers.trees.RandomForest` method implemented in WEKA was used. Based on the random forest algorithm developed by Breiman (2001), this algorithm builds unpruned random trees, without limiting the tree depth. This classifier is incapable of handling missing attribute values, it introduces an imputed value instead: usually the mean value for continuous numerical attributes, or the most common category for nominal attributes (other strategies for building models with some data missing have been presented in Section 5.1.1.3). This practice was acceptable for all attributes used, except FOSTA: $\sim$ 80% of instances in all patch sets were missing the FOSTA-based conservation score, mostly as a result of lack of functionally-equivalent proteins (a minimum of 4 FEPs was required to build a multiple sequence alignment, as presented in Section 4.3.2.7). Thus in order to avoid building a model on many imputed values, all instances with missing FOSTA values were removed from the training and testing datasets when this method was used.

Let p be the total number of attributes provided for every data point (class value excluded). For the P_{nr_res}, p was 8/9/9/10 for $STR, STR+F, STR+B$ and ALL model settings, and 9/10/10/11 for P_{nr_9} and P_{nr_14}. The default value for m_{try}, introduced in Section 5.1.1.7, is $\sqrt{p}$, rounded to the closest integer value. In other words, for the range of p from 8 to 11, the method's default is $m_{try} = 3$, however a range of values from 2 to 9 was also tested.

The common recommendation for the tree-number optimisation is to increase the number of trees until the OOB error stops decreasing. Unfortunately, this could not

[12] i.e. all attribute values are adjusted to range between -1 and 1

be achieved here; the WEKA implementation of the random forest method is prohibitively memory-demanding and allocating 20G of RAM was insufficient to build random forests with more than 150 trees for this training dataset. Therefore only a range from 50 to 150 trees per ensemble were used. Finally, although strictly there is no need to perform cross-validation when training a random forest[13], a 10-fold cross-validation procedure was employed here in order to obtain other performance measures (i.e. correlation coefficient, F-measure, sensitivity, etc.), which would facilitate comparison with other machine learning methods.

5.2.4 Preparing the benchmarking dataset

A dataset of PQS biological units was obtained following the data-quality filtering procedure developed for the verification of the PQS_{all} dataset, presented in Section 4.2.1.2. The PQS server was discontinued in August 2009, and it stopped automatically updating the list of biological units by including new PDB structures in early 2010. This last release of PQS containing 61265 structures was downloaded, and filtered to eliminate low-quality data and monomeric structures. As previously mentioned in Section 4.2.1.1, the PQS_{all} dataset consisted of protein-protein complexes gathered from PQS, not later than March 2009. Therefore removing all complexes present in PQS_{all} from complexes obtained from the 61265 structures, yielded a complementary dataset termed PQS_{bench}, including only complexes released by PQS after March 2009 and before it ceased.

[13]OOB error, introduced in Section 5.1.1.7, is calculated on instances omitted during the training step, equalling to 1-accuracy (the error rate) during cross-validation

5.2.5 Preparing interface predictions from other classifiers

Several previously published protein-protein interface prediction tools were used to obtain predictions of likely interfaces for the protein chains in the PQS_{bench} dataset: PPI-Pred is currently unavailable for testing, ProMate was accessed through the web page for batch queries[14] using the default combination of scores and extracting amino acids coloured according to their probability of comprising an interface (by setting the temperature factor in the PDB file to this value). SPPIDER predictions were obtained from `http://sppider.cchmc.org/`, using the SPPIDER II classifier. PIER predictions were obtained from `http://abagyan.ucsd.edu/PIER/pier.cgi` as downloadable comma-separated value files. meta-PPISP (and PINUP scores used within) were obtained from `http://pipe.scs.fsu.edu/meta-ppisp.html`.

5.2.6 Benchmarking

Each surveyed classifier provided residue-level predictions as numerical values. Thresholds identical to the ones used in the original papers were chosen for all the methods to indicate a positive prediction (residue predicted as interface): $p > 70$ for ProMate, predicted by minimum of 5 neural networks for SPPIDER, minimum score of 30 for PIER and $p > 0.34$ for meta-PPISP. In the case of PINUP, the original classification was based on clustered patches, but since meta-PPISP's output provides only raw PINUP scores (scaled to 0.0 to 1.0 range), a 0.5 cut-off was chosen to differentiate interface from non-interface residue.

[14]`http://bioinfo.weizmann.ac.il/promate/`

5.3 Results and discussion

After careful consideration, it was decided that none of the available methods surveyed in Section 5.1.2 was suitable for the task at hand, and an in-house interface prediction method should be developed to be added to the SAAPdb pipeline, for both methodological and technical reasons.

In terms of methodology several common weaknesses were identified, in particular training on one type of interface (transient in Neuvirth *et al.* (2004) and obligate in Bradford *et al.* (2006)); optimising models for lower specificity in order to obtain higher sensitivity, thus increasing the false positive rate, e.g. Chen and Zhou (2005); and finally, using only sequence data to obtain a more diverse training set. All sequence-only predictors (such as Res *et al.* (2005), Ofran and Rost (2003; 2007)) were eliminated; as SAAPdb focuses on structural effects and thus all residues that will be modelled for putative interface sites are, by definition, mapped to available protein structure. Consequently, ignoring structural information to widen the range of proteins on which the model is trained makes no sense in the context of this project. For references and further discussion on the sequence-only interface prediction, see Section 4.3.2.7.

While inconsistencies in methodologies presented above were considered suboptimal, the prevailing reason for developing a novel prediction was its availability, and insurance of future maintenance. The majority of surveyed methods provided access through a web page, however none of the methods that performed well in predicting interfaces not used during the cross-validation step had freely accessible source code.

The previous chapter describes in detail how an extensive and up-to-date dataset of interfaces was built, and which features were chosen as predictors of interface regions on the surface of protein structures, based on that dataset and previously published literature. This chapter reports how these features were transformed into a machine-readable format, followed by testing of various prediction approaches, finally presenting a comparison of developed models to existing predictors on a novel benchmarking dataset of complexes. This prediction method was built in order to predict likely interfaces at the residue-level as a part of the SAAPdb pipeline, ultimately compensating for the sparse interface-damaging SAAPs owing to the lack of protein-protein interface data in the PDB.

5.3.1 Protein-protein interface data

The robustness 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. At the same time, interface data are limited and there is no gold-standard dataset to train and test models, in a way easily comparable with other publicly-available methods. Ideally, one should have three mutually independent subsets of a dataset of interfaces, each sufficiently large and representative in respect of features of the full dataset. These three subsets would then be used for parameter evaluation (choosing the minimal set of features to be used for classification purposes), training the classifier, and testing the performance of the classifier, respectively. In reality, this is rarely possible and the widely accepted compromise is to choose parameters based on the whole dataset, followed by cross-validation, where the whole dataset is divided into training and testing subsets several times in order to maximise the size of the training set, without adding bias to it. For more details on data sampling, see Section 5.1.1.2.

At the same time, performance of similar methods is often reported based on the testing set during the cross-validation. Considering each method uses a slightly different definition of the interface and different, often partially overlapping datasets, comparing methodologies from various authors is very difficult. Indeed, so far there has only been one attempt to compare different interface-prediction tools on a dataset of protein-protein complexes on which no method was trained (Zhou and Qin, 2007).

First, they showed that testing on an independent set of interfaces (25 CAPRI targets) none of which was included in cross-validation, yielded significantly lower sensitivity and specificity than reported in the original publications introducing these methods. This result, along with impaired performance when tested by de Vries and Bonvin (2008) (marked by "this work" in their Table 3), strongly indicates that comparing sensitivity and specificity scores obtained from methods' original publications (unless these have been trained on the same data) should be avoided as unreliable.

Further, Zhou and Qin (2007) compared performance of the same methods on two datasets: CAPRI and Enz35, the latter containing several interfaces on which some methods used in benchmarking were trained. Figure 1 in Zhou and Qin (2007) indicates that choosing a single sensitivity value and ordering methods by their specificity is dataset-specific and consequently should not be used to compare performance: order of tools in terms of accuracy stayed the same when they compared Enz35 at coverage 50% and CAPRI at coverage 30%. However if both are observed at coverage 50%, the order of methods is different.

Both arguments served as an incentive to build a novel benchmarking dataset, to be used for an extensive methodology comparison presented in Section 5.3.5. Lastly, to ensure a dataset of interfaces no other method was trained on (including the method presented in this chapter) for benchmarking, the PQS_{nr} dataset was based on a slightly out-of-date version of the PQS list of biological units. This allowed the accumulation of newly solved structures of complexes, then used during the benchmarking step.

5.3.2 Choosing patch sizes

The surface of the protein is fragmented and each segment is then assigned to the interface or non-interface class. While the majority of methods use patches of surface residues (Jones and Thornton, 1997; Chakrabarti and Janin, 2002; Bahadur *et al.*, 2003; Neuvirth *et al.*, 2004; Caffrey *et al.*, 2004; Bradford and Westhead, 2005), there is no consensus on the optimal patch definition or size; for a review of patch-based methods see de Vries and Bonvin (2008). Typically, patches are continuous in nature, often mutually overlapping, consisting of adjacent residues on the surface of the same protein chain. The size of structure-based patches can be defined based on an empirical interface size (Jones and Thornton (1997) define the patch size for every complex to be equal to the observed interface size in that complex), fraction of the protein surface (Bradford and Westhead (2005) define patch size as a circle with $6-8\%$ of the surface area in a complex), or based on a fixed sphere radius of e.g. 10Å (Neuvirth *et al.*, 2004) around patch centres (usually uniformly distributed over the surface of a complex). Non-patch-based methods usually identify features on a residue- or atom-level. This approach can, in effect be viewed as a special case of minimal patch size – each residue is a disjoint patch; for examples of non-patch-based interface predictors, check Liang *et al.* (2006) and Qin and Zhou (2007).

The way the surface of a protein is sampled from fragments influences their classification into surface or interface fragments. The advantage of using small fragments, (i.e. patches with one residue at a time) is the lack of overlap between neighbouring patches, and an unambiguous classification of training examples into interface or surface classes[15]. Larger patches on the other hand, overlap to some extent with other patches and with the interface. This requires a ranking system for patches predicted to be in the interface: there are bound to be several patches labelled as interface and a metric has to be introduced to distinguish more likely hits among them. Also, to label the training examples, a (somewhat arbitrary) threshold has to be introduced defining which patches will be labelled as interface in the training phase: here a 50% surface overlap with the known interface was used. The advantage of larger patches is that some kind of shape property can be defined, providing more signal from which the classifier can learn. All points considered, it made sense to survey a range of patch sizes when searching for the most appropriate model.

[15]a one-residue patch can either be an interface residue, or be a surface residue

The results presented in this chapter cover a range of sizes for structure-based patches, from single-residue patches in the P_{nr_res}, to small patches in P_{nr_9} and large patches P_{nr_14}. These three datasets had 500543, 398531 and 319658 patches, respectively. None of the datasets had a minimum size of patch specified, i.e. a P_{nr_14} patch could contain examples with only one residue provided no other surface residues were found within the 14Å distance. The single-residue dataset of patches contains a lot more patches owing to the way the datasets were built: P_{nr_res} contains a patch for every surface residue (residues with $rASA^m > 5$), while in P_{nr_9} and P_{nr_14} a patch was created for every patch centre (residues with $rASA^m > 25$).

Patch size was deliberately chosen to be smaller than the average interface, thus not using the $r = 16$ (see Table 5.5), because the final prediction is destined to be used at a residue level in SAAPdb; the aim was to get a decently performing predictor, rather than perfect the prediction of the whole interface. Moreover, P_{nr_9} and P_{nr_14} patches resemble 'hot regions' (residues within 10Å distance), defined by Keskin *et al.* (2008). Several hot regions comprise an interface; effects of mutations in residues within a hot region are cooperative, while effects among hot regions are considered additive. Thus, mutations within each patch can be observed as an independently evolving unit, once interface prediction is added to the SAAPdb pipeline.

