University of
Strathclyde
Science

A robust machine learning approach for the prediction of allosteric binding sites

A thesis submitted to the University of Strathclyde for the degree of Doctor of Philosophy

by

Antony Dimitri Vassileiou

2016

Strathclyde Institute of Pharmacy and Biomedical Sciences

University of Strathclyde

Glasgow

Declaration of Authenticity and Author's Rights

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed: T. V

Date: 06 / 03 / 2017

Abstract

Allosteric regulatory sites are highly prized targets in drug discovery. They remain difficult to detect by conventional methods, with the vast majority of known examples being found serendipitously. Herein, a rigorous, wholly-computational protocol is presented for the prediction of allosteric sites.

Previous attempts to predict the location of allosteric sites by computational means drew on only a small amount of data. Moreover, no attempt was made to modify the initial crystal structure beyond the *in silico* deletion of the allosteric ligand. This behaviour can leave behind a conformation with a significant structural deformation, often betraying the location of the allosteric binding site. Despite this artificial advantage, modest success rates are observed at best. This work addresses both of these issues.

A set of 60 protein crystal structures with known allosteric modulators was collected. To remove the imprint on protein structure caused by the presence of bound modulators, molecular dynamics was performed on each protein prior to analysis. A wide variety of analytical techniques were then employed to extract meaningful data from the trajectories. Upon fusing them into a single, coherent dataset, random forest – a machine learning algorithm – was applied to train a high-performance classification model.

After successive rounds of optimisation, the final model presented in this work correctly identified the allosteric site for 72% of the proteins tested. This is not only an improvement over alternative strategies in the literature; crucially, this method is unique among site prediction tools in that is does not abuse crystal structures containing imprints of bound ligands – of key importance when making live predictions, where no allosteric regulatory sites are known.

Acknowledgements

I would like to firstly express my gratitude to my primary supervisor, Dr Blair Johnston, who was positive and encouraging from the first day of the project to the last. The level of freedom afforded to me was uncommon among Ph.D. projects, and I am hugely grateful to have been trusted to work so independently. I thank my secondary supervisor, Dr Mark Dufton, for sharing some wonderful discussions; his signature enthusiasm is infectious, and it brightened many a day of mine.

I thank Dr Nizar Al-Shar'i, Dr Nahoum Anthony and Dr Murray Robertson for all their help with training and computer programming. Without licence to incessantly pester them for assistance it would not have been possible to carry out this work in the allotted time.

Many wonderful colleagues have been in and out of the office over the years, and their inexhaustible supply of kindness, wit and support (not to mention cake) kept me going whenever my own will faltered. I thank them all and wish them the very best, wherever they have gone or will go from here.

Members of my extended family, not a scientist among them, deserve recognition for repeatedly asking me to describe the project to them in simple terms (and patiently listening to my babbling failure to do so). Unbeknownst to them, they were helping me to distil my work down to its most important points and learn to communicate them to “normal people.” I thank every one of them.

A final thanks to my partner for her profoundly unwavering support. I simply could not have got through this project without having her to return home to.

Abbreviations

AChBP	- mollusk acetylcholine binding protein
AChR	- human nicotinic acetylcholine receptor
AMP	- adenosine monophosphate
AMPK	- AMP-activated protein kinase
ASD	- Allosteric Database
ATP	- adenosine triphosphate
AUC	- area under the curve
BD	- sum of bin population by distance
CPU	- central processing unit
DPM	- diphosphomevalonate
ENM	- elastic network model
FAK	- Focal adhesion kinase
FNR	- false negative rate
FPR	- false positive rate
GB	- generalised Born
GPCR	- G-protein-coupled receptors
GPU	- graphics processing unit
HPC	- high performance computing
HS	- hydrophobicity score
HTS	- high-throughput screening
KNF	- Koshland-Nemethy-Filmer
k-NN	- k-nearest neighbour
LJ	- Lennard-Jones
MCC	- Matthews correlation coefficient
MD	- molecular dynamics
MDS	- multi-dimensional scaling
MM	- molecular mechanics
MWC	- Monod-Wynam-Changeux
NDC	- number of different conformational bins
NMA	- normal mode analysis

OOB	- out-of-bag
PARS	- Protein Allosteric and Regulatory Sites
PB	- Poisson-Boltzmann
PD	- partial dependence
PDB	- Protein Data Bank
PGD	- Protein Geometry Database
PME	- particle-mesh Ewald
POM	- percent-of-maximum
QM	- quantum mechanics
RDC	- ratio of disallowed conformations
RF	- random forest
RMSD	- root mean square deviation
RMSF	- root mean square fluctuation
ROC	- receiver operating characteristic
SASA	- solvent-accessible surface area
SID	- Simple Intrasequence Differences
SPACER	- Server for Predicting Allosteric Communication and Effects of Regulation
SVM	- support vector machine
TNR	- true negative rate
TPR	- true positive rate
VDW	- van der Waals

Table of Contents

1. Introduction

1.1 Allostery

Allostery is a biological phenomenon observed mostly in proteins where the binding of a ligand at one site transfers an effect to another. It is the mechanism by which the majority of functions in living cells are regulated, and understanding its behaviour is an essential bridge between the molecular and cellular domains(1, 2). The significance of allostery has become increasingly recognised, indicated by the popular caption used to describe it: "the second secret of life"(3). Though simple to imagine, the process has proven to be a terrific challenge to comprehensively characterise.

In this section, the concept of allostery is first introduced in broad terms before focussing on it from the perspective of medicinal chemistry. The potential benefits of allosteric modulators as drugs are discussed, as well as difficulties the pharmaceutical sector has experienced in developing them.

1.1.1 Definition and Overview

Allostery was first proposed as a concept in the 1960's(4). Two models emerged at that time, and prevailed for decades since, that aimed to describe the allosteric mechanism. The Monod-Wyman-Changeux (MWC) model(5) proposed the existence of at least two distinct protein states, termed R and T, in thermodynamic equilibrium with each other. The two states possessed varying affinity for substrate binding, with R being the high-affinity state and T being the low-affinity state. An allosteric effector was defined as a ligand that stabilised the R state (allosteric activator) or the T state (allosteric inhibitor) upon binding at another location on the protein, shifting the equilibrium one way or another. A simplified version of allosteric activation within the MWC model is shown in Figure 1.1.

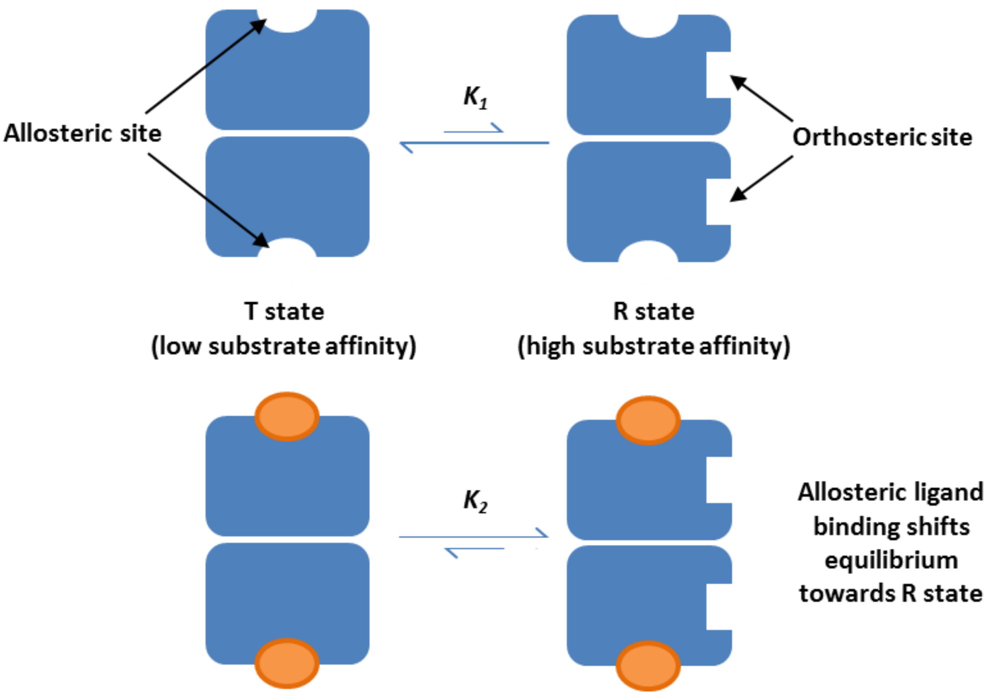


Figure 1.1: *A simplified scheme of the MWC model for allosteric activation. There is a pre-existing equilibrium between the R- and T-states which possess a high- and low-affinity for the orthosteric ligand, respectively. Upon binding of the allosteric activator ligand, the equilibrium is shifted in favour of the R state, effectively increasing the protein's affinity for the orthosteric ligand.*

The competing Koshland-Nemethy-Filmer (KNF) model(6) disputed the notion of a pre-existing equilibrium, suggesting instead that allosteric proteins existed only in the T state in the absence of allosteric effector. It suggested that the inherent flexibility in the protein caused it to adapt its conformation around the effector as binding took place, resulting in a final, allosterically bound R state. This mechanism, dominated by kinetics, is what made the KNF model become known as the induced-fit or sequential model. A simplified version of allosteric activation within the KNF model is shown in Figure 1.2.

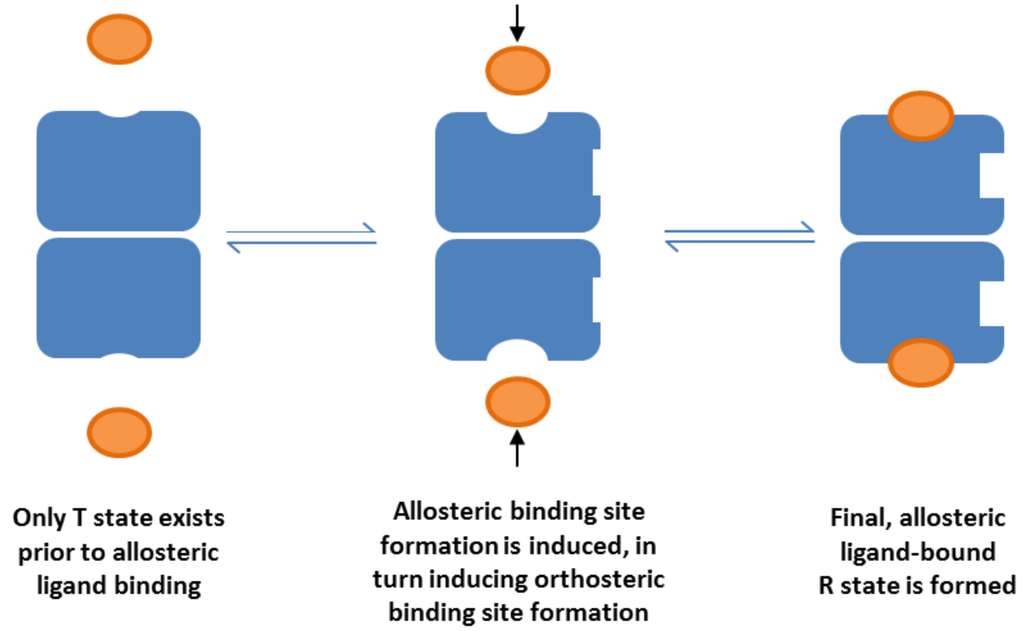


Figure 1.2: *A simplified scheme of the KNF model for allosteric activation. There is no pre-existing equilibrium between R- and T states. As the allosteric activator ligand approaches, it induces the formation of its binding site. The conformational shift in turn forms the orthosteric site. The fully-formed R state exists only in complex with the allosteric activator.*

Modern views have expanded on these principles. Both of the classical models attributed allostery only to homo-oligomers, but today it is thought that all dynamic proteins are capable of displaying some form of allosteric behaviour(7). Since the proposition of the pre-existing equilibrium by the MWC model, the vast majority of proteins have been shown to be dynamic in nature(8–13), a property that has been related to protein function and allostery(8, 12, 13). Furthermore, it was shown that allostery could take place without conformational change between the R and T end-states(14). This case was an important example of allostery that could not be characterised through the viewpoints of either MWC or KNF, implicating the intermediary dynamics as the cause of the change in function. Other research before and since has also shown that the two models can be reconciled(15–17). Today the prevailing thermodynamic view approaches proteins as ensembles of related conformations mapped onto an energy landscape, with lower energy conformations occupying a larger proportion of the ensemble. Allostery is considered a consequence of an effector-binding event influencing the peaks and troughs of the energy landscape, in turn causing a redistribution of the conformer population, as illustrated in Figure 1.3. For obvious reasons, this model is known as 'population shift.' Like the MWC model, it allows for pre-existing populations of both active and inactive conformations, even in the absence of modulator (*cf*. kinetics-based models where the effector binding event causes the formation of activated/inhibited conformations). Recent reviews discuss this idea in great detail(2, 18, 19).

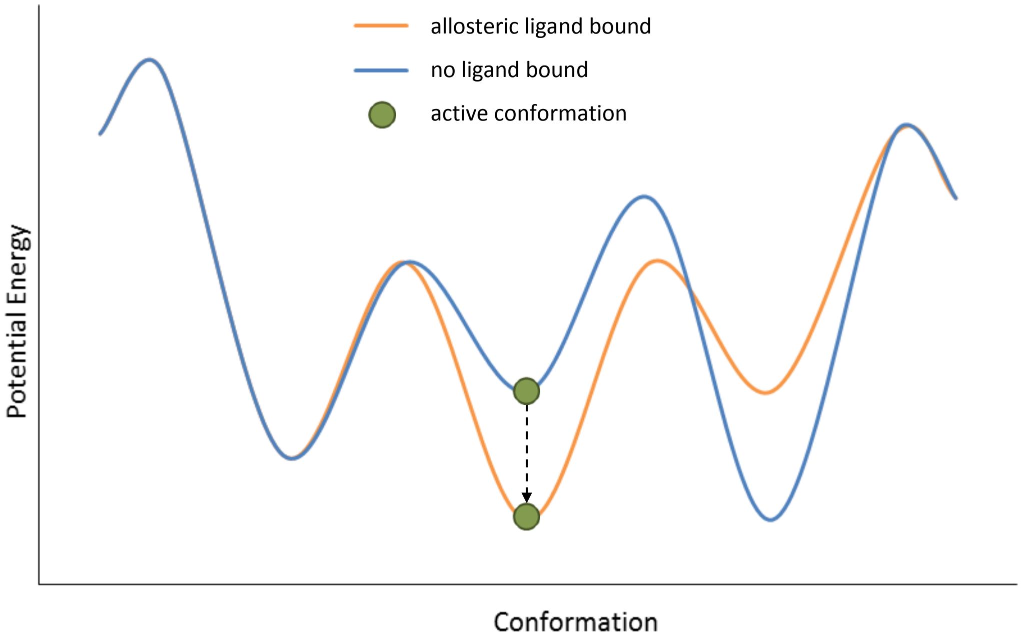


Figure 1.3: *In the ensemble-based model for allostery a protein is considered as the whole population of different conformations it adopts, with lower-energy conformations taking up larger proportions. The binding of allosteric modulators triggers a shift in the energy landscape, altering the population of active conformations. In this figure, upon allosteric ligand binding (orange line), the energy of the active conformation has lowered; hence, the ligand increased the proportion of active conformations and behaved as an activator.*

One clear message emanating from the literature is the need to unite the different aspects of allostery under a single theory. Indeed, recent studies have made strides in reconciling the old viewpoints with the new(20, 21). One such study by Tsai and Nussinov took an interesting step towards this(22), wherein the prevalent models of allostery (R-state/T-state transitions, population shifts, free energy landscapes, structural pathways) were examined together. Despite seemingly key differences in the models(15) such as the question of correlation or causation between effector binding and activated/inhibited conformation, it was shown that, in fact, these can all be distilled down to the same set of theoretical descriptors. Put differently, the phenomenon of allostery has been studied from many different angles, each observing different pieces of the same puzzle. This project began with the central idea that a wealth of knowledge of allostery already exists, but it must be brought together coherently in order to make full use of it.

1.1.2 Biochemical Context

Allosteric binding sites are naturally exploited in cells. A classic example of the biological use of allostery is haemoglobin. A haemoglobin protein comprises four subunits, each bearing a binding site for oxygen. When an oxygen molecule binds to a subunit, conformational changes are induced throughout the remaining subunits, enhancing their affinities for oxygen(23, 24). The overall result is the favourable saturation of haemoglobin with oxygen. This phenomenon, illustrated in Figure 1.4, is termed allosteric cooperativity – at the partial pressure of oxygen exhibited in the lungs, haemoglobin would be unable to become fully saturated with oxygen without it, resulting in it being a far less efficient oxygen transporter(24).

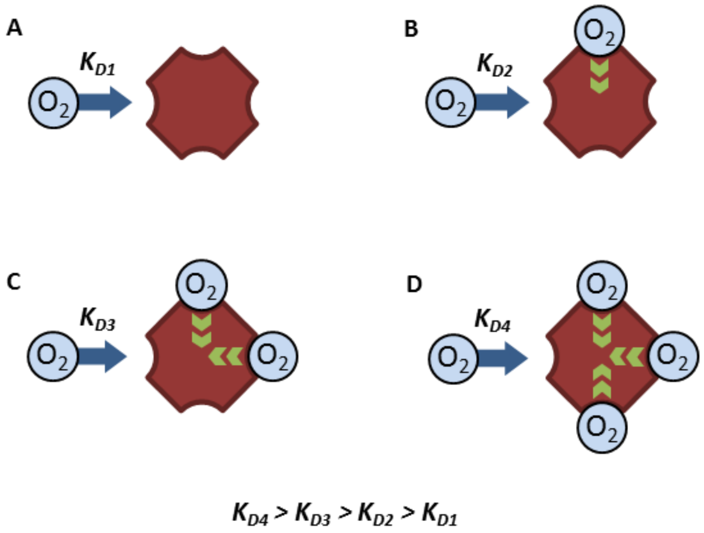


Figure 1.4: *A scheme illustrating allosteric cooperativity in haemoglobin. Binding of a molecule of O_2 at A enhances the affinities of the remaining binding sites in the haemoglobin. This effect is*

compounded with each successive O_2 molecule, allowing haemoglobin to become very easily saturated.

Another notable example of allostery in nature is feedback regulation, where proteins are allosterically inhibited by a downstream product of the pathway they are involved in(25–27), resulting in the maintenance of constant levels of material in the cell. This is illustrated in a hypothetical example in Figure 1.5.

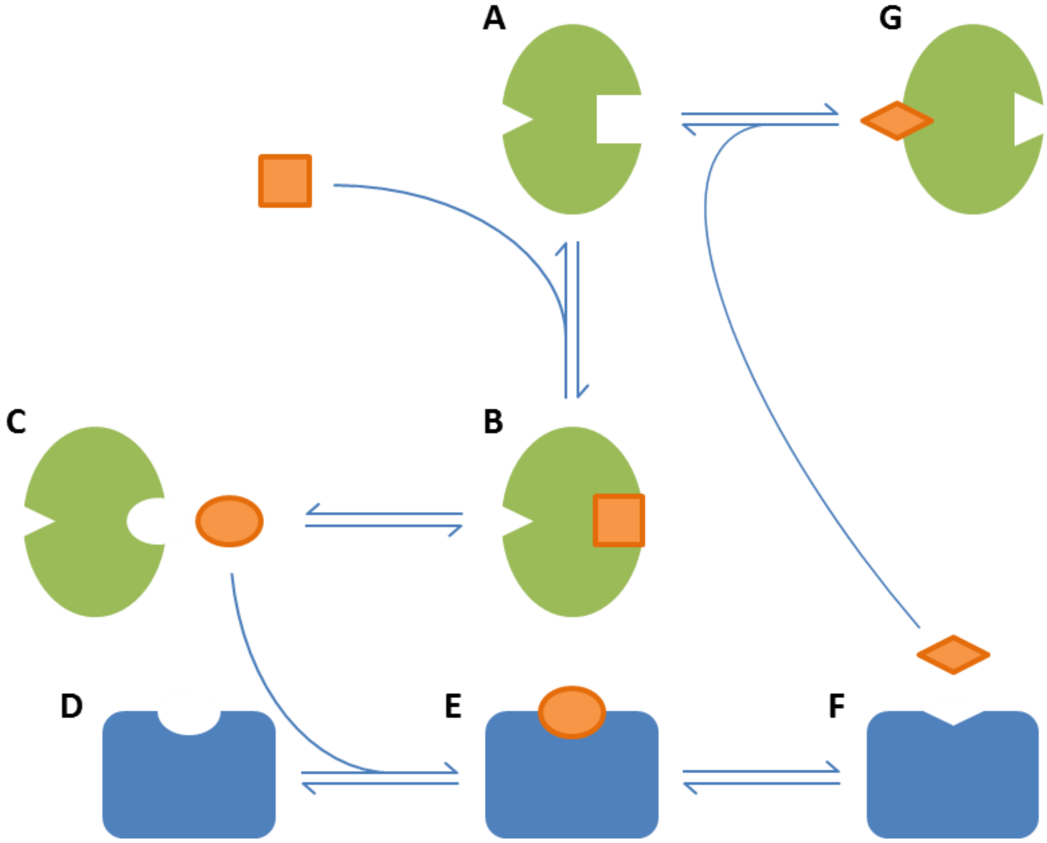


Figure 1.5: *A scheme illustrating feedback regulation. Enzyme **A** converts its substrate, via complex **B**, to the product at **C**. This is taken up as the substrate by a second enzyme **D**, forming, via complex **E**, the final product at **F**. This product allosterically inhibits enzyme **A**, forming the inactive complex **G**. A high concentration of final product results in a high inhibitory effect on its own formation, restricting further production. Mechanisms of this type are exploited in nature to maintain homeostasis in cells.*

Allosteric sites not utilised by biological substrates also exist in proteins; these are highly prized targets in drug discovery. Thanks to their position distal to the endogenous binding site of a protein (hereafter referred to as the orthosteric site), ligands can bind at these sites without competing for position with a natural substrate. However, it is worth noting that it is possible for a ligand to bind to a protein allosterically but still exhibit competitive inhibition(28–30), indicating that the communicative properties of coupled sites in proteins can apply in both directions (*i.e.* binding at the orthosteric site can influence binding at the allosteric site, as well as vice versa).

Incidentally, haemoglobin was one of the first proteins for which the development of allosteric drugs was targeted(31, 32). While the four oxygen-binding sites of haemoglobin display cooperative behaviour amongst each other, there are further allosteric sites on the protein that are not associated with biological substrates. Drugs have been developed to bind to these sites, promoting either oxygen uptake or oxygen release(33, 34).

1.1.3 Advantages of Allosteric Modulators

Where nature has not harnessed them for its own purposes, allosteric sites have been under less evolutionary pressure to remain conserved across protein subtypes than orthosteric sites(35). This means that allosteric ligands have the potential to be more selective(36–38) – an important implication for drug discovery, which is frequently troubled with drugs having a myriad of off-target effects(39, 40). Off-target effects are a near-ubiquitous phenomenon of pharmaceuticals, but a well-known and generic example of the problem is found in anti-cancer drugs. Many of these target the ATP-binding pocket of kinases; due to ATP's prevalence in all cells, and it being the natural, endogenous substrate of a vast number of proteins (most with highly conserved ATP-binding pockets), it is very difficult to design an inhibitor that selectively binds to the ATP-binding pocket of one kinase over the many others present in cells. This leads to the disruption of multiple signalling pathways beyond the target, ultimately causing the array of all too well-known adverse side-effects in patients.

The analogy of a dimmer switch is often used to describe the effect of allosteric modulators. While orthosteric inhibitors generally shut down a protein's activity (akin to a standard 'on/off' switch) for as long as they are bound, allosteric modulators are indeed capable of *modulating* protein activity(36, 37, 41) – that is, enhancing or dampening activity without fully locking it in an 'on/off' position. In terms of population shift, this phenomenon can be viewed as the proportion of active protein conformations in the ensemble being shifted to a multitude of different ratios, rather than being reduced to an insignificant presence.

Allosteric modulators often exhibit a saturation or 'ceiling' effect(37, 42) – a point of modulator concentration beyond which no further allosteric activity occurs. Figure 1.6, reproduced from the review by May *et al.*(37), shows the effect on the fraction of binding of orthosteric ligand, **A**, of increasing concentrations of positive, neutral or negative allosteric modulator, **B**. The cooperativity factor, α, quantifies the magnitude of change in ligand affinity at a binding site induced by the binding of a ligand at another. Values greater than 1 correspond to an increase in affinity, and values below 1 to a decrease. The ceiling effect is revealed through a levelling off of the fractional binding of **A**, despite further increases in the concentrations of **B**.

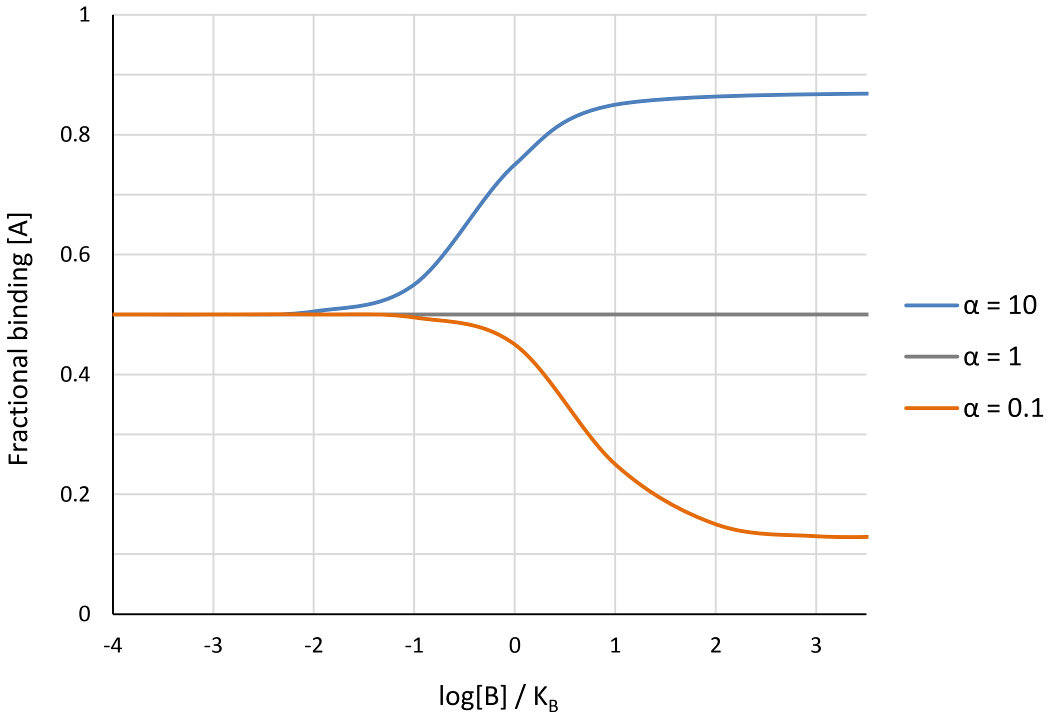


Figure 1.6: *The ceiling effect exhibited by allosteric modulators, **B**. At a fixed concentration of orthosteric ligand, **A**, an increase in allosteric effect, positive or negative (measured here by the influence on the fraction of binding of **A**), can be seen on increasing concentration of **B**. However,*

there comes a point where the increase in effect levels off. This phenomenon is not observed with orthosteric binding. Adapted from (37).

The ceiling effect has great potential to be exploited when developing new drugs: large doses of allosteric modulator can, instead of having a toxic effect by causing too great a change in protein activity, act as reservoirs that will maintain a prolonged therapeutic effect, provided they are not rapidly cleared from the cell. In other words, an effect equivalent to controlled release can be built in to the allosteric mode of action at the mechanistic level, without the need for a special formulation. Figure 1.7 illustrates this point with a plot of effect on a hypothetical target protein against time. At a glance this may appear similar to a plot of plasma concentration against time, but the difference is key: the link between efficacy and concentration breaks for allosteric modulators at concentrations beyond the onset of the ceiling effect.

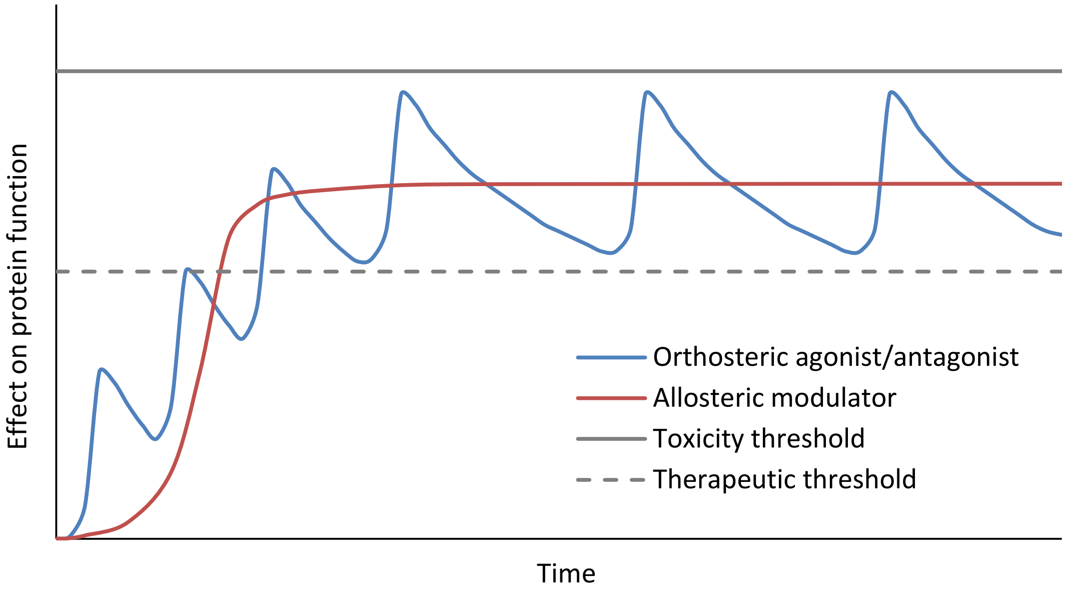


Figure 1.7: *A plot of effect on a hypothetical target protein against time. For conventional orthosteric agonists or antagonists (blue) repeated doses must administered to maintain therapeutic levels, whereas for allosteric modulators exhibiting a ceiling effect within the therapeutic range (red) one reservoir dose can be administered, resulting in behaviour much like a controlled-release formulation.*

The above applies to chronic treatments, where constant levels of protein activity must be maintained. However, for physiological processes requiring fluctuating levels of active protein, naturally controlled by messenger molecules like hormones or neurotransmitters, another exploit exists. So long as an allosteric modulator exhibits no medicinal efficacy of its own, it can only influence the extent and effect of binding of the endogenous substrate. Regardless of the time of administration of the allosteric modulator, the natural cycles of receptor activity can be conserved(36, 37, 43). The plots in Figure 1.8 and Figure 1.9 demonstrate this for the treatment of a messenger molecule deficiency by orthosteric and allosteric modes of action, respectively. The same concept applies in reverse for the treatment of an excess of messenger molecule.

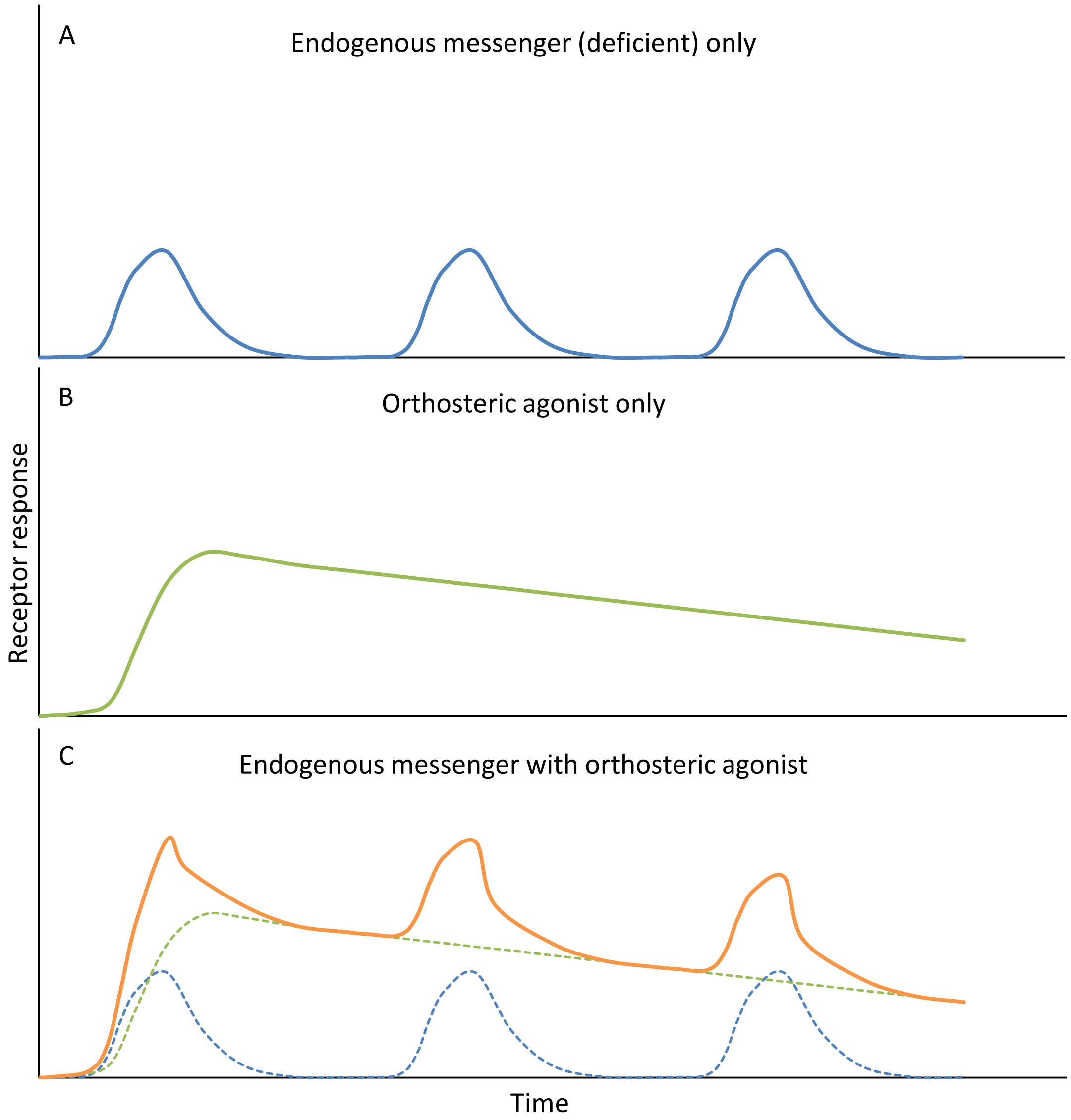


Figure 1.8: ***A**, naturally fluctuating but deficient receptor activity induced by an endogenous messenger; **B**, receptor activity due to orthosteric agonist; **C**, receptor activity treated with orthosteric agonist is boosted but natural fluctuation is disrupted. Adapted from* (43).

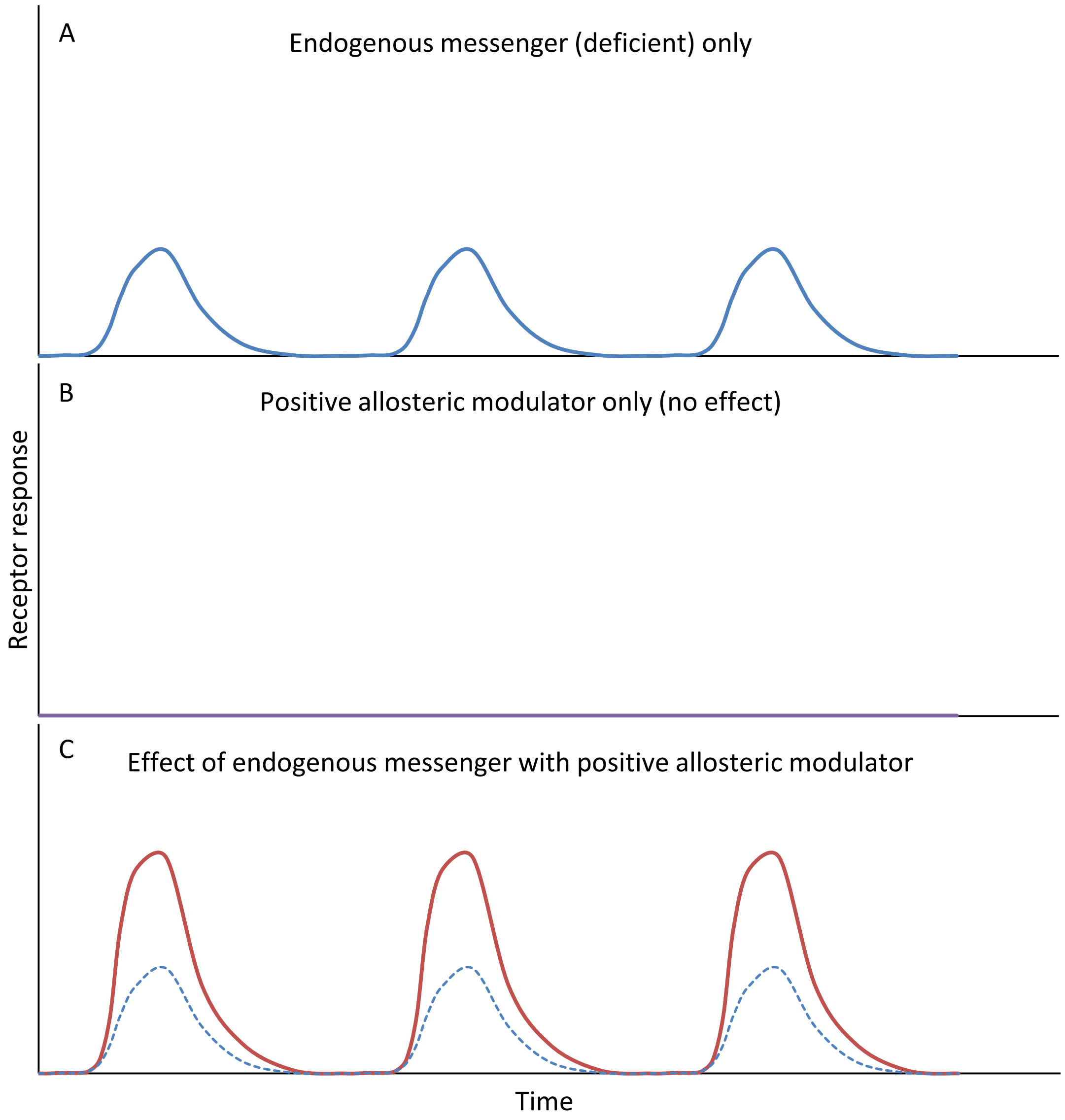


Figure 1.9: ***A**, naturally fluctuating but deficient receptor activity induced by an endogenous messenger; **B**, allosteric modulator displays no efficacy of its own; **C**, receptor activity treated with positive allosteric modulator produces a superior mimic of physiological fluctuation. Adapted from* (43).

A further subtlety of allosteric binding is the notion of *probe dependence*. In this context a 'probe' is an orthosteric ligand, usually one of an array. A single allosteric modulator can have a variable influence on the potency of individual orthosteric 'probe' ligands(44). Hypothetically, an allosteric modulator could cause a ten-fold increase in potency of one orthosteric ligand and simultaneously cause a hundred-fold decrease in potency of another. Viewed with the ensemble model, an allosteric binding event causes a redistribution of conformational populations, in effect creating a similar but distinct orthosteric binding site; this has an altered set of responses to interacting ligands(45).

A 2012 study demonstrated a potential use for probe dependence(46). Acetylcholine, a neurotransmitter, is metabolised to form an inert compound, choline. In the study it was found that an allosteric modulator was able to induce activity from choline, thereby modulating activity of the receptor. This mechanism of metabolite activation provides a novel pathway only available to allosteric modulators.

The situations discussed here demonstrate great potential advantages of allosteric drugs, but share the assumption that a suitable allosteric drug exists *and has been located*. In reality there are many hurdles in the way of allosteric drug development – that is, over and above those of conventional orthosteric drug development. The next section briefly highlights these, in particular the identification of an allosteric site, which is the focus of this project.

1.1.4 Challenges of Discovery

Historically, the identification of novel allosteric sites has proven challenging. Attempts have been made with high-throughput screening (HTS), a time-consuming and expensive process at the best of times, but allosteric hits have been few in number and generally happened upon serendipitously(47). HTS tends to perform poorly when searching for allosteric modulators, generating vast amounts of false positive data that are difficult to distinguish from genuine hits(48). Additional methods, such as kinetic assays and X-ray crystallography, are generally required to tease out the latter from the former(48, 49), even then with variable success rates.

Another complication arises from probe dependence. This dependence can be of sufficient variability to modulate activity in both directions, *i.e.* to induce both activation and inhibition at the active site, depending on the allosteric ligand bound(50). This can clearly confuse the interpretation of experimental results on a series of potential modulators. The issue can also cause problems further down the drug discovery pipeline: even after a hit compound has been identified for targeting an allosteric site, its activity could flip between activating and inhibiting modes as it is modified(51).

Computational methods have long aided the drug discovery process. A vast array of methods exists, with examples including molecular visualisation, pharmacophore modelling and ligand docking. A recent review by Sliwoski *et al*. provides a wide survey of methods(52). In terms of the drug discovery pipeline, they are most often used to inform decisions on compound selection for screening, thus increasing the likelihood of achieving a hit. An industry-standard example is the application of Lipinski's rule of five(53) to filter compound libraries for ones possessing drug-like properties.

An interesting publication recently used machine learning to distinguish the properties of allosteric and non-allosteric ligands(54), concluding that allosteric ligands are generally more rigid and lipophilic than non-allosteric ligands, though the magnitude of the differences varied by target class. This work is clearly applicable to the enrichment of compound libraries for targeted allosteric modulator screens, and indirectly suggests that allosteric sites possess complementary properties.