5.3.2.1 Small patches

The minimal patch size was determined empirically, based on the smallest observed interface in the PQS_{nr} dataset. The dataset of non-redundant chains was sorted by the size of interfaces identified using the solvent accessibility decrease criterion (Section 4.1.2). Each interface was then manually inspected using `RASMOL` (Sayle and Milner-White, 1995), in ascending order of interface sizes, to find the minimum size likely to present a biologically plausible contact between two protein chains, rather than crystal contacts PQS failed to identify and remove. The smallest interfaces satisfying several conditions[16] contained 6 or 8 amino acids, see Figure 5.3, which, if the interface was considered to be an ideal circle on the protein surface, corresponded to a patch radius of 9Å. Therefore, when building small patches using the algorithm defined in Section 5.2.1, a patch radius $r = 9$Å was used, aiming for patches which should be smaller or the same size as the smallest interface.

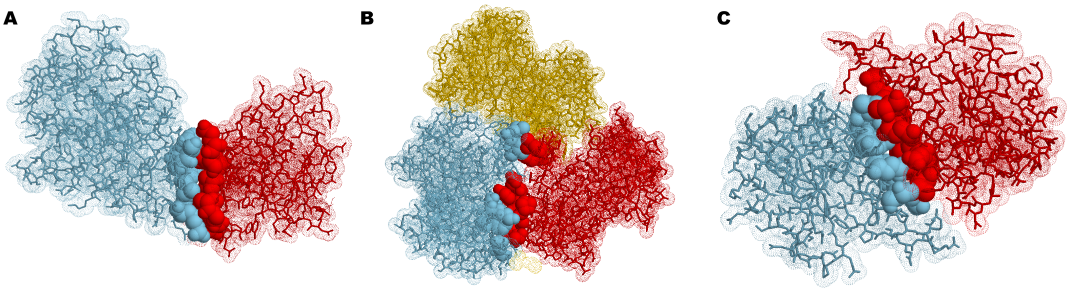

Figure 5.3: Smallest observed interfaces.
Interface residues are shown in spacefill mode. A) PDB ID: 3CR3 (heterodimer) 6 and 10 interface residues in chains A and D, respectively. B) PDB ID: 1CZY (homotrimer) 8 residues on chains A and C. C) PDB ID: 2QMS (homodimer) 8 interface residues on chains A and B.

5.3.2.2 Large patches

Large patches were created to resemble the minimum patch size needed for an interface core to occur according to Bogan and Thorn (1998). This was 600Å^2 per chain, which corresponded to a circular patch with radius of $r = 14$Å.

5.3.2.3 Single-residue patches

Single-residue patches were used in the following section to characterise interface and surface residues in PQS_{nr}, for all properties except planarity. By definition, shape indicators need coordinates for the set of nearby residues, to determine the local shape of the protein surface.

[16]surface complementarity, tight interchain packing and literature confirmation of a biological unit

5.3.3 Combining interface attributes

Each patch was initially assigned two types of values: a class value (interface or surface patch, defined in Section 5.2.2) and eight attribute values each corresponding to a sequence-, structure- or profile-based property identified in Section 4.3.2.8 to be a valid predictor: amino acid propensity, hydrophobicity, planarity, secondary structure, disulphide bonds, hydrogen bonds, sequence conservation based on FOSTA alignment and sequence conservation based on BLAST alignment.

All interface properties except the class and the secondary structure were numerical by definition; Table 5.6 summarises the ranges of patch values per attribute. Secondary structure was originally a categorical attribute with three possible outcomes for P_{nr_res}: helix, strand or other; or four possible values for P_{nr_9} and P_{nr_14}: helix, strand, helix/strand, or other. This attribute was transformed into three/four binary attributes as described in Section 5.2.2.2, consequently increasing the number of structural interface features to 8/9, respectively.

Table 5.6: Attributes used for model building.
str, *F* and *B* refer to attribute combinations, listed in the text below.

Attribute	Variable type	Value range
Propensity $^{(str)}$	Numerical (continuous)	[-1.04, 2.37]
Hydrophobicity $^{(str)}$	Numerical (continuous)	[-4.50, 4.50]
Planarity $^{(str)}$	Numerical (continuous)	[0.00, 7.88]
Secondary structure - helix $^{(str)}$	Numerical (binary)	(0,1)
Secondary structure - strand $^{(str)}$	Numerical (binary)	(0,1)
Secondary structure - helix/strand $^{*(str)}$	Numerical (binary)	(0,1)
Secondary structure - other $^{(str)}$	Numerical (binary)	(0,1)
Hydrogen bonds $^{(str)}$	Numerical (continuous)	[0.00, 1.00]
Disulphide bonds $^{(str)}$	Numerical (continuous)	[0.00, 1.00]
FOSTA conservation $^{(F)}$	Numerical (continuous)	[0.00, 1.00]
BLAST conservation $^{(B)}$	Numerical (continuous)	[0.00, 1.00]
Class	Categorical (binary)	(I, S)

* defined for patches in P_{nr_9} and P_{nr_14}

5.3.4 Choosing the most appropriate machine learning method

5.3.4.1 Survey of classifiers using WEKA

Before focusing on a particular machine learning method, all supervised classifiers implemented in WEKA (Hall *et al.*, 2009) were trained on the dataset P_{nr_9} (using ALL attributes) without changing default parameter values, to validate the choice of the dataset and the attributes. The overall performance of unoptimised classifiers on the P_{nr_9} dataset, presented in Figure 5.4, was comparable to previously published protein-protein interface prediction methods in Table 5.3, confirming sufficient size of the P_{nr_9} dataset and predictive power of the implemented interface attributes.

All methods displayed relatively high specificity (~ 0.9) with somewhat lower sensitivity (~ 0.4). In other words, classifiers were more likely to miss an interface residue than report a false positive: a favourable behaviour when the aim of the classifier is to indicate mutations in predicted interface sites. The best single indicator of a model's performance is the Matthew's correlation coefficient (MCC)[17]. Again, the majority of methods have similar, albeit mediocre MCCs of ~ 0.4, indicating the chosen parameters have some (although by no means exhaustive) predictive value. No clear preference for numerical (regression models) versus binary classification (trees and other rule-based models) methods was noticeable from performance measures.

Further, none of the produced models was an obvious choice significantly outperforming other methods used in this survey, nor the previously built methods for protein-protein interface prediction, introduced in Section 5.1.2.

Considering that no WEKA models were particularly successful 'off-the-shelf', when compared either with previously published interface predictors listed in Table 5.3 or compared with other models built in this survey, further

[17] as it uses all four counts from the confusion matrix for a binary classification problem: TP, TN, FP and FN, for more details see definition of MCC in Section 5.1.1.4

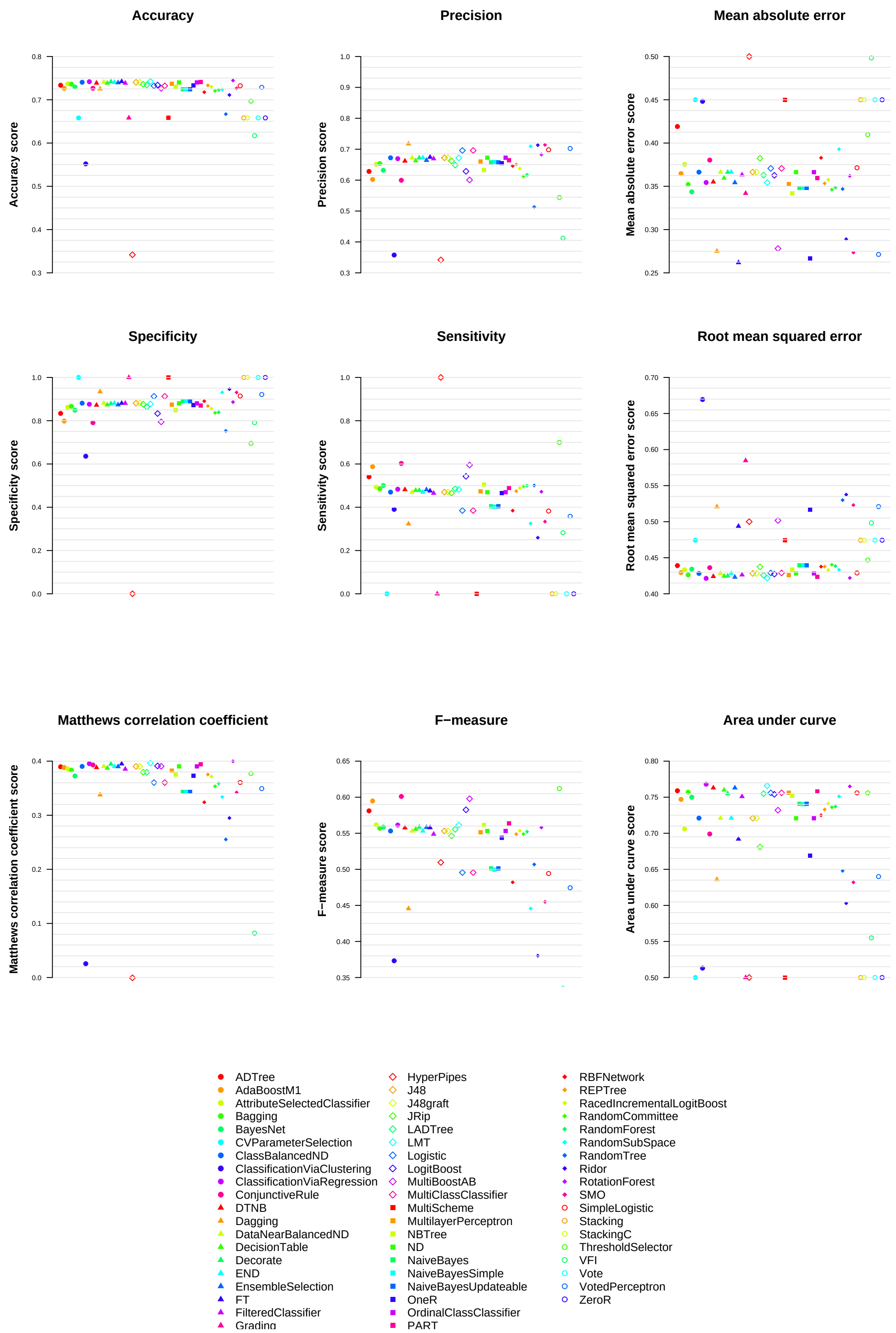

Figure 5.4: A survey of machine learning tests used on interface data. All nine performance measures were defined in Section 5.1.1.4. Each presented score for a model is an average of 10 scores obtained during 10-fold cross-validation.

analysis focused on optimising two learners, chosen for their different learning methodology, and previous strengths shown on similar data mining problems.

A multilayer perceptron was chosen as a representative of the numerical models, providing a numerical score between 0 (non-interface surface) and 1 (interface): this method, introduced and discussed below in Section 5.3.4.2, has already proven its efficacy in protein-protein interface prediction in SPPIDER, cons-PPISP and the Fariselli method. A random forest was chosen as a binary classifier representative: this method was shown to outperform decision trees, and at the same time it provides the structure of the model in a tree-like form which is more intuitive to interpret in a human-readable fashion (unlike the weights in the hidden layer of multilayer perceptrons). Random forests have proven useful recently for similar structural predictions, the efficacy in prediction of protein-protein interfaces is discussed below in Section 5.3.4.3.

5.3.4.2 Neural network prediction

Predicting protein-protein interfaces by building a neural network was investigated by using the `MultilayerPerceptron` method in `WEKA`, with the classification cut-off set to 0.5. Originally the model was set to have 5 hidden layer nodes ($H = 5$), but $H = 50$ was also tested with no improvement in performance, yet a 10-fold increase in the time required to build the model. Therefore all results presented in Table 5.7 contained five nodes in the hidden layer (for more details on the experimental setup refer to Section 5.2.3.1), varying three patch sizes, and five combinations of attributes (the four introduced in Section 5.3.3, plus $STR+F$ with all missing FOSTA conservation scores removed, hereafter referred to as $STR+F^{*}$). The latter attribute setup was used to test whether the lack of missing values affects performance of this neural network; it is worth noting here that this dataset is significantly smaller, consisting of around 20% of the patches.

Table 5.7: Neural networks performance, averaged over 10-fold cross-validation. Performance measures: accuracy (ACC), precision (PREC), specificity (SPEC), sensitivity (SENS), Matthew's correlation coefficient (MCC), F-measure (F), root mean squared error (RMSE), mean absolute error (MAE), area under the curve (AUC); SR stands for single-residue patches. Highest and lowest score in every column are coloured blue and red, respectively, all scores are averages over 10-folds of cross-validation.

Patch radius	Attributes			ACC	PREC	SPEC	SENS	MCC	F	RMSE	MAE	AUC
	structural	FOSTA	BLAST									
SR	✓	✓	✓	0.751	0.585	0.964	0.148	0.200	0.237	0.423	0.358	0.652
SR	✓		✓	0.751	0.583	0.964	0.148	0.198	0.236	0.423	0.358	0.651
SR	✓	✓		0.749	0.584	0.966	0.137	0.188	0.222	0.426	0.362	0.636
SR	✓	✓*		0.760	0.597	0.969	0.138	0.198	0.225	0.417	0.348	0.661
SR	✓			0.749	0.582	0.966	0.135	0.186	0.219	0.426	0.363	0.631
9	✓	✓	✓	0.735	0.653	0.892	0.415	0.355	0.507	0.426	0.363	0.745
9	✓		✓	0.736	0.653	0.892	0.417	0.356	0.509	0.426	0.363	0.745
9	✓	✓		0.733	0.649	0.893	0.406	0.347	0.500	0.429	0.367	0.737
9	✓	✓*		0.745	0.649	0.903	0.395	0.352	0.491	0.421	0.354	0.746
9	✓			0.733	0.649	0.893	0.405	0.346	0.499	0.429	0.368	0.735
14	✓	✓	✓	0.759	0.707	0.864	0.574	0.462	0.634	0.409	0.334	0.806
14	✓		✓	0.759	0.708	0.865	0.574	0.462	0.634	0.409	0.334	0.806
14	✓	✓		0.756	0.703	0.862	0.569	0.455	0.629	0.412	0.339	0.800
14	✓	✓*		0.766	0.699	0.877	0.551	0.458	0.617	0.403	0.325	0.808
14	✓			0.755	0.703	0.863	0.567	0.454	0.627	0.412	0.340	0.799

* instances with missing FOSTA value removed - 21% of the original dataset remained

Table 5.7 presents the results of this range of models, grouped by increasing patch size. First, the overall prediction model quality (measured by the increase in MCC, F, AUC, and decrease of RMSE and MAE, for definitions see Section 5.1.1.4) increases as the patch size grows. Single-residue patches had the highest specificity, at the expense of very low sensitivity: this model highly favours outputting the 'S' class, thus the significant drop in the correlation coefficient. Using current definitions of interfaces and attributes, residue-based prediction of interfaces is not efficient.

For all three patch sizes, the best overall performance measures were achieved when at least some evolutionary information was added: ALL, $STR + B$ and $STR + F^*$, clearly outperformed STR. Finally, the removal of patches lacking FOSTA-based conservation ($STR + F^*$) improved MCC and AUC scores and reduced both error rates, when compared with $STR + F$. In other words, modelling interfaces using a larger training dataset where one of the attributes was often missing (FOSTA values were missing for 79% of instances), is less profitable than eliminating the instances with missing values, even when the `MultilayerPerceptron` method claims to handle missing values[18].

To conclude, the multilayer perceptrons developed here for P_{nr_14} with ALL, $STR + B$ or $STR + F^*$ attributes seemed promising candidates for benchmarking against cons-PPISP and SPPIDER, the results of which are presented below, in Section 5.3.5.

5.3.4.3 Random forest prediction

Random forest provides a majority vote class based on an ensemble of trees each independently trained to predict an interface on a subsample of the training dataset. As previously mentioned in Section 5.1.1.7, random forest has two adjustable parameters, which have been tested over a range of values for the P_{nr_9} dataset, and results

[18]the input node for the attribute with the missing value will output zero for that instance

are presented in Table 5.8. All tests in this section have been performed on significantly smaller datasets; during model building, this algorithm is unable to process instances with missing values, so by default it would substitute missing FOSTA-based (all $\sim 80\%$ of them) conservation scores with the mean value for this attribute. To avoid that, all instances with missing FOSTA values have been removed, creating datasets with 104989, 85277, 69865 patches for P_{nr_res}, P_{nr_9} and P_{nr_14}, respectively. None of the other attributes had a significant frequency of missing values to need filtering. The models were built on the full datasets as well (data not shown here), yielding 2-5% lower accuracy, specificity, precision and correlation coefficient, and a 1% increase in sensitivity with respect to the values reported in Table 5.8.

The results of the random forest parameter space search showed minor changes in performance when the number of trees and randomly chosen attributes are varied. Although there was an improvement when the number of trees was increased from 100 to 125 and 150, it did not justify the significantly longer time to build these models. In terms of m_{try}, varying between the choice of a random 2, 3 or 4 attributes during each split made no discernible difference, but when increased to $m_{try} \geq 5$, the performance deteriorated. The grey line listed twice, first in Table 5.8 and again in Table 5.9 represents the same test, to be used as the baseline for comparison of different parameter combinations. To conclude, the default parameters of $T = 100$ and $m_{try} = 3$ proved to be optimal, and were used further to test performance for various patch sizes and attribute combinations.

For all three patch sizes in Table 5.9, random forest models show the largest oscillations in performance when the list of predictors used during training is changed: structure-only based models have the lowest prediction power, with some improvement when either FOSTA- or BLAST-based conservation is added, and the best performance is obtained when both conservation scores were added (the ALL attribute setup). While there is obviously some overlap between these two conservation scores, they complement each other[19]. However, contrary to the expected performance

[19]one reflecting long-term evolution, the other only functionally-equivalent proteins

Table 5.8: Random forests performance - surveying the parameter space.
Performance measures: accuracy (ACC), precision (PREC), specificity (SPEC), sensitivity (SENS), Matthew's correlation coefficient (MCC), F-measure (F), out-of-bag error (OOB); m_{try} stands for the number of randomly chosen attributes in every split; T is the number of trees. Highest and lowest score in every column are coloured blue and red, respectively, all scores are averages over 10-folds of cross-validation.

Patch	Attributes			Parameters		ACC	PREC	SPEC	SENS	MCC	F	OOB
radius	structural	FOSTA	BLAST	m_{try}	T							
9	✓	✓	✓	3	50	0.758	0.673	0.903	0.440	0.394	0.532	0.248
9	✓	✓	✓	3	75	0.760	0.679	0.905	0.440	0.399	0.534	0.243
9	✓	✓	✓	3	100	0.760	0.679	0.906	0.439	0.398	0.533	0.242
9	✓	✓	✓	3	125	0.761	0.681	0.906	0.440	0.400	0.534	0.240
9	✓	✓	✓	3	150	0.761	0.681	0.907	0.439	0.399	0.534	0.240
9	✓	✓	✓	2	100	0.760	0.680	0.907	0.437	0.397	0.532	0.243
9	✓	✓	✓	3	100	0.760	0.679	0.906	0.439	0.398	0.533	0.242
9	✓	✓	✓	4	100	0.760	0.678	0.905	0.441	0.399	0.535	0.244
9	✓	✓	✓	5	100	0.758	0.673	0.904	0.437	0.393	0.530	0.245
9	✓	✓	✓	7	100	0.757	0.670	0.902	0.438	0.391	0.530	0.246
9	✓	✓	✓	9	100	0.756	0.667	0.900	0.439	0.389	0.529	0.246

Table 5.9: Random forests performance - surveying data subsets.
Performance measures: accuracy (ACC), precision (PREC), specificity (SPEC), sensitivity (SENS), Matthew's correlation coefficient (MCC), F-measure (F), out-of-bag error (OOB); SR stands for single-residue patches; m_{try} stands for the number of randomly chosen attributes in every split; T is the number of trees. Highest and lowest score in every column are coloured blue and red, respectively, all scores are averages over 10-folds of cross-validation.

Patch	Attributes			Parameters		ACC	PREC	SPEC	SENS	MCC	F	OOB
radius	structural	FOSTA	BLAST	m_{try}	T							
SR	✓	✓	✓	3	100	0.755	0.537	0.944	0.194	0.208	0.285	0.249
SR	✓		✓	3	100	0.749	0.502	0.939	0.184	0.184	0.269	0.254
SR	✓	✓		3	100	0.737	0.453	0.913	0.213	0.170	0.290	0.267
SR	✓			3	100	0.710	0.370	0.875	0.218	0.114	0.274	0.294
9	✓	✓	✓	3	100	0.760	0.679	0.906	0.439	0.398	0.533	0.242
9	✓		✓	3	100	0.752	0.665	0.906	0.413	0.373	0.509	0.252
9	✓	✓		3	100	0.750	0.651	0.894	0.433	0.374	0.520	0.255
9	✓			3	100	0.733	0.608	0.881	0.405	0.327	0.486	0.271
14	✓	✓	✓	3	100	0.795	0.747	0.894	0.604	0.528	0.668	0.209
14	✓		✓	3	100	0.780	0.725	0.888	0.573	0.492	0.640	0.222
14	✓	✓		3	100	0.780	0.718	0.882	0.582	0.492	0.643	0.224
14	✓			3	100	0.764	0.691	0.871	0.555	0.453	0.616	0.243

presented in Table 4.3, it seems the BLAST-based conservation ($STR + B$) is more informative than the FOSTA-based one ($STR + F$). Not surprisingly, adding two non-orthogonal features (in ALL) improves the performance: a feature typical for the random forest.

Single-residue models clearly favour surface over interface class, outputting many TN and FN and not many FP and TP. Consequently, these models have low MCC indicating almost non-existent correlation between the predicted and the real class value. To conclude, the optimal random forest interface prediction is obtained for large patches (P_{nr_14}) with one, or preferably both conservation scores added to the structural interface features.

5.3.4.4 Random jungle

An alternative implementation of random forest algorithm called Random Jungle, `RJ` (Schwarz *et al.*, 2010) was initially tested[20], but was abandoned for inferior speed and lack of missing value handling capabilities, in comparison with the `WEKA` implementation. However, RJ provides a list of importance measures, a convenient feature not currently supported by `WEKA's` random forest method. The importance measure was introduced in Section 5.1.1.7: in short, it is a numerical indication of how much performance deteriorates if that attribute were to be removed from model building.

An `RJ` model was built for the P_{nr_9} dataset using ALL interface features, and default 100 trees and $m_{try} = 3$. The order of interface feature importance provided by the `RJ` algorithm is presented in Table 5.10:

[20]the main difference being `WEKA RandomForest` builds an ensemble of trees using `RandomTree`, while `RandomJungle` uses the `CART` algorithm to create trees

Table 5.10: Interface attributes, ordered by importance.
The higher importance score indicates more likely misclassification if that attribute is randomly permuted among instances in the model.

Importance	Attribute
7096	Amino acid propensities
5530	Planarity
5403	BLAST-based conservation
4704	FOSTA-based conservation
1846	Hydrophobicity
327	Hydrogen bonds
289	Secondary structure (C)
281	Secondary structure (H)
258	Secondary structure (E)
16	Disulphide bonds

To conclude on various machine learning models built above, all implemented methods indicate that models benefit from including sequence-based, structural and profile-based attributes. While the attribute with the highest importance is purely sequence-based, the second most informative interface feature is planarity which requires a high-quality protein structure to be available. Aside from hydrophobicity and planarity, other structural features had little effect: these properties might be poorly correlated with the appearance of interfaces, or simply not have been appropriately defined. Further, in the current experimental setup, the performance increases with the patch size, similarly to the work by Porollo and Meller (2007) smoothing attribute values over spatial neighbours increases the ability to predict protein-protein interfaces.