A more general issue with many methods in drug discovery is the false assumption that proteins are static structures. While they remain a highly trusted data source for drug discovery from which many structural and mechanistic conclusions are drawn, the vast majority of protein crystal structures come with an underappreciated degree of ambiguity. All too often, estimations and arbitrary decisions about the observed electron densities result in an uncertainty of atomic coordinates(55, 56). Moreover, proteins are highly dynamic in nature, assuming a large ensemble of conformations not captured by X-ray crystallography.

Knowledge of multiple protein conformations can be highly relevant, even pivotal to research. An illustrative example of this can be found in an investigation of human nicotinic acetylcholine receptor (AChR) (57, 58). Researchers crystallised small molecule AChR inhibitors in complex with mollusk acetylcholine binding protein (AChBP), which served as a structural and functional proxy for AChR. The crystal structures adopted one closed-loop conformation that was significantly different from the open-loop conformation observed in

AChBP complexes with large inhibitors such as snake neurotoxins. These results led to the hypothesis that AChBP was highly dynamic and that various open- and closed-loop conformations were viable pharmacological targets.

The above example is a fortunate one, where researchers were able to produce multiple X-ray crystal structures of their protein of interest. In doing so, they revealed different pocket conformations and gained valuable insight into the protein's behaviour. Such a position, with multiple conformers revealed through crystallography, is often not reached for other proteins that are more difficult to crystallise.

Molecular dynamics (MD) could potentially be used to model multi-conformational protein ensembles, filling the experimental knowledge gap described above. Indeed, MD has been successfully used to provide mechanistic insight into allosteric sites, though these have invariably been isolated cases requiring long/multiple simulations(59–62). MD has never been used to systematically predict allosteric sites without prior knowledge of them.

1.2 Molecular Modelling

Molecular modelling is an umbrella term referring to theoretical techniques used to reproduce aspects of molecular behaviour. Today the vast majority of it is carried out with computers by applying physical and mathematical laws to modelled systems. The computational power and time required to perform these calculations is such that sophisticated molecular modelling techniques are often restricted by available hardware. It has thus been a historically limited field, but has seen rapid expansion in recent years as the power/cost ratio of computers has improved.

Molecular modelling at an atomic scale consists of two main categories, *quantum mechanics* (QM) and *molecular mechanics* (MM). Though it was not used in this project, QM is briefly introduced here for context. MM was extensively used; specifically, it was applied to simulate the physical motions of molecular systems over time, a process known as *molecular dynamics* (MD). MM is introduced in this section for comparison to QM, with a more detailed look at MD following in section 1.3.

1.2.1 Quantum Mechanics

A trade-off exists in molecular modelling between the accuracy of a technique and the computational cost of executing it. The more approximations one is willing to make, the larger the system that can be modelled in the same time. The defining difference between QM and MM is in the handling of electrons within the models: in QM, electrons are treated explicitly. Orbital energies and coefficients can be modelled, as can bond breaking/making events and heats of formation of molecular conformations(63, 64). This is ultimately achieved through solving (approximations of) Schrödinger's equation, in many cases from first principles; such methods can thus be described as *ab initio*, though empirical and semi-empirical methods – *i.e.* methods utilising some stored parameters rather than deriving them from first principles – exist as well(64).

A prime example of an approximation used to speed up calculation time is the Born-Oppenheimer approximation(63, 64), where the Schrödinger equation is vastly simplified by splitting it into electronic and nuclear components. The rationale begins from the knowledge that forces acting on electrons and nuclei are due to their respective electric charges, which are of the same order of magnitude. Any changes to these particles' momenta due to these forces must therefore also be of the same order of magnitude; thus one can reasonably assume that the electrons and nuclei have similar momenta. Given that the mass of a proton – the least massive possible nucleus – is approximately 2000 times greater than that of an electron, and that momentum is the product of mass and velocity, nuclei must have a velocity approximately 2000 times smaller than electrons. With this knowledge in hand, two approximations can be made when considering the motion of a molecular system:

- On the timescale of electronic motion, nuclei can be considered as effectively stationary
- On the timescale of nuclear motion, electrons can be considered to instantaneously adopt their ground-state (lowest-energy) configuration

These approximations can be used to isolate the electronic and nuclear components of the Schrödinger equation. By solving the electronic component first, the solution can be used to solve for the nuclear component separately. Nuclear motion can still be modelled by assigning a range of coordinates for the nucleus' position and incorporating a repulsion term(64). Even with this extra complication, the two successive calculations are far quicker to compute than the full equation.

Many other approximations are utilised within QM, such as the variational method and perturbation theory(64), all serving to decrease the great computational cost of calculations by sacrificing some accuracy. However, further, far more drastic approximations become necessary when modelling larger systems such as DNA, RNA or proteins. Systems of this scale remain beyond the reach of quantum mechanics today, and even further so if the explicit modelling of surrounding solvent is desired. A recent review aimed at non-experts clearly and concisely discusses many of the concepts mentioned in this section on both QM and MM(65).

1.2.2 Molecular Mechanics

MM seeks to ease the computational cost associated with QM, chiefly by accounting for electron presence implicitly. In MM models, each atom is treated as a single, classical particle of fixed point mass and charge. Atomic motion is then approximated using Newtonian physics. In such a system, a set of terms known as a force field is used to describe the energies of each type of interaction.

Most force fields are defined such that the total energy of the system can be described by the sum of various interaction energies between all atoms with it: bond stretching, bond bending, bond torsion, van der Waals (VDW) interactions and electrostatic interactions. A final 'miscellaneous' energy component covers a host of further corrections specific to each force field. While the exact form of each force field can vary, most break down into these main components, summarised in Equation 1.

$$E_{total} = E_{bonds} + E_{angles} + E_{dihedrals} + E_{vdw} + E_{elec} + E_{misc} \tag{1}$$

Common force fields include CHARMM(66), GROMOS(67) and AMBER(68). These are continuously being updated and refined to better reproduce known experimental results and utilise ever-increasing processing power(69–71). The latest release of the AMBER force field at the outset of this work, ff12SB(72), was used here. The basic form of the AMBER force field is shown in Equation 2.

$$\begin{aligned} U = & \sum_{bonds} k_r(r - r_0)^2 \\ & + \sum_{angles} k_\theta(\theta - \theta_0)^2 \\ & + \sum_{dihedrals} k_\varphi \left(1 + cos(n\varphi - \delta)\right) \\ & + \sum_{\substack{nonbonded \\ pairs\ ij}} \epsilon_{ij} \left[\left(\frac{R_{i,j}}{r_{i,j}}\right)^{12} - 2\left(\frac{R_{i,j}}{r_{i,j}}\right)^{6} \right] \\ & + \sum_{\substack{nonbonded \\ pairs\ ij}} \frac{q_i q_j}{4\pi\epsilon_0 r_{i,j}} \end{aligned} \tag{2}$$

where U = potential energy of the modelled system, k_r = bond stretching force constant, r = bond length, r_0 = reference bond length, k_θ = bond angle force constant, θ = bond angle, θ_0 = reference bond angle, k_φ = torsional barrier, n = periodicity, φ = torsional angle, δ = dihedral phase, ϵ_{ij} = LJ well depth, $R_{i,j}$ = interatomic distance of minimum potential between atoms i and j, $r_{i,j}$ = interatomic distance between atoms i and j, q_i/q_j = atomic charge of atom i/j, ϵ_0 = dielectric constant.

Bonds are modelled as Hookean springs with equilibrium lengths, angles and force constants(63, 73). A periodic function defining the oscillations in energy of conformations through a 360° bond rotation is used to model dihedral angles. Non-bonded interactions are generally split into a (VDW) component, modelled with a Lennard-Jones (LJ) ‘12-6’ potential, and an electrostatic component based on Coulomb’s inverse square law. These interactions are illustrated in Figure 1.10. The E_{misc} term can be thought of as a ‘clean up’ term that accounts for weaknesses in the preceding terms. Its components vary depending on the exact functional form taken by the rest of the force field, but cross-terms are generally required: that is, in reality, a bond’s ability to bend will depend on how stretched it is; a bond’s torsional barrier will depend on the angles and lengths of vicinal bonds(63).

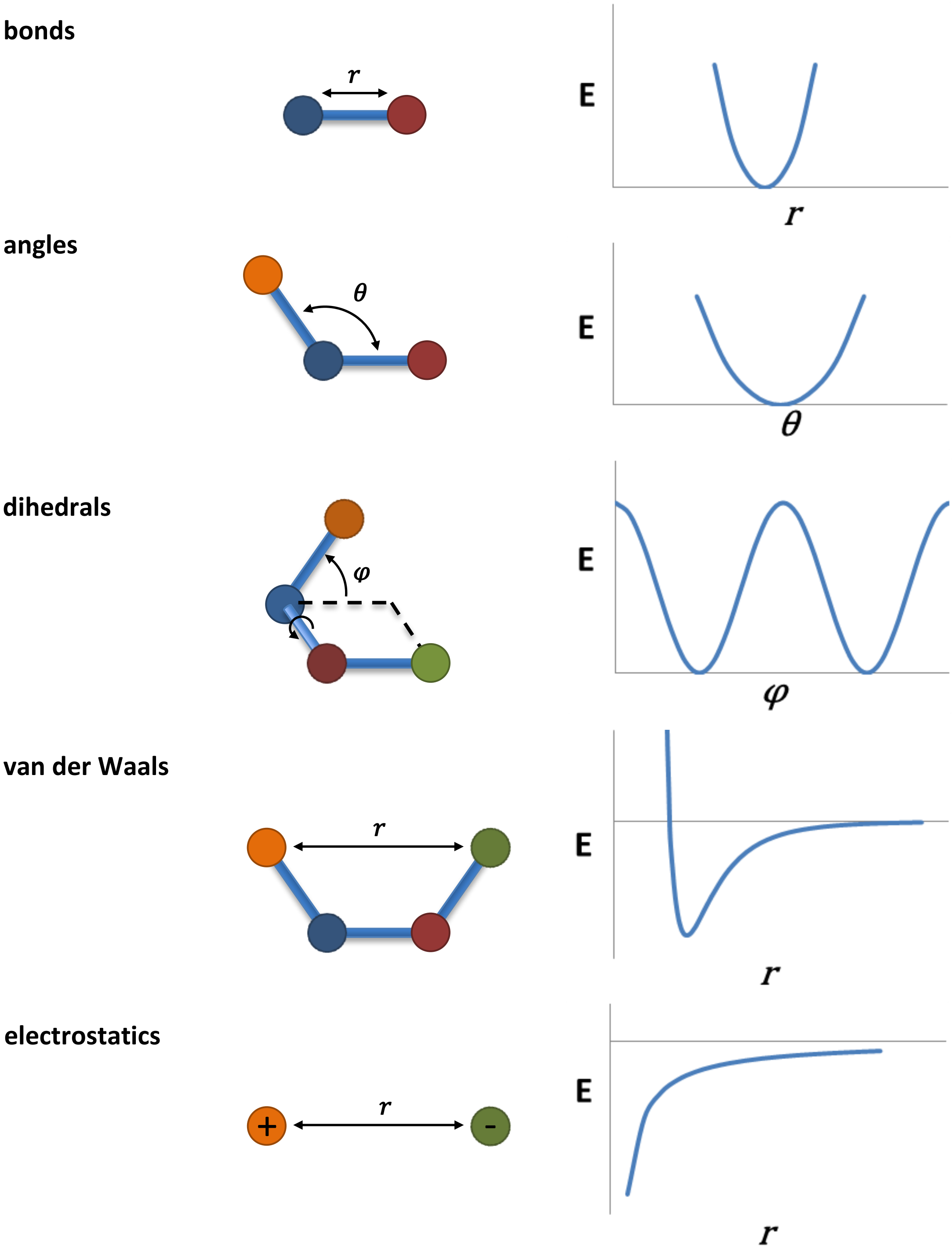


Figure 1.10: *The main interatomic interaction energies described by the component terms of a force field equation. A sample energy profile is shown alongside each term: bond lengths and angles are modelled with quadratic, Hookean terms; dihedral angles are modelled with a periodic cosine function; van der Waals forces are modelled with a '12-6' Lennard-Jones potential and electrostatics are modelled with a Coulombic inverse-square term.*

With electron presence accounted for implicitly by these terms, far fewer particles need to be simulated; a force field therefore incurs a far lower computational cost than using QM, allowing for far larger and/or longer molecular simulations. Equally though, with no explicit electrons, energetic calculations based on MM can be highly inaccurate(64, 74), and chemical reactions – which involve breaking and making bonds – cannot be modelled at all. Hybrid methods, employing MM to efficiently model the system at large and QM to model the reaction site in detail, have been developed for investigating reactions involving large molecules, such as enzyme catalysis. Due to the non-trivial nature of getting the QM-based portions of such models to properly communicate with the MM-based portions, QM/MM methods are often considered a field in their own right(75–77).

1.3 Molecular Dynamics

Molecular dynamics (MD) is the process of applying a force field to a given molecular system and integrating it with respect to time to determine the momenta of the system's atoms. The result is a full simulation of the system's dynamics that can be viewed with aid of visualisation software or processed with further analyses as required.

As is discussed in this section, the time steps over which force field equations can be integrated without collapsing the system due to errors is of the order of femtoseconds. Thus, if a simulation of meaningful length is to be produced, many successive solutions must be calculated. Two factors determine the time required to produce a MD simulation: on the software side, there is the efficiency of the algorithms solving the force field equation at each time step. On the hardware side, the power of the processor performing the calculations is just as important a factor, if not more so. These concepts are covered in this section, as are the stages of preparation involved in preparing and running a MD simulation in AMBER.

1.3.1 Software

The MM force fields mentioned 1.2.2 are implemented in MD software packages. Developers of each force field usually also develop their own MD software to implement it, so each is generally optimised to operate with its native force field. However, a level of cross-compatibility exists in that most software packages are capable of working with at least some force fields other than their own. This is important for users because this type of software is highly specialised and not trivial to operate. If a user is familiar with one software package but wishes to use a 'foreign' force field that is better suited to model their system of interest, they need not learn to operate new software.

1.3.2 Solvent Treatment

It is clear that the surrounding medium of any molecular system exerts a huge influence on its behaviour both in geometric and energetic terms; it is therefore important to incorporate solvent where appropriate, such as in simulations of biomolecules. Solvent molecules can be introduced either implicitly or explicitly.

In implicit solvation models, no new atoms are introduced to the modelled system. Instead the solvent is considered as a continuous medium with averaged properties(64, 78, 79), accounting for its influence on the solute through supplemental equations to the base force field. Perhaps the most important solvent property, particularly for most biological simulations where the solvent is water, is the dielectric constant. The Poisson-Boltzmann (PB) equation can be applied to such a model, treating the solute atoms as low-dielectric particles surrounded by a high-dielectric medium(79) and computing the interaction. The PB equation is highly accurate, but very expensive to compute. To remedy this, an approximation to the PB equation is often used that is quicker to compute, named the Generalised Born (GB) equation. This assumes an ideal case of the PB equation where the solute is perfectly spherical, adjusting the radius of each solute particle to match a predetermined energy of solvation(78).

Explicit solvation is more intuitive in theory: a layer of solvent molecules is added to the system, surrounding it in all dimensions. In this scenario, some action must be taken to prevent the solvent molecules from scattering into empty space as soon as MD commences; the most common solution is to apply periodic boundary conditions. In effect, the modelled system is treated as a unit cell and surrounded by exactly translated images of itself in all directions. In the central, ‘real’ cell, any particle crossing a boundary has an exact copy reintroduced on the opposite side (Figure 1.11). In this way the solute can experience conditions akin to those of bulk solution with a relatively small number of solvent molecules.

For periodic systems, the particle mesh Ewald (PME) method is the standard method for handling electrostatics(80). This is a grid-based method that splits interactions into short- and long-ranged, handling the former ‘normally,’ that is spatially, based on Coulomb’s inverse square law. The latter are handled with a fast Fourier transform. Overall this method remains accurate and, thanks to the fast Fourier transform, is highly efficient when faced with large numbers of particles(80, 81). This is of great significance for modelling explicitly solvated systems.

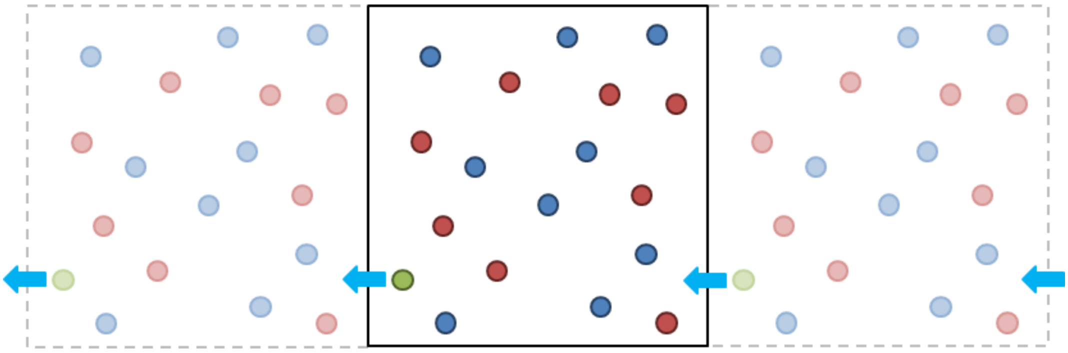


Figure 1.11: *A model system shown with periodic boundaries in one dimension. The green particle is on trajectory that will cross the cell boundary; at the same time, an identical copy of the particle will enter the system on the opposite side.*

Most MD software packages contain algorithms to aid in constructing explicitly solvated system. In AMBER, the LeAP program can automatically surround an inputted system with a solvent shell of specified depth. The fewer solvent molecules added the lesser the increase in computational expense of simulation; however, the dimensions of the unit cell must be large enough that the solute system is unable to interact with itself through the periodic boundary, a situation shown in Figure 1.12.

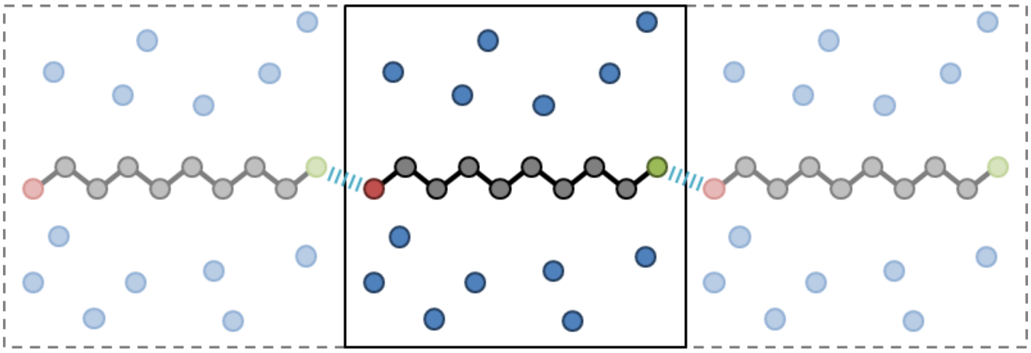


Figure 1.12: *A molecular system with periodic boundaries that are too small. The two terminal atoms of the solute molecule, while far apart in reality, can interact with each other through the boundary, resulting in a highly unrealistic model.*

Implicit solvation methods have the significant advantage of not introducing more particles, and so result in systems that are far quicker to simulate. Since the number of solvent particles required for a sufficiently large explicit solvent shell increases exponentially with the size of the system (cubically, in a 3D system), this can quickly become problematic. In situations where particularly large systems or particularly long simulations are required quickly, implicit methods may be most suitable.

However, if the resources are available to simulate the equivalent system in explicit solvent, it is often the superior choice for biomolecules. The dynamic behaviour of these large systems is highly responsive to solvent effects: proteins especially often require non-averaged solvent phenomena such as explicit solvent-solute hydrogen bonding and fluctuating local solvent densities that can only be captured with an explicit model(82).

1.3.3 Energy Minimisation

Since approximations in force fields tend to rely on systems being in near-equilibrium states, a model system must generally be brought to such an energy state before performing MD. This refinement step is of particular importance if any artificial alterations, such as addition of explicit solvent, have been made.

The potential energy of an entire system is a function of each atom's coordinates within it, and so is far too complex to solve and impossible to visualise for all but the very smallest systems(64, 83). Without being able to produce the global minimum conformation directly, minimisation procedures must operate by making small, iterative adjustments to atomic positions such that the net force acting on them is reduced each time. Since force is the gradient of energy, the system energy is minimised when a position of zero (or as close as possible to zero) net force is achieved. Figure 1.13 illustrates this idea, as well as an undesired consequence of it: with minimisations always seeking to reduce the energy gradient, they can only move towards the local minimum, which may not be the global minimum.

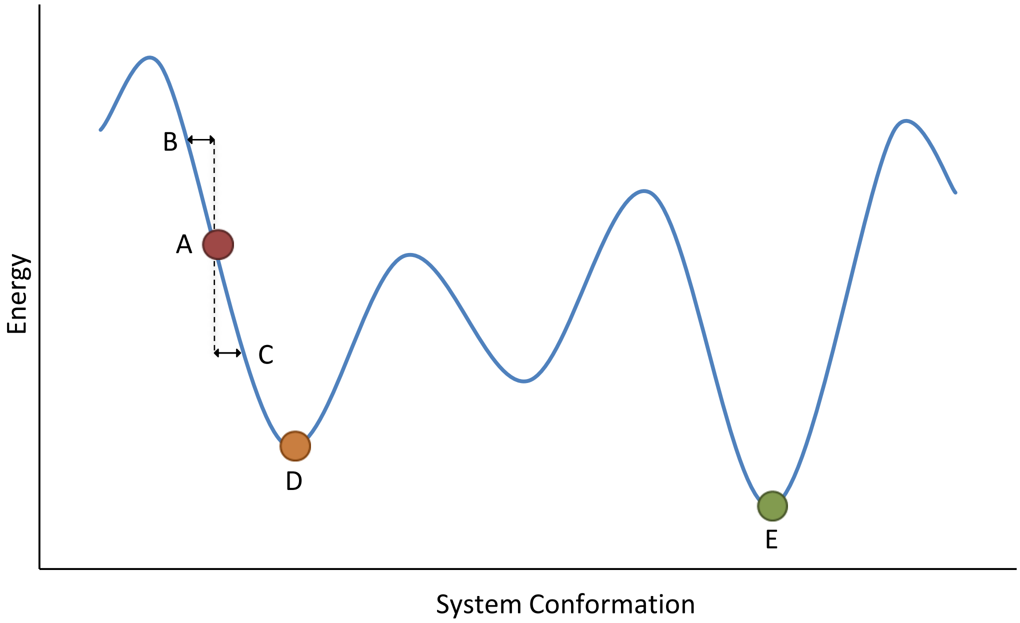


Figure 1.13: *A hypothetical potential energy surface for a molecular system is shown, reduced to one coordinate. From the initial conformation,* ***A****, energy minimisation algorithms only yield lower energy conformations, so conformation* ***B*** *cannot be reached, but conformation* ***C*** *can be. Further iterations will drive the system to the local minimum,* ***D****, but not to the global minimum,* ***E****, since the latter requires high energy barriers to be crossed.*

Many minimisation algorithms exist(64), each of them varying in computational complexity and robustness. No one method is best overall; generally, one is used to make initial refinements before switching to another that is more suitable for converging on the minimum. A common two stage-minimisation was employed throughout this work: the *steepest descent* method followed by the *conjugate gradient* method.

The steepest descent method utilises information from the first derivative of the potential energy surface (*i.e.* the gradient) and so is termed a *first order* method. Its name is intuitive: it operates by moving in the direction of the steepest slope. The method is simple and robust, working well for high energy conformations where direction steepest descent is generally clear. However, it becomes inefficient in shallow valleys(64), so is not ideal for converging on the minimum. For this purpose, the conjugate gradient method is employed. This method is also first order, but also retains knowledge of previous first derivatives to instruct the next step direction, which adds to the computation per step. However, this allows the method to reach the minimum more efficiently, meaning there is efficiency to be gained from its combined use with a quick and robust initial method like steepest descent.

Second order methods also exist, the most popular being the Newton-Raphson method(64, 83). These methods can be very powerful, but at the cost of great computational cost that scales poorly with system size; they are generally unsuitable for large biomolecular systems of the type found in this work.

For proteins, particularly those with coordinates based on crystallographic structures, it is standard practice to run three sequential stages of minimisation (each consisting of a steepest descent phase and a conjugate gradient phase). In the first, a restraining force is applied to all heavy atoms, effectively anchoring them to their initial state. This allows the efficient minimisation of hydrogen positions, which will often have been added artificially and so are most likely to be involved in bad contacts. In explicitly solvated systems, the solvent molecules – also added artificially – are generally left unrestrained as well. The second phase lifts the restraints from protein sidechains, the next most poorly resolved regions of the system. Finally, the whole system is unrestrained and minimised.

1.3.4 Heating, Equilibration and Production Phases

After minimisation of a model system, MD can begin. At this point the system is, in effect, at 0 K since there are no velocities associated with its atoms. To initiate motion in the system, all atoms are artificially assigned (small) velocities in randomised directions. With initial velocities applied, the system can be solved in terms of the force field equation and new atomic positions determined. However, without any further interference, the system will not behave like it would at any specific temperature. Special thermostatic algorithms have been developed that impose a given temperature on the system.

The Langevin thermostat(84) is a popular choice for regulating temperature for the purposes of system heating and equilibration. This scheme stochastically applies pseudorandom forces to all atoms in the system. These forces follow a Maxwell-Boltzmann distribution. The scheme therefore simulates imaginary collisions with gaseous particles at thermal equilibrium, evenly dispersing temperature with time. The net effect of this on the system is an active thrust towards a thermally equilibrated state. It is highly suited for the initial application of temperature.

For biomolecules, MD simulations are performed at temperatures in the region of 300 K. It is standard practice to stagger the heating of such systems, heating in portions of approximately 100 K and allowing a small amount of simulation time for the system to relax before the next increase in temperature. For periodic systems, the system's pressure, which will have invariably fluctuated wildly throughout the heating procedure, must also be brought under control. This is achieved by maintaining a constant temperature and gradually altering the dimensions of the unit cell until the system reaches the desired equilibrium pressure (generally 1 bar for biomolecules).

After reaching baric equilibrium the unit cell dimensions can be fixed once more. Before running production MD – that is, MD simulation that is considered experimental rather than preparatory – it is good practice to simulate the system for a further period of time as a final measure to ensure the system is stable and well equilibrated. This final equilibration should ideally be as long as possible, but in reality is often determined by the user's hardware, and how much simulation time they are willing to discard.

For production MD, it is common to switch over to the Berendsen thermostat(85). This is less invasive than Langevin dynamics, maintaining the overall kinetic energy of the system for a given temperature by scaling velocities rather than altering both scale and direction through artificial collisions. It allows more variation in temperature throughout regions of the system, making it beneficial for systems free of artefacts, where natural variation over time is desired, but is weak for 'ironing out' anomalies. It is therefore best suited for production use after a thorough equilibration. Moreover, Langevin dynamics have been shown to be unstable when used for long simulations(86).

1.3.5 Integration Step Size

A fundamental consequence of using a force field equation is that errors quickly accumulate as the time step of each integration increases(64). This is because each particle can only be represented by a single set of parameters in any one step. One can envisage two atoms at a distance with velocities directed toward one another. With a large time step, the next snapshot of system could show them overlapping, as determined by their initial velocities. Of course, in reality they would repel each other as they approached, reducing velocity and eventually changing direction altogether. Without intermediate time steps to capture this developing repulsion (chiefly by the increasing '12' term of the LJ potential), the simulation would model the collision unrealistically. This scenario is pictured in Figure 1.14.

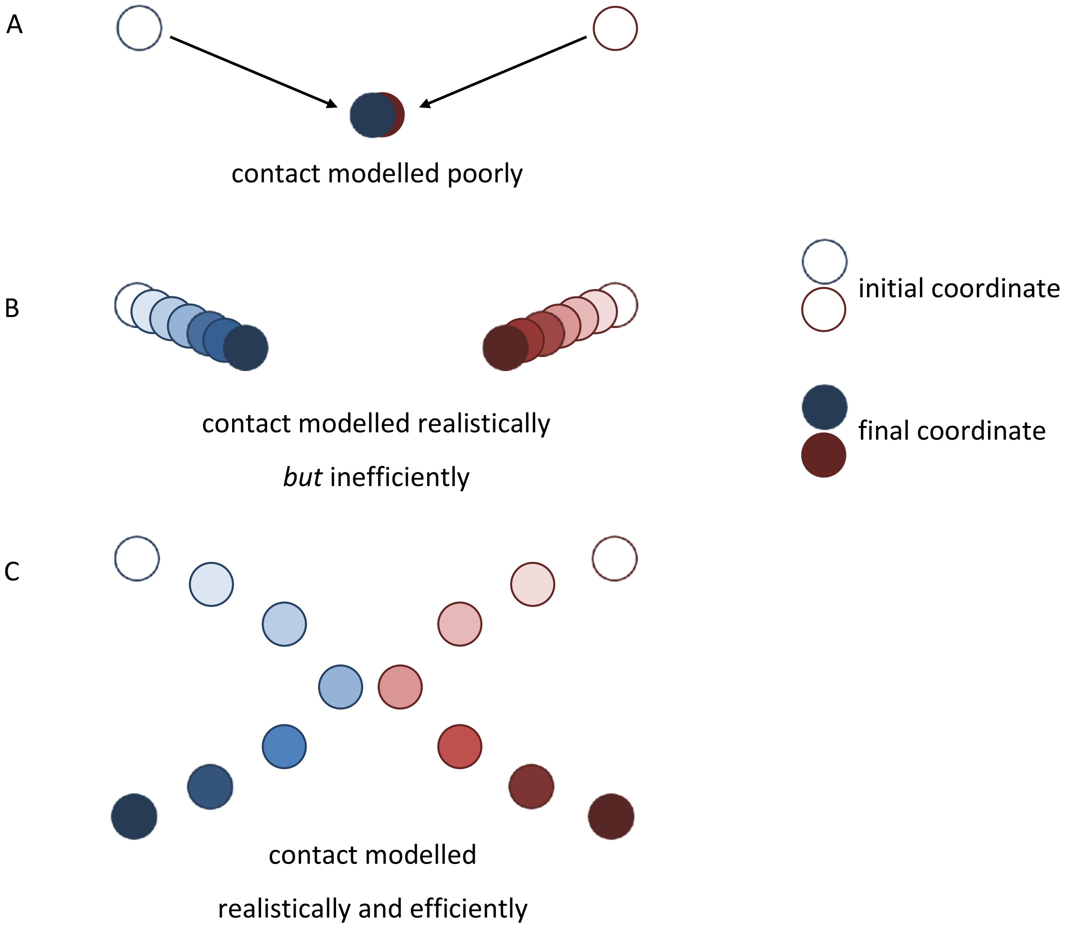


Figure 1.14: *Two atoms (red and blue) in a simulation. Initially, they are distant but on trajectories directed towards each other.* ***A****, an excessively large time step is used to determine new positions, resulting in overlapping atoms;* ***B****, an excessively small time step is used – while this avoids the bad contact of the two atoms, it is highly inefficient;* ***C****, an appropriate time step is used, resulting in efficient modelling and realistic collision handling. Adapted from*(64).

At best this yields unrealistic, high-energy results, though it can cause total collapse of the system if such artefacts cause fatal errors when solving the force field equation. Using the above example of two approaching atoms, two perfectly overlapping atoms have an interatomic distance of zero; this will cause the simulation to crash as soon as it attempts to divide by this value. In practical terms, it can be argued that a fatal error is in fact the more preferable scenario since the user is immediately alerted to a program crash; it would be worse to waste time and resources producing a ‘surviving’ but nevertheless unrealistic (and unusable) simulation. This can be avoided by keeping the time steps over which calculations are made sufficiently small.

At stated, the smaller the time step, the greater the accuracy, but also the greater the number of successive solutions required to simulate for the same time. An optimum balance of accuracy and efficiency therefore exists (Figure 1.14B *cf.* Figure 1.14C). A popular rule of thumb is to allow around 10 time steps for the highest-frequency motion in the system. For biomolecules this is the C–H stretch term, which vibrates with a frequency of approximately 10 fs^{-1}, so a 1 fs step is commonly used.

This presents a problem when studying large biomolecules such as proteins, where far larger, lower frequency motions also occur, spanning timescales of over 15 orders of magnitude (Figure 1.15). A very large number of integration steps must be calculated to produce a simulation of adequate duration to begin to capture these motions.

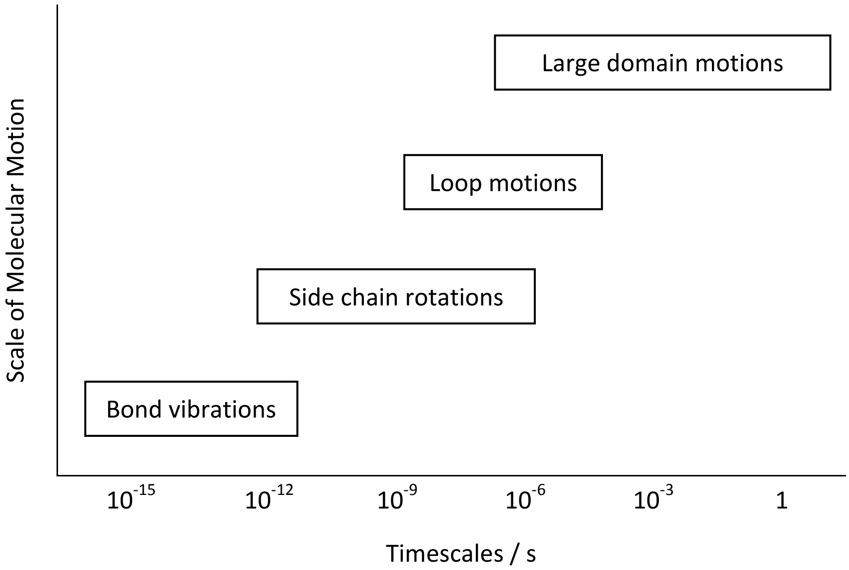


Figure 1.15: *Timescales of molecular motions in proteins. Adapted from* (87).

This means that any method allowing for a larger time step will significantly speed up the rate of MD simulation. Such a method has indeed been developed, named the SHAKE algorithm(88). This works by constraining bonds to their equilibrium lengths, and is usually applied to all covalent bonds to hydrogen since these are the fastest fluctuating. This allows the time step to be raised to 2 fs, halving simulation time. For protein simulations, the use of the SHAKE algorithm has become standard practice, and it was used in this manner for all simulations in this work.

It is worth noting a 2015 publication that presents stable results for a method, implemented in AMBER, that raises the MD time step to 4 fs(89), immediately halving the time taken to produce a simulation again. In the future, as more such evidence accumulates in the literature, it is likely that the 'standard' MD simulation of the type run in this project will move beyond the SHAKE algorithm to utilise methods such as this.

1.3.6 GPU Acceleration

The harnessing of the graphical processing unit (GPU) for rapid numerical calculations has significantly boosted the capabilities of MD(90–92). The boost is of such scale that a single, affordable desktop computer can perform simulations at speeds previously only attainable by high-performance computing (HPC) clusters.

GPUs are a particularly noteworthy advance for the MD community because, despite their exceptional computing power, they remain relatively affordable. This is because their development is aggressively backed by the multibillion-dollar computer gaming industry. The result is not only a market where GPU cards are improving quickly and constantly, but also one where they are priced for individual consumers rather than for research institutions. In terms of 'bang for buck,' the GPU is a far superior platform for performing MD simulations than a standard central processing unit (CPU). Figure 1.16 demonstrates this, displaying benchmarked performances of various modern GPU cards on an example molecular system against a 20-core CPU. A CPU with this many cores is a specialist piece of equipment that most laboratories will not possess. For comparison, the vast majority of personal computers today contain a GPU of some description.

This figure is regularly updated as new GPU cards are released and can be found at http://ambermd.org/gpus/benchmarks.htm. The GTX series in particular, which is primarily marketed (and priced) for computer gamers, still matches the performance of the more expensive Kepler series (cards beginning with 'K' in Figure 1.16), which is marketed to researchers for numerical calculations.

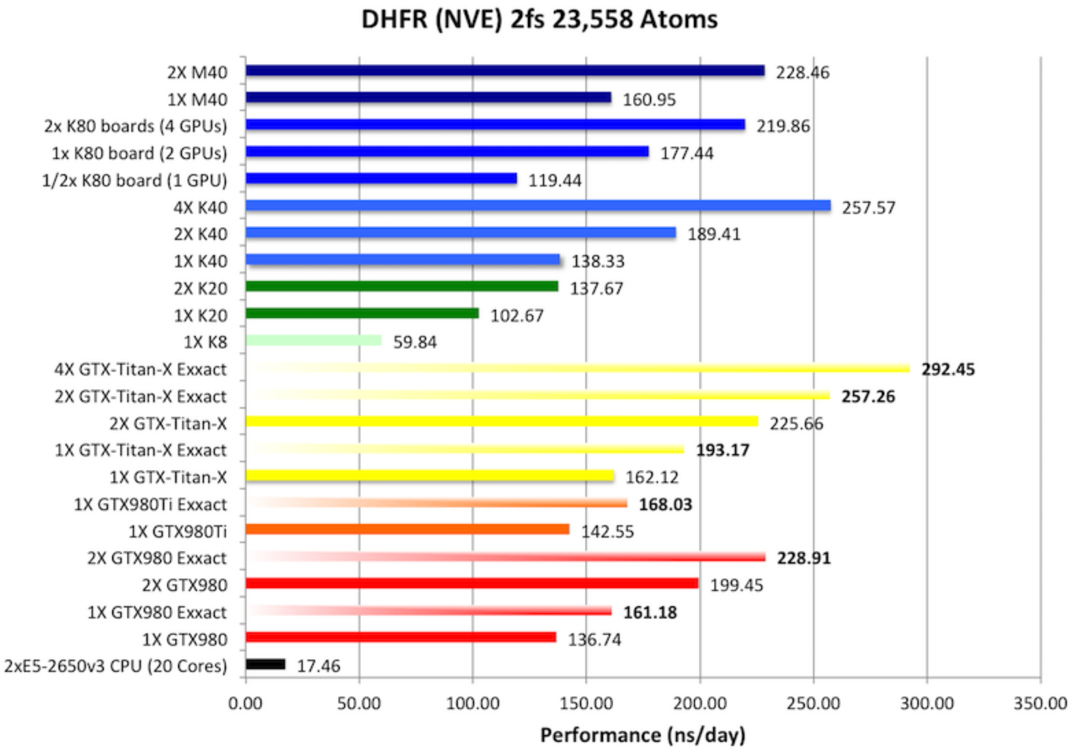


Figure 1.16: *Performance of various hardware setups for the simulation of the same molecular system in AMBER. The chart's bars are ordered by processor release date and number of processors in the setup. The GTX series in particular, which is primarily marketed for computer gaming, still matches the performance of the (more expensive) Kepler series. More significantly, a single modern GPU can perform MD simulations at nearly an order of magnitude greater speed than 20 CPU cores. This figure is regularly updated as new GPU cards are released and can be found at http://ambermd.org/gpus/benchmarks.htm.*

Even the GTX980 series GPU, the oldest in the list (though still a high-performance GPU in the gaming community at the time of writing), outperformed the CPU cluster in testing by nearly an order of magnitude. GPUs can be used in parallel, as they were in many of the displayed benchmarks, though they are far from 100% efficient. However, for producing many independent simulations – as was required for this project – a small number of GPUs can match and even outperform a HPC cluster.

1.4 Random Forest

This section begins with a generic introduction to machine learning before focussing on Random Forest (RF), the particular technique used in this project.

1.4.1 Machine Learning

Machine learning is a field concerned with making decisions or predictions based on supplied data. There are both significant similarities and significant differences between machine learning and statistics(93) – an area with which the reader is likely more familiar. There is no fundamental distinction between the two; indeed, statistics shares the goal of making decisions or predictions based on supplied data with machine learning. The two are better segregated by their origins(94): statistics is a long-established field of mathematics, while machine learning emerged more recently as a field of computer science. They can be thought of as alternative approaches to the same problems developed by people with different areas of expertise.

The following is perhaps the closest to a dividing line between statistics and machine learning: whereas statistics applies explicit functions to a dataset in order to solve defined mathematical equations, machine learning applies an algorithm that operates on-the-fly, with the form of the final model adapting to each specific dataset. In other words, statistics fits data to a pre-defined mathematical model. If the chosen model is appropriate and aligns well with reality, there is a high chance that the statistical analysis will yield accurate decisions or predictions. Machine learning, on the other hand, fits a model to data; only the algorithm by which it achieves this is pre-defined.

Machine learning methods are thus defined by the algorithm that is performed on a given dataset. A range of such algorithms have been developed, such as support vector machines(95) (SVM), artificial neural networks(96), genetic algorithms(97), fuzzy logic(98) and random forest (RF), the latter of which was used in this project and is covered in detail in the following sections. No algorithm is perfect, and its performance varies depending on the dataset it is put to work on: a quick search of the literature reveals that no method outperforms all others for all datasets, with different comparative studies each finding that different algorithms performed best(99–105).

RF was chosen as the machine learning algorithm for this project, though there is no reason why another machine learning technique could not have been applied in an analogous manner. There are several intricacies to RF that were advantageous for this work, although it is worth noting an additional, more pragmatic reason for its selection: the group has found historical success with RF in a variety of contexts(106–108) and a retains a relatively high level of expertise within it. This project was highly multi-disciplinary, requiring considerable knowledge of medicinal chemistry, structural biology and molecular modelling as well as machine learning. Initial support in the understanding and operation of RF was invaluable and saved a significant amount of time.

1.4.2 RF Background

Random forest (RF) is a machine learning algorithm first proposed in 2001 by Breiman(109). It is used to develop predictive models for tackling classification and regression problems. The terms *classification* and *regression* are used commonly in the context of predictive modelling. A classification problem is one where data, based on known variables, must be assigned labels from a limited, pre-defined set, while a regression problem requires the data to be assigned numerical values.