Contrary to conclusions reached in Chapter 4, both neural networks and random forests show improved performance when structural data are complemented by BLAST, when compared with $STR + F$. It was expected that using fewer homologues more likely to be conserved in terms of protein function will be more informative in context of interface prediction than broadening the alignments by less recently diverged homologues, some of which may have evolved to have alternative function and thus potentially display alternative evolutionary restrictions on the contact residues.

It turns out that the abundance of homologues in BLAST-based alignments[21] compensates for the introduced diversity. While for the neural networks direct comparison cannot be made ($STR+F$ has lots of missing values so it is not surprising $STR+B$ outperforms it; $STR+F^*$ is trained on a much smaller dataset than $STR+B$), in the case of random forests this trend is clearly discernible. While it was no surprise that data containing information about the long-term evolution was more informative, this trend also proved to be very convenient as FOSTA data are scarce, and regular updates (to make them more abundant) are computationally costly.

5.3.5 Comparison of interface prediction methods

The goal of this section is to compare machine learning approaches presented in Sections 5.3.4.2–5.3.4.3 with already existing methods, in a clear and reproducible way. As previously mentioned, virtually every experiment differs in the set of interfaces used, definition of interfaces and patches, attributes, and/or what constitutes a successfully predicted instance, an independent evaluation of methods should be performed using classical benchmarking, previously introduced in Section 5.1.1.5.

Further complicated by no standard datasets of protein-protein interfaces, it was not surprising to find only one review paper (Zhou and Qin, 2007) reporting comparison of several interface-predicting tools. In it, Zhou and Qin showed significantly lower performance on an independent set of proteins that has not been used in the training set in any of the methods compared: accuracy was reduced by between 4.8 and 18.2 percentage points, while coverage showed an insignificant decrease. However, both the datasets they used were fairly small, and the larger Enz35 dataset overlapped with some of the tested methods, thus raising questions about applicability of these benchmark results on future structural data.

[21] $E < 0.01$, for more details on how alignments were built and which sequences they contained, see Section 4.2.4.1

5.3.5.1 Benchmark dataset of complexes

One approach to ensure all the complexes in the benchmarking dataset have not been used before, is to obtain complexes added to the PDB *after* the last evaluated method was published. Here, the complexes for the PQS_{all} dataset, (defined in Section 4.2.1.1) were obtained in March 2009, and PQS was revisited in January 2010, extracting all complexes published in the PDB and automatically added to PQS in the meantime. This yielded 91529 chains, a 4% increase over $PQS_{filtered}$. The same percentage increase was obtained when the chains were culled into clusters of $>$ 25% sequence similarity using PISCES (introduced in Section 2.2.5). This indicates that the PDB is steadily growing in terms of new protein families.

4204 chains in 1306 novel protein complexes passed the data quality filters listed in Figure 4.2. These chains were all used for benchmarking, i.e. no same-cluster representative selection by PISCES was neccessary, yielding a benchmark dataset used to test the performance of various classifiers, hereafter termed PQS_{bench}. Conveniently, all prediction models evaluated here precede structures included in PQS_{bench}, ensuring a truly independent test of generalisation powers for various interface-predicting models.

5.3.5.2 Performance of various predictors

Protein-protein interface-predicting methods listed in Table 5.3 were applied to obtain their predicted classes for all residues in the PQS_{bench} dataset. The PPI-Pred online tool was inaccessible owing to technical reasons, thus eliminating this predictor from the analysis. Further, the PINUP website has been discontinued, allowing the download of the stand-alone predictor. However, this method is indirectly assessed through the meta-PPISP tool: meta-PPISP provides a linear combination of PINUP and two other prediction models listing their scores with the consensus class.

The results of the remaining surveyed methods: ProMate, PIER, PINUP, meta-PPISP and SPPIDER were compared with the three best-performing neural networks and three random forests chosen in Sections 5.3.4.2 and 5.3.4.3, respectively. Their performance on the PQS_{bench} dataset is listed in Table 5.11. Only residues labelled as interface or surface by the method were considered, since PQS_{bench} only includes interface and non-interface surface residues.

Table 5.11: Compared performance of several interface classifiers. Previously published methods are shown separated from classifiers developed in Sections 5.3.4.2–5.3.4.3
ACC=accuracy, PREC=precision, SPEC=specificity, SENS=sensitivity, MCC=Matthew's correlation coefficient, F=F-measure. The highest and the lowest score in every column are shown in blue and red, respectively.

Method	ACC	PREC	SPEC	SENS	MCC	F
ProMate	0.780	0.401	0.987	0.031	0.058	0.057
PIER	0.754	0.511	0.932	0.214	0.207	0.302
SPPIDER	0.759	0.472	0.783	0.676	0.410	0.556
PINUP	0.772	0.459	0.927	0.220	0.199	0.298
meta-PPISP	0.755	0.499	0.902	0.300	0.245	0.375
NN(ALL)	0.729	0.785	0.878	0.545	0.455	0.644
NN(STR+B)	0.727	0.787	0.881	0.538	0.452	0.639
NN(STR+F*)	0.726	0.781	0.876	0.541	0.449	0.640
RF*(ALL)	0.771	0.803	0.922	0.522	0.500	0.633
RF*(STR+B)	0.760	0.789	0.920	0.497	0.474	0.610
RF*(STR+F)	0.769	0.793	0.917	0.526	0.495	0.632

* instances with missing FOSTA value removed - 21% of the original dataset remained

ProMate (Neuvirth *et al.*, 2004) is a Bayesian predictor trained exclusively on transient interfaces, therefore testing it on a mixed set of both obligate and transient interfaces (i.e. PQS_{bench}), might not be the best indication of how well it performs the duties for which it was built. However, this evaluation is informative in the SAAPdb context: it provides a measure of how well this classifier would perform within the SAAPdb structural pipeline as PQS_{bench} represents an average set of chains to be added to the SAAPdb database. Although it displayed admirable performance during cross-validation (70% accuracy with 63% sensitivity), testing it on somewhat different data indicated this model has been overfitted: Zhou and Qin

(2007) achieved accuracy of 38% for 50% sensitivity using the Enz35 dataset. Similarly, ProMate did not perform well on PQS_{bench}: with the default $p > 0.70$ score (as suggested by Neuvirth *et al.*), it failed to predict a single interface residue for 49.8% of chains in PQS_{bench}, favouring lower probability output scores, and in turn predicting most test instances as surface residue.

PIER (Kufareva *et al.*, 2007) groups heavy atoms of a protein into 32 categories, based on their chemical properties, then trains a partial least squares regression model. PLS regression combines principal component analysis and multiple regression in order simultaneously to predict several mutually dependent class attributes, using a large number of independent training attributes (Geladi and Kowalski, 1986). Kufareva *et al.* (2007) reported 60% accuracy for 50% sensitivity during cross-validation; however, when tested on PQS_{bench} PIER outputs a high rate of 'surfaces', displaying poor correlation with the actual class ($MCC = 0.2$).

PINUP (Liang *et al.*, 2006) is a linear combination of three scores: amino acid propensities (found to have the strongest individual prediction power of the three scores used), conservation score and energy-score; the latter also being a linear combination of several elements originally developed by Liang and Grishin (2004) in order numerically to characterise representative amino acid conformations (i.e. rotamers). Trained on the same dataset as ProMate, it was reported to achieve 44.5% accuracy and 42.2% coverage during leave-one-out cross-validation. In other words, it predicts interfaces more often than ProMate, and therefore, is more often wrong in prediction of interfaces. In the Zhou and Qin (2007) review, PINUP slightly outperformed ProMate on both of the datasets, justifying the choice of all three scores which seem more appropriately to represent interfaces than ProMate's attributes of choice.

The PINUP raw score was obtained during meta-PPISP analysis of PQS_{bench} (see below), and a 0.50 cut-off point was introduced to indicate a residue predicted as interface by PINUP, avoiding the patch-building, clustering, and patch-ranking-based residue scoring. In this setup, PINUP showed a low correlation with the actual interface residues in the PQS_{bench} set, rendering it inappropriate for interface prediction based on this revisited performance.

Finally, the last tool used during benchmarking, building a predictor straight from features of interfaces (in contrast to meta-PPISP which reuses previously developed predictors) was SPPIDER (Porollo and Meller, 2007). This is a neural network trained on a combination of structural and sequence-based scores with added difference between observed and predicted (by SABLE) solvent-accessibility. This proved to be the most robust of all the surveyed methods: during cross-validation Porollo and Meller achieved 63.7% accuracy with 60.3% sensitivity, which deteriorated to 33% accuracy for 50% sensitivity in the Zhou and Qin benchmarking on Enz35. While this classifier shows somewhat lower specificity than PIER and ProMate, indicating it is more likely to predict a false positive, the overall performance measured by MCC and the F-measure show significant improvement over the other two methods. The success of SPPIDER in benchmarking additionally confirms that the solvent-accessibility of a residue should be added to the list of interface predictors.

As mentioned above, one meta-predictor was also surveyed: meta-PPISP is a linear combination of three previously defined interface-prediction models (Qin and Zhou, 2007), chosen for their maximal methodological difference in order to make the linear regression as informative as possible. Combining the cons-PPISP (Chen and Zhou, 2005), Promate and PINUP scores, it expectedly outperforms each of these methods utilised alone on the PQS_{bench} dataset, however the MCC and F-measure indicate it is still inferior to SPPIDER and all six models developed on the PQS_{nr} dataset.

In summary, ProMate, PIER and PINUP were not very efficient in predicting interface sites on the PQS_{bench} dataset, while meta-PPISP displayed only slight improvement. The six models built in this chapter all clearly outperform these previously published methods and, when composite performance measures (MCC and F-measure) are considered, are superior interface predictors even to the current state-of-the-art structure-based interface prediction method: SPPIDER. Therefore, the best among the six, RF*(ALL) will be added to the SAAPdb as the first mutation effect based on a predictive model.

5.4 Conclusions

As previously stated, the main motivation behind this project was not to discover something novel or revolutionary; rather, the focus was on completing existing knowledge, and providing a robust method which would be easily applicable and maintainable within the SAAPdb pipeline. In that respect, this work shows that paying attention to small inconsistencies is well worthwhile: after removing the most common pitfalls observed in previously published methods, a predictor outperforming the competition was created, even without full model optimisation (addressed below).

The dataset of interfaces gathered in the previous chapter was not without flaws: during the manual inspection of small interfaces in Section 5.3.2.3, it was noticed that PISA and PQS often differ in the orientation of monomers predicted in homodimers, often with the PISA assembly making more sense in terms of biological activity, i.e. creating larger and thus more stable interfaces, not occluding functional sites, etc. As previously mentioned, there is a plan to re-build the predictors on PISA complexes before incorporating this work into the SAAPdb.

Further, the class of non-interface surface residues is based on a negative observation; in other words, there is no guarantee that the residues not involved in contacts we are testing for at the moment are not involved in contacts with other, currently unknown, interacting partners. However, once sufficient knowledge is obtained on *all* protein-protein interactions, the need for this type of interface-predictor will disappear as well. This methodological flaw is widely accepted in all interface-prediction models, and the only current way to address it is to gather large amounts of training instances for the model to process and be able to differentiate between the two classes.

On a similar note, sampling the surface in the search of interface-characteristic features using patches might result in obtaining patches simultaneously including pieces of several interfaces, when a complex consists of more than two chains. The interface datasets mostly consisted of binary interfaces; nevertheless this issue should not be ignored. To that end, patches with varying sizes have been prepared: favouring smaller patch radii was expected to minimise the occurrence of multiple interface patches. Surprisingly, it turned out this issue is not relevant within the experimental setup presented in this chapter: larger patches yielded better performance, with no corrections for multiple interfaces in the same patch.