Before examining RF in detail, a brief introduction to some concepts of predictive modelling is presented, facilitated by an example dataset: Fisher's Iris data(110). It is a simple table containing information on a collection of flowers of the *iris* genus, and comes pre-loaded with the freely available R statistics package(111), used heavily throughout this work. This dataset is widely used in tutorials on classification and machine learning, and is a good way to introduce the concepts described before applying them in the context of allosteric sites.

The Iris dataset contains 150 samples of 3 species of *iris* (50 of each): *iris setosa*, *iris versicolor* and *iris virginica*. For each flower there are 4 data points: the measured lengths and widths of its petals and sepals. The dataset can be said to contain n cases or observables, where n = 150, and p descriptors or variables, where p = 4 (sepal length, sepal width, petal length and petal width). There is also a classifier variable which takes one of 3 independent, categorical values: the species of each flower. A sample of the dataset annotated with the above information is presented in Table 1.1.

descriptors/variables

classifier

cases/observables

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.3	3.3	6.0	2.5	*I. virginica*
5.8	2.7	5.1	1.9	*I. virginica*

Table 1.1: *A 6 case sample of Fisher's Iris dataset, containing 4 features – sepal length, sepal width, petal length and petal width – and a classifier: the species of each flower. The full set contains 150 cases: 50 of each species.*

Attempting to predict the species of another iris by comparing its petal and sepal dimensions to those in the dataset is an example of a classification problem. The four descriptors of the flowers can be considered predictor variables, or simply predictors, *i.e.* they are the variables upon which a prediction is based. The classification of the flowers (*i.e.* the species) is the response variable, or simply the response.

A regression problem is one where a continuous numerical response is desired instead of a categorical one, for instance the height of the flower stem instead of its species.

Used to solve both classification and regression problems, RF operates by generating an ensemble of independent decision trees (or a forest), each with elements of randomness incorporated. A query can then be run through each of the trees and the results fed back as a prediction. For classification problems, the majority vote of all the trees is taken as the prediction by default. For regression problems, the arithmetic mean of predictions across all trees is taken. This work approached the task of predicting allosteric binding sites as a classification problem, categorising each examined amino acid as either part of an allosteric binding site or not – in other words, a response with two categories: *true* or *false*. For this reason, RF will be described in a classification context here. The core algorithm employed by RF is detailed below.

1.4.3 RF Algorithm

The RF algorithm is described in four mains steps. Initially, a complete $n \times p$ dataset is required, where n is the number of cases and p is the number of predictors. A response variable y is also required for each case.

1. Draw a 'bootstrapped' sample of the training set with replacement

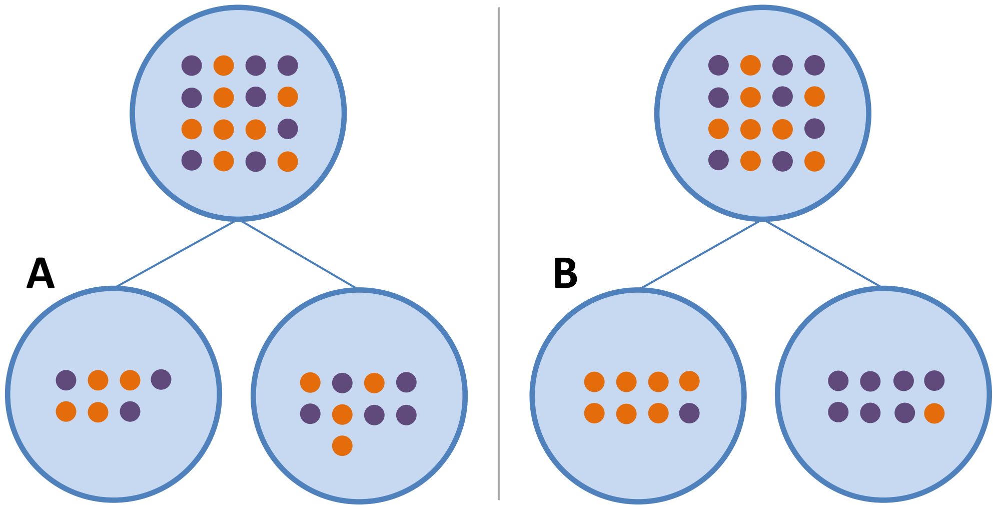
Bootstrapping is the process of drawing a randomised subset of data. During the construction of a bootstrap, a single datum is sampled and returned to the original pool. The process is repeated until the desired bootstrap sample size is reached. In this way, it is highly probable that the bootstrapped sample contains some duplicated data points and some that have been left out; it is a random permutation of the original data.

2. Optimally split the sample into two subsets by the best of m_{try} randomly selected descriptors, where $m_{try} << p$

For classification, the default and generally optimal value of m_{try} is $\sqrt{p}$. For each selected descriptor, the data are split by applying a threshold value. The quality of the split is then appraised (more detail on this criterion given below) before repeating the split with every possible threshold. The best split for each descriptor is determined, and the overall best split across all descriptors retained.

The criterion for determining the quality of a data split is the Gini impurity. This measure, derived from the Gini Index used to measure inequality in a society(112), quantifies the relative proportions of data classes in a sample. The Gini impurity for a set of data is equal to 1 minus the sum of squares of the relative class proportions within it. The quality of a data split is determined by the sum of Gini indices of each child node, each weighted by the proportion of the dataset they represent. Figure 1.17 shows a node containing data points of two classes (orange and purple dots), with two different splits, **A** and **B**, performed on it.

Calculations for the relevant Gini impurities are shown to illustrate the concepts described here.



$$\textit{Gini impurity} = 1 - \left(\left(\frac{class\ 1\ data\ points}{total\ data\ points}\right)^2 + \left(\frac{class\ 2\ data\ points}{total\ data\ points}\right)^2\right)$$

$$\begin{aligned}\textit{Gini impurity at parent node} &= 1 - \left(\left(\frac{8}{16}\right)^2 + \left(\frac{8}{16}\right)^2\right)\\ &= 1 - 0.25 - 0.25\\ &= 0.50\end{aligned}$$

*Gini impurity of split **A***

$$\begin{aligned}&= Gini(child\ 1) \times \frac{7}{16} + Gini(child\ 2) \times \frac{9}{16}\\ &= 0.489 \times \frac{7}{16} + 0.493 \times \frac{9}{16}\\ &= 0.214 + 0.277\\ &= 0.491\end{aligned}$$

*Gini impurity of split **B***

$$\begin{aligned}&= Gini(child\ 1) \times \frac{8}{16} + Gini(child\ 2) \times \frac{8}{16}\\ &= 0.218 \times \frac{8}{16} + 0.218 \times \frac{8}{16}\\ &= 0.109 + 0.109\\ &= 0.218\end{aligned}$$

Figure 1.17: *A node of data points is depicted, with the two classes coloured purple and orange. Two splits, A and B, have been performed on the parent node. Split B yielded a lower Gini impurity and so is considered the superior split.*

3. Repeat the splitting procedure with the subsets of cases at each node until full length trees form, *i.e.* single classes populate the terminal nodes

4. Generate new decision trees (*i.e.* repeat steps 1-3), each with a new bootstrapped sample until n_{tree}, the specified number of decision trees to be generated, has been reached.

With a 'forest' of decision trees generated, an unclassified case can be passed through the model, with each tree funnelling it through decisions to a terminal node associated with a class label. Each tree votes on the case's class, with the majority vote constituting the model's prediction. Alternatively, one can retain the proportions of votes and yield predicted probabilities of class membership for each case.

1.4.4 Features of RF

- **Internal Validation**

There are distinct advantages to RF that make it highly suited for the data involved in this work, the first being its ability to perform internal validation. Since any one decision tree in the forest is built only upon a bootstrap of the dataset, cases that remain excluded from the bootstrap – so called *out-of-bag* (OOB) data – are used to conduct on-the-fly testing, storing the percentage of misclassified cases. As standard, this OOB error rate is calculated for every tree and aggregated, providing a good estimation of the model's overall performance as it is generated.

The OOB error can be used to monitor the growth of a RF model. As the number of trees is increased, the OOB error initially decreases sharply. This is due, on one hand, to the inherent weakness of any single semi-random decision tree's ability to perform as a classification model, and on the other to the great 'wisdom of crowds' benefit from considering all trees together as an ensemble. The OOB error tends to level off after sufficient trees have been added. For many datasets, the accepted default value of 500 trees has proven sufficient to reach this point of convergence, while others have required more. There are diminishing returns to be gained from increasing the number of trees further beyond this point(113).

The OOB validation also alleviates the burden of validation from a separate dataset. Usually, a predictive model must be validated against a suitably sized set of external data; while there is no gold standard, a typical size for this is 20% of the training set. Creating such a dataset requires a portion of training data to be set aside and so can no longer be used to train the model. With RF, though the whole training set is utilised overall, each individual tree is validated against its OOB data. External data partitioning is therefore not critical, allowing all of the available data to be used either for training or for carrying out a true, unseen test for confirmation of performance. In this project, where data points were both scarce in quantity and laborious to accumulate, this was a significant advantage.

- **Variable Importance**

It is possible to obtain a measure of the importance of each variable in a RF model to its performance. The premise of the measure is to adopt a null hypothesis at each node in the decision tree: if the chosen variable is a weak predictor, then randomly rearranging its values – a procedure which severs any link between the predictor and the response – will only weakly affect the accuracy of the model's predictions. The more important a variable is to the model's performance, the greater the reduction in accuracy upon permutation. This analysis is performed as the model is generated, using the OOB data to determine prediction accuracy before and after permutation, subtracting the latter from the former and averaging across all trees.

Care is required in the interpretation of this analysis(114). The hypothesis not only assumes null correlation between the predictor and the response, but null correlation between the predictor and other predictors. This results in variables that are correlated to truly important ones also yielding a high mean decrease in accuracy.

The analysis is also deceptive in that it cannot be reliably used to test the consequences of removing a variable from the training dataset, despite this being in essence what the analysis does. The term 'variable importance' suggests that any variable with a non-zero score is of value: removing it from the dataset would result in a drop in model performance. In reality, removing data can be beneficial to model performance, as was found in section 4.7.1. The reason for this discrepancy is that, if a model was retrained with a variable removed from the dataset, each node that would have used it to make its decision would instead select a different real variable, rather than a make its decision based on a permuted variable akin to pure noise.

- **Partial Dependence**

This is a method for quantifying the effect of a given variable on a model's positive class probability; in other words, how much more likely a *true* prediction becomes as the given variable changes value. This is usually presented as a graph of so-called *partial dependence* (PD) on a given variable plotted against the variable's values in ascending order. An example graph of this type is presented in Figure 1.18. At the same time, it is useful to note the overall 'shape' of the variable, *i.e.* what proportion of it takes on what values. The decile ranges of the example variable are marked in Figure 1.18.

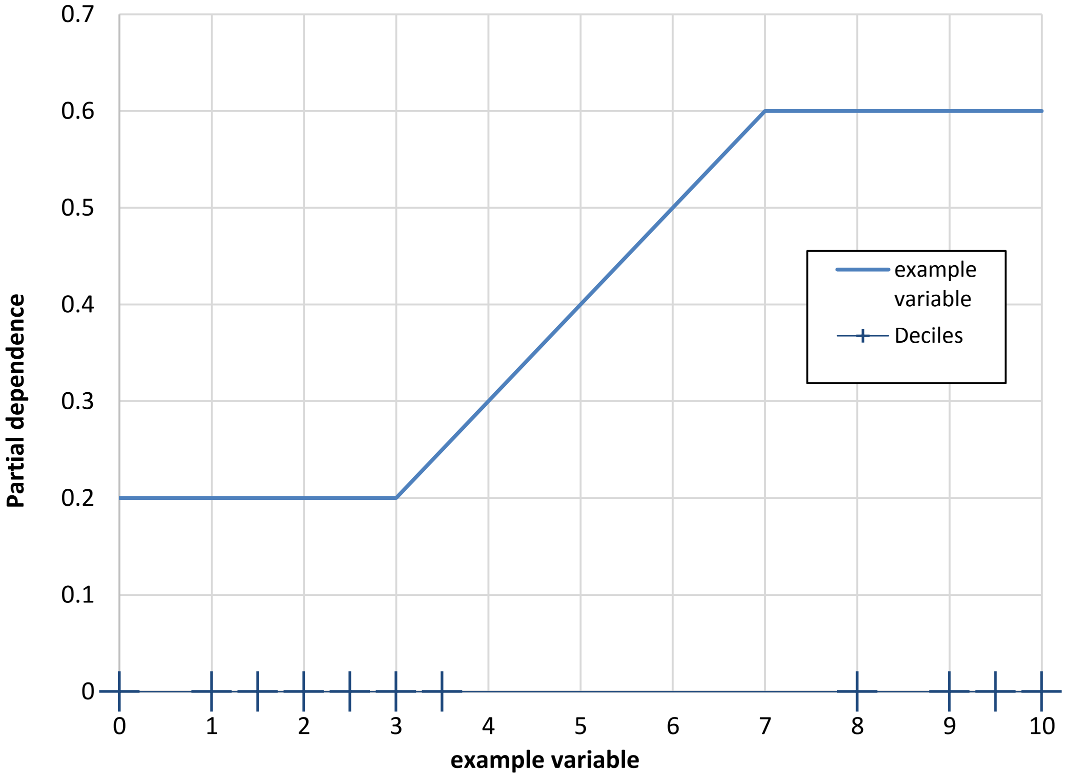


Figure 1.18: *An example partial dependence plot. The top 3 deciles correspond to a higher PD from the model, meaning that high values of this example variable are more likely to result in a true classification.*

This can provide great insight into the real information a model is detecting from a variable. In the example it can be seen that the top 3 deciles of the variable attracted a greater PD from the model. The conclusion to be drawn from this would be that higher values of this variable are a marker of *true* classification.

- **Proximity Measure**

This calculation determines the frequency with which a pair of cases reaches the same terminal node in all trees of a RF model. This type of data can be considered as a kind of 'distance' between each pair of cases, with closer (*i.e.* more similar) ones having a greater proximity value. Using multi-dimensional scaling (MDS) the data can be reduced to 2-3 dimensions, allowing the user to visualise the proximities of all cases at once. This is a powerful method for determining how well the model was able to separate the classes. However, it requires the pairwise comparison of all n cases in the training dataset, which in turn requires the generation of a $n \times n$ matrix. This requires exponentially greater computational expense to calculate as the value of n increases; for this project, where n = approximately 32 000, it proved unfeasible to perform on all models and was reserved only for the final one.

- **Stability**

A more general advantage of RF lies in its robustness to noise in the training data relative to other models. It is apt to say that a RF model is "evolved" rather than fitted. Instead of handling all predictors at once, each decision tree in RF operates by looking at a small, randomly selected subset of predictors at each node, ignoring the rest. For each predictor, the splitting point yielding the optimum Gini impurity is calculated, and the one producing the lowest of these is taken forward. A small gain is made, and the process is repeated. The predictors can be thought of as constantly competing for selection in the micro-environment of each node. Noisy predictors, less able to split the data into the correct classes, are naturally weeded out as the tree is constructed. Including noisy predictors in the training set does not drastically affect the model's ability to isolate the most useful ones; that is, unless there are so many noisy predictors that the useful ones are rarely/never selected. This innate resilience of RF allowed for a less stringent approach to variable inclusion for the project – a very welcome notion when it was not known in advance what data would prove useful.

As a caveat to the above it should be made clear that, while RF is highly capable of handling variable noise, it is can still be affected by variable correlation. Highly correlated variables contain the same or similar information; a RF model based on a highly correlated dataset will comprise more nodes splitting on what is effectively the same real information. This leads to an artificially high emphasis being placed on the correlated data for their actual predictive capability.

1.4.5 Method Optimisation

RF has only a small number of parameters that are modifiable by the user, and default settings for these have already been well established that are often optimal(109, 115). However, these defaults were reached using datasets of roughly equal balance: that is, datasets containing a similar number of cases from each class (such as Fisher's Iris dataset, containing 50 cases of each species). Machine learning methods, including RF, tend to suffer a drop in performance when dealing with highly imbalanced datasets(116).

The dataset developed for this project contained a large class imbalance, with approximately 95% of residues being classed as *false*. This meant that a model could achieve a formal accuracy of approximately 95% by simply classifying all cases as *false,* despite being entirely useless. For this reason, an optimisation of the RF parameters was required for this project. Accuracy, along with many other evaluation measures, are described in more detail in section 1.4.7.

1.4.6 Dealing with Class Imbalance

There is a consensus in the literature that one's approach to RF should be modified in cases of high class imbalance(116–118), but no standard procedure is known that benefits all situations(116). The imbalance can be dealt with by applying a weight to the minority class, known as class-weighted RF(117). A class-weighted RF is constructed in the same manner as a standard RF, but with a modification made to the splitting procedure at each node, where the penalty to the Gini criterion for misclassifying the minority class is increased. A second weight is applied when the forest comes to vote on the classification of an unknown case: the number of votes for the minority class is multiplied by a constant, resulting in fewer trees needing to vote for the minority class to gain an overall majority. This leads to more cases being classified as the minority class.

Another method for addressing imbalanced datasets is balanced RF(117). Before growing a forest of decision trees, cases are artificially added to or removed from the training set, known as over-sampling and down-sampling, respectively. When over-sampling, identical copies of random cases of the minority class are created and added to the training set until the desired class balance is reached. This results in a training set with multiple identical cases, which could perturb any underlying patterns in the data. To minimise this problem over the forest as a whole, over-sampling is performed individually for each decision tree, smoothing out the number of copies made of each case.

Alternatively, the class balance can be redressed with down-sampling. Here, the minority class is sampled once in its entirety and cases of the majority class are randomly sampled (without replacement) until the desired class balance is reached. This results in a training set that is both balanced and composed of unique cases, unlike over-sampling, but also in a training set that does not contain all of the available cases of the majority class. As with over-sampling, the training set is reconstructed for each decision tree, so the over- and under-representation of cases over the whole forest is again minimised.

Though both tend to boost the performance of 'naïve' RF with highly imbalanced datasets, down-sampling has been shown to generally outperform over-sampling(117). Neither weighted RF nor balanced RF (by down-sampling) has been shown to be the superior technique over the other. However, balanced RF has the added benefit of being more computationally efficient, since each decision tree requires only a proportion of the whole training set to grow. For this reason, balanced RF was chosen as the methodology to be applied in this project.

1.4.7 Quantifying Predictive Power

The OOB error rate of a RF model is a highly convenient and robust method for quantifying predictive power, but this is a special measurement unique to RF. There are many other methods for achieving this. The results of any classification prediction can be summarised as a table of correctly- and incorrectly predicted class counts. This is termed a confusion matrix, and is of the form shown in Table 1.2.

		Predicted Result	
		False	**True**
Observed Result	**False**	a	b
	True	c	d

Table 1.2: *The form of a confusion matrix – the most compact output format of a classification prediction.*

Ideally, $b = 0$ and $c = 0$ in the above table, *i.e.* the class of all cases is correctly predicted.

There are many evaluation measures available to quantify the predictive power of a classification model that are functions of the numbers in Table 1.2. None are considered perfect for comprehensively capturing the information contained in the confusion matrix and so are generally not considered in isolation(115, 117, 119, 120). In fact, there is no standard evaluation technique for classification models; instead, analyses are tailored to the context of a model's application, using whatever measures are deemed suitable. Nevertheless, the most well-known measures of this type remain a good starting point, and are described below.

- **Precision**

$$Precision = \frac{d}{d + b} \tag{3}$$

Precision calculates the proportion of cases correctly called true out of the total number of cases predicted *true*(119, 120). It ranges from a minimum value of 0 when d = 0, *i.e.* no correctly predicted *true* cases, to a maximum of 1 when b = 0.

- **Recall / sensitivity / true positive rate (TPR)**

$$Recall\ OR\ Sensitivity\ OR\ TPR = \frac{d}{d+c} \tag{4}$$

This measure, with different names in different contexts, calculates the proportion of cases correctly called *true* out of the total number of *true* cases(119, 120), with minimum and maximum values of 0 and 1, respectively. Note that the false positive rate (FPR) can be determined by taking 1 – TPR.

- **Inverse recall / specificity / true negative rate (TNR)**

$$Inverse\ recall\ OR\ Specificity\ OR\ TNR = \frac{a}{a+b} \tag{5}$$

Equivalent to TPR but operating on the negative class, this measure compares the number of the cases correctly called *false* to the total number of *false* cases(120), with minimum and maximum values of 0 and 1, respectively. Note that the reciprocal false negative rate (FNR) can be determined by taking 1 – TNR.

- **F measure**

$$F\ measure = \frac{(1+\beta^2) \times Precision\ \times Recall}{(\beta^2 \times Precision\ ) + Recall} \tag{6}$$

The F measure was devised to find a compromise between precision and recall(119, 120). The β value is treated as a factor applying relative importance on recall against precision. The formula is balanced when β = 1 and corresponds to the harmonic mean of precision and recall. Its values range from 0 to 1.

Since the above measures do not incorporate all 4 values a, b, c, d they fundamentally cannot fully characterise the confusion matrix in isolation. The following measures do:

- **Accuracy**

$$Accuracy = \frac{a+d}{a+b+c+d} \tag{7}$$

Accuracy is the ratio of correct predictions over all predictions. It incorporates all four values of the confusion matrix into a single descriptor, but can be wildly misleading for imbalanced sets(121). For example, if real *true* cases made up a twentieth of the dataset, as they do in this project, a model could class every single case as *false*, and thus be completely ineffective, yet 95% accurate.

- **G mean**

$$G\ mean = \sqrt{TPR \times TNR} = \sqrt{\frac{d}{d+c} \times \frac{a}{a+b}} \tag{8}$$

Since b and c represent the number of incorrectly classified cases, the larger the G mean of the confusion matrix, the greater the predictive power of the model. Note that this is the geometric mean of TPR and TNR, a measure recommended by Kubat *et al.*(122), rather than the similarly named G-measure(120), which is the geometric mean of precision and recall, and gives similar values to the F measure.

- **Cohen's kappa**

$$Cohen's\ kappa = \frac{\frac{a}{t} + \frac{d}{t} - \left(\frac{a+b}{t} \times \frac{a+c}{t} + \frac{c+d}{t} \times \frac{b+d}{t}\right)}{1 - \left(\frac{a+b}{t} \times \frac{a+c}{t} + \frac{c+d}{t} \times \frac{b+d}{t}\right)} \tag{9}$$

where $t = a + b + c + d$. Cohen's kappa(123) was designed to measure the agreement between two models. Used in this context the 'agreement' is between prediction and reality. It also incorporates chance, *i.e.* it accounts for class-imbalanced scenarios such as this one, where a high agreement on false cases is likely through chance. It ranges from -1 to 1, where a value of 1 indicates perfect agreement and a value of -1 indicates perfect disagreement. In the context of a predictive model, the latter case would indicate that all cases are being classified oppositely, in which case one could simply flip the class labels and achieve a perfect prediction. The true minimum in terms of predictive power is therefore 0, indicating purely random agreement. There has been some dispute over the usefulness of Cohen's kappa(124, 125), with some believing it to underestimate agreement in many cases. However, it can be accepted that a high Cohen's kappa value corresponds to high agreement.

- **Matthews Correlation Coefficient (MCC)**

$$MCC = \frac{ad - bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}} \tag{10}$$

First developed by Matthews(126), the MCC is considered among the best evaluation measures of its kind. It accounts for class imbalance, like Cohen's kappa, but is not associated with the same underestimation of agreement. Matthews himself used it for comparing predicted and observed secondary protein structure; it has since been used widely in the scientific community in many different contexts(127–129). Like Cohen's kappa, it ranges from -1 to 1, where a value of 1 indicates perfect agreement and a value of -1 indicates perfect disagreement.

- **Receiver operating characteristic (ROC)**

First used during World War II by British radio operators to help discriminate between random interference and signals due to approaching German bombers, the receiver operating characteristic (ROC) curve has since been applied in many scientific fields with great success. Triballeau *et al.* (and references therein) provide an excellent introduction to the ROC curve method in the context of drug discovery(130).

To produce a ROC curve for a given set of predictions, one must have a numerical value associated with each prediction signifying the probability of its positive classification. In the context of RF, this information is available in the form of the proportion of votes cast by decision trees in favour of a *true* classification. These numbers are readily extracted for each predicted case. At this point, a cut-off value is introduced, above which all cases are classified as true. Clearly, if this value is above the highest proportion of *true* votes, all cases are classified as *false*. A confusion matrix can be plotted for this, yielding a TPR of 0 and a TNR of 1. The cut-off value is then incrementally lowered until a case is classified as *true*. A new set of TPR and TNR values can be now be calculated. The process is then repeated until all cases are finally classified as *true* (TPR = 1, TNR = 0). Each instance of the confusion matrix is then plotted in a graphical space of TPR against FPR (FPR = 1 – TNR), yielding the final ROC curve. An example curve is shown in Figure 1.19.

In this space, all curves necessarily begin at the origin and proceed to a final point at (1,1). A perfect prediction extends vertically to the point (0,1). A straight, diagonal line through the graph corresponds to random noise, *i.e.* there is a completely zero-sum trade-off between sensitivity and specificity. As with negative MCC and Cohen's kappa values, any line curving below the diagonal is an anti-predictor; one can flip the class labels and thus flip the ROC curve into a 'positive' one above the main diagonal.

The predictive power represented by ROC curves can be elegantly quantified by calculating the area under the curve (AUC). A perfect prediction therefore has an AUC of 1, and completely random prediction has an AUC of 0.5. The example model in Figure 1.19 has an AUC of approximately 0.8.

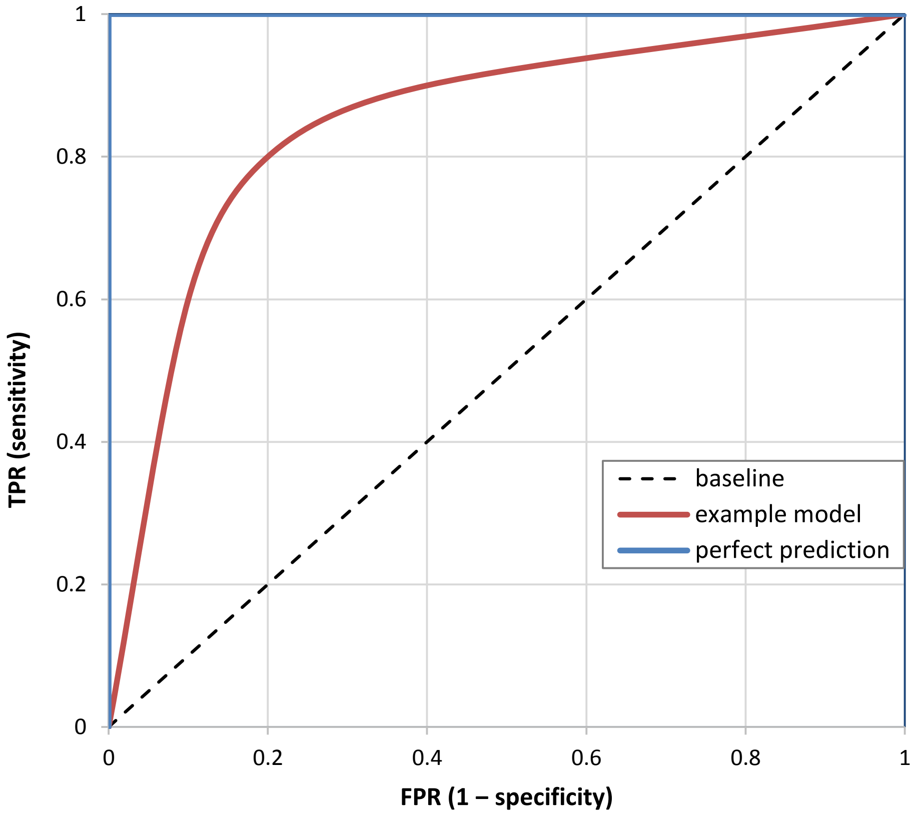


Figure 1.19: *An example ROC curve (red). A perfect prediction (blue) produces a curve reaching (0,1).*

The other analyses described in this section have utilised only a single confusion matrix with a fixed threshold value of 0.5 defining the class of each case (*i.e.* a majority of RF trees needed to classify). While this can often be a sensible threshold to use, it is not necessarily optimal. The ROC curve, in essence, monitors the changes in the confusion matrix as this threshold is dialled from 1 to 0. A sharp increase in sensitivity indicates a good threshold; for the example model plotted in Figure 1.19, the threshold used at (0.2, 0.8) would appear optimal. This type of information is highly informative, and is also captured to an extent by the AUC. In this project, the ROC was used extensively to track the performance of model iterations as more data became available.

For all of the discussed measures that consider both classes (precision, TPR and TNR only consider one) there is an important caveat: both classes are treated with equal importance. In other words, the same penalty applies for misclassifying a *true* result as a *false* result. Depending on the context of a prediction, this may not be desirable or appropriate. For example, in document retrieval (such as Google searching), one correct hit is all that is needed. If the top result of a search is correct, it does not matter if the next hundred results are totally irrelevant. The user will find what they are looking for and the document retrieval can be considered a success.

1.4.8 Allosteric Context

For this project it was important not to lose sight of the overall goal when analysing predictive models. Rather than becoming mired in imperfect statistical measures, it often proved more prudent to judge the quality of a prediction by manual, visual inspection. Results were often mapped onto a 3D image of the protein, rendering residues predicted as *true* and *false* in different colours. The allosteric modulator could also be included for test cases where its binding mode and location was already known. A glance from a trained eye at such an image invariably proved enough to estimate the quality of a prediction without calculating any evaluation measures. This is exemplified in Figure 1.20 with UDP-glucose dehydrogenase (PDB code: 3PJG), an arbitrarily chosen protein with a known allosteric inhibitor. Two hypothetical predictions are mapped onto the protein and, without applying any evaluation measures, it is clear that model **B** is the stronger predictor of the allosteric site, since only it classified a concentrated patch of residues in the vicinity of the allosteric ligand as *true*.

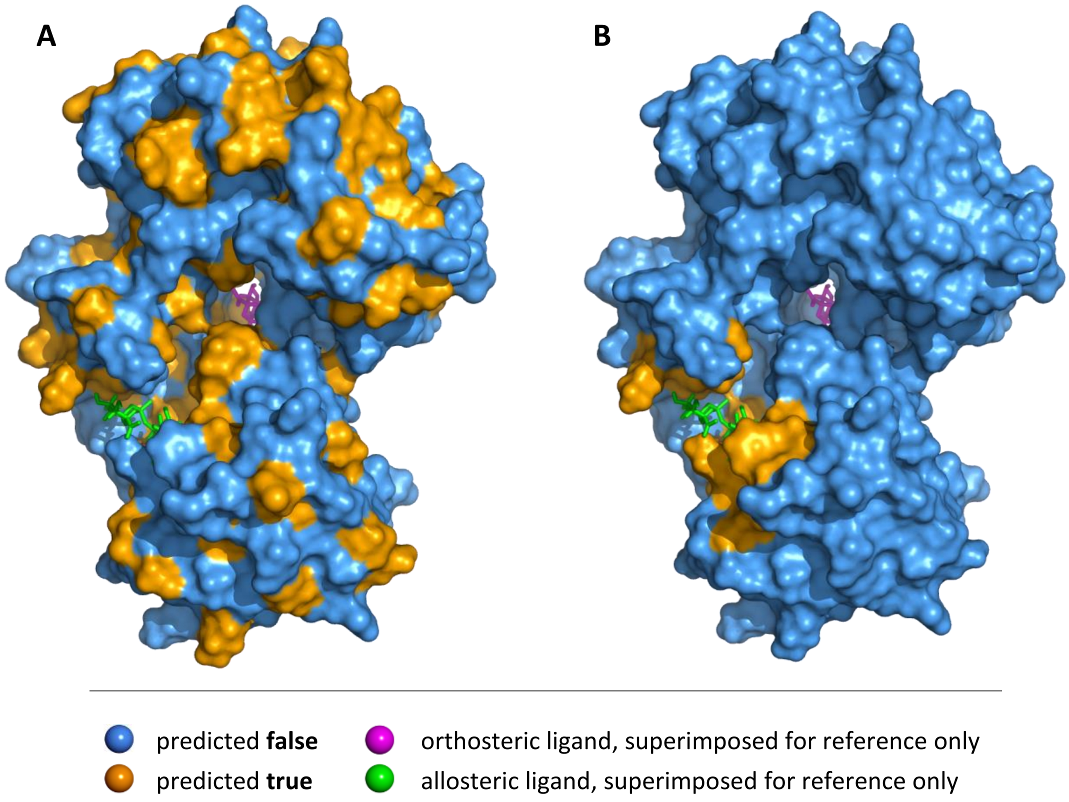
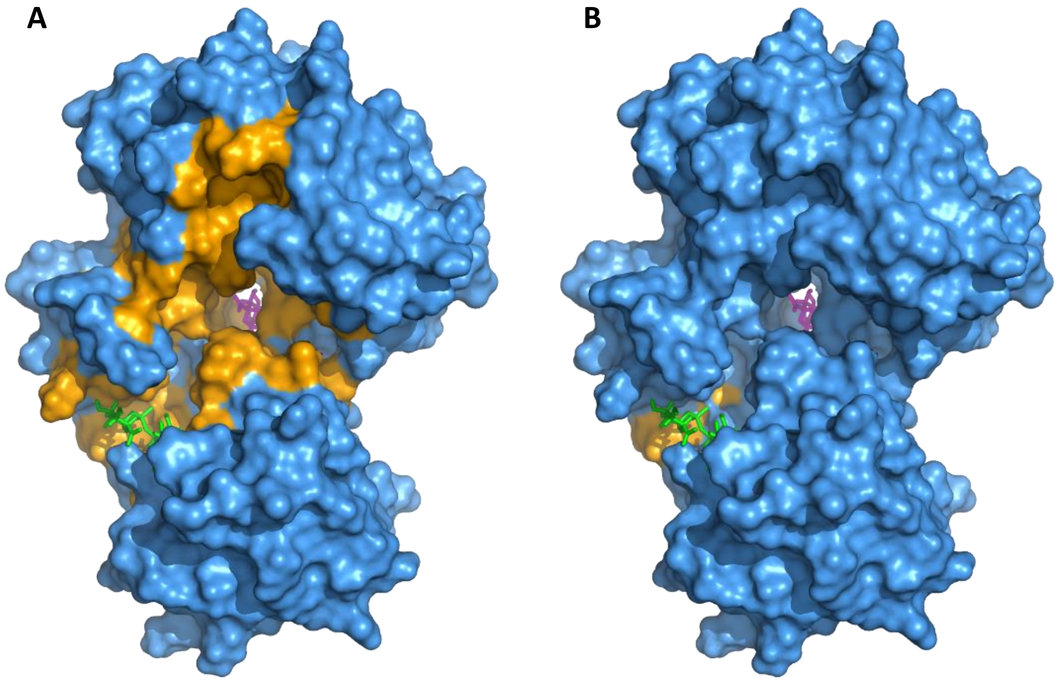

A
B
predicted **false**
predicted **true**
orthosteric ligand, superimposed for reference only
allosteric ligand, superimposed for reference only

Figure 1.20: *Two hypothetical predictions,* ***A*** *and* ***B****, of allosteric site locations for UDP-glucose dehydrogenase are mapped on to the protein. No evaluation measures are presented for these predictions to illustrate that it is clear by eye alone that model* ***B*** *is a significantly stronger predictor than model* ***A****.*

The goal of the model was to provide a medicinal chemist with a site to target: flagging every last residue within a pocket as true was not critical, so long as some were highlighted and few others in the protein were. In some cases, a statistically poor prediction could be enough to point the chemist to the correct site, for instance, a sole residue being (correctly) flagged as true in the entire protein. Once more using hypothetical predictions mapped on to UDP-glucose dehydrogenase, Figure 1.21 demonstrates this scenario where the statistically poorer prediction is, at least arguably, the more useful one.



G mean =	0.826	G mean =	0.277
Cohen’s kappa =	0.192	Cohen’s kappa =	0.138
MCC =	0.296	MCC =	0.273

🔵 predicted **false** 🟣 orthosteric ligand, superimposed for reference only
🟠 predicted **true** 🟢 allosteric ligand, superimposed for reference only

Figure 1.21: *Two hypothetical predictions **A** and **B**, of allosteric site locations are presented for the same protein. Model **A** highlights many residues near the correct site, but also many other areas of the protein. Model **B** has only selected a single residue and is the statistically weaker prediction, though it is arguably more useful in that it unambiguously leads the scientist to the correct site.*

In essence, a model can perform well against a battery of statistical measures without necessarily performing well when it comes to directing the scientist to an allosteric site, and *vice versa*. This is simply a reflection of the fact that any statistical measure answers the highly specific mathematical question that its equation poses, regardless of what the human interpreter would perhaps like them to answer. However, these statistical measures can still serve as useful guides provided they are treated as such, rather than gold standards.

Figure 1.21 raises another pertinent issue, namely the problem of classifying residues in the orthosteric site. For the purposes of calculating the evaluation measures in Figure 1.21, residues in orthosteric sites were deemed to be *false*. Consequently, model **A** was statistically penalised for mostly classifying the orthosteric site as *true* and model **B** was equally enhanced (model **A** remained statistically superior in spite of this). Had the orthosteric site instead been labelled as *true*, the differences in performances would have appeared even starker, though clearly a more accurate picture lies somewhere in between. This issue is discussed further below.

1.4.9 Classifying Orthosteric Sites

Throughout this work a binary classifier (*true*/*false*) was used to define individual protein residues as either allosteric or non-allosteric. In using a binary classifier, the problem of how to handle the residues of an orthosteric site arose. It was not quite satisfactory to classify them either as *true* – in other words, the same category as allosteric site residues – or as *false*, suggesting that they do not belong to a site at all. Treating them as *true* would result in RF models training to detect these residues as well as allosteric ones, when it is known that they do not have the same properties(131). Moreover, a predictive model could prove statistically strong without returning a single correct allosteric site, so long as it could still detect orthosteric sites. Equally, treating the orthosteric site as *false* would result in overly harsh statistics, since there is a clear difference between a model that highlights orthosteric sites and a model that highlights scattered, nonsensical residues corresponding to no site at all. Both choices had their drawbacks, but there remained an advantage to keeping the classification problem binary. Firstly, it allowed for the use of much simpler analyses in examining the results, besides which there was no guarantee that approaching the problem with three classes would yield an improvement. On the contrary, RF classification models tend to perform better with fewer classes(132).

More importantly than any of the above, one must consider the purpose of a model such as this. In any 'live' situation where an allosteric site is being sought for a given protein, one would almost certainly have knowledge of the orthosteric site's location. Any prediction made by a model highlighting residues of an orthosteric site can therefore be safely disregarded irrespective of the predicted classification: it is the one area of a protein that, by definition, cannot be allosteric.

The datasets of classification problems comprise two parts, predictors and response. Generally, the response is definitively valid and true. For example, in the context of the Iris dataset, there is no ambiguity surrounding the real species of each individual flower. They are known, having been identified manually. With a known response, one can apply machine learning, such as RF, to test the connection between it and the predictors. A weak prediction simply reflects the weak connection of the chosen variables to the response, and a strong one reflects the opposite. However, for this project, part of the work involved deciding upon what the response of the dataset should be. As is discussed in section 2.4.1, the chosen method was neither definitive nor flawless. In analysing the predictions of any constructed model, one had to be wary of the quality of the data being fed into it: if the response values of the dataset contained noise, a statistically strong predictive model could simply have been a strong predictor of noise. Equally, a model appearing statistically weak could in reality be performing fairly well.

1.5 Existing Methods for the Prediction of Allosteric Binding Sites

There is evident value in the ability to predict allosteric sites, and a vast range of such methods has been described in recent years. Generally, methods either treat proteins in terms of their sequence or structure. Sequence-based methods track properties such as evolutionary significance of sequence positions in phylogenetic trees(133) or Shannon entropy, a theoretical measure of information gain(134). Aside from MD, there are other methods that treat proteins in structural terms, such as the elastic network model (ENM)(135), which reduces protein structure to a series of nodes interconnected by harmonic springs, and the related normal mode analysis (NMA)(136), which assumes a harmonic potential energy landscape in the vicinity of the given structure. Both of these methods offer fast alternatives to the application of a more complex, computationally expensive MM-based force field. A recent review and references therein do a good job of capturing the current landscape of predictive methods(137).

Below is more detailed description of a selection of predictive methods that are more closely comparable to the work of this project. Specifically, they share a common starting point for a prediction: a ligand-free, PDB-formatted protein structure.

1.5.1 PARS

In 2012 Panjkovich and Daura developed a method for the prediction of allosteric sites in proteins based on flexibility(138). Recently, the authors published a web interface for open use of their methodology, named Protein Allosteric and Regulatory Sites (PARS)(139), available at http://bioinf.uab.cat/pars.

A set of proteins with known allosteric sites was gathered by the authors. NMA was performed on each protein in the apo state, using the results to derive B-factors as an estimation of protein flexibility. This procedure was repeated in the presence of the allosteric ligands, and the two sets of B-factors were compared. The procedure was considered to have detected an effect on protein flexibility due to the presence of the allosteric ligand if the majority of B-factors were found to be significantly different by a Mann-Whitney test.

To perform the same procedure with unknown allosteric sites, proteins were first filtered through LIGSITE^{csc}, a web server designed to detect pockets on proteins surfaces(140). Up to eight candidate pockets were docked with a dummy ligand in turn, calculating B-factors for each complex. Each set of B-factors was then compared to those of the apo protein as before.

The authors concede that this methodology relies on LIGSITE^{csc} to select the correct cavity, and that it failed to do this for over a third of their original set of 91 proteins. Their reported success rate of 65% is based on the remainder of the set.

As well as producing the B-factor comparison, the web implementation performs a complementary analysis measuring the structural conservation of each pocket across protein families. The rationale for incorporating this stems from earlier work by the group which found that structural conservation is prevalent in pockets across protein families – though to a lesser extent than orthosteric sites – and could aid in the identification of allosteric sites(141). The results are presented as two scores per candidate pocket: a p-value from the Mann-Whitney test and a percentage conservation score. The algorithm ranked the real allosteric pocket in the top position in 44% of the test set, and in the top three positions in 73% of the test set.