The obvious way to obtain an improved prediction is to use ever growing number of complexes in the PDB; based on the results presented in the last two chapters, there are additional routes to achieve improved prediction performance: (i) testing patches larger in size, (ii) including solvent-accessibility and protrusion as predictors, (iii) expanding amino acid propensities to profile-based propensities, (iv) testing various thresholds for $rASA_o$ interface-surface overlap. Expansion of this project in these directions was stopped owing to time restrictions.

The field of protein-protein interface prediction is extensive: constantly new methods, interface-distinguishing features and models appear. Nevertheless, it seems the efficacy of these prediction methods has reached a plateau: even the best ones have accuracy of 75-80%, specificity of 80-90%, correlation between the predicted and the observed class from 0.4 to 0.5, and F-measure ~ 0.6. With the limited data available on which to train models, one approach to improve this level of performance is either to narrow down the range of interfaces analysed and predicted (e.g. focus on either transient or obligate interfaces), or to impose functional or structural restrictions on the dataset (e.g. by considering only one family of proteins such as antibodies, or just transmembrane proteins). However, the widely accepted enhancement is to start combining methodologically-different interface predictors in meta-predictors, similar to the approach of Qin and Zhou (2007) in meta-PPISP.

Chapter 6

Conclusions

The SAAPdb project aims to gather and analyse single amino acid polymorphisms, and assess them in terms of effects these mutations are likely to have on protein function, stability, folding and interactions. When the work presented in this thesis started in October 2007, this resource was just published (Hurst *et al.*, 2008), and a new release was being prepared[1], processing mutation data from 12 sources listed in Table 2.1, testing for 16 structural effects and sequence conservation (listed in Table 2.2). In the meantime, SAAPdb has undergone some major changes and a new release is expected shortly. Consequently all results presented here are based on the 2008 release.

This chapter is structured around the two main goals of the SAAPdb project: data collection and analyses (Section 6.1), and the expansion of SAAPdb methodology (Section 6.2), ending with a look into the future in Section 6.3.

[1]made available to public in early 2008 via `http://www.bioinf.org.uk/saap/db/`

6.1 Analyses of mutations

After the publication of the first draft of the human genome in 2000, and the development of cheaper and faster second-generation sequencing platforms in 2005, a lot of focus has turned to human genomic variation, resulting in an explosion of data sources and tools processing these data.

This thesis focuses on exploring data already existing in SAAPdb. Chapter 3 presents an analysis motivated by a specific evolutionary phenomenon: compensation of mutations through epistatic interactions with other mutations, often also termed 'fitness reversal'. This analysis found that compensated mutations occur in more solvent-accessible residues, and on average have milder effects when compared with uncompensated disease-associated mutations (DAMs) (Barešić *et al.*, 2010), which was in accordance with previous findings by Ferrer-Costa *et al.* (2002). Finally, based on the conservation of residues surrounding DAMs, it showed compensation appears through random genetic drift, while uncompensated DAMs tend to occur in more conserved structural environments.

A similarly conceived analysis[2] was performed on a set of mutations in kinase domains, provided by our collaborators (Izarzugaza *et al.*, 2011), but has not been described in this thesis. This highly populated family of proteins is involved in signal transduction, cell-cycle regulation and tumourigenesis. It has been shown that kinases have unusually conserved structural features considering the heterogeneous spectrum of sequences they adopt (Knight *et al.*, 2007): a compromise between the restrictions at the structural level neccessary to perform a specific task (i.e. binding ATP and transferring the phosphate group), and the specificity in binding to a wide range of proteins targeted for phosphorylation. While the SAAPdb analysis showed some family-specific trends, i.e. an increase in pathogenic kinase mutations in interface residues and more neutral than pathogenic kinase mutations creating

[2] a subset of SAAPdb mutations of specific interest was compared with the remaining (background) mutations, resulting in subset-specific preferences for the structural effects.

destabilising voids in the protein structure, the unexpected finding was a significant decrease in the number of structural consequences detected for kinase mutations, when compared with the general frequency of annotation in SAAPdb mutation data (Izarzugaza *et al.*, 2011).

This fact has an interesting implication: while being broad, the spectrum of structural features for which SAAPdb tests could still be expanded, since clearly some kinase-specific restrictions on mutation positions exist, which SAAPdb is not capable of identifying. Finally, the work on CPDs and kinases resulted in a series of scripts automatically extracting a SAAPdb subset and creating a summary of trends for it, which is potentially useful for future collaborations with experimental groups.

On the level of mutation entries, an obvious enhancement would be to add genomic-level information, where available. For example, codon usage information might be useful to experimentalists surveying known mutants in order to create novel mutants, while the information on the allele frequencies might be informative during drug design, targeting a special structural effect. Additionally, rather than just covering SAAPs (i.e. mutations within the exons), SAAPdb would benefit from adding single nucleotide variations in promotor regions, enhancers, mRNA splice sites, regions of mRNA stabilised by secondary structure, etc. Indeed, Martin group has plans to expand SAAPdb and provide mutations in non-coding regions of the genome as well.

The primary limitation in expanding the coverage of publicly-available mutation data is the lack of a standardised data format. While an appropriate format exists in the form of the LOVD system (Fokkema *et al.*, 2011) and is widely recommended as a versatile and secure tool for human variation storage, its use is limited: only 54% (842/1550) of LSMDBs in the Human Genome Variation Society's repository[3] have been built using LOVD. Further, this tool is less than ideal for use during the SAAPdb data gathering step, as it does not provide any option for bulk download of mutation

[3] `http://www.hgvs.org/dblist/glsdb.html`

data or web services access to the data. Consequently, considerable time is spent building dataset parsers, rather than utilising human variation data to identify new trends.

6.2 Methodology utilised to analyse single amino acid polymorphisms

SAAPdb links genomic-level information on single amino acid polymorphisms with the phenotype data (i.e. the level of pathogenicity of a single amino acid polymorphism), by providing information on the intermediate level: it sets out to explain which structural effects are caused by the single nucleotide variation that leads to the observed phenotypic effect, or lack thereof.

Chapters 4 and 5 present a novel structural analysis, set out to increase the coverage of mutations in protein-protein interfaces, stemming from the relatively low fraction of multichain protein complexes in the Protein DataBank. Trained on a range of previously identified (and then confirmed in Chapter 4) sequence-based, structural and evolutionary parameters, the RF*(ALL) random forest model presented in Chapter 5, when assessed on an independent set of interfaces not seen by any of the predictors during training, outperforms all previously developed interface-predictors. This also authenticates the choice of the PQS_{nr} dataset as a representative set of protein-protein interfaces, as well as the features identified in Chapter 4.

It is worth noting here that this new `predicted interface` category is the first predictive structural effect to be added to SAAPdb; all others are based on calculations performed from structural or sequence information, and thresholds imposed on these calculations. Once it is incorporated into the next version of the SAAPdb, it will be interesting to see what will be the gain in coverage of mutations in interface, in comparison with using three previously implemented categories: `interface`, `pqs` and `binding`, all of which rely on the presence of structures of complexes. Provided this addition to the pipeline proves useful, other similar prediction methods can be developed to complement the existing set of structural categories.

6.3 Future prospects

Unfortunately, the way SAAPdb was originally built is becoming increasingly incompatible with the rapid accumulation of mutation data. In effect, each time mutation data are updated, the whole database is rebuilt from the ground up, every time with more data to process. A more logical approach would be to apply structural analyses to one mutation (or all mutations in one protein) at a time, thus enabling incremental updates. Moreover, this might be offered as a web service in which a user provides a novel single amino acid polymorphism and, provided the appropriate protein structure is available, gets the range of structural effects, calculated in real time, as an output. Further, SAAPdb currently outputs a categorical output based on a cut-off for every effect, e.g. a `void` category will output 'affecting' if the largest created void in the structure is greater than 275Å^3, and 'ok' for all values below that threshold. Some of the SAAPdb categories (e.g. `void`, `clash`) would be more intuitively described on a continuous scale.

To conclude, at present[4], even after many other tools providing structural details on single amino acid polymorphisms have been made available (for an overview see Section 1.2.3), SAAPdb offers a plethora of additional information for a mutation besides its mapping to protein sequence and structure: a large selection of precalculated structural effects and visualisation of mutations mapped onto the protein structure. However, as the field and the amount of gathered knowledge on SAAPs grows, the focus is shifting to pathogenicity prediction. Indeed, some preliminary work on the predictive power of SAAPdb categories performed by McMillan (2009) and Ledda (2011) showed promising results: performance of classification with regards to

[4]three years after its original publication

pathogenicity was comparable to, or better than, methods reviewed in Section 1.2.3.1, and this aspect of the project is being actively developed at the moment. Once models of satisfactory performance are built, the Martin group can move on to test them on experimentally-obtained, independently-evaluated sets of mutations by participating in the Critical Assessment of Genome Interpretation (Callaway, 2010).

In the meantime, the work presented in this thesis can be finely tuned and expanded to enhance the understanding of the topic. All results obtained for compensated mutations will have to be recalculated once the new, significantly enriched, dataset is provided within the new release of SAAPdb. Moreover, Figure 3.7 indicated that the FEP sequences containing neutralised mutation that is detrimental to humans can be grouped into prokaryotic, eukaryotic and mammalian FEPs; it will be interesting to expand this study and check whether these three groups have different preferences in terms of compensation mechanisms. The protein-protein interface predicting model can be enhanced in several ways, several of which have been proposed in Section 5.4, covering further parameter optimisation, considering additional interface predictors and alternate input data encoding.

As previously stated, the motivation behind the interface prediction modelling project was to expand the list of effects covered by the SAAPdb. The model RF*(ALL) presented in the Section 5.3.5.2 outperformed the previously published interface residue classifiers on the benchmark dataset, and this model will be added to the next update of the SAAPdb as the 18^{th} category. Unfortunately, at the time of writing of this chapter[5], the SAAPdb was under severe reconstruction, addressing issues listed at the beginning of this section. This initiative will result in more robust database, and easier addition of new mutation datasets, and categories of effects. Once this upgrade is completed, the `predicted interface` category will be incorporated into the structural pipeline of the SAAPdb. Next, propensities of neutral and disease-associated SAAPs for occurence in predicted protein-protein interfaces will be calculated, and compared with the trends of previously implemented interface categories, presented

[5]September 2011

in Figure 1.8. In addition, a test will be performed to assess whether compensated pathogenic deviations occur more/less often in predicted interface sites than uncompensated disease-associated SAAPs.

The field of bioinformatics is still relatively new and growing. Covering the recent advances across this field proves to be challenging for a person in regular contact with it, let alone for the less computer-adept among the wet-lab scientists. Yet, the latter are exactly the target group for whom most of the tools are currently developed. The future of *in silico* analyses is in their easy use and combination, which is facilitated through platforms like ICENI (Cohen *et al.*, 2005) and Taverna (Hull *et al.*, 2006): methods incorporated into one of these frameworks are easily combinable into workflows of *in silico* experiments. In turn, this eliminates data parsing (storing and preparation), a tedious step when standardised data formats such as XML and JSON are not utilised.

In summary, this thesis has investigated CPDs, showing that they have characteristics which, on average, are distinct from uncompensated PDs, giving us insite into mechanisms of evolution. Secondly it has developed a new form of analysis (prediction of interface residues) that can be incorporated into the SAAPdb pipeline and, as part of that, has developed a carefully filtered set of protein interfaces for training prediction methods.

Bibliography

Adzhubei, I. A., Schmidt, S., Peshkin, L., Ramensky, V. E., Gerasimova, A., Bork, P., Kondrashov, A. S. and Sunyaev, S. R. (2010). A method and server for predicting damaging missense mutations, *Nature Methods*, **17**, 248–249.

Alpaydin, E., (2009). *Introduction to Machine Learning.* The MIT Press, 2nd edition.

Altschul, S. F. (1993). A protein alignment scoring system sensitive at all evolutionary distances, *Journal of Molecular Evolution*, **36**, 290–300.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990). Basic local alignment search tool, *Journal of Molecular Biology*, **215**, 403–410.

Altschul, S. F., Wootton, J. C., Gertz, E. M., Agarwala, R., Morgulis, A., Schäffer, A. A. and Yu, Y.-K. (2005). Protein database searches using compositionally adjusted substitution matrices, *FEBS Journal*, **272**, 5101–5109.

Amberger, J., Bocchini, C. A., Scott, A. F. and Hamosh, A. (2009). McKusick's Online Mendelian Inheritance in Man (OMIM), *Nucleic Acids Research*, **37**, D793–D796.

Bader, G. D., Donaldson, I., Wolting, C., Ouellette, B. F., Pawson, T. and Hogue, C. W. (2001). BIND–The biomolecular interaction network database, *Nucleic Acids Research*, **29**, 242–245.

Bahadur, R. P., Chakrabarti, P., Rodier, F. and Janin, J. (2003). Dissecting subunit interfaces in homodimeric proteins, *Proteins*, **53**, 708–719.

Bahadur, R. P., Chakrabarti, P., Rodier, F. and Janin, J. (2004). A dissection of specific and non-specific protein-protein interfaces, *Journal of Molecular Biology*, **336**, 943–955.

Baker, E. N. and Hubbard, R. E. (1984). Hydrogen bonding in globular proteins, *Progress in Biophysics and Molecular Biology*, **44**, 97–179.

Bao, L., Zhou, M. and Cui, Y. (2005). nsSNPAnalyzer: Identifying disease-associated nonsynonymous single nucleotide polymorphisms, *Nucleic Acids Research*, **33**, W480–W482.

Barbui, T., Finazzi, G., Rodeghiero, F. and Dini, E. (1983). Immunoelectrophoretic evidence of a thrombin-induced abnormality in a new variant of hereditary dysfunctional antithrombin III (AT III 'Vicenza'), *British Journal of Haematology*, **54**, 561–565.

Barešić, A., Hopcroft, L. E. M., Rogers, H. H., Hurst, J. M. and Martin, A. C. R. (2010). Compensated pathogenic deviations: Analysis of structural effects, *Journal of Molecular Biology*, **396**, 19–30.

Barešić, A. and Martin, A. C. R. (2011). Compensated pathogenic deviations, *Biomolecular Concepts*, **2**, 281–292.

Berg, J. M., Tymoczko, J. L. and Stryer, L., (2006). *Biochemistry*. W. H. Freeman, 6th edition.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. and Bourne, P. E. (2000). The Protein Data Bank, *Nucleic Acids Research*, **28**, 235–242.

Bluhm, W. F., Beran, B., Bi, C., Dimitropoulos, D., Prlic, A., Quinn, G. B., Rose, P. W., Shah, C., Young, J., Yukich, B., Berman, H. M. and Bourne, P. E. (2011). Quality assurance for the query and distribution systems of the RCSB Protein Data Bank, *Database: The Journal of Biological Databases and Curation*, **2011**, bar003.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S. and Schneider, M. (2003). The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003, *Nucleic Acids Research*, **31**, 365–370.

Bogan, A. A. and Thorn, K. S. (1998). Anatomy of hot spots in protein interfaces, *Journal of Molecular Biology*, **280**, 1–9.

Bork, P., Jensen, L. J., von Mering, C., Ramani, A. K., Lee, I. and Marcotte, E. M. (2004). Protein interaction networks from yeast to human, *Current Opinion in Structural Biology*, **14**, 292–299.

Bradford, J. R., Needham, C. J., Bulpitt, A. J. and Westhead, D. R. (2006). Insights into protein-protein interfaces using a Bayesian network prediction method, *Journal of Molecular Biology*, **362**, 365–386.

Bradford, J. R. and Westhead, D. R. (2005). Improved prediction of protein-protein binding sites using a support vector machines approach, *Bioinformatics*, **21**, 1487–1494.

Bragg, W. L. (1913). The structure of some crystals as indicated by their diffraction of X-rays, *Proceedings of the Royal Society of London*, **A89**, 248.

Breiman, L. (1996). Bagging predictors, *Machine learning*, **24**, 123–140.

Breiman, L. (2001). Random forests, *Machine learning*, **5**, 5–32.

Bromberg, Y. and Rost, B. (2007). SNAP: Predict effect of non-synonymous polymorphisms on function, *Nucleic Acids Research*, **35**, 3823–3835.

Brünger, A. T. (1992). Free R value: A novel statistical quantity for assessing the accuracy of crystal structures, *Nature*, **355**, 472–475.

Bullock, A. N. and Fersht, A. R. (2001). Rescuing the function of mutant p53, *Nature Reviews Cancer*, **1**, 68–76.

Caffrey, D. R., Somaroo, S., Hughes, J. D., Mintseris, J. and Huang, E. S. (2004). Are protein-protein interfaces more conserved in sequence than the rest of the protein surface?, *Protein Science*, **13**, 190–202.

Callaway, E. (2010). Mutation-prediction software rewarded, *Nature News*, page doi:10.1038/news.2010.679.

Capriotti, E., Calabrese, R. and 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**, 2729–2734.

Chakrabarti, P. and Janin, J. (2002). Dissecting protein-protein recognition sites, *Proteins*, **47**, 334–343.

Chen, H. and Zhou, H.-X. (2005). Prediction of interface residues in protein-protein complexes by a consensus neural network method: test against NMR data, *Proteins*, **61**, 21–35.

Cohen, J., McGough, A. S., Darlington, J., Furmento, N., Kong, G. and Mayer, A. (2005). RealityGrid: An integrated approach to middleware through ICENI, *Philosophical Transactions. Series A, Mathematical Physical and Engineering Sciences*, **363**, 1817–1827.

Connoly, M. L. (1983). Analytical molecular surface calculation, *Journal of Applied Crystallography*, **16**, 548–558.

Cordell, H. J. (2002). Epistasis: What it means, what it doesn't mean, and statistical methods to detect it in humans, *Human Molecular Genetics*, **11**, 2463–2468.

Cornette, J. L., Cease, K. B., Margalit, H., Spouge, J. L., Berzofsky, J. A. and DeLisi, C. (1987). Hydrophobicity scales and computational techniques for detecting amphipathic structures in proteins, *Journal of Molecular Biology*, **195**, 659–685.

Cowperthwaite, M. C., Bull, J. J. and Meyers, L. A. (2006). From bad to good: Fitness reversals and the ascent of deleterious mutations, *PLoS Computational Biology*, **2**, e141.

Cuff, A. L., (2004). *P53: Linking Sequence, Structure and Disease*. PhD thesis, University of Reading.

Cuff, A. L., Janes, R. W. and Martin, A. C. R. (2006). Analysing the ability to retain sidechain hydrogen-bonds in mutant proteins, *Bioinformatics*, **22**, 1464–1470.

Cuff, A. L. and Martin, A. C. R. (2004). Analysis of void volumes in proteins and application to stability of the p53 tumour suppressor protein, *Journal of Molecular Biology*, **344**, 1199–1209.

Cuff, A. L., Sillitoe, I., Lewis, T., Clegg, A. B., Rentzsch, R., Furnham, N., Pellegrini-Calace, M., Jones, D., Thornton, J. and Orengo, C. A. (2011). Extending CATH: Increasing coverage of the protein structure universe and linking structure with function, *Nucleic Acids Research*, **39**, D420–D426.

David, F. and Yip, Y. (2008). SSMap: A new UniProt-PDB mapping resource for the curation of structural-related information in the UniProt/Swiss-Prot Knowledgebase, *BMC Bioinformatics*, **9**, 391.

Dawson, K., Thorpe, R. S. and Malhotra, A. (2010). Estimating genetic variability in non-model taxa: A general procedure for discriminating sequence errors from actual variation, *PLoS One*, **5**, e15204.

Dayhoff, M. O., Schwartz, R. M. and Orcutt, B. C., (1978). *Atlas of Protein Sequence and Structure*, pages 353–358. National Biomedical Research Foundation.

de Vries, S. J. and Bonvin, A. M. J. J. (2008). How proteins get in touch: Interface prediction in the study of biomolecular complexes, *Current Protein & Peptide Science*, **9**, 394–406.

DePristo, M. A., Weinreich, D. M. and Hartl, D. L. (2005). Missense meanderings in sequence space: a biophysical view of protein evolution, *Nature Reviews Genetics*, **6**, 678–687.

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

Dong, Q., Wang, X., Lin, L. and Guan, Y. (2007). Exploiting residue-level and profile-level interface propensities for usage in binding sites prediction of proteins, *BMC Bioinformatics*, **8**, 147–147.

Edgar, R. C. (2004a). Local homology recognition and distance measures in linear time using compressed amino acid alphabets, *Nucleic Acids Research*, **32**, 380–385.

Edgar, R. C. (2004b). MUSCLE: A multiple sequence alignment method with reduced time and space complexity, *BMC Bioinformatics*, **5**, 113.

Edgar, R. C. (2004c). MUSCLE: Multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research*, **32**, 1792–1797.

Edgar, R. C. and Batzoglou, S. (2006). Multiple sequence alignment, *Current Opinion in Structural Biology*, **16**, 368–373.

Fariselli, P., Pazos, F., Valencia, A. and Casadio, R. (2002). Prediction of protein-protein interaction sites in heterocomplexes with neural networks, *European Journal of Biochemistry*, **269**, 1356–1361.

Ferrer-Costa, C., Gelpí, J. L., Zamakola, L., Parraga, I., de la Cruz, X. and Orozco, M. (2005). PMUT: A web-based tool for the annotation of pathological mutations on proteins, *Bioinformatics*, **21**, 3176–3178.

Ferrer-Costa, C., Orozco, M. and de la Cruz, X. (2007). Characterization of compensated mutations in terms of structural and physico-chemical properties, *Journal of Molecular Biology*, **365**, 249–256.

Ferrer-Costa, C., Orozco, M. and de la Cruz, X. (2002). Characterization of disease-associated single amino acid polymorphisms in terms of sequence and structure properties, *Journal of Molecular Biology*, **315**, 771–786.

Finn, R. D., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S. R., Sonnhammer, E. L. L. and Bateman, A. (2006). Pfam: Clans, web tools and services, *Nucleic Acids Research*, **34**, D247–D251.

Fisher, R. A. (1935). The logic of inductive inference, *Journal of the Royal Statistical Society series A*, **98**, 39–54.

Fokkema, I. F. A. C., Taschner, P. E. M., Schaafsma, G. C. P., Celli, J., Laros, J. F. J. and den Dunnen, J. T. (2011). LOVD V.2.0: The next generation in gene variant databases, *Human Mutation*, **32**, 557–563.

Friedler, A., Hansson, L. O., Veprintsev, D. B., Freund, S. M. V., Rippin, T. M., Nikolova, P. V., Proctor, M. R., Rüdiger, S. and Fersht, A. R. (2002). A peptide that binds and stabilizes p53 core domain: Chaperone strategy for rescue of oncogenic mutants, *Proceedings of the National Academy of Sciences of the USA*, **99**, 937–942.

Friedler, A., Veprintsev, D. B., Hansson, L. O. and Fersht, A. R. (2003). Kinetic instability of p53 core domain mutants: Implications for rescue by small molecules, *Journal of Biological Chemistry*, **278**, 24108–24112.

Futschik, M. E., Chaurasia, G. and Herzel, H. (2007). Comparison of human protein-protein interaction maps, *Bioinformatics*, **23**, 605–611.

Geladi, P. and Kowalski, B. (1986). Partial least squares regression: A tutorial, *Analytica Chimica Acta*, **185**, 1–17.

Goldstein, B. A., Hubbard, A. E., Cutler, A. and Barcellos, L. F. (2010). An application of Random Forests to a genome-wide association dataset: Methodological considerations & new findings, *BMC Genetics*, **11**, 49.

Guharoy, M. and Chakrabarti, P. (2005). Conservation and relative importance of residues across protein-protein interfaces, *Proceedings of the National Academy of Sciences of the USA*, **102**, 15447–15452.