1.5.2 SPACER

In 2011, with the notion of population shift in mind, Mitternacht and Berezovsky proposed a method to quantify the coupling of the intrinsic motions of a protein to potential binding sites(142, 143). This procedure was later implemented on web server named SPACER (Server for Predicting Allosteric Communication and Effects of Regulation)(144), available at http://allostery.bii.a-star.edu.sg/.

Putative binding sites are first located by moving a probe ligand over the protein in a course-grained Monte Carlo docking simulation. Using a geometric measure called local closeness(145) to quantify the connectivity of nodes in contact with the probe, the results are then clustered to produce a refined list of sites. Binding leverage, defined as the energetic strain experienced by ligand-protein contacts due to the motions of low frequency normal modes (determined by NMA), can then be calculated. Binding leverage is intended to quantify the energetic cost of deforming a ligand-bound site: a high binding leverage indicates a site highly correlated with the conformational states being analysed – in this case low frequency normal modes, which are known to be relevant to allostery(146–148). No performance figures were presented for this analysis, since it used only for a single example case.

1.5.3 Allosite

In this work(149), descriptors from fpocket(150), a program for the detection and characterisation of cavities in proteins, were derived for a set of proteins with allosteric sites (fpocket is discussed further in section 2.4.12). The descriptors were calculated for all detected cavities in the set of proteins, classifying each as either an allosteric site or not. A support vector machine (SVM) – an alternative machine learning method to RF – was trained to classify data produced by fpocket and thus predict which cavities were allosteric sites. The authors report a success rate of approximately 83% on their test set, and have made their model available for use as a web server named Allosite, available at http://mdl.shsmu.edu.cn/AST/. A study by Warmuth *et al.* utilised SVM in a drug discovery context, and their publication contains a short description of the technique(151).

The proteins used with Allosite were taken directly from the Allosteric Database (ASD)(152), a database of proteins with known allosteric activity created by the same group. The ASD is discussed further in section 1.6.

Allosite is conceptually the closest to the work carried out in this project, in that it uses a machine learning algorithm to predict the location of allosteric sites. However, there are key differences between Allosite and this work. The full consequences of these differences are made clear in Chapters 4 and 5, where this project's results are presented and discussed. In particular, section 5.8 details the results of a test of Allosite's performance and compares it to the models generated over the course of this project.

1.5.4 Summary

There are similar concepts involved across the discussed predictive methods. All involve an initial detection of protein cavities/pockets, followed by some quantitative measurement. This is then compared either to the same protein's apo state or to a pre-existing training set. Of these methods, Allosite is most similar to this work since it also uses machine learning to class each tested cavity as allosteric or not.

An important distinction between the above methods and this work lies in the treatment of the initial protein structures. With the above methods, no alterations are made to the inputted structure, which tends to be, near invariably, a crystal structure. Certainly, the protein structures used to train and test these methods were collections of crystal structures originally complexed with allosteric modulators. While these modulators were deleted *in silico*, no steps were taken to address the perturbations to local protein structure. It is highly likely that these methods were simply detecting this 'imprint' left behind by the deleted ligand, rather than any underlying signal of an allosteric site. In any situation where a live prediction is made – that is, on a protein where no allosteric site is known – no such imprint would be present in the structure, and the model's performance would be hindered. Figure 1.22A illustrates this concept of ligand imprinting and how it can allow predictive models to 'cheat' by betraying the position of the allosteric site. Figure 1.22B inserts an extra step that is required to correct for this issue.

This work naturally avoided the obstacle of ligand imprinting by working with proteins' MD trajectories, rather than their original crystal structures. Before any data were derived for use in a predictive context, the proteins had been through a thorough minimisation and equilibration procedure, as described in section 2.3.

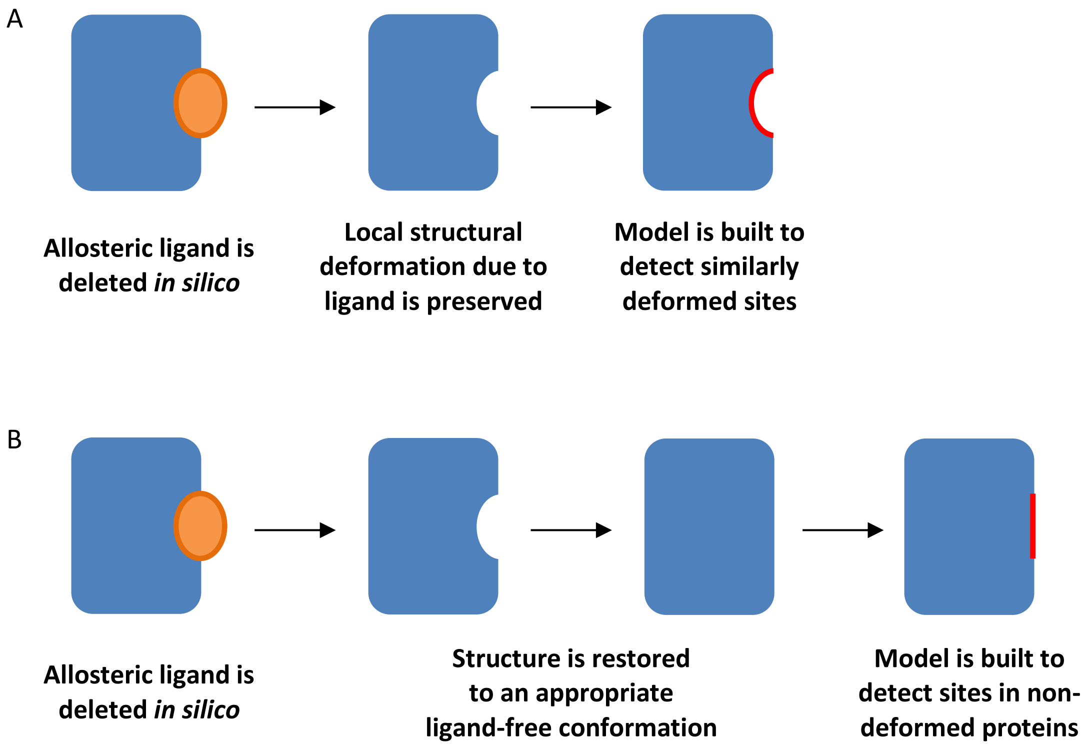

A
Allosteric ligand is deleted *in silico*
Local structural deformation due to ligand is preserved
Model is built to detect similarly deformed sites
B
Allosteric ligand is deleted *in silico*
Structure is restored to an appropriate ligand-free conformation
Model is built to detect sites in non-deformed proteins

Figure 1.22: ***A***, *a scheme of the generic procedure commonly followed by developers of allosteric site-predicting models. While care is taken to remove allosteric ligands from proteins before using them to construct the model, the deforming effect of the ligand's presence on the surrounding protein structure is not removed. Subsequent models are able to make use of this 'imprint' to detect the location of the site.* ***B***, *a scheme of a generic procedure that bypasses this issue; in this work the structural deformation of proteins was restored through MD.*

1.6 Work plan

Before any predictive models and subsequent analysis could take place, a dataset had to be constructed. Proteins with known 3D structures, allosteric sites and orthosteric sites had to be collected. Repositories such as the PDB and ASD, briefly discussed below, were to be used for this task.

The ASD is a database of proteins with known allosteric activity(152). Initially it appeared to be an ideal resource for constructing this project's dataset. However, the ASD's criteria for the inclusion of a protein are far less stringent than those for this project. Gaps in data entries are tolerated and filled in as the database is updated. This is rightly so, since the ASD is intended to be a comprehensive database of proteins with allosteric properties, annotating entries with a variety of contextual information such as known modulators, binding affinities, physicochemical properties, and therapeutic areas and related biological data. However, in many cases the gaps included a confirmed 3D protein structure or the location of the allosteric binding site. This prevented the straightforward use of the entire ASD for this project. Nevertheless, it could still be used as a starting point.

After collecting a set of suitable proteins – a process that would have to ultimately be performed manually – MD would be performed on each. The generated trajectories would then be subjected to a succession of analyses (described in detail in the Methods section), each quantifying some property of the protein on a per-residue basis. These would be formatted akin to the Iris dataset, which is compatible with RF. The table described here is exemplified in Table 1.3, which is of equivalent structure to that of the Iris dataset in Table 1.1.

descriptors/variables

classifier

Cases/observables

Residue	Analysis 1	Analysis 2	Analysis m	Allosteric
Tyrosine	1.1	100.4	1.69	False
Leucine	1.8	115.8	1.21	False
Leucine	3.2	40.7	0.87	False
Phenylalanine	2.1	22.3	0.17	True
Glycine	4.1	42.2	2.55	False
Residue n	1.7	0.0	1.94	True

Table 1.3: *An example of the structure of the* m x n *dataset created over the course of the project. Here, the cases were residues of proteins with known allosteric sites; the descriptors were the outputs of the various analyses performed on the proteins' MD trajectories; the response was a* true/false *assignment denoting whether the residue was part of the allosteric site or not.*

With a dataset constructed, RF models could be trained to predict which residues were part of an allosteric site, given the corresponding descriptors. This procedure would inevitably require successive rounds of optimisation, using previous results to direct the construction of future models. Crucially, none of the descriptors could require knowledge of the location of an allosteric site, allowing them to be calculated in an identical manner for proteins where the allosteric site was unknown.

2. Methods

2.1 Protein Selection

A dataset of proteins with known allosteric modulator sites was compiled. These were obtained from the PDB, ASD and ChemBL: all freely available online databases. For a protein to be included in the set, an unequivocal definition of its allosteric- and orthosteric binding sites was required. An available, experimentally solved crystal structure of the allosteric modulator-protein complex was necessary, as was an equivalent complex with the natural substrate or another orthosteric ligand. In some cases a ternary complex of the protein, substrate and allosteric modulator were available; this was also acceptable and could be used to define both binding sites. Though it may appear obvious, experimental evidence of activity-modulating behaviour on the part of the allosteric molecule was also required. As is discussed in this section, cases arose that appeared to meet these conditions at first glance, but had to be ruled out after closer inspection.

2.1.1 Mevalonate Kinase

The above criteria proved impossible to translate into an automated filter; instead, simple text searches were used to initially filter the databases before manually selecting proteins for inclusion. Manual inspection proved to be important, as exemplified with the case of mevalonate kinase from *s. pneumoniae*. Andreassi *et al.* published a crystal structure(153) of the kinase bound to diphosphomevalonate (DPM), a known allosteric inhibitor of mevalonate kinase (PDB code: 2OI2). The study found that, despite it displaying the behaviour of an allosteric inhibitor in kinetic studies, DPM bound to the orthosteric site under the crystallisation conditions employed. The crystal structure, though it could indeed be described as a complex of the protein and a non-competitive inhibitor, revealed no allosteric binding site, and so was of no use to this project. This could potentially be picked up by an automated filter. Indeed, it is listed in the ASD and was included in the dataset used to construct the Allosite model.

2.1.2 Focal Adhesion Kinase

In some cases only a complex of a different isoform was available with an orthosteric ligand. As the orthosteric complex was required only to assign an approximate location of the site, this was considered adequate so long as there was a high degree of sequence identity and structural overlap between the two isoforms. One example of this is focal adhesion kinase (FAK). An allosteric site is known for the *Homo sapiens* isoform (PDB code: 4EBV), but, at the time of writing, the only complex available with an orthosteric ligand was from *Gallus gallus* (PDB code: 2J0L). However, they share a sequence identity of 96% and the RMSD resulting from structural superimposition was 2.27 Å (Figure 2.1). It was deemed acceptable to include the *Homo sapiens* isoform in the dataset, using the ligand from the *Gallus gallus* isoform to define the orthosteric site.

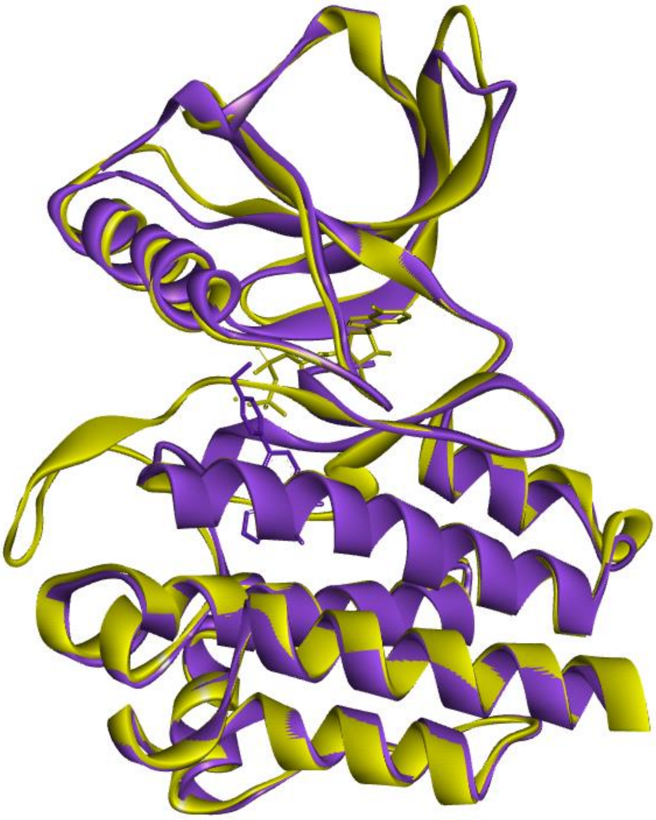


Figure 2.1: *The superimposed images of Homo sapiens (purple) and Gallus gallus (yellow) isoforms of FAK in ribbon form, including respective ligands shown in stick form. With little discernible difference in structure at the orthosteric site and an overall sequence identity of 96% and a RMSD of 2.27 Å, it was considered appropriate to use the Gallus gallus ligand to define the orthosteric site of the Homo sapiens isoform.*

2.1.3 AMP-Activated Protein Kinase

Another protein excluded after close inspection was AMP-activated protein kinase (AMPK) (PDB code: 2V92). The protein is critical in maintaining energetic homeostasis by controlling the relative concentrations of AMP and ATP in the cell. In mammals it contains multiple adenyl-binding sites, which are competed for by AMP and ATP(154). By responding to the relative concentrations of AMP and ATP, AMPK can trigger catabolic and anabolic cascades as required. Of importance to this project, there is a further site that AMP binds to with very high affinity, increasing the activity of the protein, thus defining AMP as an allosteric activator. However, the allosteric mechanism is believed to operate via a rearrangement of quaternary structure triggered by the allosteric binding event; this in turn inhibits the dephosphorylation of key residues which must remain phosphorylated to activate the protein(155). This means that the allosteric- and orthosteric sites are not coupled directly, but rather through a covalent bond-forming phosphorylation event that cannot be modelled with classical MD; for this reason, it was excluded from the dataset. It may be considered appropriate to include AMPK in other allosteric datasets, depending on the context of their use. For reference, this protein was included in the Allosite dataset.

2.1.4 Cdc34

In 2011 Ceccarelli *et al.* published the solved X-ray crystal structure of human Cdc34(156), an enzyme in the ubiquitin-proteasome system responsible for conjugating ubiquitin to substrates, bound to an allosteric inhibitor (PDB code: 3RZ3). Indeed, the inhibitor bound at a site 19 Å distal from the catalytic Cys93 residue and had a marked effect on Cdc34 activity. However, the group later published the Cdc34/ubiquitin/inhibitor complex (PDB code: 4MDK), where it was revealed that the putative allosteric inhibitor had been trapping Cdc34 and ubiquitin in an inactive conformation, with ubiquitin binding directly at the allosteric ligand's location(157). This meant that the ligand was not exerting its inhibitory effect at a distance but rather behaving much like a cofactor, facilitating the binding of ubiquitin at its own location. Depending on the breadth of one's definition, this could be considered a subtype of allostery, but the mechanism was distinct from the other proteins investigated, and was thus considered inappropriate to allow it into the dataset. For reference, this protein was also included in the Allosite dataset. Like the case of AMPK preceding this, it depends on one's definition of allostery whether this case ought to be discounted or not.

2.1.5 AMPA Receptor Subunit GluA2

Occasionally, mutations are deliberately made to proteins to help facilitate crystallisation(158) (PDB code: 1U7T). Whilst this makes the job of the crystallographer easier, the full effect of mutation(s) on the overall conformation of the protein is not always known. For this reason, mutated proteins were excluded from the dataset, unless there was evidence that the mutant retained both its activity and its responsiveness to the allosteric modulator. This was the case with the ligand binding domain of GluA2, an AMPA receptor subunit(159) (PDB code: 4U4X), which was included in the dataset.

2.1.6 Glycogen Phosphorylase

Two different forms of glycogen phosphorylase were identified that could potentially be included in the dataset. These proteins were liver glycogen phosphorylase A from *Homo sapiens* (PDB code: 1FA9) and muscle glycogen phosphorylase B from *Oryctolagus cuniculus* (PDB code: 1H5U). Though 87% identical in sequence, the structures used contained distinct allosteric sites. 1FA9 contained the native AMP as its allosteric ligand, and 1H5U instead contained a novel allosteric inhibitor. The two structures are superimposed in Figure 2.2 for reference.

By virtue of being similar isoforms, it is possible that these proteins, in reality, share allosteric sites. This is already known to be true for AMP, which binds to both isoforms at the same site and modulates activity; in fact, it is has a greater effect in the muscle isoform than it does in the liver isoform where it has been crystallised(160). However, since the two sites are distinct, one can be treated as the '1FA9 site' and the other as the '1H5U site.' By treating the proteins in this way – that is*,* as entirely unrelated proteins – predictive models would not receive any artificial assistance in identifying either site as a result of the other being included in the dataset.

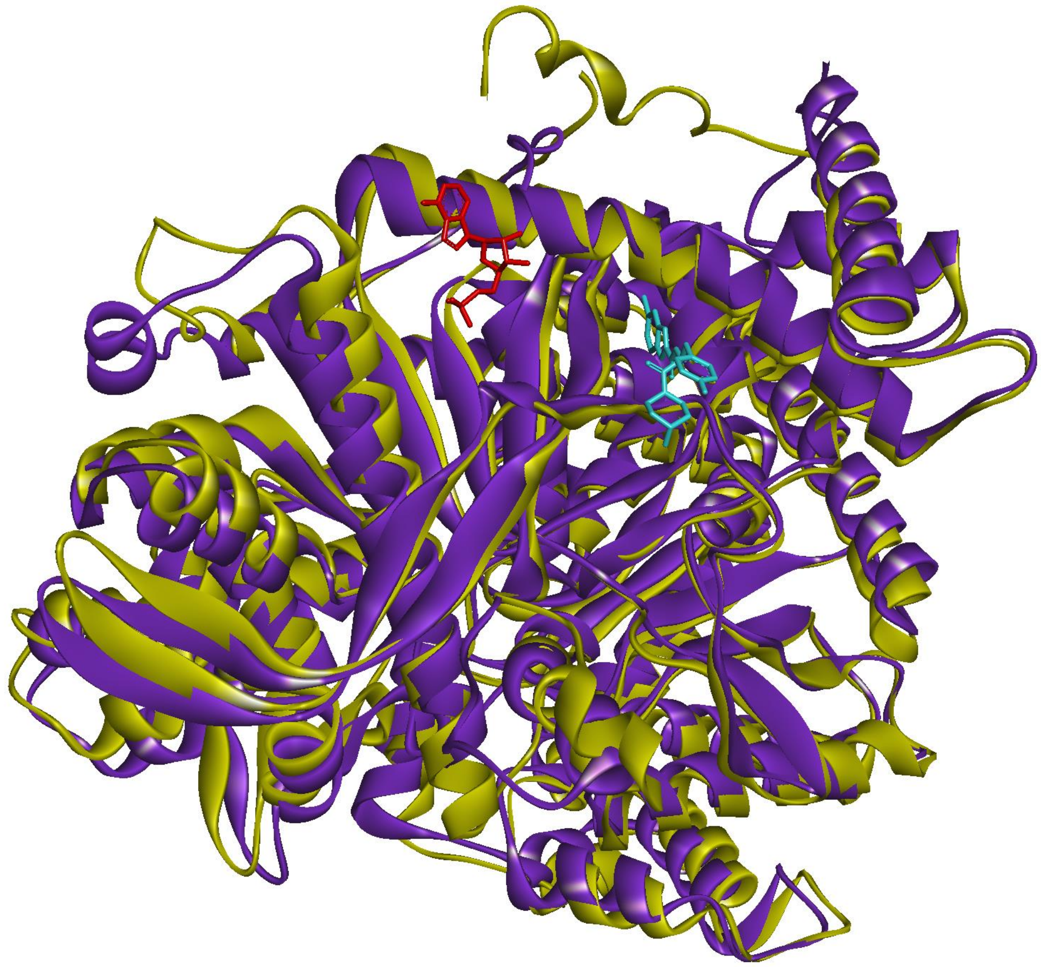


Figure 2.2: *The superimposed images of two glycogen phosphorylase isoforms. Liver glycogen phosphorylase A (PDB code: 1FA9) is shown as a yellow ribbon, with its allosteric ligand in red. Muscle glycogen phosphorylase B (PDB code: 1H5U) is shown as a purple ribbon with its allosteric ligand in turquoise. Though the structures were 87% similar, neither protein's presence in the dataset would be able to artificially aid a model in detecting the other's allosteric site.*

2.2 Sequence Alignment

In order to examine the diversity of the chosen set of proteins, pairwise sequence alignments of all studied proteins were performed. In order to achieve this, the ClustalW algorithm(161) was implemented in Pipeline Pilot with the aid of Dr Murray Robertson. The created protocol performed all alignments and produced a percentage-identity matrix: these values were used to render the plot presented in section 4.2.1.

2.3 Molecular Dynamics

Protein structures were downloaded as .pdb files from the PDB. Structures were prepared for MD by first removing irrelevant ions and other artefacts of the crystallisation process, *e.g.* crystallisation buffer solute. Where residues at the N- and C-termini of a chain were missing, the first/last present residue was capped with an acetyl/N-methyl group, respectively, to block the introduction of artificial ionic charges. Small missing loops in the sequence were filled in by selecting the highest scoring pose produced by the Loop Refinement protocol within Discovery Studio (default settings). To prevent the introduction of larger artificial errors, large missing loops (>20 residues in length) were not filled in, instead capping the loop termini. Water molecules present in the crystal were retained if they were at the protein surface or buried beneath it.

The AMBER 12 suite(72) was used to carry out all MD simulations. Using tleap, the command line version of the LEaP module within AMBER, the ff12SB forcefield was applied to all protein atoms. These were solvated in a 6 Å deep octahedral shell of TIP3P water molecules(162) before neutralising the net charge of the system with the appropriate number of Na^+ or Cl^- ions. Further Na^+ and Cl^- ions were added to bring the system to a salt concentration of 150 mM to approximate cytosolic ion content(163). Periodic boundary conditions were applied, with the PME(164) method used to calculate long-range electrostatic interaction at a cutoff distance of 10 Å. All bonds involving hydrogen atoms were constrained to their equilibrium lengths with the SHAKE algorithm(88), allowing the stable use of 2 fs time-steps in all simulations.

A 3-stage minimisation was carried out, each consisting of 50 000 steps, the first 250 of which used the steepest-descent algorithm and the rest conjugate gradient. The first stage allowed free movement only for hydrogens, water and ions; a harmonic restraining force of 100 kcal mol^{-1} Å^{-2} was applied to all other atoms. The second stage lifted the restraint from protein sidechain atoms and the third allowed the complete, unrestrained movement of all atoms.

The minimised system was heated in 4 mini-stages. Maintaining a constant volume, the first stage heated the system to 100 K over 20 ps; the second stage heated up to 200 K over 40 ps and the third to 310 K over 80 ps. The system was then switched to constant pressure and equilibrated for a further 80 ps at 310 K. The Langevin thermostat(84) was used throughout the heating process.

As an equilibration measure, a further 5 ns of dynamics was carried out on the system before switching over to the Berendsen thermostat(85) for the production phase. Systems were simulated for at least 50 ns, saving coordinates every 2500 steps. The trajectories of the first 50 ns of Berendsen dynamics were used in any subsequent analyses (10 000 saved frames).

2.4 Trajectory Analyses

The following analyses were performed on every protein for which a 50 ns trajectory of production MD was generated. The resulting data were used to populate a single spreadsheet to be used as input for machine learning. Many of the analyses produced a measurement for each residue per trajectory frame. In such cases the numbers were condensed into single values such as the mean, median and standard deviation.

No one analysis was intended to be a 'silver bullet' capable of identifying all allosteric sites for all proteins. All that was required of each analysis was the characterisation of some aspect of protein residues in the context of their environments; in fact, it was expected that some metrics would ultimately prove useless for allosteric site prediction.

In addition, not all residues in a site defined by the method used in this work were necessarily relevant to binding or activity. This, as well as no one analysis being ideal for the identification of allosterically important residues, meant that there was little to gain from manually inspecting every analytical result of every protein in the hope of detecting clear, overall trends. Due to the sheer volume of data and the coarseness associated with any one analytical result, one would inevitably miss many important subtleties, and over-fit what little was observed. It was better simply to understand what each analysis was capable of revealing, using RF to sift through the data in detail and detect trends emerging from it.

Where appropriate, a plot of the output of each analysis for pyruvate dehydrogenase kinase 2 (PDB code: 2BU2) – a protein chosen arbitrarily from the constructed dataset – is included as part of its description.

2.4.1 Allosteric and Orthosteric Site Definition

There is no standard definition of a binding site. Clearly the concept of a binding site refers to the region of the protein that is involved in the interaction between it and a ligand, but for this work any definition had to be resolvable to the level of individual residues, *i.e.* every residue of a protein had to be discretely classed as 'in' or 'out' of a site. The method also had to be consistent, as it was to be applied to as large and as diverse an array of proteins as possible. The chosen method satisfied these criteria, though was by no means flawless; in all likelihood, some irrelevant residues will have been classed as part of sites, and some possibly important residues will have been left out.

For each protein, a selection sphere was applied to every atom of the orthosteric and allosteric ligands. Any residue residing wholly or partially within the spheres was classed as part of the relevant site. In cases where a residue was within both sites, the allosteric classification was given priority. This definition procedure is illustrated in Figure 2.3.

The classification of residues as part of an allosteric site could not be achieved in this way for proteins with no known allosteric site; this is the variable that RF models would use as a response.

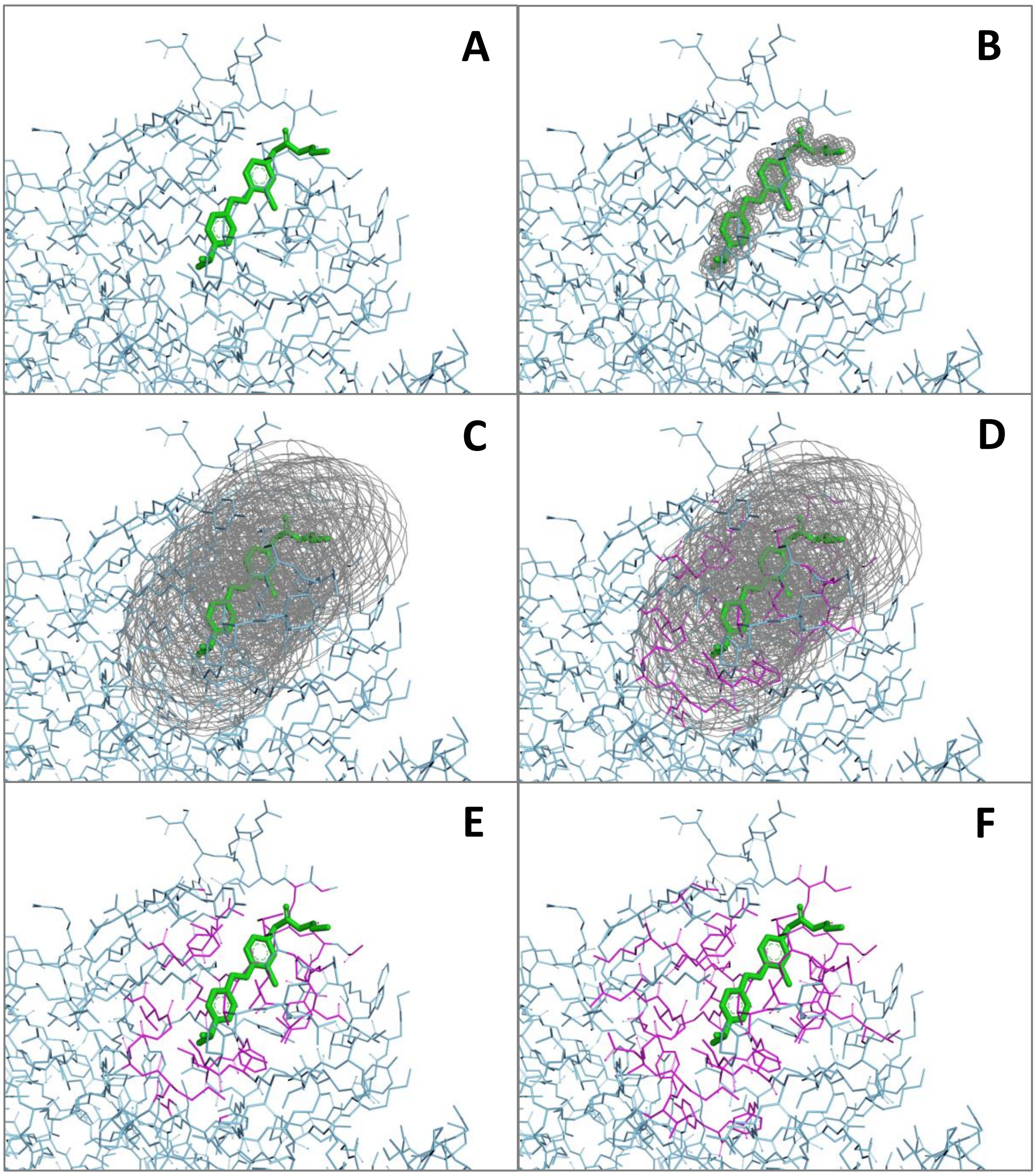


Figure 2.3: *An illustration of the site definition process.* ***A****, the crystal structure of a ligand in its binding site.* ***B****, a radius is applied to every atom of the ligand.* ***C****, the radii are extended to the desired length – in this case 7 Å.* ***D****, all atoms located within the radii are selected.* ***E****, the radii are removed for clarity.* ***F****, the selection is extended to any partially-selected residues; these residues are defined as being part of the binding site.*

2.4.2 Percent-of-Maximum Distance to the Orthosteric Site

For each protein, the furthest alpha carbon atom from the orthosteric site was determined, defining the ‘site’ as the geometric centre of all residues previously classed as part of the site. The distances from the orthosteric site to all alpha carbons in the protein were then determined and normalised to determine the percent-of-maximum (POM) distance. These measures were produced with a custom Python script.

The reasoning behind this was to uniformly quantify the distance between residues and the orthosteric site: perhaps a trend in the distance between allosteric and orthosteric sites would emerge. No significant trend was observed in this work, though it certainly worth tracking this information in the future if the set of proteins analysed is expanded upon.

2.4.3 Hydrophobicity Score

The hydrophobicity score (HS) is a simple constant assigned to each residue based on its chemical identity. These values, originally generated by Kyte and Doolittle(165), do not change depending on the specific environment of an individual residue. However, they are based on the averaged physicochemical properties of each amino acid side-chain across proteins. From a computational perspective, these data are similar to residue names in that they remain constant for each amino acid regardless of its environment, though residue names are treated as completely independent categorical data (*i.e.* with no relation to one another), whilst the HS are numerical data. This allows a machine learning algorithm to relate, for example, valine, with a HS of 4.2, more closely to leucine (HS = 3.8) than lysine (HS = -3.9). Table 2.1 lists the complete set of HS data used for this project.

Residue	Hydrophob. Score	Residue	Hydrophob. Score
ALA	1.8	LEU	3.8
ARG	-4.5	LYS	-3.9
ASN	-3.5	MET	1.9
ASP	-3.5	PHE	2.8
CYS	2.5	PRO	-1.6
GLN	-3.5	SER	-0.8
GLU	-3.5	THR	-0.7
GLY	-0.4	TRP	-0.9
HIS	-3.2	TYR	-1.3
ILE	4.5	VAL	4.2

Table 2.1: *The hydrophobicity scores used for each amino acid.*

2.4.4 Mass-Weighted Residue Fluctuation

The average structure of the trajectory was calculated then energy-minimised to produce a representative conformation of the protein over its trajectory. The root mean square fluctuation (RMSF) of each residue's centre of mass from its position in the minimised average structure was calculated. Calculations and minimisations were performed using the ptraj and sander modules within AMBER, respectively. The minimised average structure produced here was also used for all subsequent analyses referring to such a structure. A typical output of the analysis is shown in Figure 2.4, with the residues of the allosteric site highlighted. Greater values correspond to more flexible regions of the protein, which explains a common artefact of this analysis: chain terminals exhibit extremely large fluctuations. This is simply because chain terminals are only chemically tethered at one end and so are free to make large movements with no great energetic penalty; for the same reason these residues tend not to be part of any binding site (though they sometimes are).

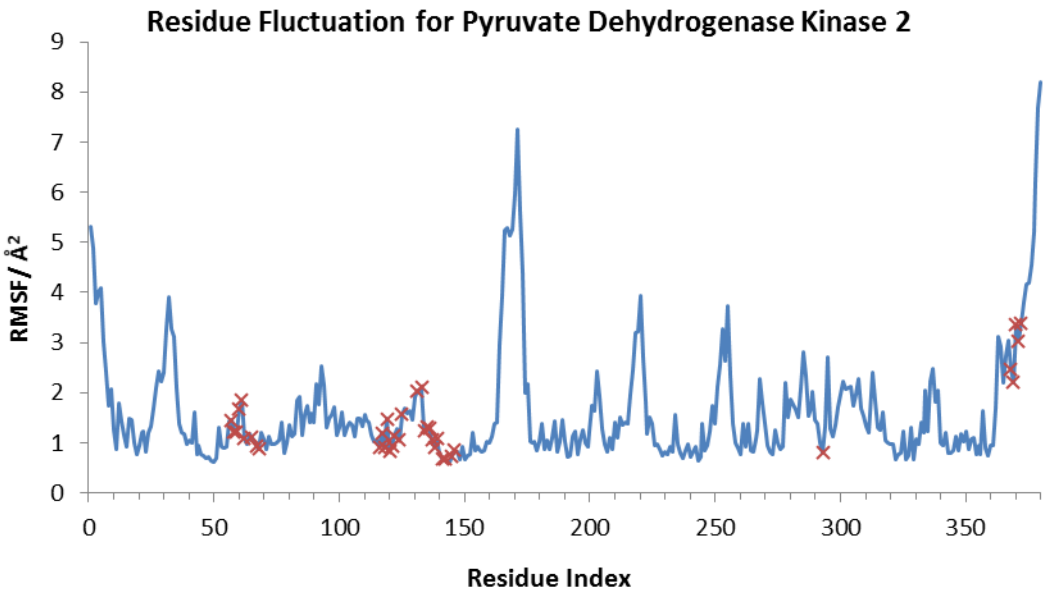


Figure 2.4: *The residual fluctuation values for pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses. As well as the terminals, there are clear segments within the protein that display large fluctuations, indicating the presence of flexible loops.*

While it is tempting to focus merely on areas of large fluctuation such as residues 164-173, it is important to examine the whole spread of data; for example, areas of low fluctuation generally map to areas of defined secondary structure. For this particular protein, the allosteric site comprises mainly α-helices, which could explain why most of the residues exhibited a low fluctuation. Of course, it may well be anomalies in such patterns – for example, a residue within an α-helix with a high fluctuation, despite its ordered environment – that are indicative of something important to allostery.

2.4.5 Correlated Motion

Correlated motion tracks the movement of each residue in relation to every other residue. For each residue, the vectors representing its movements from frame to frame are determined. A correlation analysis is then performed on each pair of vectors, resulting in a square matrix of values. This procedure can be performed within the ptraj module of AMBER. To reduce the matrix to single values per residue, the values were averaged, yielding an overall value of generic correlated motion.

A further descriptor was derived by averaging a residue's correlations only with the residues of the orthosteric site. This produced an average value of correlation to the orthosteric site, which was perhaps more relevant than correlation to the protein as a whole.

The above correlations of motion were also performed for the main chain atoms of each residue only. The rationale for this was that the averaged motions of a whole residue would generally be dominated by the motions of the sidechain; in order to investigate main chain movements in any way these had to be filtered out. This resulted in a total of four descriptors of various types of correlated motion.

2.4.6 Solvent-Accessible Surface Area

The solvent-accessible surface area (SASA) was calculated using Hubbard and Thornton's naccess program(166) (version 2.1.1). The program operates by applying a VDW radius to the coordinates of each atom in the protein and 'rolling' a probe sphere of a specified radius – in this case 1.4 Å, the VDW radius of water – over the surface. The total area of each residue accessed by the probe is logged and outputted. The SASA of each residue for every frame of a trajectory was calculated; the total SASA was also split into polar and apolar components. All output data was then summarised, taking a minimum, maximum, mean (Figure 2.5), median and standard deviation value for each residue over the course of the trajectory.

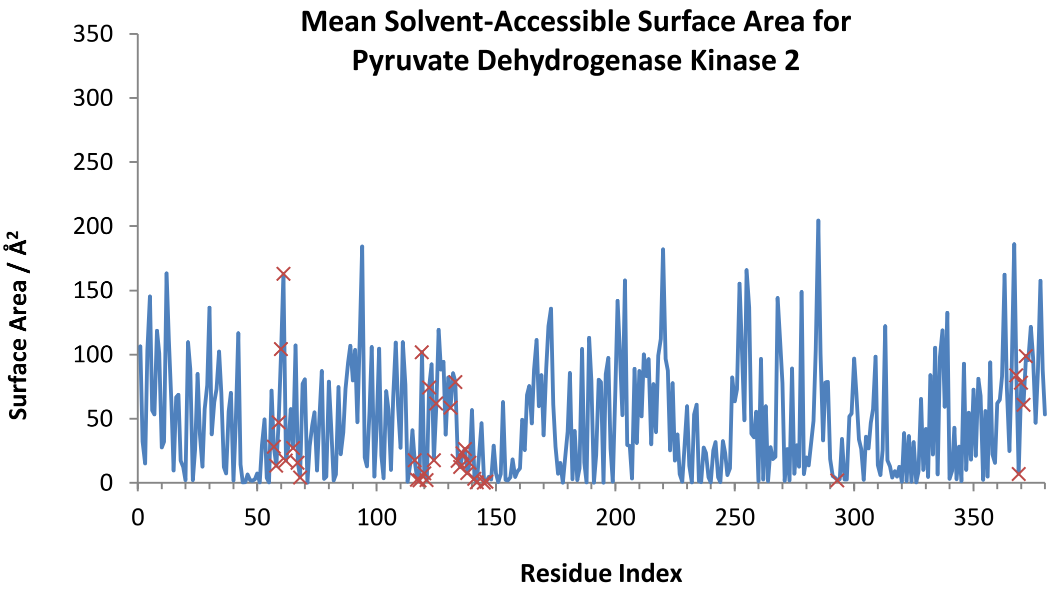


Figure 2.5: *The mean solvent-accessible surface areas (SASA) for pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses. Analogous plots could be produced for the other summary values.*

From Figure 2.5 it can be seen that the vast majority of allosteric residues had SASA values of less than 120 Å^2 for the example protein, with some approaching 0 Å^2 (*i.e.* a buried residue). This trend can be rationalised: since a ligand tends to fit into a concave pocket of a protein, one would expect the majority of residues within the pocket to exhibit a middle-to-low accessible surface area. Of course, this is a broad and generalising observation; as discussed earlier, it is more appropriate to leave the data processing to RF.

This analysis is directly affected by the chemical identity of the residues being analysed. Clearly, residues with large side-chains such as tryptophan, arginine and tyrosine will have a larger SASA than an equivalently-exposed glycine or alanine.

2.4.7 Hitting Time and Commute Time

Hitting times and commute times are measurements with origins in information theory(167), measuring the number of time steps taken to send information from one point to another. In 2007, Chennubhotla and Bahar related these communicative properties to the equilibrium fluctuations of residues on proteins(168). In this context, the hitting time *(j,)* is defined as the expected number of arbitrary time steps it takes to send information from residue i to residue j. The commute time *(i,)* incorporates the ‘return journey’ of the information, defined as the sum of *(i,)* and *(j,)*. These two may not be equal in value, thus hitting time is directional in nature while commute time is not:

$$C(i,j) = H(i,j) + H(j,i) = C(j,i) \tag{11}$$

The study found that highly functional residues, including catalytic residues and secondary structure elements, displayed shorter hitting times, *i.e.* a fast relay of information. According to the authors, the commute times are indicative of more generic properties of signal transduction. For this methodology a single protein structure is modelled as an elastic network - the minimised average structure of a trajectory was used in this work.

Determining the hitting and commute times first required the calculation of the interaction strength, or affinity for each pair of residue i and j. This was defined as:

$$a_{ij} = \frac{N_{ij}}{\sqrt{N_i N_j}} \tag{12}$$

where N_{ij} is the total number of contacts made between atoms in residues i and j based on a cutoff distance of 4 Å, and N_i and N_j are the total number of heavy atoms in residues i and j, respectively. Based on a_{ij} the local interaction density d for each residue j was defined as:

$$d_{\mathrm{j}} = \sum_{i=1}^{n} a_{ij} \tag{13}$$

From the populated affinity matrix A = {a_{ij}} and degree matrix D = diag{d_{j}} the stiffness or Kirchhoff matrix Γ was calculated by:

$$\Gamma = D - A \tag{14}$$

The hitting time is given by:

$$H(j,i) = \sum_{k=1}^{n} \left(\Gamma_{ki}^{-1} - \Gamma_{kj}^{-1} - \Gamma_{ji}^{-1} + \Gamma_{jj}^{-1}\right) d_k \tag{15}$$

while commute time is given by:

$$C(i,j) = \left(\Gamma_{ii}^{-1} + \Gamma_{jj}^{-1} - 2\Gamma_{ij}^{-1}\right)\sum_{k=1}^{n} d_k \tag{16}$$

Full derivations of these expressions can be found in the original paper(168). The mean hit/commute times for each row of the outputted matrices, *<(j,i)>* and *<C(i,j)>* respectively, yielded a single, overall measure for each residue. These values are presented for an example protein below in Figure 2.6, with the residues of the allosteric site marked.