Hall, M., Franke, E., Holmes, G., Pfahringer, B., Reutemann, P. and H, W. I. (2009). The WEKA data mining software: An update, *SIGKDD Explorations*, **11**, 10–18.

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve, *Machine Learning*, **77**, 103–123.

Harper, P. L., Luddington, R. J., Daly, M., Bruce, D., Williamson, D., Edgar, P. F., Perry, D. J. and Carrell, R. W. (1991). The incidence of dysfunctional antithrombin variants: Four cases in 210 patients with thromboembolic disease, *British Journal of Haematology*, **77**, 360–364.

Hazes, B. and Dijkstra, B. W. (1988). Model building of disulfide bonds in proteins with known three-dimensional structure, *Protein Engineering, Design and Selection*, **2**, 119–125.

Headd, J. J., Ban, Y. E. A., Brown, P., Edelsbrunner, H., Vaidya, M. and Rudolph, J. (2007). Protein-protein interfaces: Properties, preferences, and projections, *Journal of Proteome Research*, **6**, 2576–2586.

Hendrickson, W. A. and Ogata, C. M., (1997). Phase Determination from Multiwavelength Anomalous Diffraction Measurements. In Charles W. Carter, J. (ed.), *Macromolecular Crystallography Part A*, volume 276 of *Methods in Enzymology*, pages 494–523. Academic Press.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks, *Proceedings of the National Academy of Sciences of the USA*, **89**, 10915–10919.

Henrick, K., Feng, Z., Bluhm, W. F., Dimitropoulos, D., Doreleijers, J. F., Dutta, S., Flippen-Anderson, J. L., Ionides, J., Kamada, C., Krissinel, E., Lawson, C. L., Markley, J. L., Nakamura, H., Newman, R., Shimizu, Y., Swaminathan, J., Velankar, S., Ory, J., Ulrich, E. L., Vranken, W., Westbrook, J., Yamashita, R., Yang, H., Young, J., Yousufuddin, M. and Berman, H. M. (2008). Remediation of the Protein data bank archive, *Nucleic Acids Research*, **36**, D426–D433.

Henrick, K. and Thornton, J. M. (1998). PQS: a protein quaternary structure file server, *Trends in Biochemical Sciences*, **23**, 358–361.

Hirosawa, M., Totoki, Y., Hoshida, M. and Ishikawa, M. (February 1995). Comprehensive study on iterative algorithms of multiple sequence alignment, *Computer applications in the biosciences: CABIOS*, **11**, 13–18.

Holton, J. M. and Frankel, K. A. (2010). The minimum crystal size needed for a complete diffraction data set, *Acta Crystallographica Section D, Biological Crystallography*, **66**, 393–408.

Hu, Z., Ma, B., Wolfson, H. and Nussinov, R. (2000). Conservation of polar residues as hot spots at protein interfaces, *Proteins*, **39**, 331–342.

Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M. R., Li, P. and Oinn, T. (2006). Taverna: A tool for building and running workflows of services, *Nucleic Acids Research*, **34**, W729–W732.

Hurst, J. M., McMillan, L. E. M., Porter, C. T., Allen, J., Fakorede, A. and Martin, A. C. (2008). The SAAPdb web resource: A large scale structural analysis of mutant proteins, *Human Mutation*, **30**, 616–624.

Hwang, H., Vreven, T., Janin, J. and Weng, Z. (2010). Proteinprotein docking benchmark version 4.0, *Proteins: Structure, Function, and Bioinformatics*, **78**, 3111–3114.

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. and Valencia, A. (2011). Characterization of pathogenic germline mutations in human protein kinases, *BMC Bioinformatics*, **12**, S1.

Janin, J., Bahadur, R. P. and Chakrabarti, P. (2008). Protein-protein interaction and quaternary structure, *Quaterly Reviews of Biophysics*, **41**, 133–180.

Jerez, J. M., Molina, I., Garca-Laencina, P. J., Alba, E., Ribelles, N., Martín, M. and 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.

Jones, S. and Thornton, J. M. (1995). Protein-protein interactions: A review of protein dimer structures, *Progress in Biophysics and Molecular Biology*, **63**, 31–59.

Jones, S. and Thornton, J. M. (1996). Principles of protein-protein interactions, *Proceedings of The National Academy of Science of the USA*, **93**, 13–20.

Jones, S. and Thornton, J. M. (1997). Analysis of protein-protein interaction sites using surface patches, *Journal of Molecular Biology*, **272**, 121–132.

Jordan, I. K., Kondrashov, F. A., Adzhubei, I. A., Wolf, Y. I., Koonin, E. V., Kondrashov, A. S. and Sunyaev, S. (2005). A universal trend of amino acid gain and loss in protein evolution, *Nature*, **433**, 633–638.

Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features, *Biopolymers*, **22**, 2577–2637.

Kanehisa, M. (1997). A database for post-genome analysis, *Trends in Genetics*, **13**, 375–376.

Karchin, R., Diekhans, M., Kelly, L., Thomas, D. J., Pieper, U., Eswar, N., Haussler, D. and Sali, A. (2005). LS-SNP: Large-scale annotation of coding non-synonymous SNPs based on multiple information sources, *Bioinformatics*, **21**, 2814–2820.

Karlin, S. and Brocchieri, L. (1996). Evolutionary conservation of RecA genes in relation to protein structure and function, *Journal of Bacteriology*, **178**, 1881–1894.

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

Keskin, O., Gursoy, A., Ma, B. and Nussinov, R. (2008). Principles of protein-protein interactions: what are the preferred ways for proteins to interact?, *Chemical Reviews*, **108**, 1225–1244.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences, *Journal of Molecular Evolution*, **16**, 111–120.

Kimura, M. (1985). The role of compensatory neutral mutations in molecular evolution, *Journal of Genetics*, **64**, 7–19.

Kleywegt, G. J. and Jones, T. A. (1997). Model building and refinement practice, *Methods in Enzymology*, **277**, 208–230.

Knight, J. D. R., Qian, B., Baker, D. and Kothary, R. (2007). Conservation, variability and the modeling of active protein kinases, *PLoS One*, **2**, e982.

Kondrashov, A. S., Sunyaev, S. and Kondrashov, F. A. (2002). Dobzhansky-Muller incompatibilities in protein evolution., *Proceedings of the National Academy of Sciences of the USA*, **99**, 14878–14883.

Krissinel, E. and Henrick, K. (2007). Inference of macromolecular assemblies from crystalline state, *Journal of Molecular Biology*, **372**, 774–797.

Kroese, D. P., Taimre, T. and Botev, Z., (2011). *Handbook of Monte Carlo Methods*, page 772. John Wiley and Sons.

Kufareva, I., Budagyan, L., Raush, E., Totrov, M. and Abagyan, R. (2007). PIER: Protein interface recognition for structural proteomics, *Proteins*, **67**, 400–417.

Kulathinal, R. J., Bettencourt, B. R. and Hartl, D. L. (2004). Compensated deleterious mutations in insect genomes, *Science*, **306**, 1553–1554.

Kumar, P., Henikoff, S. and Ng, P. C. (2009). Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm, *Nature Protocols*, **4**, 1073–1081.

Kwok, C. J., Martin, A. C. R., Au, S. W. N. and Lam, V. M. S. (2002). G6PDdb, an integrated database of glucose-6-phosphate dehydrogenase (G6PD) mutations, *Human Mutation*, **19**, 217–224.

Kyte, J. and Doolittle, R. F. (1982). A simple method for displaying the hydropathic character of a protein, *Journal of Molecular Biology*, **157**, 105–132.

Laskowski, R. A. (1995). SURFNET: a program for visualizing molecular surfaces, cavities, and intermolecular interactions, *Journal of Molecular Graphics and Modelling*, **13**, 323–330.

Ledda, F., (2011). Predicting pathogenicity from SAAPdb categories. Unpublished.

Lee, B. and Richards, F. M. (1971). The interpretation of protein structures: estimation of static accessibility, *Journal of Molecular Biology*, **55**, 379–400.

Levy, E. D. (2007). PiQSi: Protein quaternary structure investigation, *Structure*, **15**, 1364–1367.

Li, X., Keskin, O., Ma, B., Nussinov, R. and Liang, J. (2004). Protein-protein interactions: Hot spots and structurally conserved residues often locate in complemented pockets that pre-organized in the unbound states: Implications for docking, *Journal of Molecular Biology*, **344**, 781–795.

Liang, S. and Grishin, N. V. (2004). Effective scoring function for protein sequence design, *Proteins*, **54**, 271–281.

Liang, S., Zhang, C., Liu, S. and Zhou, Y. (2006). Protein binding site prediction using an empirical scoring function, *Nucleic Acids Research*, **34**, 3698–3707.

Lo Conte, L., Chothia, C. and Janin, J. (1999). The atomic structure of protein-protein recognition sites, *Journal of Molecular Biology*, **285**, 2177–2198.

Markowski, C. A. and Markowski, E. P. (1990). Conditions for the effectiveness of a preliminary test of variance, *The American Statistician*, **44**, 322–326.

Martin, A. C. R., (1999). Solv computer program. Unpublished.

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

Martin, A. C. R., Facchiano, A. M., Cuff, A. L., Hernandez-Boussard, T., Olivier, M., Hainaut, P. and Thornton, J. M. (2002). Integrating mutation data and structural analysis of the TP53 tumor-suppressor protein, *Human Mutation*, **19**, 149–164.

McKusick, V. A., (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, L. E. M., (2009). *Post-genomic Structural Analysis of Single Amino Acid Polymorphisms*. PhD thesis, University College London.

McMillan, L. E. M. and Martin, A. C. R. (2008). Automatically extracting functionally equivalent proteins from SwissProt, *BMC Bioinformatics*, **9**, 418.

Metzker, M. L. (2010). Sequencing technologies - the next generation, *Nat Reviews Genetics*, **11**, 31–46.

Meyer, D., Leisch, F. and Hornik, K. (2003). The support vector machine under test, *Neurocomputing*, **55**, 169–186.

Mierswa, I., Lemmen, F., Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T. (2006). YALE: Rapid prototyping for complex data mining tasks, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)*.

Miller, S., Janin, J., Lesk, A. M. and Chothia, C. (1987). Interior and surface of monomeric proteins, *Journal of Molecular Biology*, **196**.3, 641–656.

Missiuro, P. V., Liu, K., Zou, L., Ross, B. C., Zhao, G., Liu, J. S. and Ge, H. (2009). Information flow analysis of interactome networks, *PLoS Computational Biology*, **5**, e1000350.

Mitchell, T., (1997). *Machine Learning.* McGraw Hill, 1st edition.

Mood, A., Graybill, F. A. and Boes, D. C., (1974). *Introduction to the Theory of Statistics*, pages 241–246. McGraw-Hill, 3rd edition.

Moore, B., Hu, H., Singleton, M., Reese, M. G., De La Vega, F. M. and Yandell, M. (2011). Global analysis of disease-related DNA sequence variation in 10 healthy individuals: Implications for whole genome-based clinical diagnostics, *Genetics in Medicine*, **13**, 210–217.

Moreira, I. S., Fernandes, P. A. and Ramos, M. J. (2007). Hot Spots–a review of the protein-protein interface determinant amino-acid residues, *Proteins*, **68**, 803–812.

Morris, A. L., MacArthur, M. W., Hutchinson, E. G. and Thornton, J. M. (1992). Stereochemical quality of protein structure coordinates, *Proteins*, **12**, 345–364.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins, *Journal of Molecular Biology*, **48**, 443–453.

Negi, S. S. and Braun, W. (2007). Statistical analysis of physical-chemical properties and prediction of protein-protein interfaces, *Journal of Molecular Modeling*, **13**, 1157–1167.

Neuvirth, H., Heinemann, U., Birnbaum, D., Tishby, N. and Schreiber, G. (2007). ProMateus–an open research approach to protein-binding sites analysis, *Nucleic Acids Research*, **35**, W543–W548.

Neuvirth, H., Raz, R. and Schreiber, G. (2004). ProMate: A structure based prediction program to identify the location of protein-protein binding sites, *Journal of Molecular Biology*, **338**, 181–199.

Ng, P. C. and Henikoff, S. (2003). SIFT: Predicting amino acid changes that affect protein function, *Nucleic Acids Research*, **31**, 3812–3814.

Nooren, I. M. A. and Thornton, J. M. (2003). Diversity of protein-protein interactions, *The EMBO Journal*, **22**, 3486–3492.

Notredame, C., Higgins, D. G. and Heringa, J. (2000). T-coffee: A novel method for fast and accurate multiple sequence alignment, *Journal of Molecular Biology*, **302**, 205–217.

Nugent, T. and Jones, D. T. (2009). Transmembrane protein topology prediction using support vector machines, *BMC Bioinformatics*, **10**, 159–159.

Ofran, Y. and Rost, B. (2003). Analysing six types of protein-protein interfaces, *Journal of Molecular Biology*, **325**, 377–387.

Ofran, Y. and Rost, B. (2007). Protein-protein interaction hotspots carved into sequences, *PLoS Computational Biology*, **3**, e119.

Ofran, Y. and Rost, B. (2003). Predicted protein-protein interaction sites from local sequence information, *FEBS Letters*, **544**, 236–239.

Pazos, F. and Valencia, A. (2008). Protein co-evolution, co-adaptation and interactions, *EMBO Journal*, **27**, 2648–2655.

Pelak, K., Shianna, K. V., Ge, D., Maia, J. M., Zhu, M., Smith, J. P., Cirulli, E. T., Fellay, J., Dickson, S. P., Gumbs, C. E., Heinzen, E. L., Need, A. C., Ruzzo, E. K., Singh, A., Campbell, C. R., Hong, L. K., Lornsen, K. A., McKenzie, A. M., Sobreira, N. L. M., Hoover-Fong, J. E., Milner, J. D., Ottman, R., Haynes, B. F., Goedert, J. J. and Goldstein, D. B. (2010). The characterization of twenty sequenced human genomes, *PLoS Genetics*, **6**, e1001111.

Perkins, J. R., Diboun, I., Dessailly, B. H., Lees, J. G. and Orengo, C. (2010). Transient protein-protein interactions: Structural, functional, and network properties, *Structure*, **18**, 1233–1243.

Pettit, F. K., Bare, E., Tsai, A. and Bowie, J. U. (2007). HotPatch: A statistical approach to finding biologically relevant features on protein surfaces, *Journal of Molecular Biology*, **369**, 863–879.

Poon, A., Davis, B. H. and Chao, L. (2005). The coupon collector and the suppressor mutation: Estimating the number of compensatory mutations by maximum likelihood, *Genetics*, **170**, 1323–1332.

Porollo, A. and Meller, J. (2007). Prediction-based fingerprints of protein-protein interactions, *Proteins*, **66**, 630–645.

Povolotskaya, I. S. and Kondrashov, F. A. (2010). Sequence space and the ongoing expansion of the protein universe, *Nature*, **465**, 922–926.

Qin, S. and Zhou, H.-X. (2007). meta-PPISP: A meta web server for protein-protein interaction site prediction, *Bioinformatics*, **23**, 3386–3387.

Rabi, I. I., Zacharias, J. R., Millman, S. and Kusch, P. (1938). A new method of measuring nuclear magnetic moment, *Physical Review*, **53**, 318.

Ramensky, V., Bork, P. and Sunyaev, S. (2002). Human non-synonymous SNPs: Server and survey, *Nucleic Acids Research*, **30**, 3894–3900.

Res, I., Mihalek, I. and Lichtarge, O. (2005). An evolution based classifier for prediction of protein interfaces without using protein structures, *Bioinformatics*, **21**, 2496–2501.

Reumers, J., Maurer-Stroh, S., Schymkowitz, J. and Rousseau, F. (2006). SNPeffect V2.0: A new step in investigating the molecular phenotypic effects of human non-synonymous SNPs, *Bioinformatics*, **22**, 2183–2185.

Rodien, P., Brémont, C., Sanson, M. L., Parma, J., Van Sande, J., Costagliola, S., Luton, J. P., Vassart, G. and Duprez, L. (1998). Familial gestational hyperthyroidism caused by a mutant thyrotropin receptor hypersensitive to human chorionic gonadotropin, *The New England Journal of Medicine*, **339**, 1823–1826.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *Nature*, **323**, 533–536.

Saar-Tsechansky, M. and Provost, F. (2007). Handling missing values when applying classification models, *Journal of Machine Learning Research*, **8**, 1625–1657.

Saitou, N. and Nei, M. (1987). The Neighbor-joining method: A new method for reconstructing phylogenetic trees, *Molecular Biology and Evolution*, **4**, 406–425.

Sayle, R. A. and Milner-White, E. J. (1995). RASMOL: Biomolecular graphics for all, *Trends in Biochemical Sciences*, **20**, 374.

Schneider, M., Fu, X. and Keating, A. E. (2009). X-ray vs. NMR structures as templates for computational protein design, *Proteins*, **77**, 97–110.

Schwarz, D. F., König, I. R. and Ziegler, A. (2010). On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data, *Bioinformatics*, **26**, 1752–1758.

Sherry, S. T., Ward, M. H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M. and Sirotkin, K. (2001). dbSNP: the NCBI database of genetic variation, *Nucleic Acids Research*, **29**, 308–311.

Shi, T., Seligson, D., Belldegrun, A. S., Palotie, A. and Horvath, S. (2005). Tumor classification by tissue microarray profiling: Random forest clustering applied to renal cell carcinoma, *Modern Pathology*, **18**, 547–557.

Shih, H. H., Brady, J. and Karplus, M. (1985). Structure of proteins with single-site mutations: a minimum perturbation approach, *Proceedings of the National Academy of Sciences of the USA*, **82**, 1697–1700.

Šikić, M., Tomić, S. and Vlahoviček, K. (2009). Prediction of protein-protein interaction sites in sequences and 3D structures by Random Forests, *PLoS Computational Biology*, **5**, e1000278.

Smith, D. K. and Thornton, J. M., (1989). SSTRUC computer program. Unpublished.

Sneath, P. H. and Sokal, R. R., (1973). *Numerical Taxonomy.* Freeman.

Stitziel, N. O., Binkowski, T. A., Tseng, Y. Y., Kasif, S. and Liang, J. (2004). topoSNP: A topographic database of non-synonymous single nucleotide polymorphisms with and without known disease association, *Nucleic Acids Research*, **32**, D520–D522.

Stone, E. A. and Sidow, A. (2005). Physicochemical constraint violation by missense substitutions mediates impairment of protein function and disease severity, *Genome Research*, **15**, 978–986.

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P. and Feuston, B. P. (2003). Random Forest: A classification and regression tool for compound classification and QSAR modeling, *Journal of chemical information and computer sciences*, **43**, 1947–1958.

The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing, *Nature*, **467**, 1061–1073.

The International Hapmap 3 Consortium (2010). Integrating common and rare genetic variation in diverse human populations, *Nature*, **467**, 52–58.

The International Hapmap Consortium (2003). The International HapMap Project, *Nature*, **426**, 789–796.

The International Hapmap Consortium (2005). A haplotype map of the human genome, *Nature*, **437**, 1299–1320.

The UniProt Consortium (2009). The Universal Protein Resource (UniProt) 2009, *Nucleic Acids Research*, **37 (Database issue)**, D169–D174.

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

Thusberg, J., Olatubosun, A. and Vihinen, M. (2010). Performance of SNP pathogenicity prediction methods, *Poster presented at The International Conference on Computational Systems Bioinformatics (CSB2010).*

Tong, W., Hong, H., Fang, H., Xie, Q. and Perkins, R. (2003). Decision Forest: Combining the predictions of multiple independent decision tree models, *Journal of Chemical Information and Computer Sciences*, **43**, 525–531.

Tuchman, M., Jaleel, N., Morizono, H., Sheehy, L. and Lynch, M. G. (2002). Mutations and polymorphisms in the human ornithine transcarbamylase gene, *Human Mutation*, **19**, 93–107.

Uzun, A., Leslin, C. M., Abyzov, A. and Ilyin, V. (2007). Structure SNP (StSNP): A web server for mapping and modeling nsSNPs on protein structures with linkage to metabolic pathways, *Nucleic Acids Research*, **35**, W384–W392.

Valdar, W. S. and Thornton, J. M. (2001). Conservation helps to identify biologically relevant crystal contacts, *Journal of Molecular Biology*, **313**, 399–416.

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

Velankar, S., McNeil, P., Mittard-Runte, V., Suarez, A., Barrell, D., Apweiler, R. and Henrick, K. (2005). E-MSD: an integrated data resource for bioinformatics, *Nucleic Acids Research*, **33**, D262–D265.

Via, A., Zanzoni, A. and Helmer-Citterich, M. (2005). Seq2Struct: a resource for establishing sequence-structure links, *Bioinformatics*, **21**, 551–553.

Wang, G. and Dunbrack, R. L. (2003). PISCES: A protein sequence culling server, *Bioinformatics*, **19**, 1589–1591.

Wang, G. and Dunbrack, R. L. (2005). PISCES: Recent improvements to a PDB sequence culling server, *Nucleic Acids Research*, **33**, W94–W98.

Welch, B. L. (1947). The generalization of “Student’s” problem when several different population variances are involved, *Biometrika*, **34**, 28–35.

Westbrook, J., Ito, N., Nakamura, H., Henrick, K. and Berman, H. M. (2005). PDBML: The representation of archival macromolecular structure data in XML, *Bioinformatics*, **21**, 988–992.

White, D., Abraham, G., Carter, C., Kakkar, V. V. and Cooper, D. N. (1992). A novel missense mutation in the antithrombin III gene (Ala387–>Val) causing recurrent venous thrombosis, *Human Genetics*, **90**, 472–473.

Wilkinson, G. N. and Rogers, C. E. (1973). Symbolic descriptions of factorial models for analysis of variance, *Applied Statistics*, **22**, 392–399.

Witten, I. H. and Frank, E., (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition.

Wright, S., (1932). *Proceedings of the Sixth International Congress of Genetics*, volume 1, pages 356–366. Brooklyn Botanic Garden, Menasha, Wisconsin. The roles of mutation, inbreeding, crossbreeding and selection in evolution.

Xu, Q., Canutescu, A., Obradovic, Z. and Dunbrack, R. L. (2006). ProtBuD: a database of biological unit structures of protein families and superfamilies, *Bioinformatics*, **22**, 2876–2882.

Yan, C., Wu, F., Jernigan, R. L., Dobbs, D. and Honavar, V. (2008). Characterization of protein-protein interfaces, *Journal of Protein Chemistry*, **27**, 59–70.

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.

Yip, Y. L., Scheib, H., Diemand, A. V., Gattiker, A., Famiglietti, L. M., Gasteiger, E. and Bairoch, A. (2004). The Swiss-Prot variant page and the ModSNP database: A resource for sequence and structure information on human protein variants, *Human Mutation*, **23**, 464–470.

Yue, P., Melamud, E. and Moult, J. (2006). SNPs3D: Candidate gene and SNP selection for association studies, *BMC Bioinformatics*, **7**, 166.

Zhang, G., Pei, Z., Krawczak, M., Ball, E. V., Mort, M., Kehrer-Sawatzki, H. and Cooper, D. N. (2010). Triangulation of the human, chimpanzee, and Neanderthal genome sequences identifies potentially compensated mutations, *Human Mutation*, **31**, 1286–1293.

Zhou, H.-X. and Qin, S. (2007). Interaction-site prediction for protein complexes: A critical assessment, *Bioinformatics*, **23**, 2203–2209.

Zhou, H. X. and Shan, Y. (2001). Prediction of protein interaction sites from sequence profile and residue neighbor list, *Proteins*, **44**, 336–343.

Zlotnick, A. (2005). Theoretical aspects of virus capsid assembly, *Journal of Molecular Recognition*, **18**, 479–490.