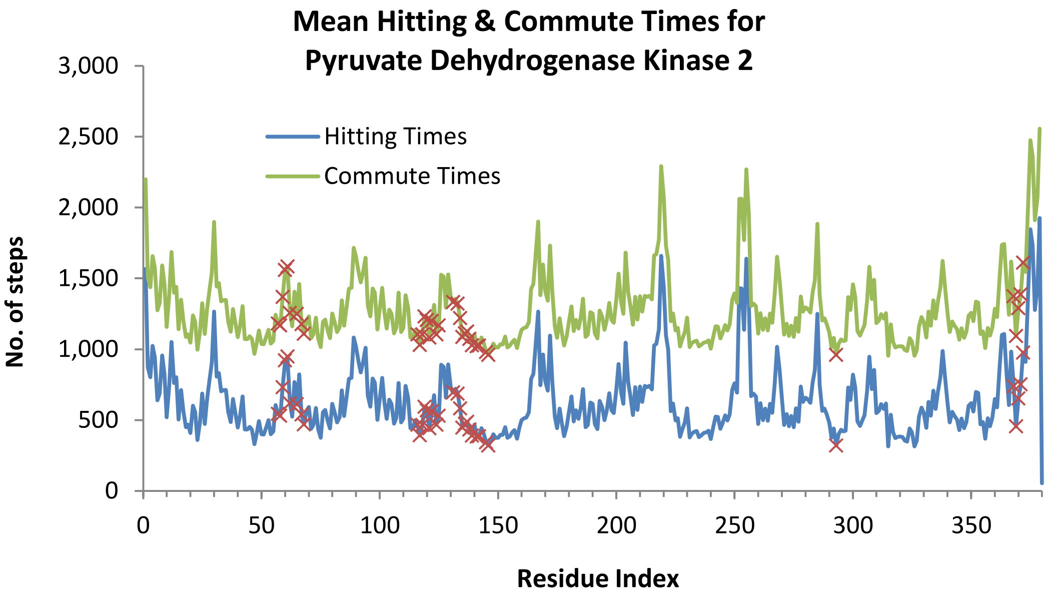


Figure 2.6: *The hitting and commute times for each residue of pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses.*

The ratio of hitting time and commute time was also taken for each residue (Figure 2.7). These numbers were taken forward with the raw numbers as a means of exposing relatively short hitting times, even in residues with longer commute times.

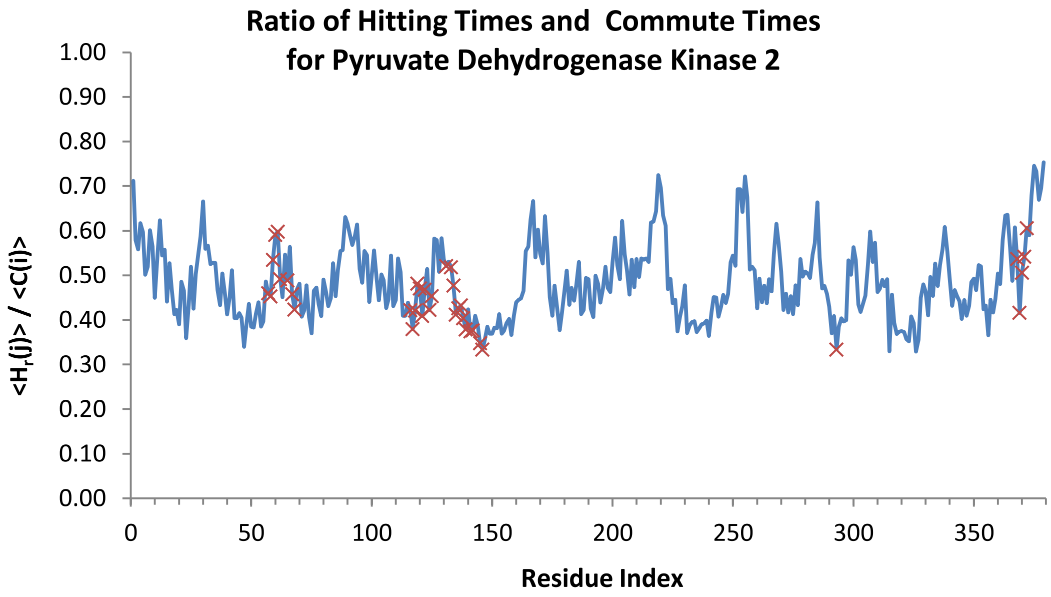


Figure 2.7: *The ratio of hitting times and commute times for each residue of pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses. This plot would be a horizontal line if the graphs in Figure 2.6****Error! Reference source not found.*** *were roportional; evidently, residues exhibit hitting times of differing length relative to their commute times.*

A and D matrices were generated using a custom Perl script; the calculation of Γ and subsequent values were performed in MATLAB using a modified script. The original Perl and MATLAB scripts were written by Dr Nahoum Anthony.

2.4.8 Correlated Energies

Introduced by Erman(169), this method relates fluctuations in energy of the surroundings of the protein to the fluctuations of the residue positions within it. It was observed that spatial energy exchange was anisotropic, with certain residues behaving more responsively to incoming energy than others. The study was able to identify so-called 'energy gates' – residues with a high tendency to propagate received energy throughout the protein. As with Chennubhotla and Bahar's hit/commute time analysis, a single protein structure modelled as an elastic network is required; the minimised average structure of a trajectory was used once more.

The procedure for computing the energy correlation of a protein's residues required the use of a Kirchhoff matrix Γ derived in the same manner as the hit/commute time methodology (Equations 12-14) though a cutoff distance of 7 Å was used in defining the affinity matrix A instead of 4 Å. At a fixed temperature, the correlation of energy fluctuations between a pair of residues *i* and *j* is proportional to the following:

$$\begin{aligned}
&2\left(\left(\Gamma_{ik}^{-1}\right)^2+\left(\Gamma_{il}^{-1}\right)^2+\left(\Gamma_{jk}^{-1}\right)^2+\left(\Gamma_{jl}^{-1}\right)^2\right)\\
&+\Gamma_{ii}^{-1}\Gamma_{kk}^{-1}+\Gamma_{ii}^{-1}\Gamma_{ll}^{-1}+\Gamma_{jj}^{-1}\Gamma_{kk}^{-1}+\Gamma_{jj}^{-1}\Gamma_{ll}^{-1}\\
&-4\left(\Gamma_{il}^{-1}\Gamma_{ik}^{-1}+\Gamma_{jl}^{-1}\Gamma_{jk}^{-1}+\Gamma_{ik}^{-1}\Gamma_{jk}^{-1}+2\Gamma_{il}^{-1}\Gamma_{jl}^{-1}\right)\\
&-2\left(\Gamma_{ii}^{-1}\Gamma_{kl}^{-1}+\Gamma_{jj}^{-1}\Gamma_{kl}^{-1}+\Gamma_{kk}^{-1}\Gamma_{ij}^{-1}+\Gamma_{ll}^{-1}\Gamma_{ij}^{-1}\right)\\
&+4\left(\Gamma_{ij}^{-1}\Gamma_{kl}^{-1}+\Gamma_{ik}^{-1}\Gamma_{jl}^{-1}+\Gamma_{il}^{-1}\Gamma_{jk}^{-1}\right)
\end{aligned} \tag{17}$$

For the full derivation of this equation, the reader is referred to Erman's original paper(169). Summation of each row of the outputted matrix yielded the overall energetic interaction of residue i with the rest of the protein; these were the final numbers taken from this analysis. As with the hit/commute times, A and D matrices were generated using a custom Perl script (written by Dr Nahoum Anthony) and the final calculations were implemented in MATLAB. Figure 2.8 shows the results of this analysis for pyruvate dehydrogenase kinase 2.

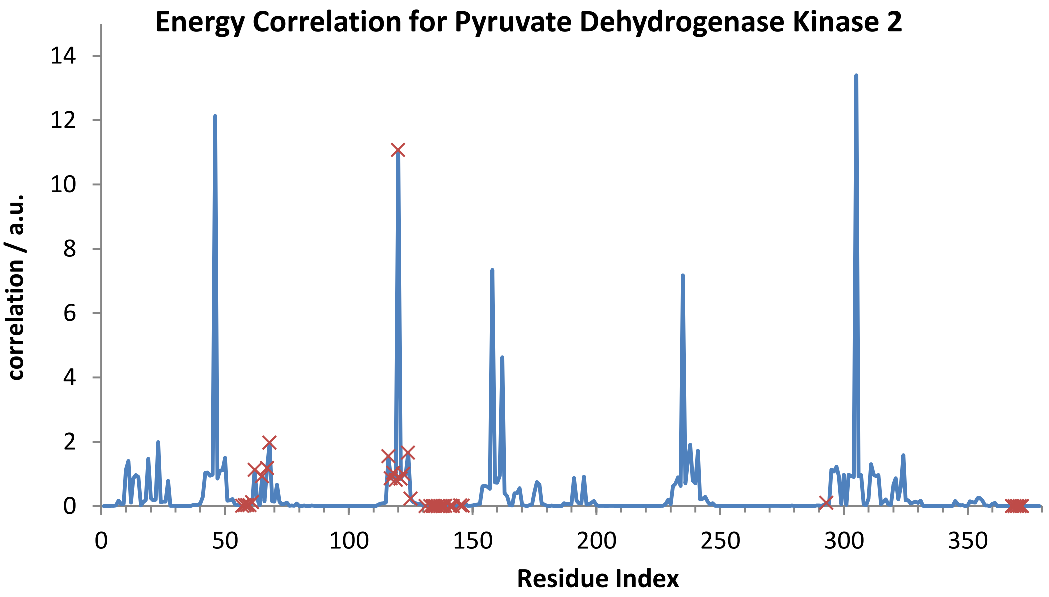


Figure 2.8: *The correlated energies of each residue of pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses. Residues more highly correlated with the rest are involved in the propagation of energy throughout the protein and, in theory, more likely to be involved in allostery.*

2.4.9 Dihedral Angle Analysis

This analysis examined the repeating main-chain dihedral angles in a protein. It was developed as part of this project and is described in detail in Chapter 3. The final outputs of the analysis were of the standard form: a single value per residue. Four such measures were produced, monitoring various aspects of the conformational behaviour of the residues.

2.4.10 Simple Intrasequence Differences

Developed by Pritchard *et al.*(170), a Simple Intrasequence Differences (SID) analysis yields a number of metrics based on the surroundings of a residue. These often provide some insight on the nature of the protein fold the residue is part of. The algorithm proceeds through the sequence's alpha carbons, defining a 7 Å sphere at each. All other residues whose alpha carbons reside within the sphere are clustered. The cluster is then scored in multiple ways before moving to the next residue; the scoring methods used were as follows:

- Count: a simple count of the cluster population. This information gives some indication of how crowded the residue's environment is. In the context of binding site prediction, it may be that residues within ideal pockets will likely have a middling Count score: too low would suggest that the residue is exposed on a flat or convex surface and too high would suggest that the residue is buried.

- Strands: similarly, the number of non-consecutive segments of the sequence within the cluster is counted.

- Highest – lowest (HL): this score subtracts the lowest sequence number in the cluster from the highest in the cluster. A high HL scores is indicative of two or more parts of the polypeptide chain, distant in terms of primary sequence, folding into the same 3-D space.

- Greatest gap (GG): here, the cluster members are ordered by sequence number. The difference between each consecutive residue number is then calculated, the largest of which is retained as the GG score. A GG score of 1 implies that only one continuous segment of residues was in the cluster. The highest GG scores are obtained when exactly 2 chain segments occupy the cluster; further chain segments bring the GG score down.

- Differential (DIFF): where there is ambiguity in the GG, it can be removed by subtracting it from the cluster's HL; this is the DIFF score. GG and HL will be similar in value when 2 chain segments are present in the cluster. Further segments will bring down the value of GG but not HL, so between all of these values a good level of insight into the local topology of a residue can be deduced.

This analysis operates on a single protein structure, and is sufficiently quick that it was feasible to analyse entire trajectories, generating SID scores per residue per frame. Minima, maxima, means, medians and standard deviations value were taken as single summary values for each trajectory. To differentiate between other differences in residues' spreads of SID scores, the range of scores was taken (*i.e.* maximum – minimum). This range was also divided by the count of discrete scores. Finally, the absolute difference between the mean and median were taken.

2.4.11 Normalised SID

The above SID data were normalised from 0-1 for each protein, allowing an even comparison between proteins of different sizes. For instance, a 250 residue protein has a maximum theoretical HL score of 249 (if both terminals are within 7 Å of each other) while a 1000 residue protein could routinely achieve scores 3 times as large, and has a theoretical maximum of 999. Independently normalising the two proteins' scores would give a maximum value of 1 to the first case and values of 1 or less to the latter, removing the skew caused by significantly different protein sizes.

However, there are also drawbacks to this approach. A GG score of 1 has the identical implication regardless of protein size: that only a single, continuous chain fragment is present in the cluster. Normalising a protein's GG scores would change the values to a proportion of the maximum, which, as mentioned above, vary from protein to protein according to sequence length. Thus, while there is unique information represented in both the raw and normalised scores, there is also unique noise. However, this did not pose a major problem for this work due the way RF naturally handled noisy data.

2.4.12 fpocket

The fpocket program(150) was written to identify cavities on the surfaces of proteins. It operates by first performing Voronoi tessellation on the protein. At each Voronoi vertex, a sphere can be drawn to fill the empty space between atoms. These spheres, termed alpha spheres, are then filtered and clustered, resulting in coherent pockets. Figure 2.9 shows the output structure of the pocket-finding algorithm performed on pyruvate dehydrogenase kinase 2 with default settings. Based on properties of the alpha sphere clusters, such as the number of them in the pocket, the sphere radii and density, a host of metrics can be calculated to characterise each pocket. Some chemoinformatic descriptors can also be determined by examining the atoms around each pocket, such as hydrophobicity and charge. In total, 20 descriptors were generated by fpocket.

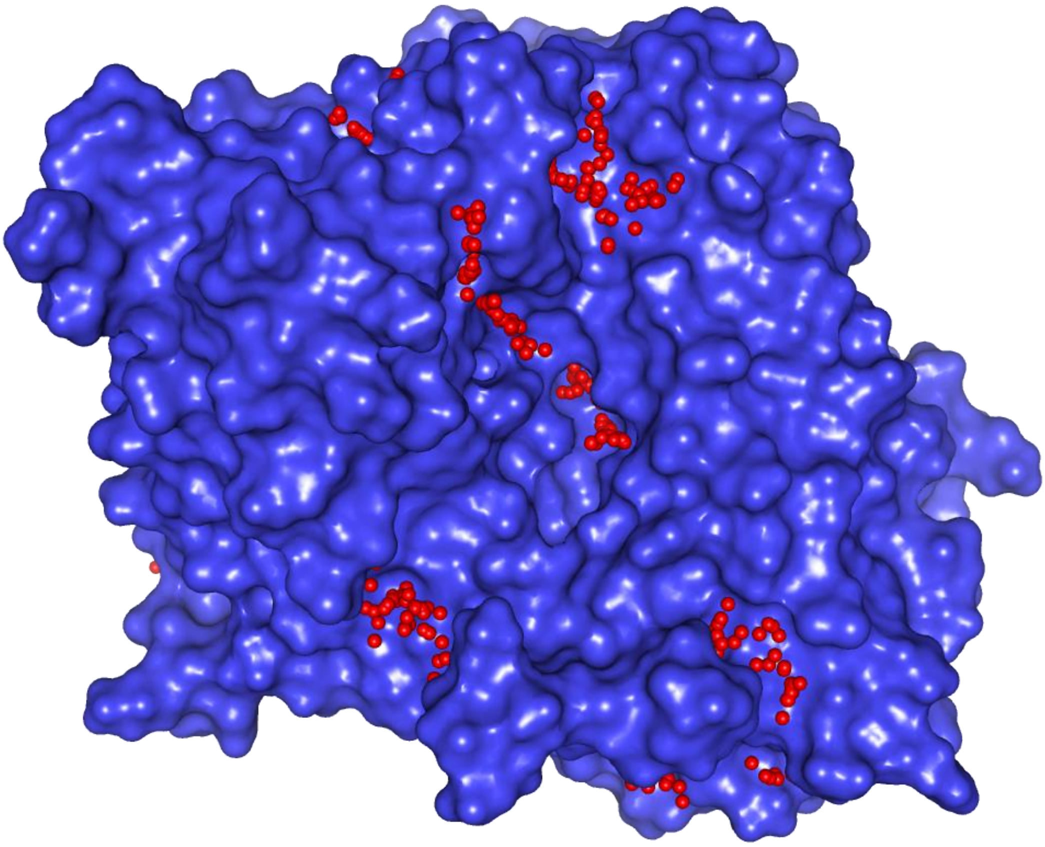


Figure 2.9: *The output of the pocket-finding portion of the fpocket analysis performed on pyruvate dehydrogenase kinase 2. The alpha spheres can be seen filling in the cavities of the protein surface.*

A complication arose with the use of this program in that the descriptors characterises entire pockets rather than individual residues. A post-processing script was written to handle this. Generally, all residues were assigned the scores of the pocket to which they belonged. In cases where they belonged to more than one pocket – *i.e.* where one side of a residue faces one pocket and the other side faces another – the residue received the scores of the pocket with the largest volume. The pocket volume was one of the descriptors calculated by fpocket and so was retrievable without any calculations.

The fpocket analysis was performed on the minimised average structure of each trajectory. However, it is worth noting that the developers added a capability for fpocket to analyse entire trajectories, frame at a time.

This extension, named mdpocket, was tested extensively. It was found that the code was unable to properly release memory as it progressed from frame to frame, and so could not analyse a complete trajectory before running out memory and crashing. Several workarounds were devised, including parallelising fractions of trajectories across a HPC facility. Ultimately, the results proved unreliable and could not be used. This is mentioned because, if mdpocket were to be developed further, the data it generated could prove very powerful. The authors were contacted; they are aware of issues with mdpocket and development is ongoing.

2.4.13 DelPhi

The DelPhi program(171, 172) is a method for calculating the distribution of electrostatic potential across a molecule's surface. It is included in Discovery Studio, a piece of software heavily used for preparation and visualisation of protein structures in this project. In its packaged form, the algorithm efficiently solves the PB equation by a grid-based method. It returns a distinct potential for each residue of the protein; this is precisely the format required for inclusion in the dataset. The execution time was of the order of minutes per structure, so the calculation could not be extended to every frame of a trajectory. Instead, potentials were calculated once for the minimised average structure of each trajectory. Figure 2.10 shows the result of the DelPhi analysis for pyruvate dehydrogenase kinase 2.

As stated previously, it has been found that allosteric ligands tends to be less polar than orthosteric ligands(54). Results from this project presented in section 4.2 complement this by showing allosteric sites to also be less polar than orthosteric sites. This is rationale enough to include a variable based on electrostatic potential; however, such a variable ought to be utilised even in the absence of a working theory, since it describes a fundamental aspect of all molecules that surely affects allosteric behaviour.

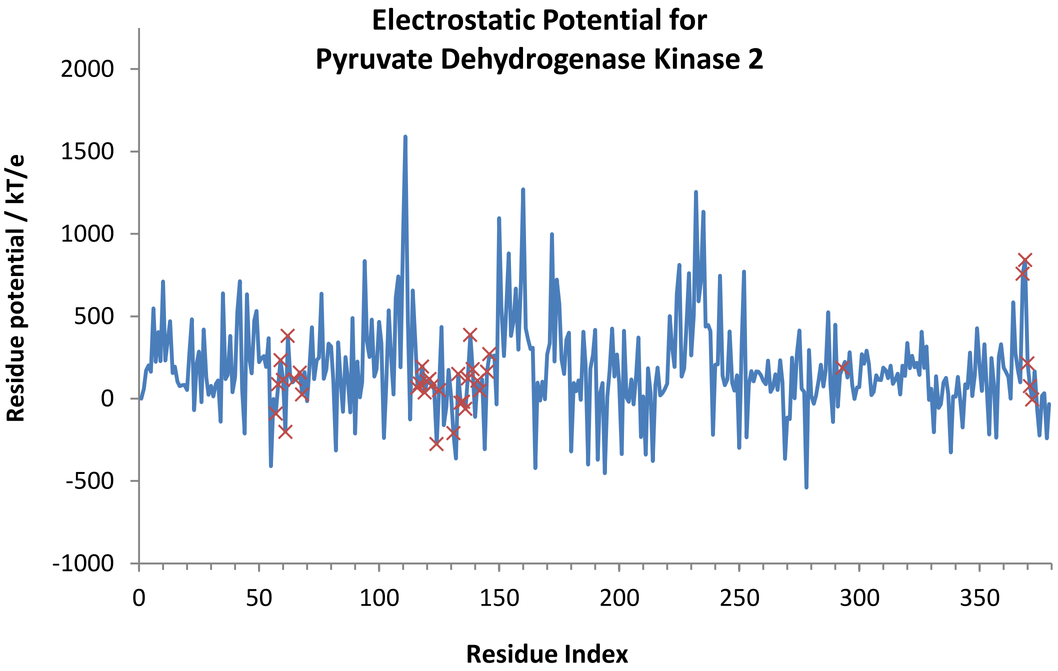


Figure 2.10: *The residue potentials calculated by the DelPhi analysis for pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses.*

2.5 Random Forest

The data generated from all analyses were compiled into a single text file using a Python script. Separate files were created for training sets and testing sets; for training sets, the response, *i.e.* a *true*/*false* value signifying whether each residue was part of the real allosteric site or not, was also included. The files were loaded into the R statistics package, as was randomForest(173) (version 4.6-7), the library containing the R implementation of RF. Random forests could then be generated at desired parameters within R. A further library, sprint(174) (version 1.0.4), was loaded which facilitated the parallelization of a selection of computationally intense R commands, including randomForest, over multiple cores and multiple computers on a cluster.

3. Metric Development: Dihedral Angle Analysis

3.1 Overview

This short chapter details the development of a novel analysis of amino acid dihedral angles over the course of a MD trajectory. The analysis outputs were tailored to the form required for the construction of the dataset, namely a single value per residue.

Ramachandran plots are a method for visualising protein backbone dihedral angles. There are three repeating dihedral angles in a peptide backbone, shown in Figure 3.1, denoted φ (phi), ψ (psi) and ω (omega). The omega angle rarely deviates from approximately 180° due to the rigidity of the amide motif; for this reason, Ramachandran *et al.* chose to exclude it and plot the phi angle of a given residue against its psi angle(175). The plots vividly demonstrated that amino acids in globular proteins tend to exhibit only a certain range of phi- and psi angle combinations, occupying only certain areas of what became known as the Ramachandran plot.

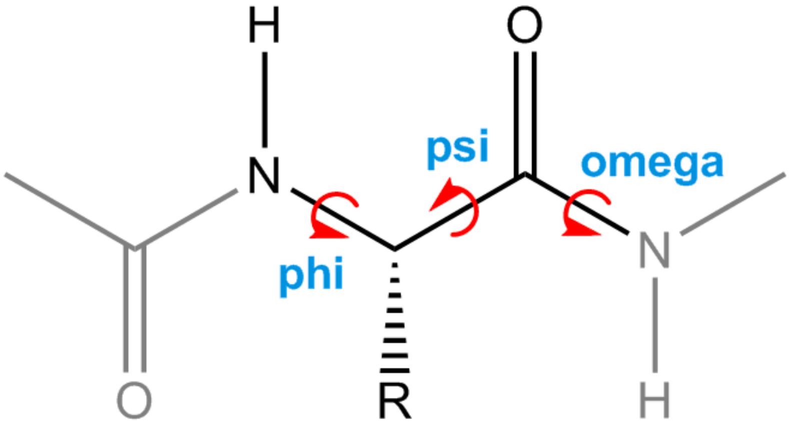


Figure 3.1: *The three repeating main chain dihedrals of a protein residue, denoted phi, psi and omega. Omega angles rarely deviate from a planar 0° or 180°.*

An example Ramachandran plot is presented in Figure 3.2. This was produced by querying the PGD (details of this procedure follow in section 3.2). Each point represents a phi-psi conformation of an alanine residue in a real crystal structure; the regions of high and low occupancy are clearly visible. High occupancy regions on a Ramachandran plot of this type are composed of those conformations that are either low in steric strain or typically found in secondary structural motifs (or both). Across the amino acids the regions corresponding to common secondary structural motifs (highlighted in Figure 3.2) are fairly consistent, though the rest of the conformational distribution varies significantly(176).

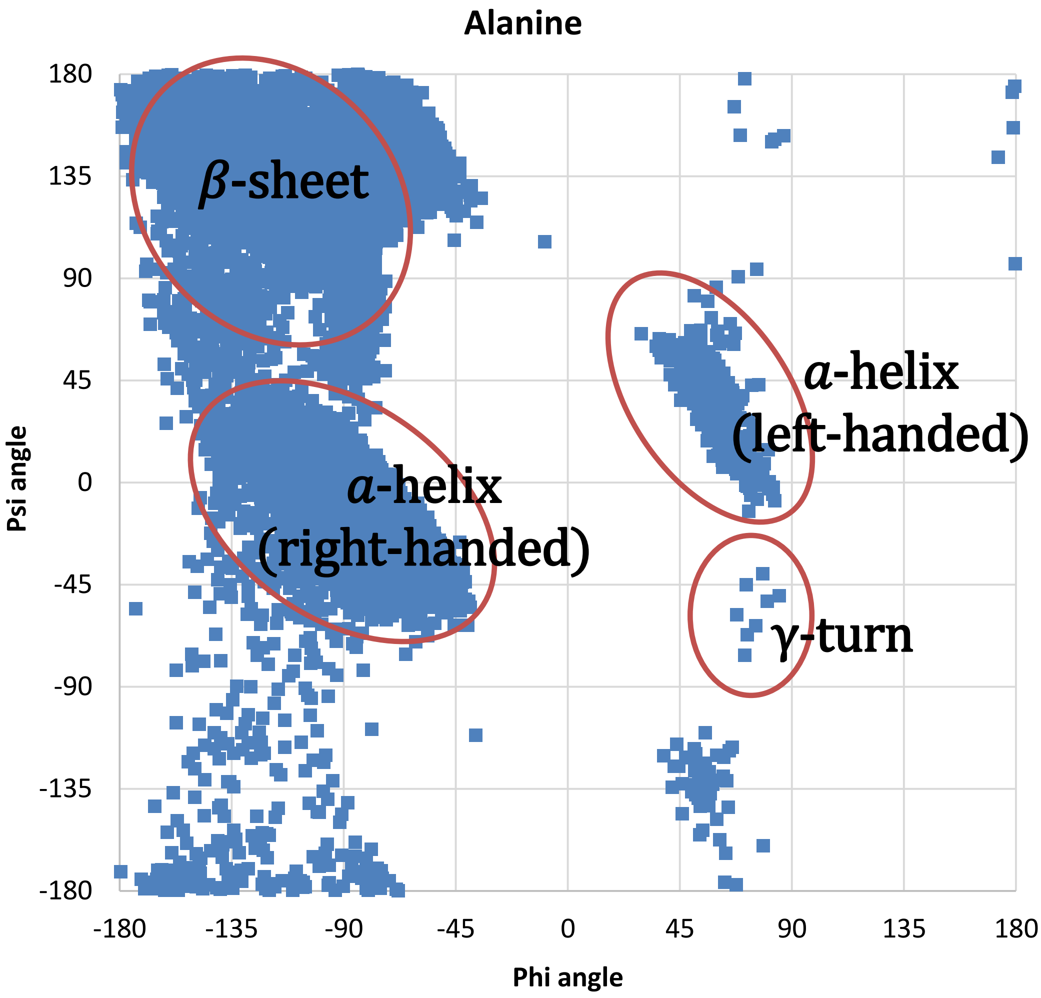


Figure 3.2: *A Ramachandran plot showing the typical conformational distribution for alanine. Each point represents a phi-psi conformation of an alanine residue in a real crystal structure. The highlighted regions corresponding to common secondary structural motifs are approximately consistent across the amino acids, though significant overall differences in conformational distributions exist for each.*

The terms ‘allowed’ and ‘disallowed’ are often used to describe the regions of high- and low occupancy, respectively, on the Ramachandran plot, though it should be made clear that this is merely convention. A small proportion of residues in proteins may be found in disallowed regions, and are often worth investigating further. Provided the anomalous conformation is not due to an error made in the observation of the structure, unusual circumstances in the local environment must be the cause of it. A particularly favourable hydrogen bond interaction or a strained junction between two highly structured regions are examples that could mitigate the energetic penalty for such a conformation being adopted(177).

It has been proposed that residues with disallowed phi-psi combinations could be a feature of allosteric sites(178). The study sought to find novel inhibitors of TEM-1 β-lactamase. Co-crystallisation with the protein revealed that some of these inhibitors bound to a cryptic allosteric site. These structures were compared with previously known high resolution crystal structures of TEM-1(179). A leucine residue within the allosteric site, which persistently adopted a disallowed conformation in the apo protein, was observed to have shifted to an allowed state in the presence of the ligands. The authors only observed this in hindsight, but suggested that this behaviour could be a marker of an allosteric site: residues under conformational strain could be relieved of it by an incoming ligand, resulting in an extra energetic incentive for a ligand to bind in the vicinity.

With this notion at its core, it was thought that useful information pertaining to allostery could be obtained by monitoring the phi- and psi angles of each residue over a MD trajectory.

The dihedral angles themselves were easily retrievable from the trajectories using the cpptraj module of AMBER. However, its output was simply two text files listing the phi- and psi-angles for each residue in the protein at each frame. There was no available method for visualising the data in phi-psi space or processing it into meaningful descriptors.

3.2 Data Retrieval

To normalise any comparison of phi- and psi angles between different amino acids, standard allowed zones were required for each amino acid. The Protein Geometry Database (PGD) was used to accomplish this task(180). The PGD (http://pgd.science.oregonstate.edu/) is linked to repositories of protein crystal structures and, through a web interface, allows the user to query these for various crystallographic parameters, including residue dihedral angles. A number of criteria can be specified to narrow the search, a significant one being the resolution of the crystal structure; this allowed data to be gathered only from high-quality experimental work.

A PGD query first filters the structures in its repositories for specified conditions, then searches for a specified structural motif comprised of at least one core residue. Preceding and succeeding residues can be added optionally. The search returns all dihedral angles for each returned residue. A further binning functionality is available for phi- and psi angles to aid the drawing of Ramachandran plots.

Queries were made for each amino acid in turn, and also to all residues preceding proline in the protein sequence, since it has been shown that these residues experience a significant and often overriding conformational influence due to the neighbouring presence of proline, regardless of their own side-chain(176). Hence, 21 queries were made in total. The filter criteria were mostly default: a minimum resolution of 1.2 Å, a maximum sequence identity of 25%, a minimum R-factor of 0.25 and a minimum R-free of 0.3.

For all queries, a 3-residue motif was specified. Residue i, the residue for which results would be collected, was set to each amino acid in turn, with residues i–1 and i+1 set to include all 20. By requiring the presence of residues i–1 and i+1, the search would return only non-terminal instances of residue i. For the pre-proline search, residue i was reset to all 20 amino acids, and residue i–1 was set to proline.

The phi- and psi angles returned by each of the 21 queries were binned into 3° squares and downloaded. All queries were performed on 5th August 2013; the raw file downloads are available in Appendix 1.

3.3 Post-processing

Unless stated otherwise, work on this analysis was carried out using MATLAB. For each query, the binned results were used to populate a 120x120 matrix, in effect forming a Ramachandran plot at 3° resolution. These were stored, treating them as experimentally-derived amino acid-specific standards of allowed and disallowed regions of phi-psi space.

The cpptraj module within AMBER was used to extract the phi- and psi angles of each residue for every frame of its trajectory, resulting in 10 000 phi-psi angle pairs per residue. For each residue in turn, the phi- and psi angles of each frame where binned into 3° squares and stored in a matrix, analogously to the generation of standards. Overlaying this data with the appropriate standard allowed for a visual indication of the conformational space the residue had sampled in the simulation. An example of this is shown in Figure 3.3, using SER134 from pyruvate dehydrogenase kinase 2, an arbitrarily chosen residue from the allosteric site.

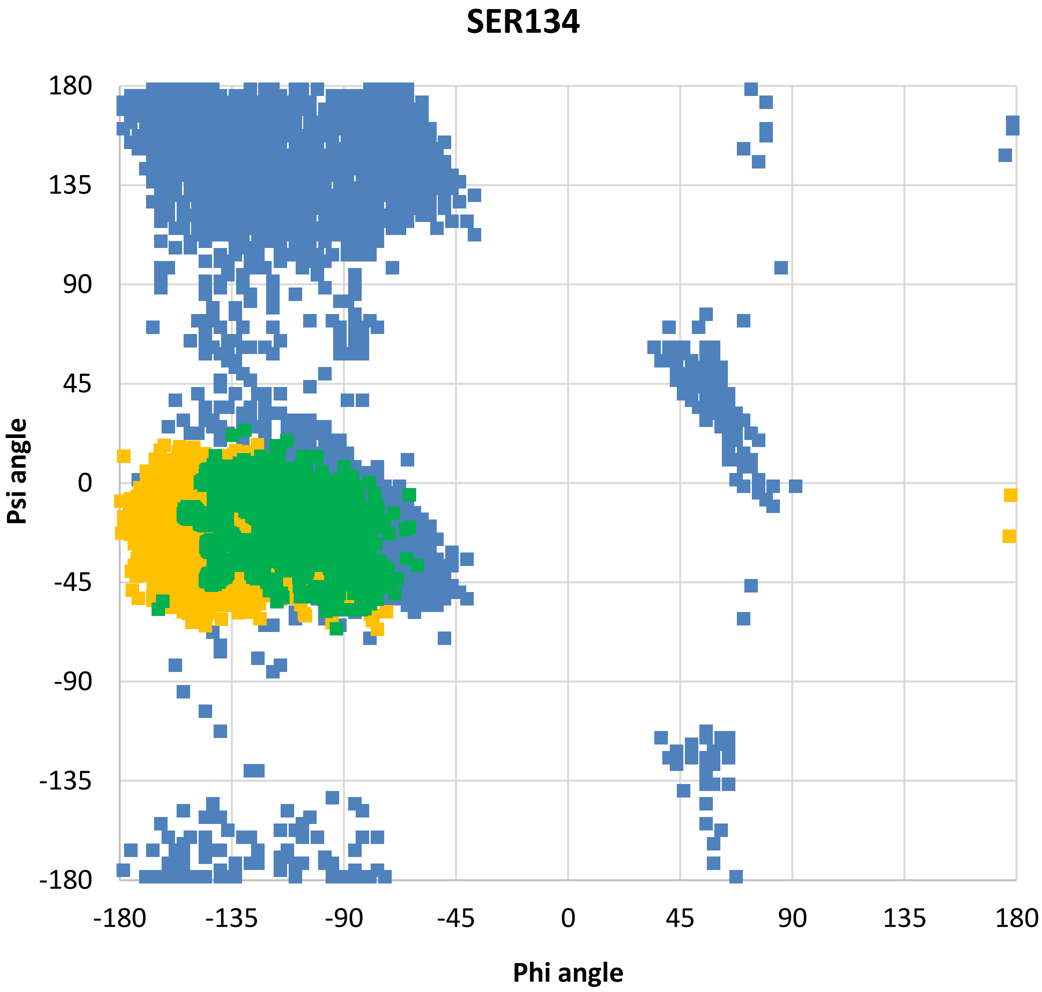


Figure 3.3: *An example of the monitoring of a residue's adopted phi-psi conformations throughout a MD trajectory. The standard allowed regions obtained for serine (blue) is shown with the allowed (green) and disallowed (yellow) conformations sampled by SER134 overlaid.*

This visual analysis conveys the journey of a residue in terms of conformational change highly effectively. Incorporating the frame numbers into the plot such that the path taken by the residue from start to finish could be seen would further enhance it. However, it was not feasible to manually view the data in this way for all residues in the dataset, of which there were approximately 32 000. Alternative analyses were required that could be run automatically. Besides logistical reasons, single values per residue were required if any information from this analysis was to be compatible with the RF training set.

3.4 Final Metrics

A number of metrics were devised to digest the vast quantities of information contained in this type of data. As stated, the final outputs were required to be one-dimensional for the purposes of use with RF. However, due to the information's complexity, it was decided to produce several metrics, each capturing some, rather than all, aspects of the data.

3.4.1 Number of Different Conformations

The first metric was a simple count of the number of different conformational bins (NDC) that were occupied by frames of the trajectory (see Figure 3.4 for output). The theoretical maximum value of this metric equals the number of frames in the analysed trajectory; if trajectories of varying frame totals or frame storage frequencies were to be analysed, the counts would have to be normalised for any comparison to be valid. This was not required for this work since all trajectories were 50 ns in duration and comprised of 10 000 frames.

The NDC score is indicative of a residue's ability to sample conformational space over the course of the trajectory. This information is related to but distinct from the residue's fluctuation, a major difference being that fluctuation accounts for the whole amino acid while the NDC score focuses on the main chain dihedral angles. The NDC score also only tracks unique conformations, whereas fluctuation can be increased by repeated oscillations around a relatively small conformational space.

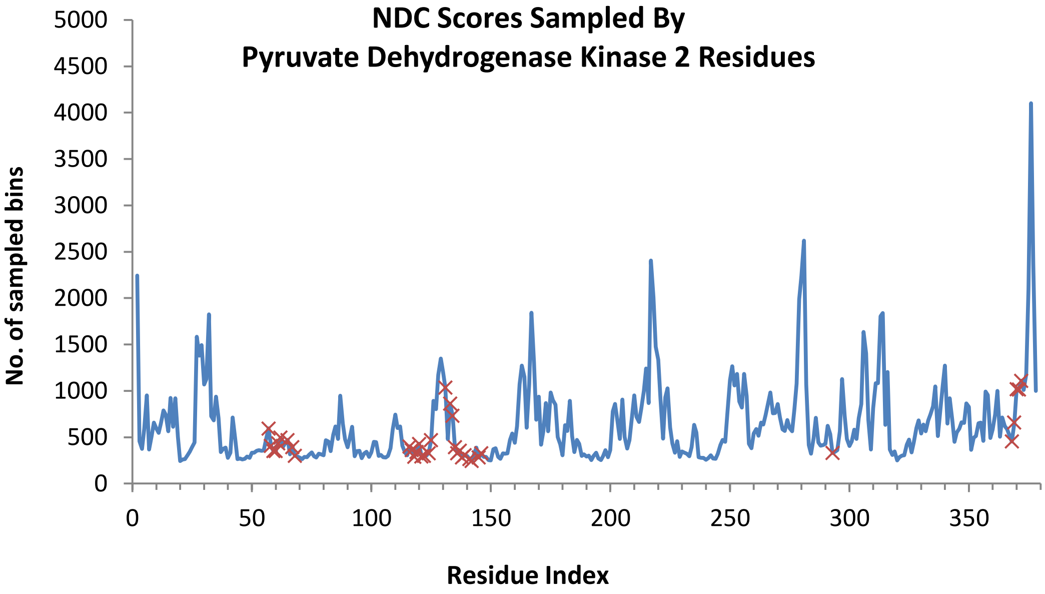


Figure 3.4: *The NDC scores of each residue of pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses.*

3.4.2 Ratio of Disallowed Conformations

As alluded to above, the NDC score does not account for the population of occupied bins, nor their positions in terms of allowed and disallowed areas of the Ramachandran plot. These aspects are taken into account for by the ratio of disallowed conformations (RDC). This measure sums the populations of each bin in a disallowed region and divides it by the total number of frames, yielding a number between 0 and 1. Over the course of a MD simulation, most residues in a protein will, transiently, sample a variety of disallowed states. Of more importance is the proportion of time spent in disallowed states, which is given by the RDC score. An example RDC output is presented in Figure 3.5.

This notion can be seen in the example result shown above in Figure 3.3: while the different phi-psi bins sampled by the residue can be clearly seen, the populations of these bins are not displayed. One cannot deduce how much time the residue spent in a disallowed state.

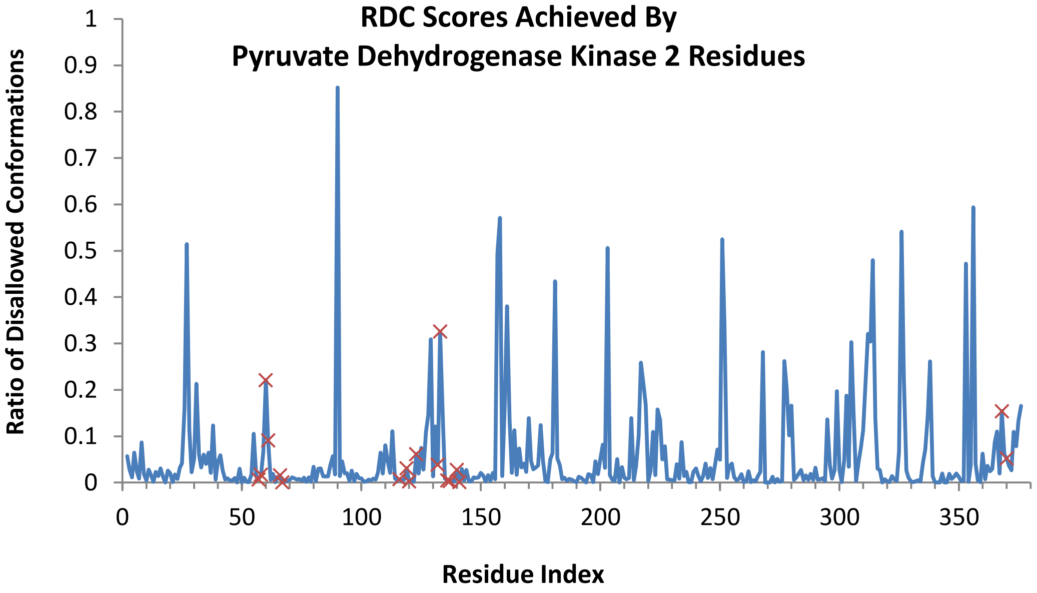


Figure 3.5: *The RDC scores of each residue of pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses.*

3.4.3 Maximum Distance to Allowed Region

A degree of coarseness exists with this type of phi-psi analysis, since the allowed regions upon which results are based are binned approximations, themselves based on a sample of known structures rather than comprehensive knowledge. Clearly, a phi-psi conformation positioned in a bin adjacent to the allowed region is far less significant than one far out in the disallowed region.

To quantify this notion, a method was required to monitor the distance of each occupied bin to the nearest allowed region. A k-nearest neighbour (k-NN) search was suitable for this purpose. Given two sets of points, A and B, the algorithm searches for the nearest point in A to each point in B. However, the distance measured by the function was the direct, Euclidean distance; there was no facility to account for periodic boundaries.

Periodic boundaries are discussed in the context of MD simulations in section 1.3.2, though the issue is apparent in the example serine residue discussed in section 3.3. Two points with phi angles close to, but under, 180° can be seen in Figure 3.6. The true, nearest allowed region lies at (-172.5,1.5) and is circled. However, the 'nearest' neighbouring point returned by a k-NN search would be further away than this. Arrows in Figure 3.6 mark these incorrect distances to the two points in question.

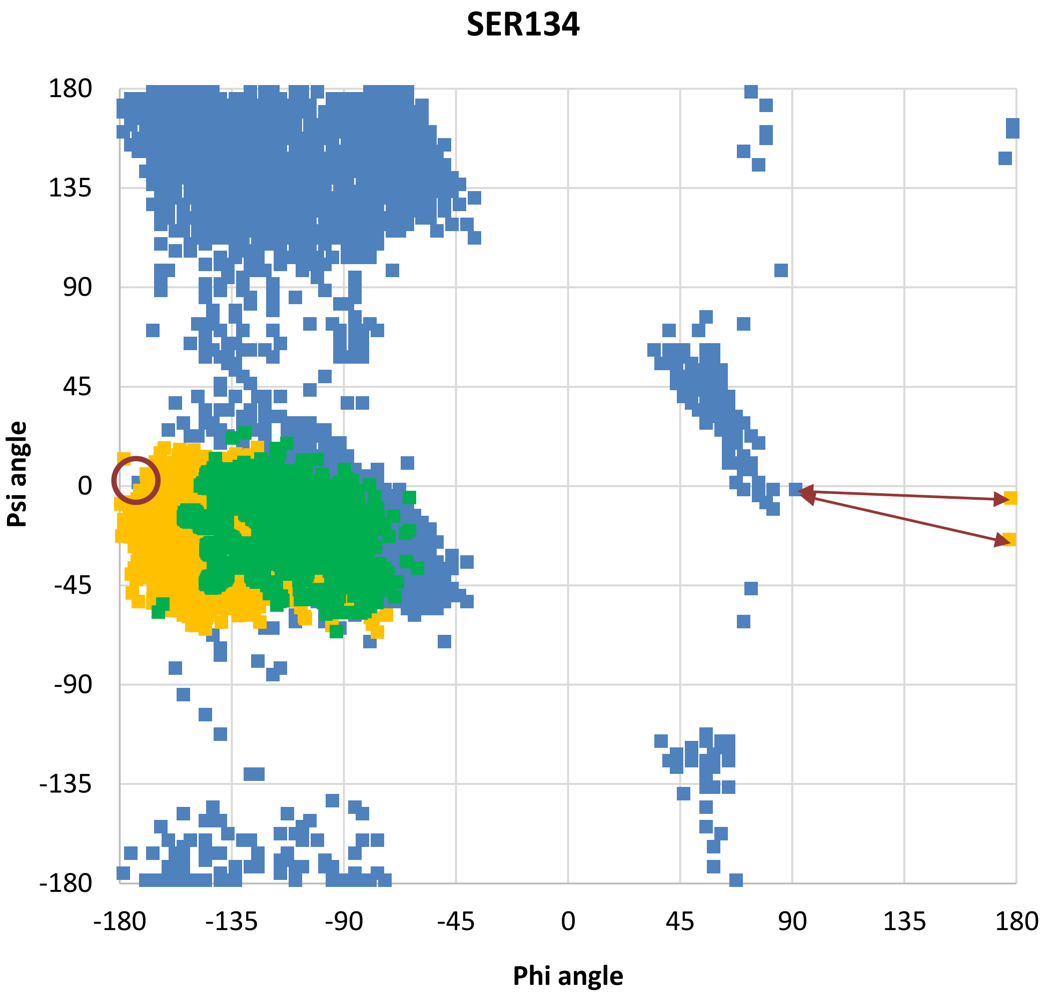


Figure 3.6: *The 'nearest' allowed region to the two points on the far right of the plot, as returned by a k-NN search, is marked with arrows. However, because the plot boundaries are periodic in reality, the true nearest allowed region is the one circled.*

A modification was required to allow the correct distances to be returned by the k-NN search. This could have either been made to the input data, allowing the 'naïve' k-NN search to return the correct points, or the function itself, allowing it to accommodate periodic boundaries: the former was chosen. For each standard 120x120 matrix of allowed regions, nine identical copies were concatenated into a single 360x360 matrix, forming a 3x3 tiling of the original. Figure 3.7 shows the result of this procedure for the serine standard.

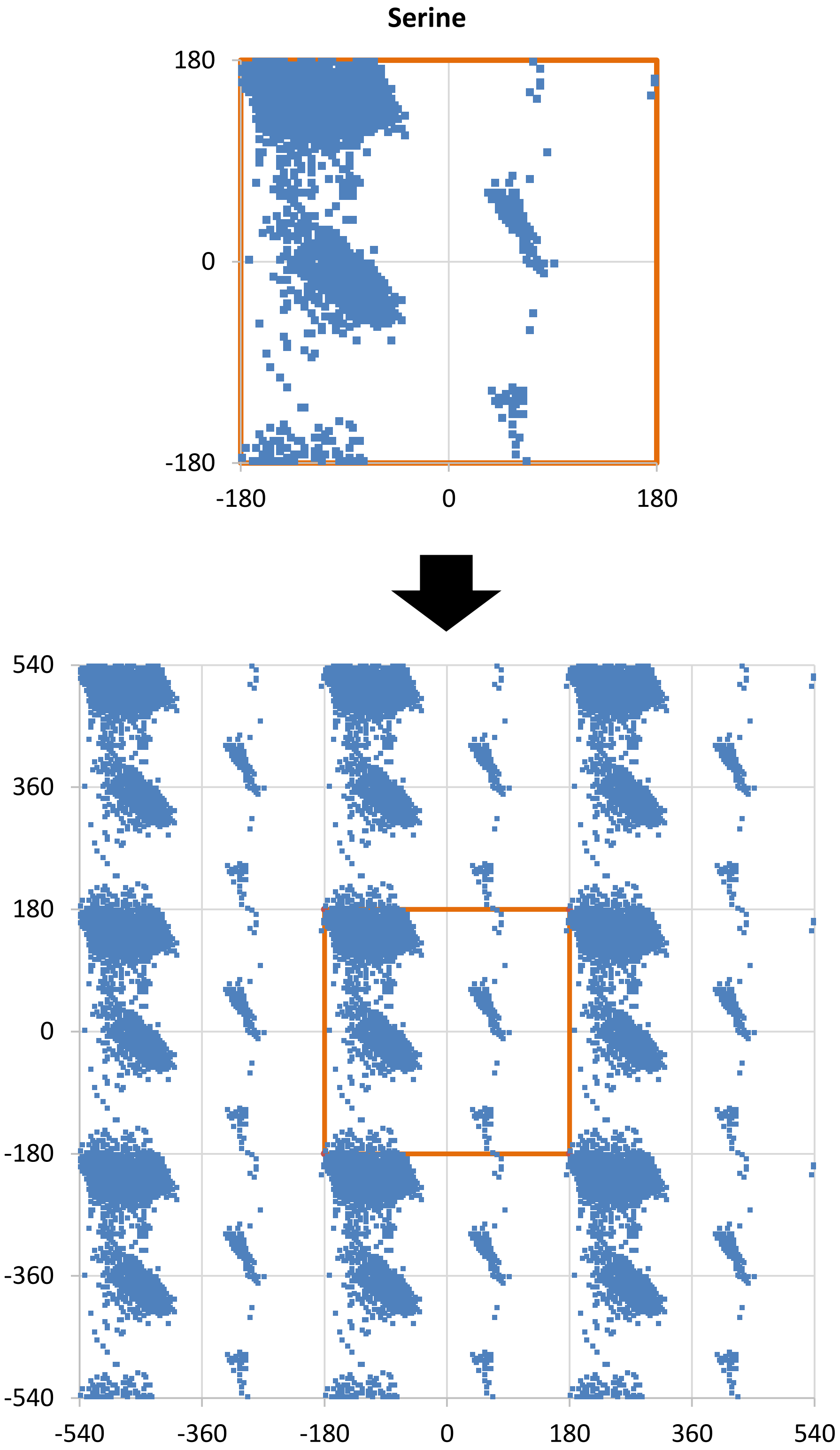


Figure 3.7: *A 3x3 tiling of the original standard allowed region for serine.*

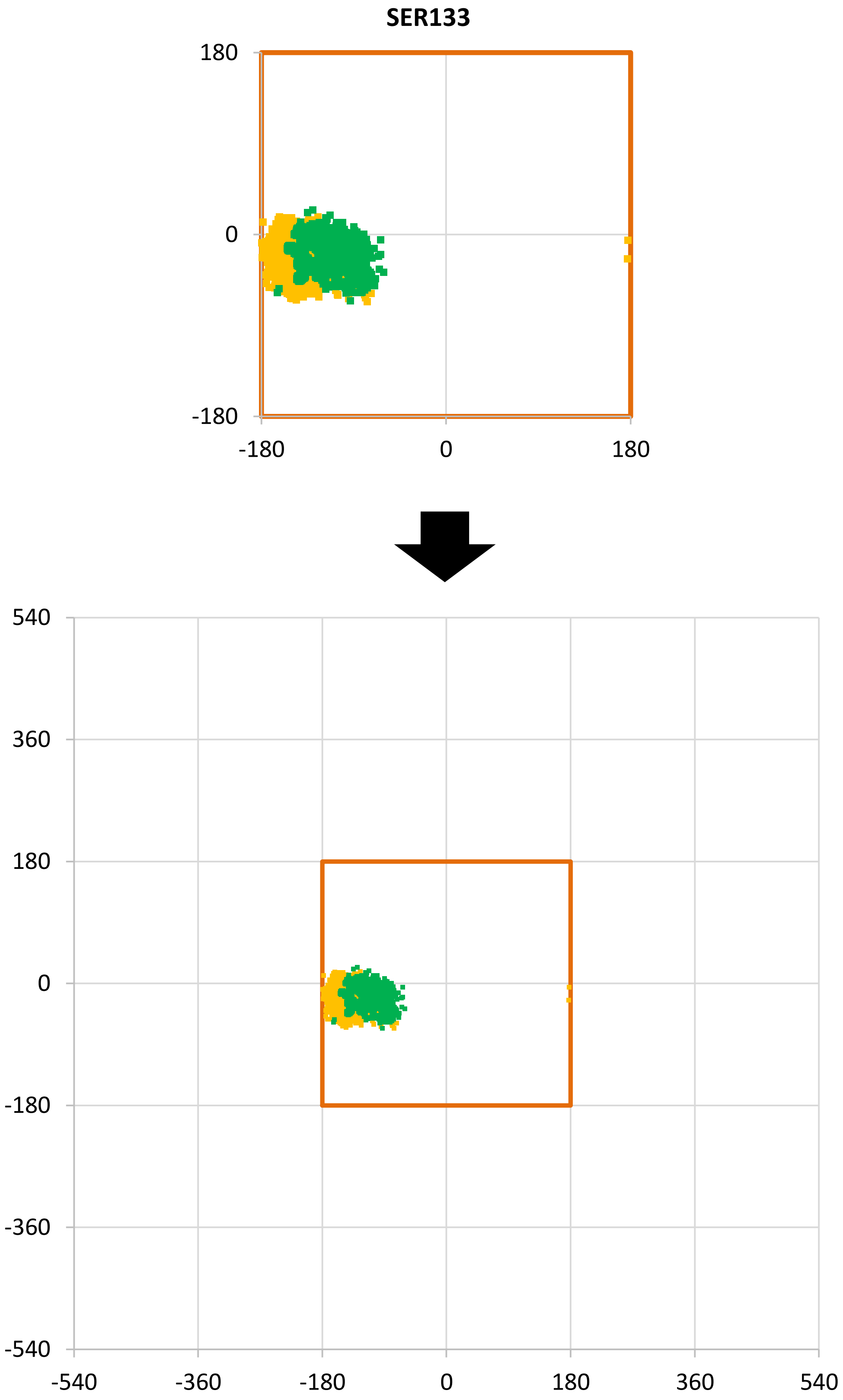


Figure 3.8: *To make it compatible with the 3x3 version of the standard allowed regions, empty tiles were concatenated to the trajectory data plot as shown.*

For trajectory data a second, empty 360x360 matrix was created, positioning the data in the central 120x120 tile, as shown in Figure 3.8.

These modified data sets contained an extra period of data in all dimensions. Since no measures were required that could theoretically cross two successive periodic boundaries, this was sufficient to allow the 'naïve' k-NN search to produce correct results despite treating the data with hard boundaries. It would iterate through all nine tiles of the standard allowed regions for each trajectory-occupied bin, with the nearest neighbouring point potentially located in one of the surrounding tiles. Figure 3.9 shows the problem with earlier example being resolved by this method.

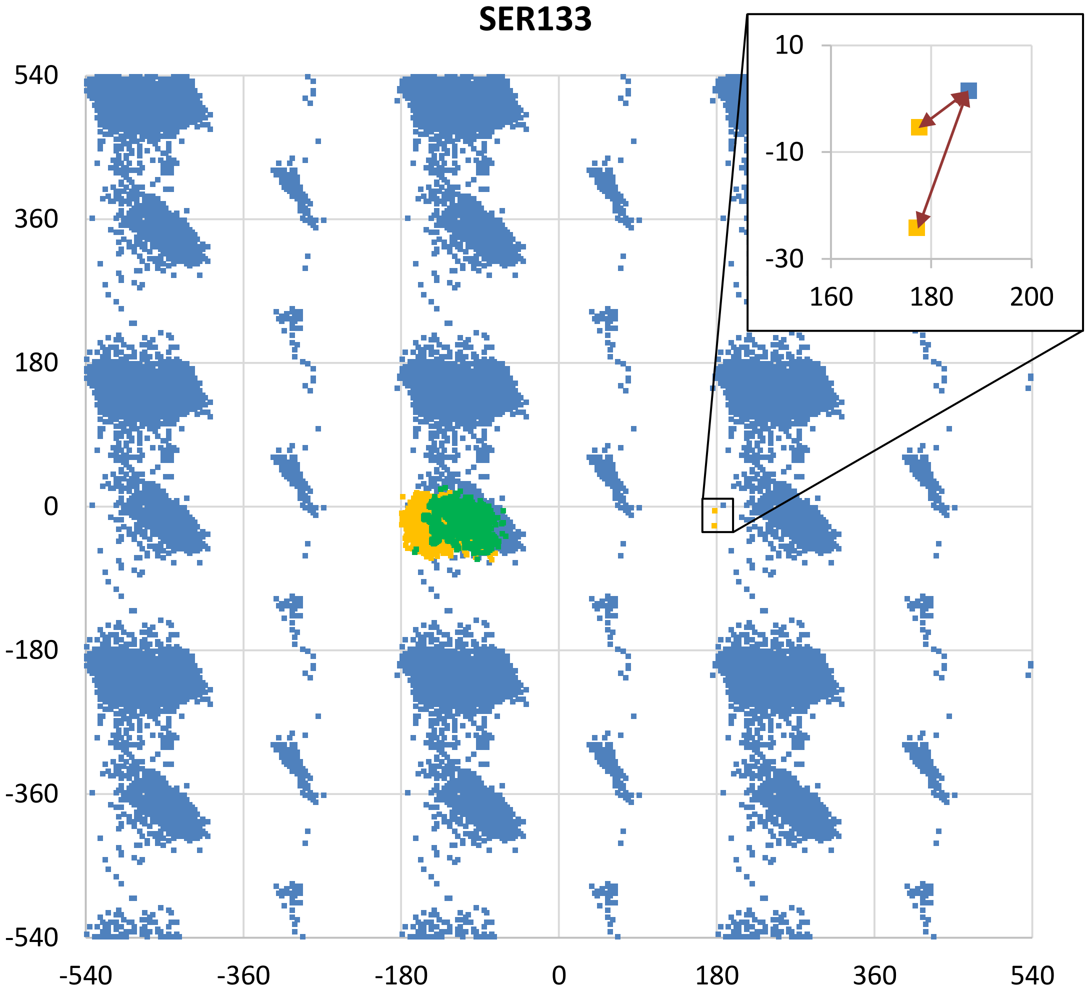


Figure 3.9: *When performed on the modified data, the k-NN search returns the correct distances for all nearest-neighbour pairs, including those that cross a periodic boundary in the original data.*

A simple metric utilising this data was the maximum distance from an allowed region (MaxDist). For each occupied bin, the distance to the nearest allowed region was found by the method detailed above and the maximum found. This could be considered a measure of the maximum degree of 'disallowance' experienced by residues, which could in turn potentially translate into an energy gain upon interaction with a ligand. An example plot of MaxDist scores is shown in Figure 3.10. It should be noted that the distance values were based on a 120x120 matrix and so do not correspond to the scale of a Ramachandran plot.

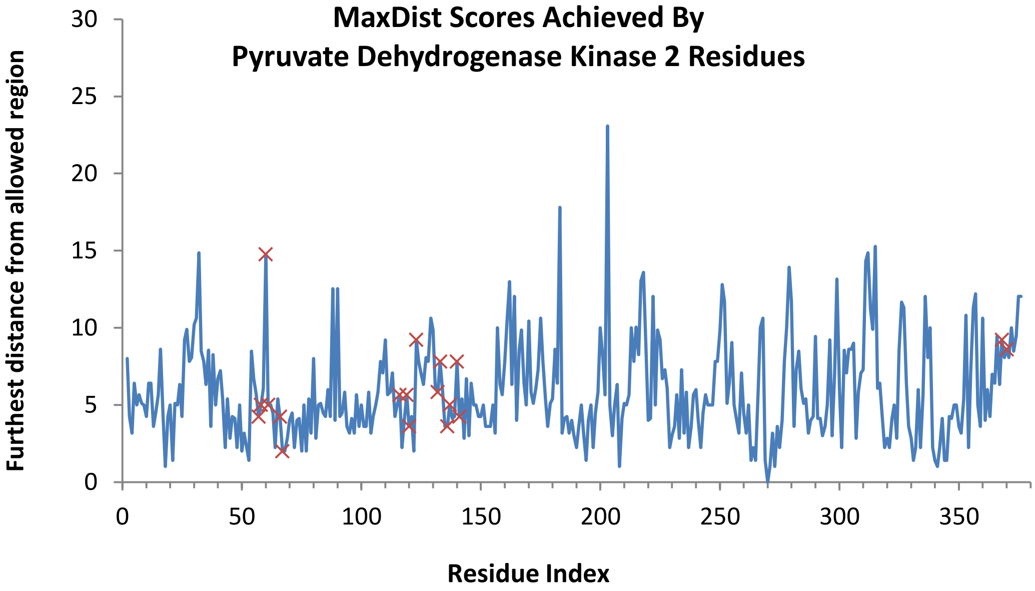


Figure 3.10: *The MaxDist scores achieved by each residue of pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses.*

3.4.4 Sum of Bin Populations Weighted by Distance

A final metric was developed to balance the number of frames a residue spent in a given phi-psi conformation with the degree of 'disallowance' of that conformation. The intention was to score residues such that those with a similar set of scores (but different phi-psi behaviours) could be further resolved. For example, residues with a large number of frames close to, but nevertheless outside, allowed regions, would exhibit high RDC scores. In many cases this could be due to the natural 'breathing' of the protein structure throughout the trajectory, while in others some local phenomenon could be the causing a shift in conformation. Similarly, residues transitioning between two low-energy, allowed states may exhibit a high MaxDist score if they pass through a high energy, disallowed state. This scenario is distinct from one in which a residue favours a disallowed conformation for an extended period of time.

Translating the above into terms of the available data, the metric had to incorporate the population of conformational bins sampled by each residue with the distance to the nearest allowed region. Thus, for each residue, the product of bin population and distance to the nearest allowed region was taken and summed for each bin sampled over the trajectory:

$$BD_i = \sum_{b=1}^{n} (p_b \times d_{AR}) \tag{18}$$

where BD_i = the final score based on bin populations and distances (BD) for residue i, p_b = the population of a bin of trajectory phi-psi conformations, b, and d_{AR} = distance to the nearest allowed region. This effectively weighted each conformation by its distance to the nearest allowed region, and allowed conformations were naturally eliminated due to the distance of zero. No normalisation was required, since measures were already being taken against the same set of standard allowed regions.

The BD score represented a combined measure of the frequency and severity of torsional 'frustration' experienced by a residue over the course of a trajectory. An example plot of BD scores is shown in Figure 3.11.

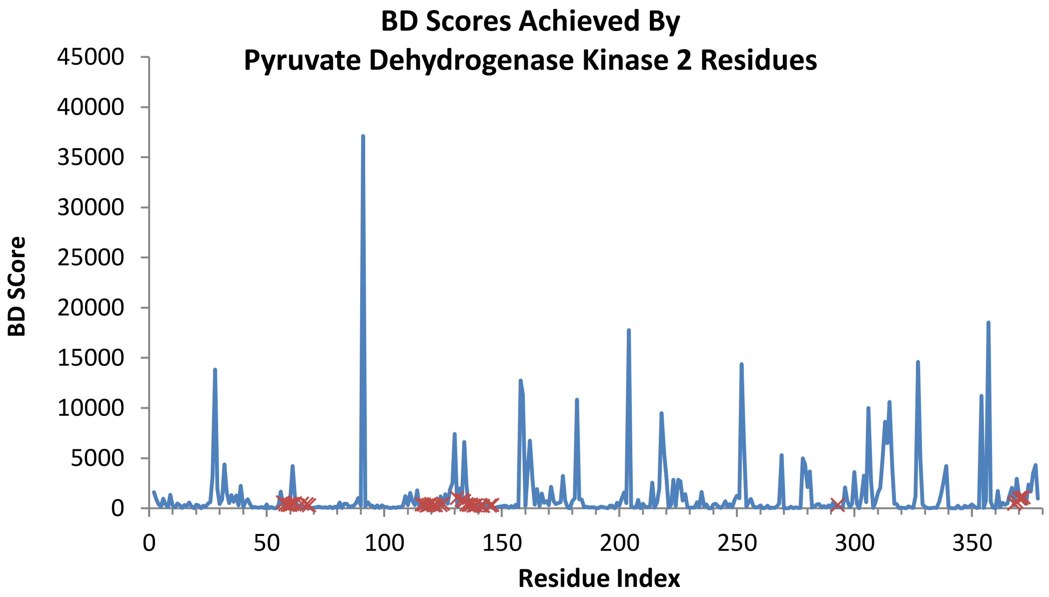


Figure 3.11: *The calculated BD scores for each residue of pyruvate dehydrogenase kinase 2. The residues of the allosteric site are marked with red crosses.*

3.5 Summary

This analysis examined the dynamic behaviour of the main chain torsional motions of protein residues. This was achieved by first defining regions of standard allowed Ramachandran space for each individual amino acid. The dihedral angles of each protein residue could then be monitored against the appropriate standard over the course of its MD trajectory.

Four metrics were developed to quantify different aspects of torsional motions. The NDC score gives an indication of the quantity of conformation space sampled over the trajectory, while the RDC score measures the proportion of the simulation each residue spent in a disallowed state. The MaxDist score monitored the severity of the disallowance or torsional 'frustration' experienced by a residue, while the BD score also incorporated its frequency.

The latter two scores in particular required unusual workarounds to obtain fully accurate data within the environment of MATLAB. However, they added significantly to the overall picture of residue behaviour. Most importantly, this novel analysis added an entirely new data type to the dataset, and would surely benefit the performance of future RF models.

4. Results and Discussion

4.1 Final Selection of Proteins

At total of 60 proteins were gathered and used in the dataset for this project. Their PDB codes are listed below.

PDB Code	***Protein***	***Species***	***Allosteric ligand in structure?***	***Orthosteric ligand in structure?***
1FA9	liver glycogen phosphorylase A	*Homo sapiens*	yes	yes
1FIY	phosphoenolpyruvate carboxylase	*Escherichia coli*	yes	no - found in 1JQN
1H5U	muscle glycogen phosphorylase A	*Oryctolagus cuniculus*	yes	yes
1JLR	uracil phosphoribosyltransferase	*Toxoplasma gondii*	yes	no - found in 1JLS
1LDN	lactate dehydrogenase	*Geobacillus stearothermophilus*	yes	yes
1PFK	phosphofructokinase	*Escherichia coli*	yes	yes
1PZP	TEM-1 beta-lactamase	*Escherichia coli*	yes	no - found in 1M40
1S9I	MAP kinase kinase 2 (MEK2)	*Homo sapiens*	yes	yes
1T4J	protein tyrosine phosphatase 1B	*Homo sapiens*	yes	no - found in 1PTY
1V4T	glucokinase	*Homo sapiens*	no - found in 1V4S	no - found in 1V4S
1W96	acetyl-CoA carboxylase	*Saccharomyces cerevisiae*	yes	no - found in 1DV2
2BKK	aminoglycoside phosphotransferase	*Enterococcus faecalis*	yes	yes
2BU2	pyruvate dehydrogenase kinase 2	*Homo sapiens*	no - found in 2BU7	yes
2I80	D-ala D-ala ligase	*Staphylococcus aureus*	yes	no - found in 2I8C
2JFZ	glutamate racemase	*Helicobacter pylori*	yes	yes
2P9H	lac operon repressor	*Escherichia coli*	yes	no - found in 1JWL
2PIT	androgen receptor	*Homo sapiens*	yes	yes
2PUV	glucosamine-6-phosphate synthase	*Candida albicans*	yes	yes
2V4Y	UMP kinase	*Escherichia coli*	yes	no - found in 2BNE
2VGB	erythrocyte pyruvate kinase	*Homo sapiens*	yes	yes
2XCW	cytosolic 5'-nucleotidase II	*Homo sapiens*	yes	yes

PDB Code	*Protein*	*Species*	*Allosteric ligand in structure?*	*Orthosteric ligand in structure?*
2XO8	myosin ATPase	*Dictyostelium discoideum*	yes	yes
2YC3	2-C-methyl-D-erythritol 4-phosphate cytidylyltransferase (IspD)	*Arabidopsis thaliana*	yes	no - found in 1W77
2ZD1	HIV-1 reverse transcriptase	*Human immunodeficiency virus 1*	yes	no - found in 1RTD
3ALO	MAP kinase kinase 4 (MKK4)	*Homo sapiens*	yes	yes
3DC2	d-3-phosphoglycerate dehydrogenase	*Mycobacterium tuberculosis*	yes	no - found in 3DDN
3ELJ	MAP kinase 8 (JNK1)	*Homo sapiens*	no - found in 3O2M	yes
3EPS	isocitrate dehydrogenase kinase/phosphatase	*Escherichia coli*	yes	yes
3F3U	proto-oncogene tyrosine-protein kinase Src (c-Src)	*Homo sapiens*	yes	no - found in 2QLQ
3FIG	alpha-isopropylmalate synthase	*Mycobacterium tuberculosis*	yes	no - found in 1SR9
3H30	Ser/Thr kinase CK2	*Homo sapiens*	yes	no - found in 1LP4
3HRF	3-phosphoinositide-dependent protein kinase-1 (PDK1)	*Homo sapiens*	yes	yes
3IFC	muscle fructose-1,6-bisphosphatase	*Homo sapiens*	yes	yes
3IJG	macrophage migration inhibitory factor	*Homo sapiens*	yes	no - found in 3IJJ
3JVR	Checkpoint kinase 1 (CHK1)	*Homo sapiens*	yes	no - found in 3TKH
3K5V	Bcr-Abl tyrosine kinase	*Mus musculus*	yes	yes
3LW0	insulin-like growth factor 1 receptor kinase	*Homo sapiens*	yes	no - found in 2OJ9
3O96	RAC-alpha Ser/Thr kinase (AKT1)	*Homo sapiens*	yes	no - found in 3CQW
3PJG	UDP-glucose dehydrogenase	*Klebsiella pneumoniae*	yes	yes
3PY1	cyclin-dependent kinase 2 (CDK2)	*Homo sapiens*	yes	yes
3R1R	ribonucleotide reductase protein R1	*Escherichia coli*	yes	no - found in 4R1R
3TYQ	NS5B polymerase	*Hepatitis C virus*	no - found in 1GX5	yes - also found in 1GX5

PDB Code	***Protein***	***Species***	***Allosteric ligand in structure?***	***Orthosteric ligand in structure?***
3U69	thrombin	*Homo sapiens*	yes and no	no - found in 4RKO
3UO9	glutaminase	*Homo sapiens*	yes	yes
3V55	Mucosa-associated lymphoid tissue lymphoma translocation protein 1	*Homo sapiens*	no - found in 4I1R	no - found in 3V4O
3ZG0	penicillin binding protein 2A	*Staphylococcus aureus*	yes	yes
3ZLK	glucose-1-phosphate thymidylyltransferase (RmlA)	*Pseudomonas aeruginosa*	yes	no - found in 1G0R
4A1Z	mitotic kinesin Eg5	*Homo sapiens*	no - found in 3ZCW and 2X2R	yes
4AVC	protein lysine acetyltransferase (PAT)	*Mycobacterium tuberculosis*	yes	yes
4BNY	3-oxoacyl-(acyl-carrier-protein) reductase (FabG)	*Pseudomonas aeruginosa*	yes	no - found in 4AG3
4BQH	Uridine diphosphate N-acetylglucosamine pyrophosphorylase (UAP)	*Trypanosoma brucei*	yes	no - found in 1JV1
4CSM	chorismate mutase	*Saccharomyces cerevisiae*	yes	yes
4EBV	focal adhesion kinase	*Homo sapiens*	yes	no - found in 2J0L
4M15	interleukin-2-inducible T-cell kinase (ITK)	*Homo sapiens*	yes	yes
4M19	dihydrodipicolinate synthase	*Campylobacter jejuni*	yes	yes
4NL1	dihydropteroate synthase	*Bacillus anthracis*	yes	yes
4P9D	deoxycytidylate deaminase	*Cyanophage S-TIM5*	yes	yes
4R5I	chaperone DnaK	*Escherichia coli*	no - found in 4R5G	yes
4RYL	protein arginine methyltransferase 3 (PRMT3)	*Homo sapiens*	yes	no - found in 1ORI
4U4X	AMPA receptor subunit GluA2	*Rattus norvegicus*	yes	yes

Table 4.1: *The 60 proteins selected for inclusion in the dataset. PDB codes shaded in a darker blue link to protein kinases. Where ligand(s) were not present in the chosen structure, they were superimposed from another structure that did contain them for the purposes of defining the binding sites.*

4.2 Preliminary Analysis of Protein Set

This section of work sought to characterise the collected protein set with the intention of validating, as far as possible, the use of it as a representative sample training set of allosteric proteins.

4.2.1 Protein Diversity

A total of 60 proteins were included in the allosteric set. Of these, 14 were protein kinases. To quantify the diversity of the set, a sequence alignment was performed on each pair of proteins using the ClustalW algorithm(181). The values from the resulting percentage-identity matrix were used to render a heat map for presentation purposes, shown in Figure 4.1. The overall similarity across the proteins was observed to be weak, with a mean identity of 15.8%. As would be anticipated, a higher mean similarity of 32.2% was observed across the protein kinases within the dataset.

An exception to the overall low similarity was found with the first and third proteins in the heat map (PDB codes: 1FA9 and 1H5U). This was anticipated: the proteins were the two isoforms of glycogen phosphorylase discussed in section 2.1.6.

An enlarged, electronic version of this matrix is available in Appendix 2, as is a version with the identities displayed numerically.

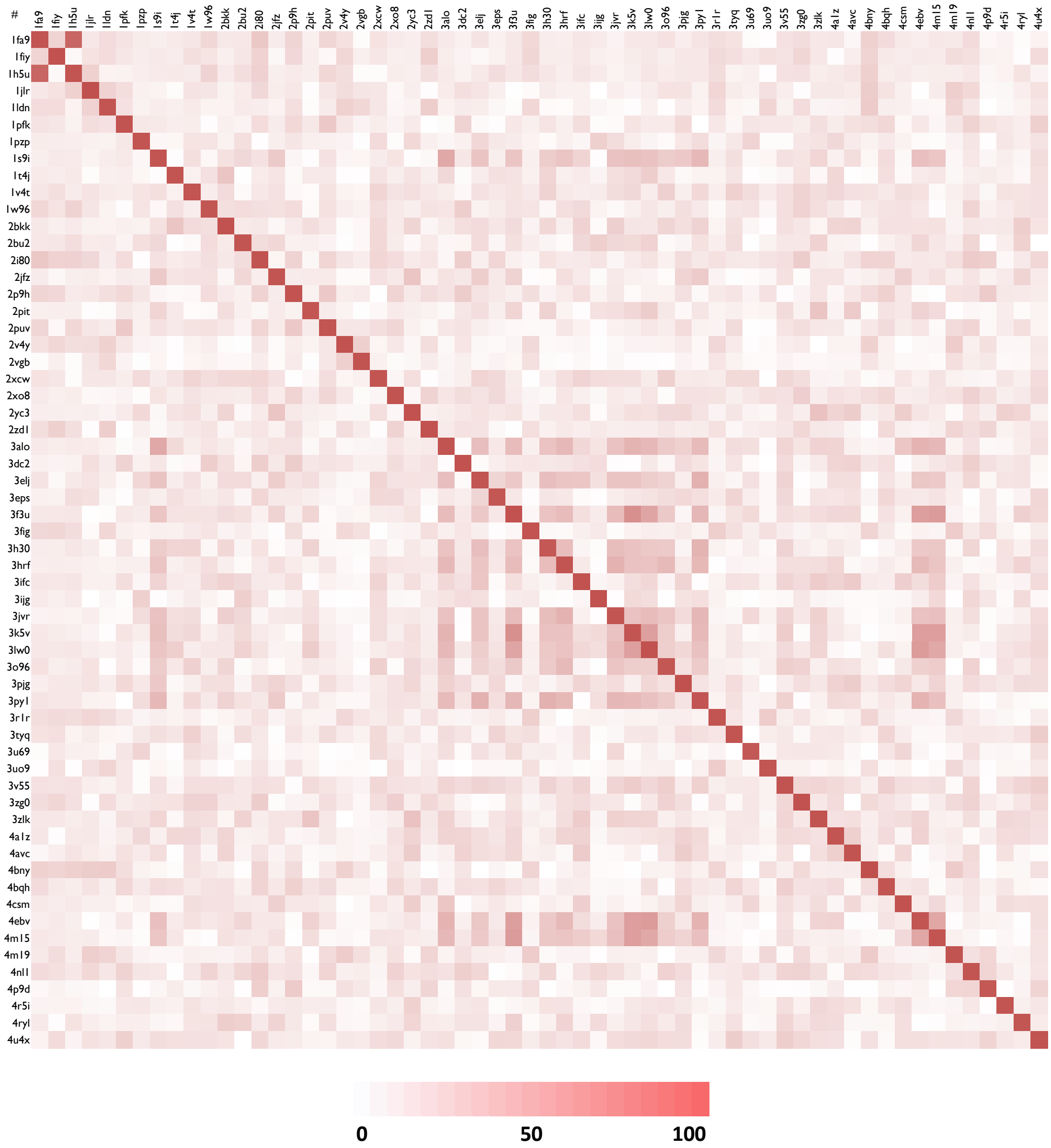


Figure 4.1: *Matrix of percentage identities for each pair of the proteins in the dataset, shown as a heat map. An enlarged, electronic version of this matrix is available in Appendix 2, as is a version with the identities displayed numerically.*

4.2.2 Residue Abundances

To validate the proportionality of the amino acid make-up in the dataset, a residue count of the entire PDB repository was also performed (PDB sequence repository download date: 21/05/2015). This would serve as an estimation of the natural abundances of amino acids.

While the PDB is certainly a large enough repository of protein sequences to provide this estimation, it must be considered that the population of residues in the PDB are likely to be skewed. Firstly, by virtue of it being a repository of protein *structures*, the PDB is likely to contain an overrepresented sample of protein sequences that are easily crystallisable. It is also likely to contain an overrepresented sample of proteins that are of high interest to researchers, such as GPCRs and protein kinases. The PDB also does not limit the number of structures of a given protein. Thus the residues of many proteins were counted multiple times. Multiple copies of proteins can also be present within one PDB entry.

While some of these issues could theoretically be addressed, the format of the data dump contained no metadata beyond the PDB code of each sequence. There was therefore no quick method to filter out sequences.

The UniProt Archive (UniParc)(182) presented an alternative to the PDB for use in this exercise. Since a UniParc entry does not require a crystal structure, it is a far more expansive than the PDB. It is also non-redundant; it is as close to a comprehensive database of protein sequences as can be found. An equivalent residue count of this database was also performed (download date: 03/06/2015).

The abundances found in the project dataset, the PDB and UniParc are shown below in table format (Table 4.2), followed by the same data in plot format (Figure 4.2), followed by a plot of the differences between the dataset and databases for each amino acid (Figure 4.3).

	Background Abundances		
Residue	**Dataset**	**PDB**	**UniParc**
ALA	8.2%	8.1%	8.9%
VAL	7.5%	7.1%	6.7%
ILE	6.3%	5.6%	5.6%
LEU	9.4%	8.9%	9.9%
MET	2.7%	2.4%	2.3%
PHE	3.7%	3.9%	3.9%
TRP	1.1%	1.3%	1.3%
TYR	3.5%	3.4%	2.9%
ARG	5.3%	5.2%	5.7%
HIS	2.1%	2.7%	2.2%
LYS	6.1%	5.9%	5.1%
ASP	5.6%	5.6%	5.4%
GLU	7.0%	6.7%	6.2%
SER	5.9%	6.2%	6.9%
THR	4.9%	5.6%	5.6%
ASN	4.1%	4.2%	3.9%
GLN	3.4%	3.7%	3.9%
CYS	1.4%	1.3%	1.3%
GLY	7.3%	7.5%	7.2%
PRO	4.4%	4.7%	5.0%

Table 4.2: *The % abundances of each amino acid found in the manually-curated 60-protein dataset, the PDB and UniParc.*

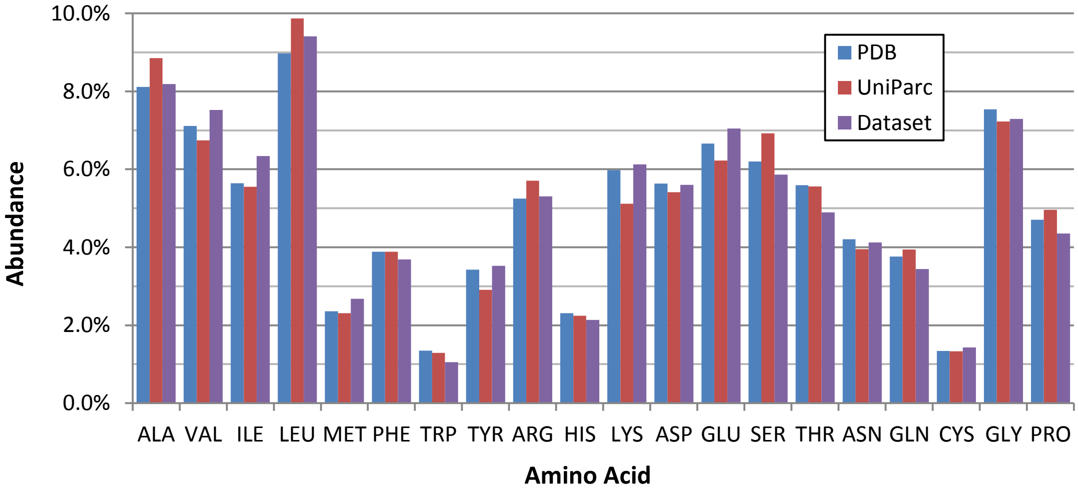


Figure 4.2: *The % abundances of each amino acid found in the dataset, the PDB and UniParc.*

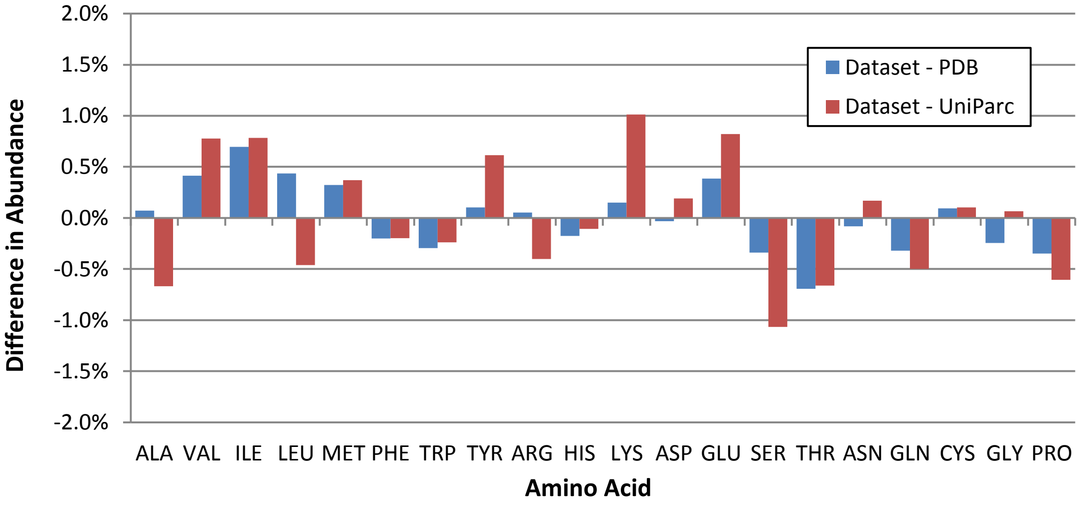


Figure 4.3: *The absolute differences in % abundances of amino acids in the dataset from those in the entire PDB and UniParc databases.*

Against the PDB, Ile, Val, Leu and Glu were the most overrepresented and Thr and Gln were the most underrepresented, but all abundances deviated from the abundance in the PDB by less than 1%. There were larger variations between the dataset and UniParc. However, considering that the dataset only contained 60 proteins whereas the PDB and UniParc contained approximately 78 000 and 93 000 000 sequences, respectively, the abundances were fairly closely aligned. It was thus decided to further investigate the constitution of allosteric and orthosteric binding sites in terms of residue abundances.

4.2.3 Constitution of Binding Sites

As previously described, residues with atom(s) within a given radius of a ligand's atom(s) were defined as part of that ligand's binding site. The identity of each residue in the dataset was logged, as was its designation as part of an allosteric site, orthosteric site or neither. The radius was then varied from 4 Å to 30 Å; this excessive upper limit was used to observe any detected local trends blending into background noise as the radius took in larger proportions of the protein. The lower bound of 4 Å was chosen as the smallest integer distance that would include residues interacting with the bound ligand through hydrogen bonding: since hydrogen atoms were not present in crystal structures, the distance between atoms involved in hydrogen bonding was often $>$3 Å.

The changing make-up of binding sites against selection radius was then compared to the background of the dataset by subtracting each amino acid's background abundance from its abundance in a defined binding site. These results are presented in Figure 4.4 and Figure 4.5. In each graph the line plots are coloured according to chemical properties (aromatic, aliphatic, acidic, etc.) and shaded for each amino acid. Electronic versions of the graphs are available in Appendix 3, should the reader wish to examine them in greater detail.

Allosteric sites (Figure 4.4) were found to contain a larger relative proportion of aliphatic and aromatic amino acids and a reduced proportion of hydrophilic and acidic amino acids, consistent with other studies that suggest an increased hydrophobic character of allosteric pockets(35). Conversely, orthosteric sites displayed a more hydrophilic profile (Figure 4.5). Met, Tyr and Gly also stood out as overrepresented. The particularly great overabundance of Gly was noteworthy. Since glycine can be considered to bear empty space as its 'sidechain,' this is evidence for the notion that orthosteric cavities are generally larger and more flexible than other areas of protein.

In both the allosteric and orthosteric cases the broad trends evolved over the lower selection radii of 4-7 Å, beyond which they began to blend into the background. Based on this result, 7 Å was selected as the most suitable radius for site definition. Interestingly, this value of 7 Å is in agreement with previous studies suggesting it to be optimal for defining the sphere of influence of an amino acid(169, 170).

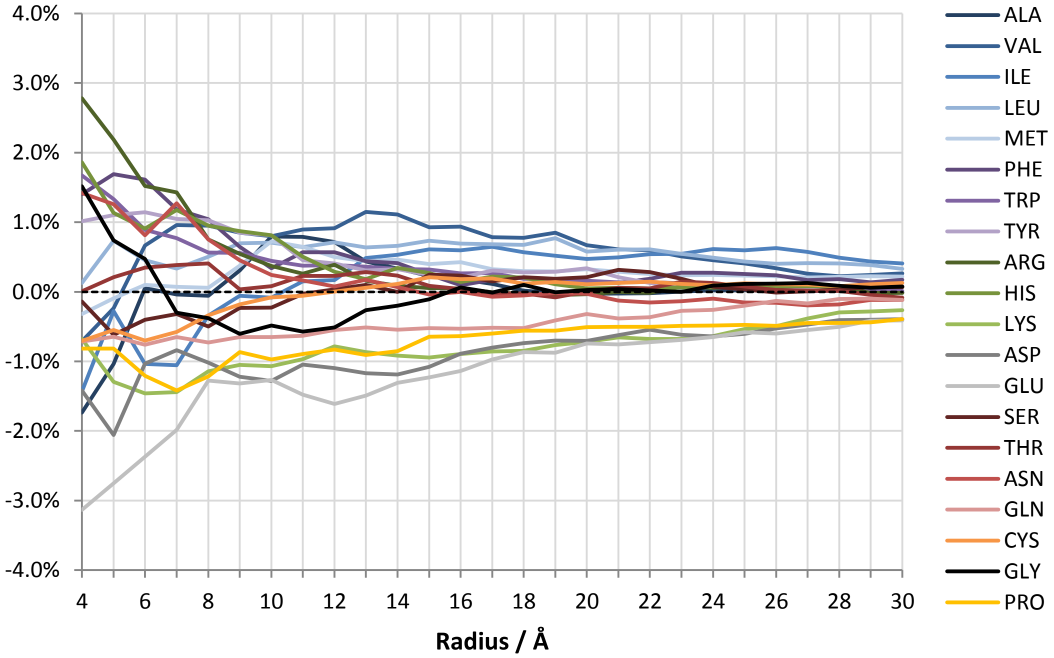


Figure 4.4: *The deviation in % abundance of amino acids found in allosteric sites from the background on varying the selection radius. Each series is coloured by the main property of the amino acid (hydrophilic, aromatic, acidic, etc.)*

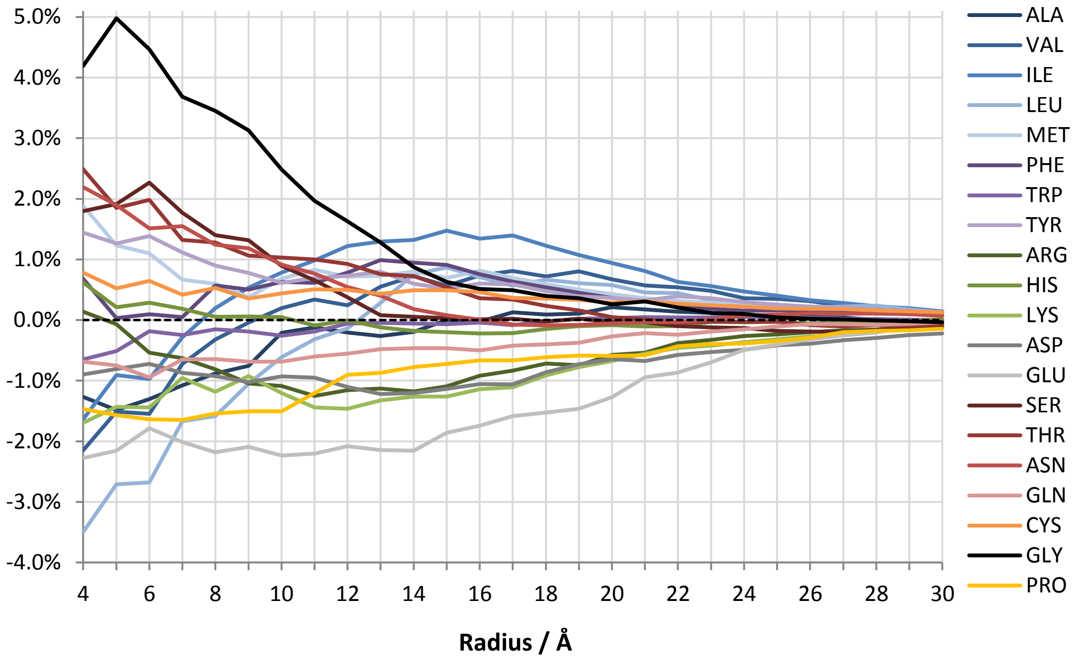


Figure 4.5: *The deviation in % abundance of amino acid in orthosteric sites from the background on varying the selection radius. Each series is coloured by the main property of the amino acid (hydrophilic, aromatic, acidic, etc.)*

An interesting way to collate some of the above observations was to examine all residues in a defined binding site together by molecular weight. For the dataset, the weighted average mass of a residue was calculated to be 129.38 g mol^{-1}. Figure 4.6 tracks the average molecular mass of residues in each type of binding site in the dataset over the range of selection radii. Allosteric sites were found to be heavier than average; this was due to the overabundance of heavy residues such as Trp, Tyr, Phe and Arg. Conversely, orthosteric sites were found to be lighter than average. This was primarily due to the great overabundance of Gly, the lightest amino acid, but also due to increases in Ser and Thr, which are also among the lighter amino acids.

These results show that allosteric cavities are generally smaller than their orthosteric counterparts, since a larger molecular mass (and so an approximately larger molecular volume) is contained in the same selection radius. This supports the complementary study by Van Westen *et al.* that found allosteric ligands to be generally smaller than orthosteric ligands(54).

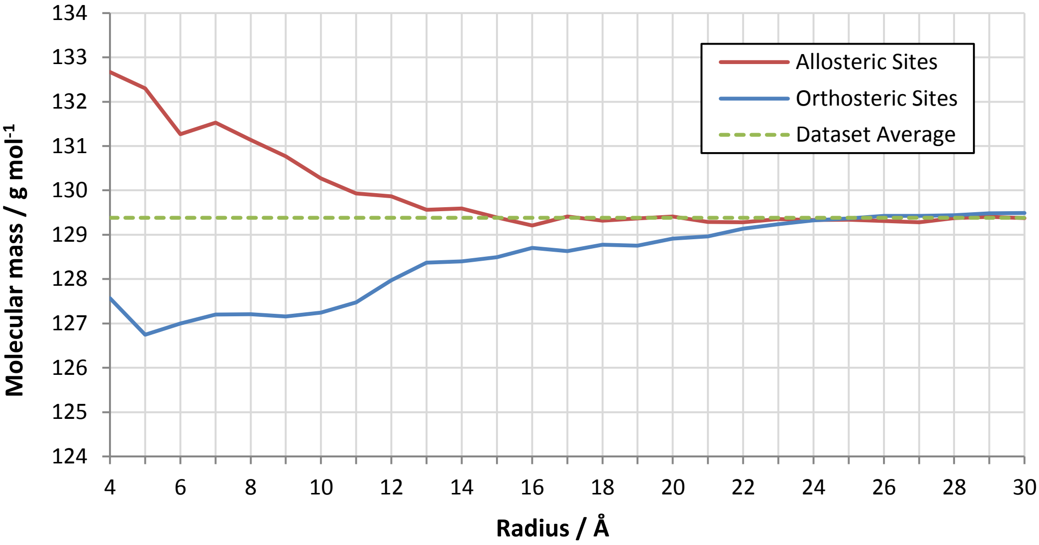


Figure 4.6: *The deviation in % abundance of each amino acid found in allosteric sites and orthosteric sites (defined by a 7 Å selection radius) from the background abundances of the dataset.*

4.3 Descriptor Generation

A 50 ns production phase of MD was performed for each of the 60 chosen proteins. The full battery of analytical methods detailed in section 2.4 was then performed on every trajectory. The data generated were archived on a per-analysis-per-protein basis. With the aid of a custom, fully modular Python script, the results of any combination of analyses on any combination of proteins could be compiled into a training set as required.

Extensive automation of the workflow (Figure 4.7) was necessary to produce the complete dataset within the timescale of this project; this was accomplished through the development of a number of custom scripts, written in Python, Bash and MATLAB code. A small number of Perl scripts written by Dr Nahoum Anthony were also used, as were Pipeline Pilot protocols written with the aid of Dr Murray Robertson. The final product was a database of individual protein residues, an array of descriptors associated with each and a final classifier designating them as part of an allosteric site or not. Appendix 4 contains all descriptors generated, formatted as a tab-delimited text file.

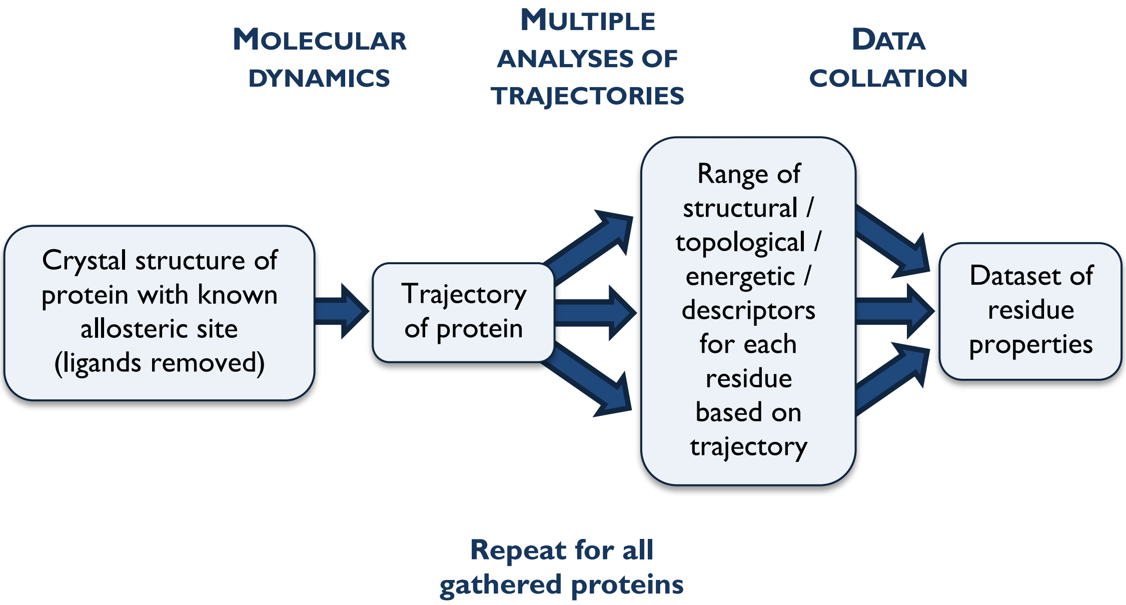

MOLECULAR DYNAMICS
MULTIPLE ANALYSES OF TRAJECTORIES
DATA COLLATION
Crystal structure of protein with known allosteric site (ligands removed)
Trajectory of protein
Range of structural / topological / energetic / descriptors for each residue based on trajectory
Dataset of residue properties
Repeat for all gathered proteins

Figure 4.7: *Schematic of the main workflow stages of dataset construction.*

The above schematic (rightly) appears simple in principle. However, in reality, this was a challenging and intensive task that much of the project's time was devoted to. As a result, it became feasible to recalculate descriptors if necessary and reformat the entire dataset as required.

In generating this quantity of data, errors and setbacks were not only expected but inevitable; without investing the time to develop the wokflow's 'agility' – its capacity to cycle through the different stages of analysis efficiently, reliably and with minimal supervision – it would have been impossible to trial successive models and test multiple hypotheses.

4.4 Random Forest Optimisation

The small number of user-modifiable parameters in RF allowed for a computationally feasible optimisation protocol on a small dataset. Individual forests were trained on 8-protein training sets and tested on two proteins, PDK1 and glutamate racemase (PDB codes: 3HRF and 2JFZ, respectively), chosen as arbitrary examples of a kinase and non-kinase from the set. Three parameters were varied: the allosteric site selection radius, the number of *false* cases sampled for each *true* case in each bootstrap and the number of trees grown per RF model. For suitable ranges, summarised in Table 4.3, every combination of values for each of these parameters was built and tested – a total of 210 models.

Radius / Å	*false*/*true* ratio	no. of trees
4	0.5:1	1000
5	0.75:1	5000
6	1:1	20000
7	1.25:1	
8	1.5:1	
9	1.75:1	
10	2:1	
	2.25:1	
	2.5:1	
	3:1	

Table 4.3: *The ranges of RF parameter values tested in the optimisation procedure.*

The lower bound of 4 Å for selection radii was retained from previous analyses. The upper bound, while far lower than the 30 Å used in the previous analysis on binding site constitution, was still chosen to be excessive, in that a radius of 10 Å was visibly a poor definition of a binding site, since it often captured large swathes of the entire protein. The optimum radius was therefore expected to be within the chosen bounds.

The range of *false*/*true* ratios to be achieved through downsampling the training set were chosen with less prior knowledge; if either bound of the range proved to be optimal, the range would have been extended further in that direction. As detailed in this section, this was not the case, so the original range is presented.

The number of trees was varied to confirm that enough trees were being used to put the models in the 'plateau' of minimum OOB error. If this was the case, there should have been little difference in performance between them. This proved to be true, as can be seen (or indeed not seen, since the points almost entirely overlap) in the following series of graphs. For this reason, the numbers of trees in each run are not labelled and are presented as equivalent data points.

The results of the optimisation protocol are presented below, graphed against a series of established statistical measures in turn. They were more revealing of the weaknesses in the majority of the evaluation measures than they were of the optimal parameters for the dataset. Selection radii are represented by colour. It has been indicated that the m_{try} parameter's default value ($\sqrt{p}$) is suboptimal for high-dimensional datasets only(115) – that is, datasets where $n << p$. For such datasets predictive power received a boost when the default value was increased. Since $n > p$ for the dataset constructed in this project, it was not deemed to be high-dimensional. For the results described below m_{try} was kept at default throughout. In explaining many of the observed trends, the values of a model's 2x2 confusion matrix a, b, c, d are frequently referred to.

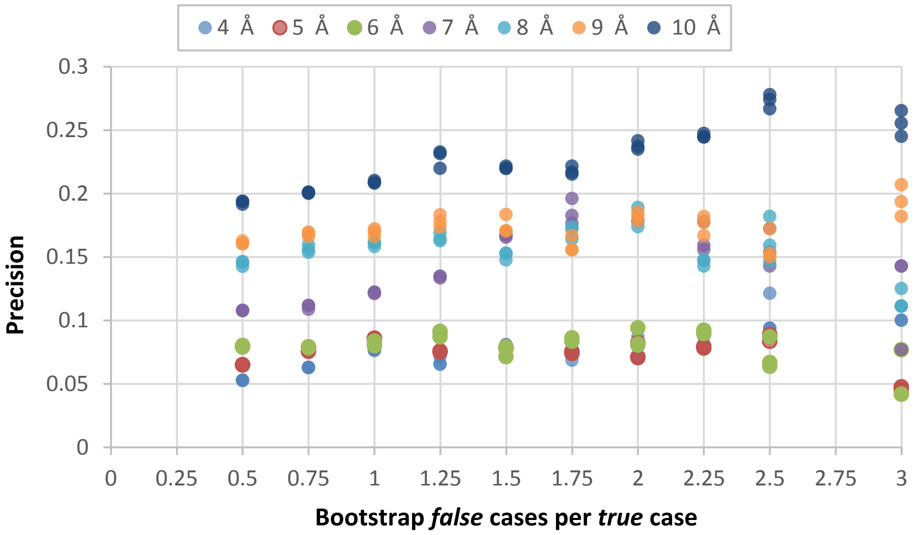


Figure 4.8: *The precision of each iteration of the RF optimisation.*

Figure 4.8 shows the precision of each iteration of the RF optimisation. The uniform increase on increasing selection radius was rationalised by precision's bias towards large values of d. As the selection radius is increased, a greater proportion of a protein's residues are classified as true, giving even weak models a greater statistical chance of correctly predicting a true residue. A slight increase in precision was observed on increasing *false*/*true* ratio. This was due to precision favouring a minimal value for b, which RF models weighted in favour of false naturally achieve. An anomalously large precision was seen for a 7 Å radius at a *false*/*true* ratio of 1.75:1, suggesting (though by no means proving) optimum parameters.

Figure 4.9 shows the TPR or recall of each iteration. The *false*/*true* ratio dominated the trend here, which was explained by a maximal TPR requiring a maximal value for d and a minimal value for c. Models weighted in favour of true – *i.e.* low *false*/*true* ratios – inevitably produce these, regardless of how many false positives are produced with them.

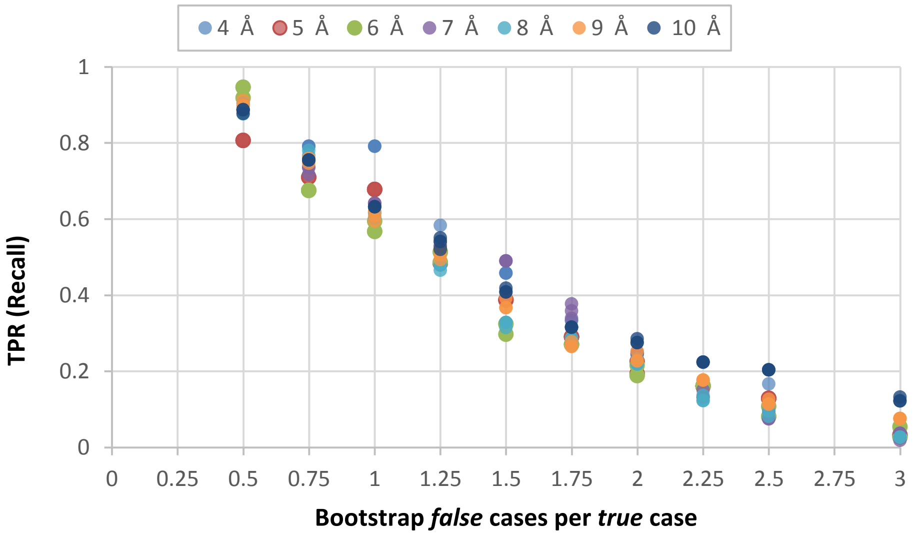


Figure 4.9: *The TPR (or recall) of each iteration of the RF optimisation.*

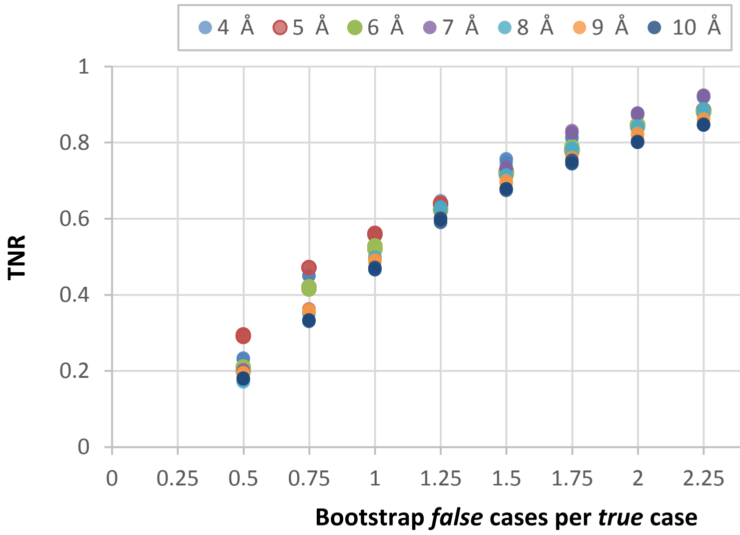


Figure 4.10: *The TNR of each iteration of the RF optimisation.*

As it is the inverse of TPR, one would expect to see the inverse trend was for TNR. This was found to be the case and is shown in Figure 4.10.

A similar trend to this was seen when accuracy was examined (Figure 4.11). As stated previously, accuracy does not account for random chance. Exacerbating the class imbalance of the training set in favour of *false* cases, *i.e.* the majority class, invariably achieved higher levels of accuracy. What can be gleaned from the data in Figure 4.11 is that, at an even balance of classes (1:1 *false*/*true* ratio), many parameter settings achieved accuracies over 50% (*i.e.* better than random chance). Whilst not a spectacular result, only a small dataset was available for this procedure and highly accurate results were not expected. Any indication that there was a genuine signal in the data was sufficient.

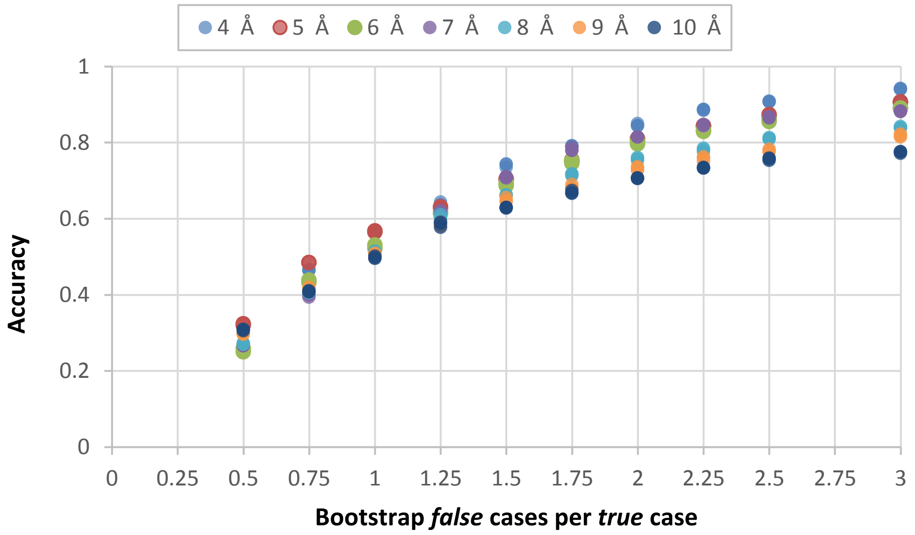


Figure 4.11: *The accuracy of each iteration of the RF optimisation.*

Figure 4.12 shows the F measure of each iteration. Here, larger selection radii yielded greatest F measures, as did lower *false*/*true* ratios. In other words, models whose training sets contained the largest proportion of residues labelled as true prevailed.

The F measure is based on precision and recall which are in turn based on d, b and c. It will thus favour models which produce the largest values of d relative to b and c. Whilst this is indeed desirable, there is little value in a model built upon a skewed dataset containing mostly 'correct' answers (a vast proportion of residues labelled true, as is the case with a 10 Å radius) which is then trained to further bias them through downsampling. The situation observed reveals analogous problems with the F measure to accuracy. Interestingly, the same anomaly in the trends can be seen as with precision: the 7 Å radius at a *false*/*true* ratio of 1.75:1 yielded a greater F measure than expected. There is no artificial reason for this result.

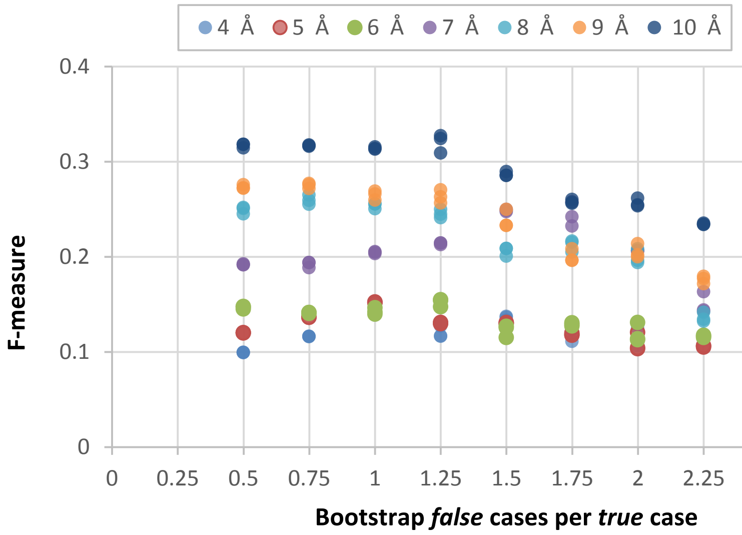


Figure 4.12: *The F measure of each iteration of the RF optimisation.*

The G means of the results are shown in Figure 4.13. Here the class balance proved highly significant, with values between 0.75 and 1.75 (closer to an even class balance) yielding the greatest G means. Although the G mean does utilise all values of the confusion matrix, this is only achieved indirectly through the TPR and TNR. No cross comparisons of a and b with c and d are involved that would help to properly account for class imbalance. It is therefore possible that this peak in G mean at a 1:1 ratio, where there was no class imbalance, was artificial. It is certainly interesting that smaller selection radii (4-5 Å) performed best – residues within these radii would certainly be of high relevance to allostery, perhaps indicating that this is the result of a genuine signal in the data.

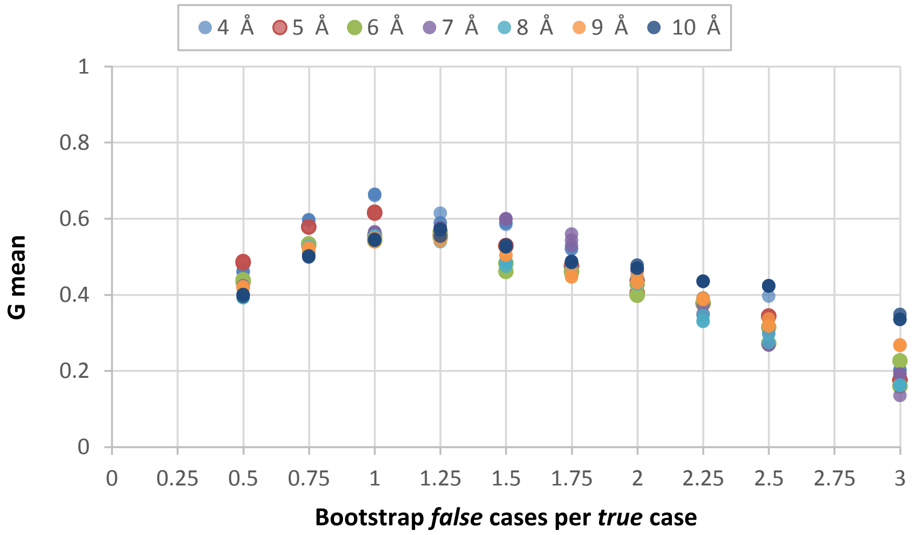


Figure 4.13: *The G mean of each iteration of the RF optimisation.*

Figure 4.14 and Figure 4.15 show the Cohen's kappa and MCC respectively for each parameter setting. Of all evaluation measures examined, these appeared to have come closest to 'weeding out' any intrinsic bias associated with using a particular set of parameters: there appeared to be no universal increase or decrease in these metrics solely based on selection radius or *false*/*true* ratio. For both metrics, the 7 Å radius at a *false*/*true* ratio of 1.75:1 certainly appeared among the best parameter choices, if not the best. The 7 Å radius at *false*/*true* ratios either side of 1.75:1 (1.5:1 and 2:1) also performed well according to these measures – this was promising evidence that a genuine signal was being observed rather than a noisy outlier. The parameters highlighted by examining the G mean (Figure 4.13) at a balanced class ratio also yielded high MCC's, but not Cohen's kappa values.

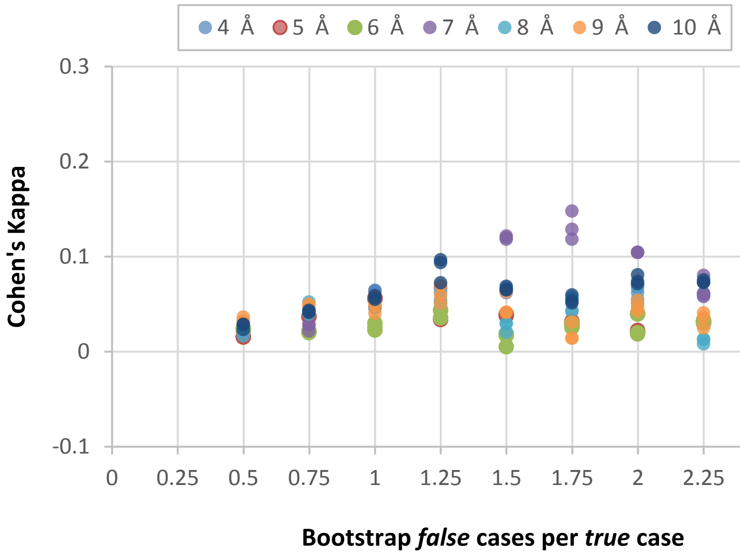


Figure 4.14: *Cohen's kappa of each iteration of the RF optimisation.*

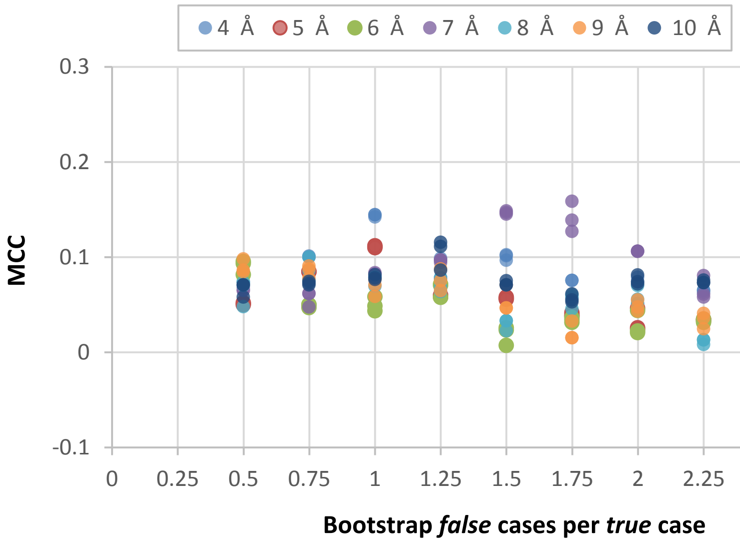


Figure 4.15: *MCC of each iteration of the RF optimisation.*

All of the applied evaluation measures were considered and, on balance, the optimal site selection radius was taken as 7 Å, and the optimal downsampling false/true class ratio was taken to be 1.75:1. However, what is most evident from the results presented here is that the majority of these statistical measures are imperfect and tend to fundamentally favour certain parameter sets. Cohen's Kappa and MCC appeared to perform best, producing results apparently free of bias.

At the time, a method for calculating the ROC for a RF model had not been implemented, so these data were unavailable for use in the above optimisation exercise. However, for future models it was used in place of these more flawed metrics to quantify performance.

4.5 Monitoring Performance by Visualisation

For this project, the majority of statistical measures based upon a model's confusion matrix were doomed to be artificially skewed in one way or another. Even for those that were not, a fundamental issue remained in that, while the models dealt in the 'currency' of individual protein residues, the real goal was to predict an allosteric *site*.

A deeper insight into the real effectiveness of a model was to be gained from viewing the prediction for each protein mapped onto its structure. In this way, one was able to scrutinise the net result of the predictions, without becoming swamped in the minutia of flawed evaluation metrics.

By using this method, it quickly became clear what was required of a prediction in order for it to be deemed a success. Most important was that the viewer of the results was directed, through use of the predictions, to the known allosteric site. Clearly, if this condition was not met, the model was a failure. After that, the fewer further sites highlighted, the better, though the viewer ought to use their knowledge to assess the druggability of a predicted site. There is also the potential to spot a 'suspicious' area that has been highlighted for an artificial reason and dismiss it. An example would be if the exact residues that had been added artificially to fill a missing loop of the protein structure were all predicted allosteric, but no other residues in the vicinity were. This would suggest that the residues had been introduced to the structure poorly and so exhibited abnormal properties that were reflected in their descriptors. Predictions where vast regions of a protein have been highlighted can also be exposed; these may have appeared as strong to statistical measures despite being less useful.

Initially, a binary colour scheme was used to highlight each residue as either *true* (allosteric) or *false* (not allosteric). This was based simply on the classification given by the majority of votes from RF trees. However, over the course of the project, as various parameters were altered and datasets expanded, some models appeared to be overly strict, with very few residues receiving a majority of *true* votes. By this binary colour scheme, such models appeared weak, but closer examination revealed that predictions could sometimes be 'rescued' by slightly shifting the threshold for a *true* classification. More important that the absolute proportion of votes classifying a residue as *true* was its proportion relative to other residues. In other words, the more pertinent question was not which residues were 'allosteric,' but rather which were '*most* allosteric.'

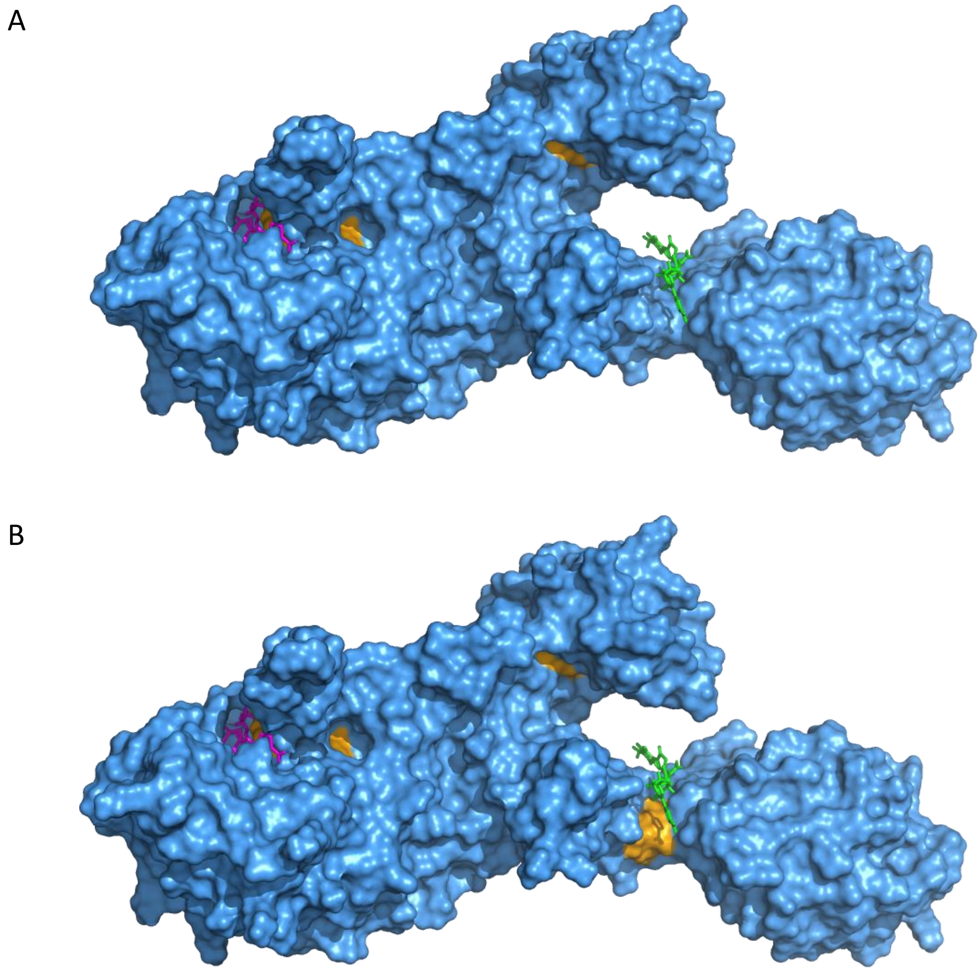


Figure 4.16: ***A****, the prediction for penicillin binding protein 2A (PDB code: 3ZG0) entirely missed the allosteric site, and must be considered a failure;* ***B****, the same prediction is again rendered, this time lowering the probability threshold for a true classification enough to 'flip' two residues – these are in close proximity to the allosteric ligand, and arguably allow this prediction to be considered successful.*

Equally, other models classified far too many residues as *true*; while they did capture the allosteric site, they also highlighted significant portions of the whole protein. A scoring function was devised to strike a balance between 'rescuing' predictions with few, or even no, majority votes for *true* classification and keeping the incidence of false positives to a minimum. As well as achieving this, the scores were required to map to colours on the RGB colour model. In this model integer values of 0-255 are used to represent proportions of red, green and blue colour; it is commonly used in computing applications that generate coloured images, including PyMOL.

Every tree in a RF model votes on the classification of each case that passes through it. Although the majority vote is generally taken forward as the overall ensemble's prediction, the proportion of votes in favour of a class – in this case, *true* – can be taken forward instead. This data can be considered as a residue's predicted probability of being *true*, with a value between 0 and 1. For each protein, the range of values was arranged in descending order and divided into three sections. The first comprised all residues with a probability greater than 0.5; in no cases did these ever make up 5% or more of the total number of residues. The second section took in as many more residues from the top of the list as required to reach 5% of the protein's residues and the remainder formed the final section. For the second and third sections, the ranges of probabilities contained within them were each normalised to form scores in the range of 0-255. The members of the first section were assigned a maximum score of 255. These scores were mapped to RGB values according to the data in Table 4.4.

Colour	Residues with prob. >0.5	Remainder of upper 5% of residues	Lower 95% of residues
Red	255	255	score
Green	0	255 – score	score
Blue	0	255 – score	255

Table 4.4: *A summary of the mapping of residue scores to RGB values for rendering.*

The result of this scoring function is summarised visually in Figure 4.17. The example protein used in Figure 4.16 is shown rendered according to this scoring function in Figure 4.18. It can be seen to retrieve the same key residues in the allosteric site. This function was applied to all proteins that were monitored by visual inspection.

The scoring function was implemented to smooth the results of the RF predictions while minimising the incidence of false positives. It achieved this by ranking residues by allosteric probability and suggesting the top 5% as the final prediction. For the purposes of monitoring model performance, any protein rendered by this method that yielded at least one red-shaded residue in the immediate vicinity of the allosteric ligand was considered a success. Any case found to be particularly ambiguous even after going through the scoring function was deemed a failure.

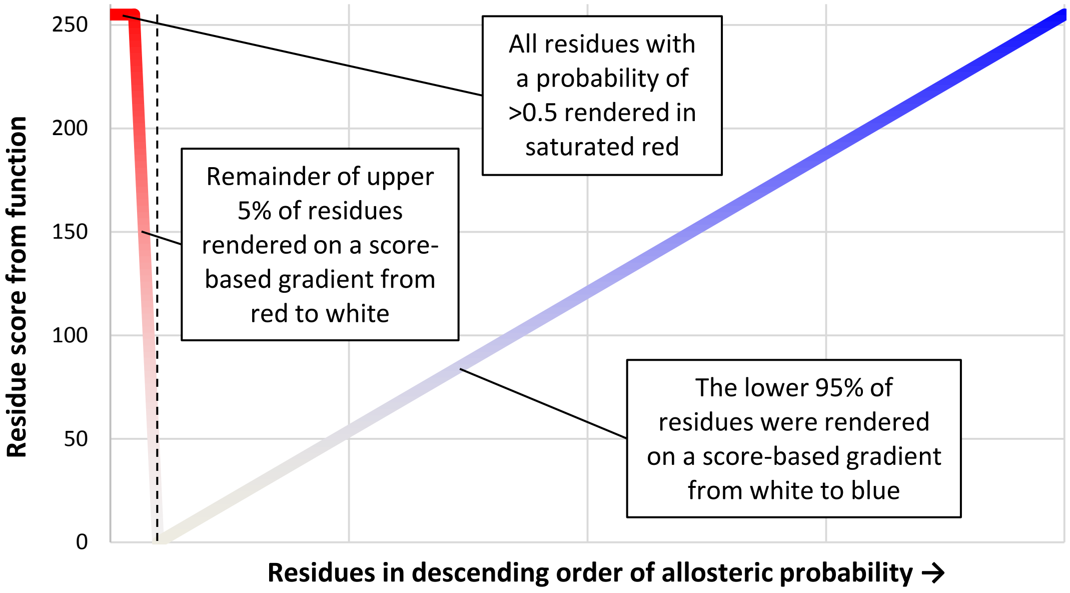


Figure 4.17: *The scoring function is summarised graphically. The colour of the line reflects the result of the function for a list of residues' probabilities.*

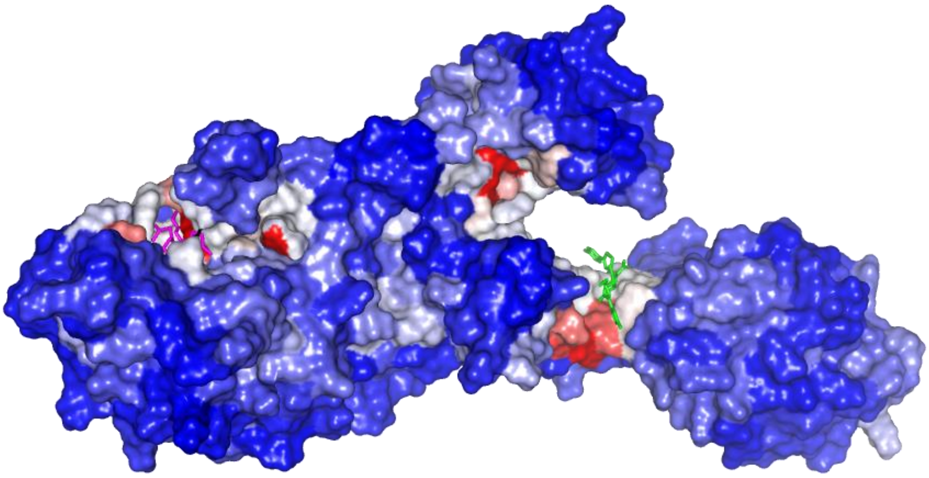


Figure 4.18: *The scoring function is applied to penicillin binding protein 2A (PDB code: 3ZG0), the same example protein shown in Figure 4.16. It can be seen revealing the same key residues in the allosteric site as when the prediction was manually altered.*

4.6 Model Iterations

RF models were constructed at various stages of the project with whatever quantities of data were available. Their performances were monitored by ROC AUC, plotted in Figure 4.19. A brief discussion of the findings contained in the graph follows. Some RF models were constructed retrospectively for the purposes of confirming trends in this exercise. The models based on a 60-protein dataset are examined more closely in subsequent sections.

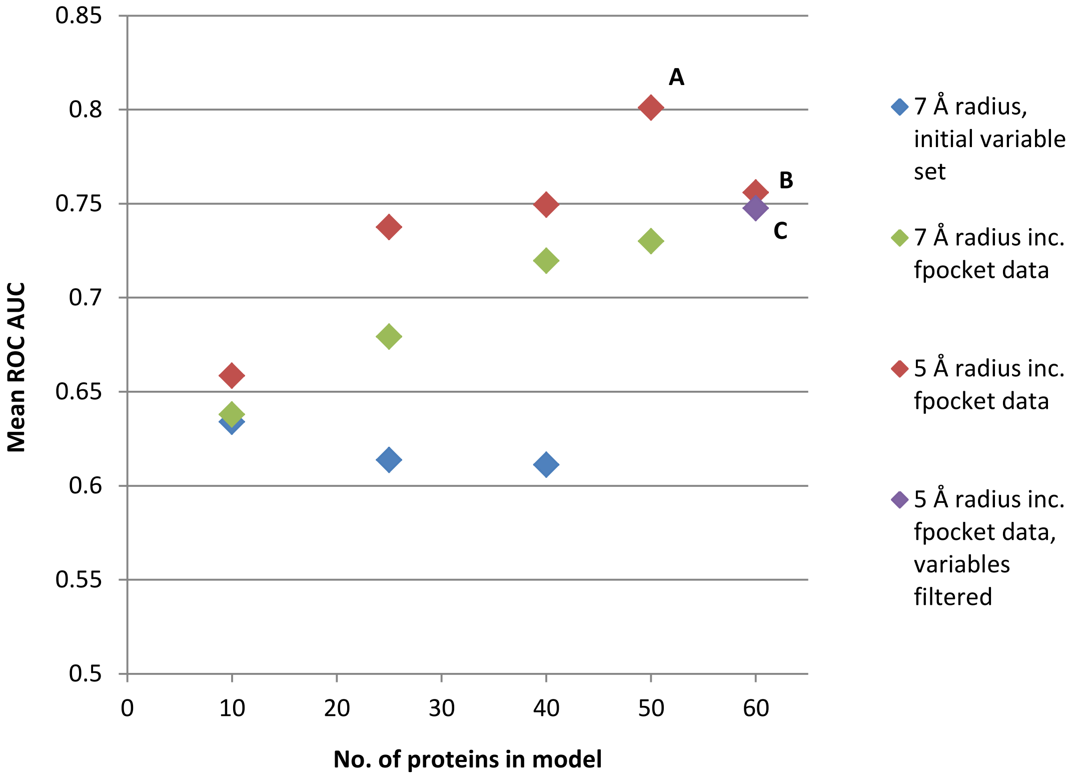


Figure 4.19: *A plot of ROC AUC values of various RF models produces over the course of the project. Some points are labelled for easier referencing in the following discussion.*

Following the initial optimisation of RF parameters, a 7 Å selection radius for allosteric sites was used initially to define the 'correct' classes of protein residues. The only predictors available at this point were SASA, fluctuation, correlated energy, hit/commute times and SID/nSID. As can be seen from Figure 4.19, the ROC AUC did not improve as more proteins were added to the dataset; in fact, it slightly dropped. This stalling of model performance as the number of cases was expanded suggested that the quantity, or quality, of the variables was the limiting factor. Indeed, upon the addition of variables based on fpocket to the set, the addition of further proteins to the dataset beyond the initial ten raised the ROC AUC considerably.

When a fully-populated dataset for 40 proteins was accumulated, the selection radius was varied to confirm whether it was still optimal. Only at 5 Å did the ROC AUC increase: consequently, comparable versions of earlier models using a 5 Å radius were calculated to monitor this trend. Indeed, a larger ROC AUC was observed using all prior datasets. Though not marked in Figure 4.19, a battery of models with varying *false*/*true* class ratios were performed – none equalled or excelled the ROC AUC of the 1.75:1 ratio already in use.

Upon expanding the dataset to include 50 proteins, the jump in ROC AUC from switching to a 5 Å radius became more pronounced (Figure 4.19, model **A**). It appeared that this radius was significantly superior to the originally selected one. However, when the dataset reached its final size containing data on the residues of 60 proteins (Figure 4.19, model **B**), the ROC AUC of the constructed RF model fell considerably.

The need to more thoroughly scrutinize model performance was apparent. With no other statistical method as reliable as the ROC AUC available, the best option was to map the predictions of the RF model back onto each protein, as discussed in section 4.5. By rendering the proteins using the devised function and inspecting the results manually, it was found that model **A** achieved successful predictions for 29 out of 50 proteins, or a 58% hit rate. Model **B**, despite the markedly lower ROC AUC, was successful for 39 out of 60 proteins, or a 65% hit rate. For comparison, model **B** was successful for 31 out of the same 50 proteins used in constructing model **A** (62%). This demonstrated the discrepancy between the ROC AUC and the effective performance as observed manually.

The weakness of the ROC, at least in this context, was that it treated every single residue equally. This is not as useful a property as it may first sound because of the class imbalance in the data. For a protein where the allosteric site has been misclassified, a high ROC AUC can still be achieved if the remainder of the protein was correctly classed *false*. By inspecting them visually, it was possible to treat all predictions that failed to detect the allosteric site as equally (that is, entirely) wrong, regardless of the accuracy of the rest of the prediction.

The failure of the mean ROC AUC as a measure of performance is further highlighted by inspecting model **C**. By mean ROC AUC, it appeared inferior to model **B**, but visual inspection revealed that it was in fact a significant improvement. The dataset for model **C** had been filtered to remove correlated variables; the next section discusses this procedure and the RF model constructed from it.

4.7 Final Model

4.7.1 Treatment of Variable Correlation

The strongest-performing model yet (Figure 4.19, model **B**) contained 137 variables. Many of these were known to be partially correlated, and it was most likely that more still were correlated to one degree or another. To remedy this, Pearson's correlation coefficient was computed using MATLAB for all variable pairs except residue names – this variable was categorical and not compatible with this analysis, and so was removed (but retained in future RF models). The result was a matrix of correlation coefficients. These quantify the linear relationship between two variables, with values ranging between -1 and 1. Perfect anti-correlation achieves a score of -1 and perfect correlation achieves a score of 1. The matrix is presented as a heat map in Figure 4.20. Variables were ordered arbitrarily but not randomly: all variables originating from each analysis were placed side by side (analyses in no particular order). For this reason, 'solid' blocks of consecutive correlated variables were likely to be observed. The map is annotated to note the variables generating prominent 'hot spots.'

Indeed, most high variable correlations were observed on an intra-analysis basis. The most striking, but perhaps least surprising, result was that the vast majority of individual SID score permutations (maximum, minimum, mean, median, etc.) were highly correlated. Moreover, a high correlation was observed between each SID score and its normalised counterpart. A further striking observation regarding the fpocket scores was that, whilst very highly correlated with one another, they were very poorly correlated with other variables. This explains why the addition of these scores to the dataset dramatically affected model performance: the information they contained was almost entirely non-redundant. Similarly, the variables describing correlated motions and correlated energies were found to be poorly correlated with the rest of the dataset, suggesting that these too were supplying unique information to model.

One new insight this analysis offered was the fair inverse correlation of SASA measures with the majority of SID-based scores. SID scores examine topological details of a residue's environment by performing various counts of surrounding residues. In this context, surface residues (with high SASA scores) could be considered to have an 'absence' of protein topology on their exposed flanks. This could contribute to overall lower SID scores and explain the observed trend.

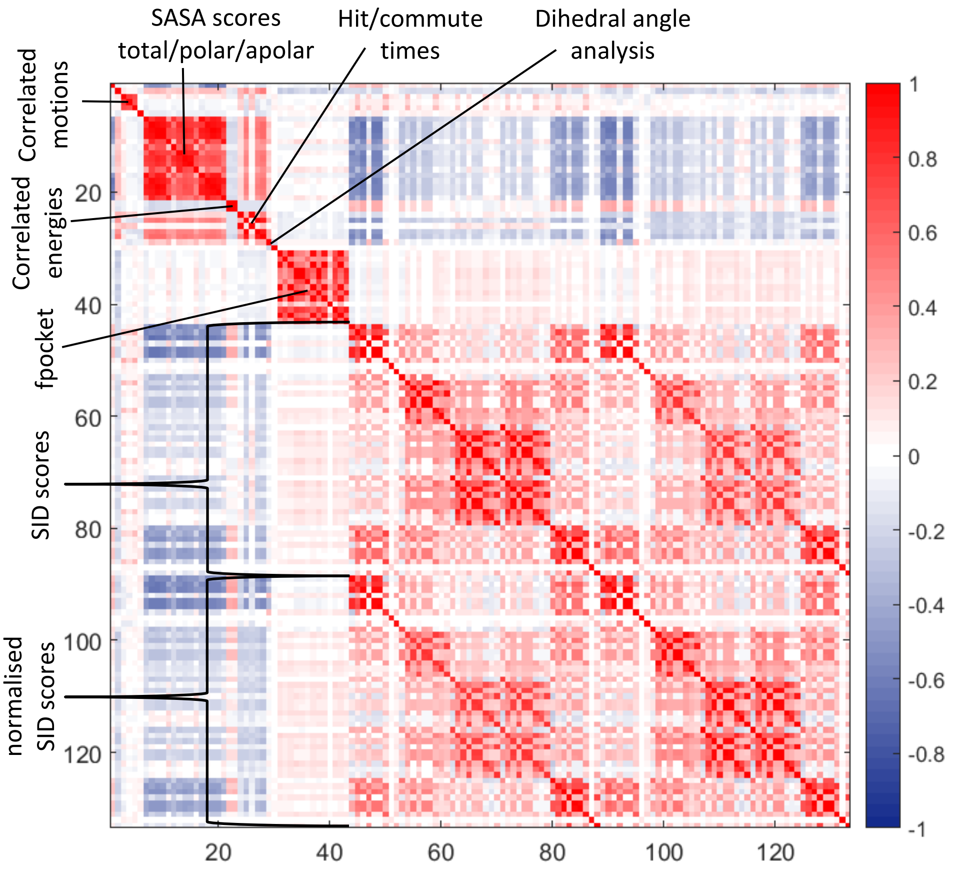


Figure 4.20: *A heat map of the pairwise correlation matrix for the 136 variables included in model* ***B****'s dataset.*

The distinction between correlated variables and anti-correlated variables is irrelevant in the context of datasets for machine learning, so the modulus of the correlation matrix was taken and a threshold of 0.8 chosen as sensible starting point. The result is shown in Figure 4.21. For those variables with correlations over the threshold, these were removed one at a time, starting with the one with the most, until no more variable pairs were over the threshold. A spreadsheet of the dataset is available in Appendix 5 explicitly detailing the complete list of variables as well as marking those that were removed by this procedure.

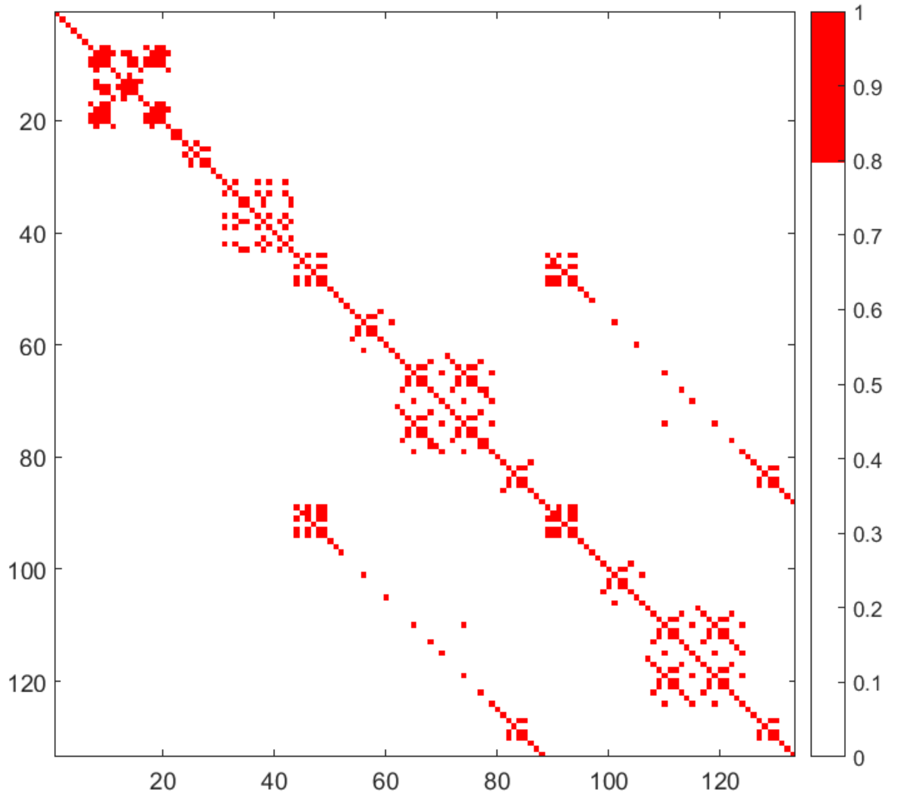


Figure 4.21: *The correlation matrix with all values above a threshold of 0.8 coloured red and the rest in white. Variables with correlations over 0.8 were removed one at a time, starting with the one with the most such correlations, until no more variable pairs were over the threshold.*

The dataset was reduced to 55 variables (including residue names) by the procedure. This was used for the construction of model **C**.

Despite this model yielding an inferior ROC AUC, manual inspection revealed it to correctly predict 43 out of 60 sites in the dataset, or a 72% hit rate. By use of the scoring function and visual inspection this is the strongest performance achieved by any model, and is the main result of the project.

With the significant boost to model performance offered by the removal of correlated variables, a lower correlation threshold of 0.7 was investigated. This resulted in a dataset reduced to 35 variables, but the RF model constructed from this achieved only a 67% hit rate (40 out of 60). For the purposes of this section of discussion, model **C**, which performed better with a correlation threshold of 0.8 than this subsequent model, will be referred to as the 'final model.' The 60 predictions of the final model, rendered onto each protein using the previously described scoring function, are available as interactive PyMOL session files as part of Appendix 6.

4.7.2 Analysis of Final Model

For the final model, a variable importance plot was generated using the inbuilt function within the randomForest library of R. This is presented in Figure 4.22. The identity of each residue was, by a great margin, the most important variable. It is possibly taken for granted how much information is encapsulated in the names of the amino acids. From the name alone, one can infer a vast quantity of information about a given residue, such as mass, volume, functionality, hydrogen accepting/donating capability and aromaticity. These basic chemical properties are the essence of all others, so it is not surprising that this, the simplest and most fundamental piece of data in the whole dataset, was the most heavily utilised by the RF model.

At the other end of the spectrum, the *STRANDS_Range.nUniques* variable (range of values multiplied by the number of unique values for the STRANDS SID number) displayed a mean decrease in accuracy of exactly 0. This means that the model's performance was entirely unaffected by permuting this variable to noise, indicating that no node in any tree selected this variable to make its split. Despite not being correlated to others in the dataset, this variable was redundant. This does not necessarily mean the variable did not contain any real information, but merely that, at any node where it was shortlisted for use, another variable was found to be able to produce a superior split. In any case, it served as an effective demonstration of RF's ability to handle extraneous data.

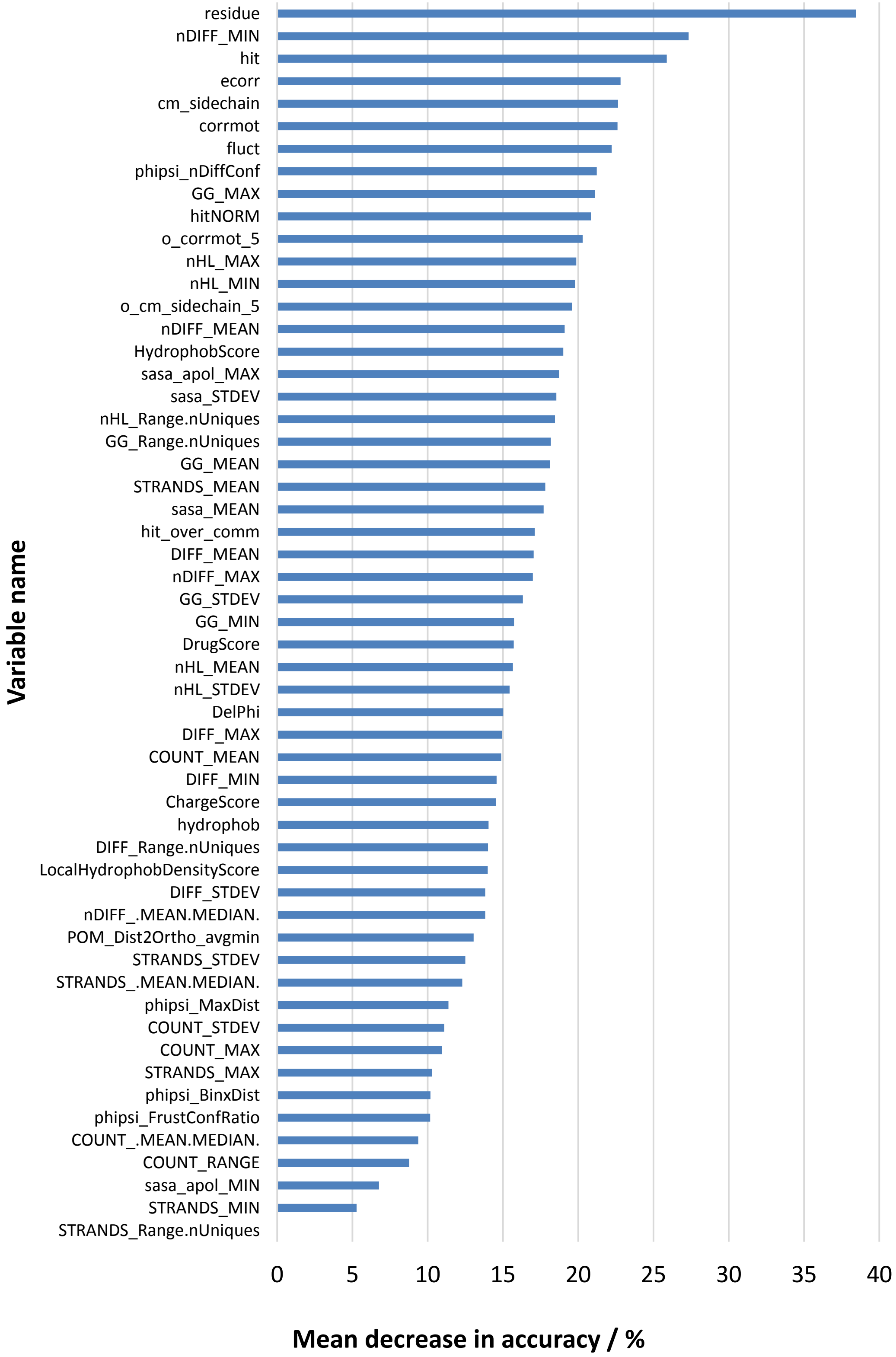


Figure 4.22: *Variable importance plot for the 55 variables included in the final RF model.*

Shorthand Variable name	Full Variable Name/Description	Mean % Decrease in Accuracy
residue	Name of residue	38.5
nDIFF_MIN	Minimum DIFF score, normalised	27.3
hit	Hitting time	25.9
ecorr	Correlated energy score	22.8
cm_sidechain	Correlated motion score (sidechain included)	22.7
corrmot	Correlated motion score (main chain atoms only)	22.6
fluct	Residue fluctuation	22.2
phipsi_nDiffConf	Dihedral angle NDC score	21.2
GG_MAX	Maximum GG score	21.1
hitNORM	Hitting time, normalised	20.9
o_corrmot_5	Correlated motion (main chain atoms only) to orthosteric site (5 Å radius definition)	20.3
nHL_MAX	Maximum HL score, normalised	19.9
nHL_MIN	Minimum HL score, normalised	19.8
o_cm_sidechain_5	Correlated motion (sidechain included) to orthosteric site (5 Å radius definition)	19.6
nDIFF_MEAN	Mean DIFF score, normalised	19.1
HydrophobScore	fpocket metric; mean hydrophobicity of pocket	19.0
sasa_apol_MAX	Maximum apolar SASA	18.7
sasa_STDEV	Standard deviation in total SASA	18.6
nHL_Range.nUniques	range of HL scores multiplied by number of unique scores, normalised	18.5
GG_Range.nUniques	range of GG scores multiplied by number of unique scores	18.2
GG_MEAN	Mean GG score	18.1
STRANDS_MEAN	Mean STRANDS score	17.8
sasa_MEAN	Mean total SASA	17.7
hit_over_comm	Ratio of hitting time and commute time	17.1
DIFF_MEAN	Mean DIFF Score	17.0
nDIFF_MAX	Maximum DIFF score, normalised	17.0
GG_STDEV	Standard deviation in GG score	16.3
GG_MIN	Minimum GG score	15.7

DrugScore	fpocket metric; composite score assessing durggability of pocket – precise basis of calculation is withheld by authors	15.7
nHL_MEAN	Mean HL score, normalised	15.7
nHL_STDEV	Standard deviation in HL score, normalised	15.5
DelPhi	DelPhi score	15.0
DIFF_MAX	Maximum DIFF score	15.0
COUNT_MEAN	Mean COUNT score	14.9
DIFF_MIN	Minimum DIFF score	14.6
ChargeScore	fpocket metric; mean pocket charge	14.5
hydrophob	Hydrophobicity score of residue	14.0
DIFF_Range.nUniques	range of DIFF scores multiplied by number of unique scores	14.0
LocalHydrophobDensityScore	fpocket metric; ratio of neighbouring apolar alpha-sphere pairs and total apolar alpha-spheres	14.0
DIFF_STDEV	Standard deviation of DIFF scores	13.8
nDIFF_.MEAN.MEDIAN.	Absolute difference of mean and median DIFF scores, normalised	13.8
POM_Dist2Ortho_avgmin	Percent-of-max distance to orthosteric site in minimised-average conformation	13.1
STRANDS_STDEV	Standard deviation in STRANDS score	12.5
STRANDS_.MEAN.MEDIAN.	Absolute difference of mean and median STRANDS scores	12.3
phipsi_MaxDist	Dihedral angle MaxDist score	11.4
COUNT_STDEV	Standard deviation in COUNT score	11.1
COUNT_MAX	Maximum COUNT score	11.0
STRANDS_MAX	Maximum STRANDS score	10.3
phipsi_BinxDist	Dihedral angle BD score	10.2
phipsi_FrustConfRatio	Dihedral angle RDC score	10.2
COUNT_.MEAN.MEDIAN.	Absolute difference of mean and median COUNT scores	9.4
COUNT_RANGE	Range of COUNT scores	8.8
sasa_apol_MIN	Minimum apolar SASA	6.8
STRANDS_MIN	Minimum STRANDS score	5.3
STRANDS_Range.nUniques	range of STRANDS scores multiplied by number of unique scores, normalised	0.0

Table 4.5: *Complete list of variables included in the final model, ordered by variable importance. A full name/description is provided for the shorthand names used in generating the preceding figure.*

It was also pleasing to see one of the novel and relatively untested variables from the dihedral angle analysis, the NDC score (number of different phi-psi conformational bins sampled over the trajectory), ranked prominently in terms of importance.

Aside from the two variables at the very ends of the plot, what can be seen is a notably gradual decline in importance. This suggests that there was no clear subset of 'core' variables in the data that was most important to the model's performance. While some relative importance between a pair of variables could be approximated with this result, the overarching conclusion one should draw from this is that all variables contributed to the model's performance to a greater or lesser extent.

With hindsight, this result was perhaps foreseeable. If allostery was a low-dimensional concept – in other words, one that could be sufficiently well characterised to make accurate predictions using only a small number of descriptors – it would most likely have been solved before now.

4.7.3 Partial Dependence of Variables

Had it been the case that a small subset of variables was responsible for the vast majority of the model's performance, those PD plots would have been examined in this section. However, as was noted in the preceding section, all variables contributed to the overall performance of the model to at least a modest extent, save for one that was entirely excluded. Based on this result, it would appear that allostery cannot be reduced to a small set of markers. This is a critical caveat but, so long as it is kept in mind, it is still worthwhile examining a selection of variables in detail.

Based on the variable importance plot in Figure 4.22, the PD plots for the three highest scoring variables and the lowest non-zero scoring variable are presented below. Of note in all of the plots is the scale of the y-axis, which invariably ranges between approximately -0.5 and -1. This is because the analysis is measuring the influence of each variable on the model's positive classification, *i.e.* on the model making a *true* prediction. Since the model is trained with a class imbalance of 1.75:1, it is overall more likely to make a *false* prediction in all situations. In order to reach a positive PD, a variable would have to be near-perfectly correlated with allosteric sites. Such a variable certainly did not exist in this dataset and, as was suggested in the preceding section, likely does not exist at all.

The PD for the most important variable, the name of the residue, is shown in Figure 4.23. Many of the trends in this plot matched the observations in section 4.2 and the complementary findings of van Westen *et al.* on allosteric ligands(54). Aromatic residues scored high and acidic residues scored low. Tryptophan in particular was the highest scoring residue here and also the residue with the greatest relative increase in abundance in allosteric sites. The spread of basic residues' scores also matched (H and R high, K low).

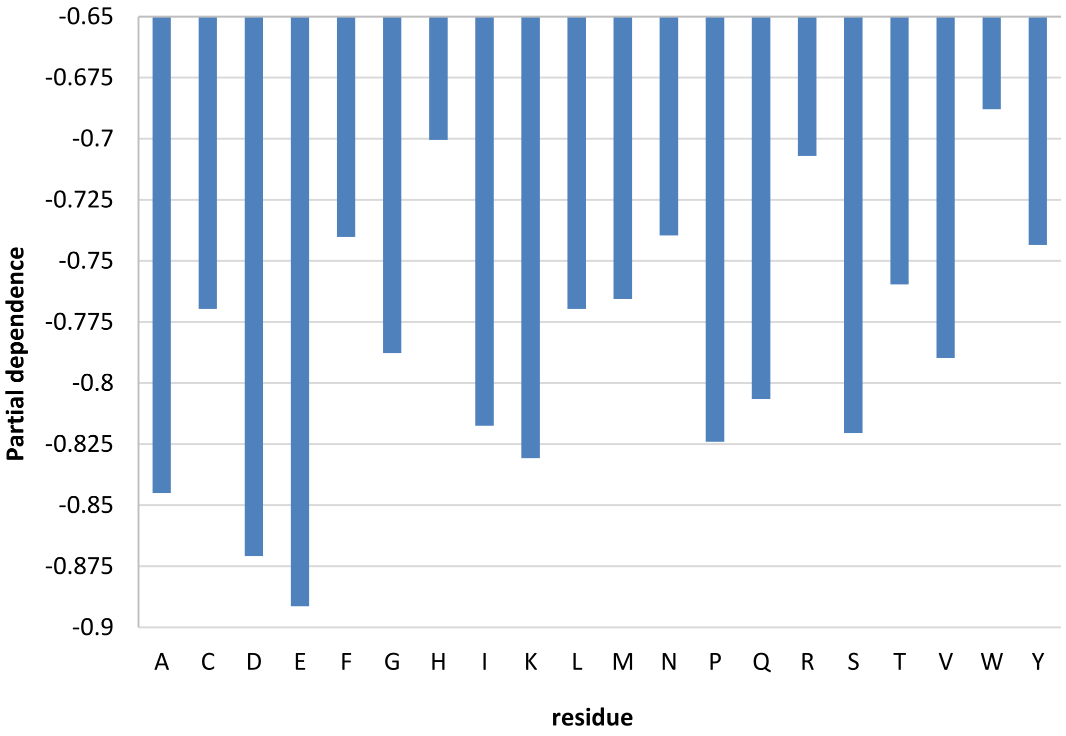


Figure 4.23: *The PD plot of the residue variable, i.e. the name of the residue. Several trends matching those observed in section 4.2, such as lower PD for acidic residues and higher PD for aromatic residues.*

Figure 4.24 shows the PD plot of the *nDIFF_MIN* variable. This is the minimum, normalised DIFF score produced by residues over their MD trajectory. The distribution of the dataset along this variable explains why it was found to be as important as it was by the RF model: a sharp increase in partial dependence occurs through the 4th and 5th deciles. In other words, this variable was able to split the data roughly in half, with the higher values being more likely to result in a *true* classification. This is a powerful split for a RF node to make in terms of Gini purity. Residues present in the vicinity of three or more sections of protein chain produce the highest DIFF scores. It is interesting that the *minimum* of such scores should appear so important; residues that deviated from their central positions for even a single frame of MD would have lost their high *nDIFF_MIN* score. This indicates that residues *consistently* present at multi-way interfaces in the protein structure (thus retaining a high *nDIFF_MIN* score) are more likely to result in a *true* classification.

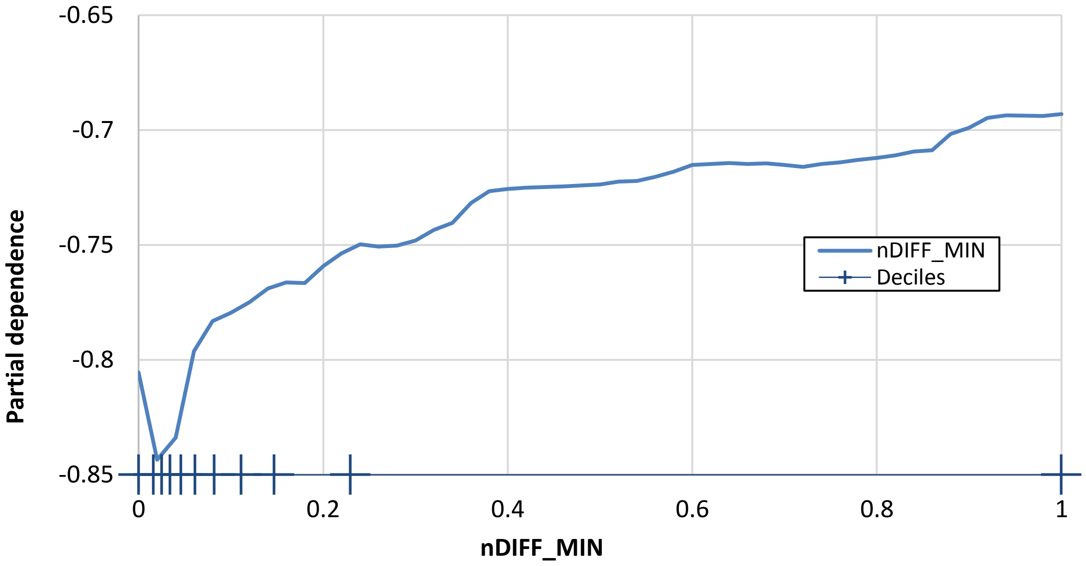


Figure 4.24: *The PD plot of the nDIFF_MIN variable. A sharp increase in PD occurred near the midpoint, meaning a powerful split could be made at RF nodes.*

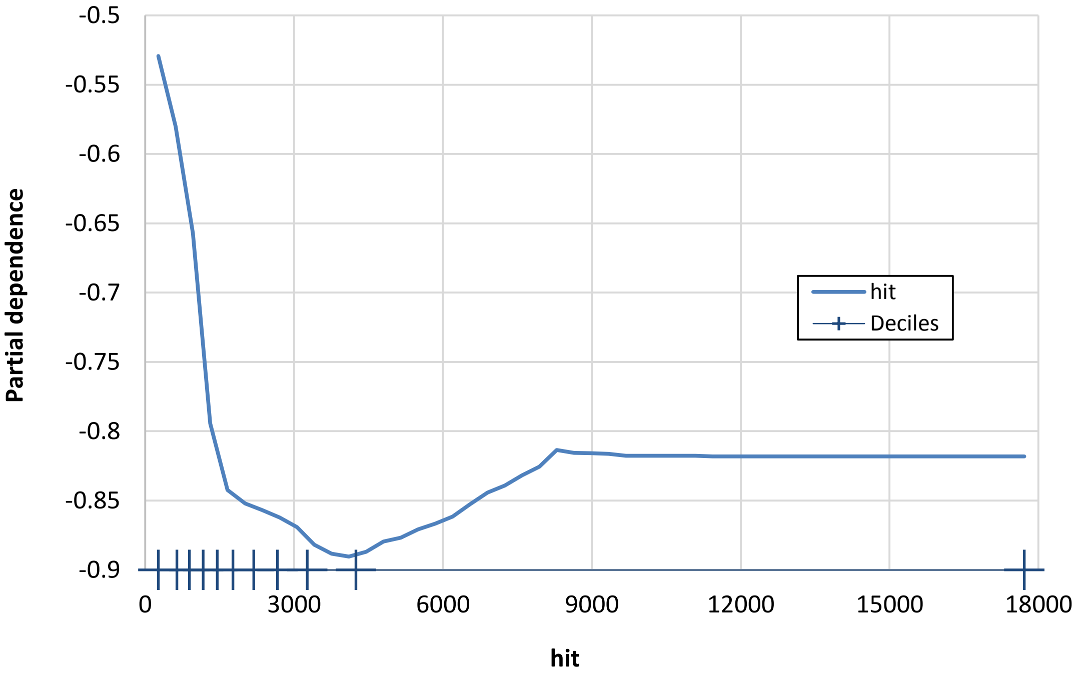


Figure 4.25: *The PD plot of the hit variable. A sharp increase in PD occurred near the midpoint, meaning a powerful split could be made at RF nodes.*

Figure 4.25 shows the PD plot of the *hit* variable (hitting time). An even more marked gradient was observed, this time descending across the lower half of the dataset. This indicates that low values more often led to *true* classifications. This correlates well with the theory that residues in allosteric sites possess low hitting times.

The PD plot of the least important variable, *STRANDS_MIN* (minimum STRANDS score of residue across trajectory frames), used by the final model is shown in Figure 4.26. The reason for this variable's relative unimportance can be seen by examining the spread of data along the x-axis. Not only did at least 60% of the data share the identical value of 1, and so could fundamentally not be split at a RF node, but there was only a very slight impact on PD over at least 90% of the data values. It would appear that the decision trees used this variable to split off the very small number of cases with a *STRANDS_MIN* score of 5, where a notable increase in PD is observed. After doing this, little information remained to be extracted from this variable.

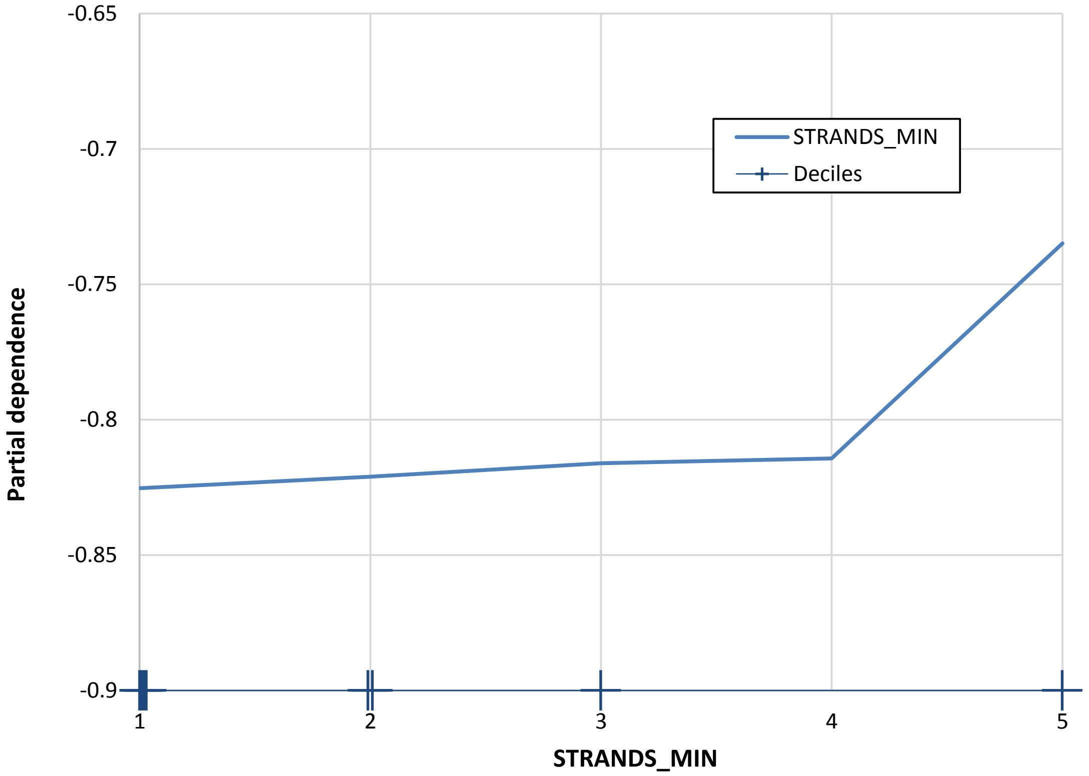


Figure 4.26: *The PD plot for the STRANDS_MIN variable. A very shallow gradient is observed over the vast majority of the dataset, meaning that the decision trees in the model were rarely able to make a strong split on this variable.*

4.7.4 Proximity Measures

The proximity measure quantifies the frequency with which a pair of cases received the same classification. Cases that often received the same classification are positioned in close spatial proximity, and *vice versa*. Thus, for a perfect model, cases would appear in a tight cluster for each real class, with a large gap segregating the clusters.

For the final model the proximity measures of the dataset's cases in the final RF model were computed. These were reduced to three dimensions by MDS. These operations are packaged into the randomForest library of R. The Spotfire data visualisation suite was used to plot this data. While it has the capability to present 3D data in an interactive, rotatable scatterplot, this does not translate well to the page. Instead, three sequential scatterplots are presented, graphing dimension 2 of the MDS vs. dimension 1 (Figure 4.27), followed by dimension 3 vs. dimension 1 (Figure 4.28), followed by dimension 3 vs. dimension 2 (Figure 4.29).

The results, while far from the described ideal, showed a dense cluster that the majority of residues (of both classes) fell into, with a wide scattering of outliers. The *true* residues deviated from their positions within the cluster significantly less than the *false* ones from all three perspectives. There remained too little distinction between the two for it to be of benefit to investigate the relationships between specific variables and positions in this dimension space, but it was clear that some enrichment was taking place in the model, with *true* predictions presenting in the cluster with a probability greater than random chance.

More than anything else, this analysis illustrated the sheer complexity of what was being asked of a predictive model. Allosteric sites have little analytical definition; they tend to be retrospectively defined after the location of a bound allosteric ligand has been determined by experiment. Even with this information known, the 'site' has no rigid physical definition. It is an onerous demand to make of an algorithm to generate a model that, based solely on data derived from a MD simulation, predicts something so poorly characterised.

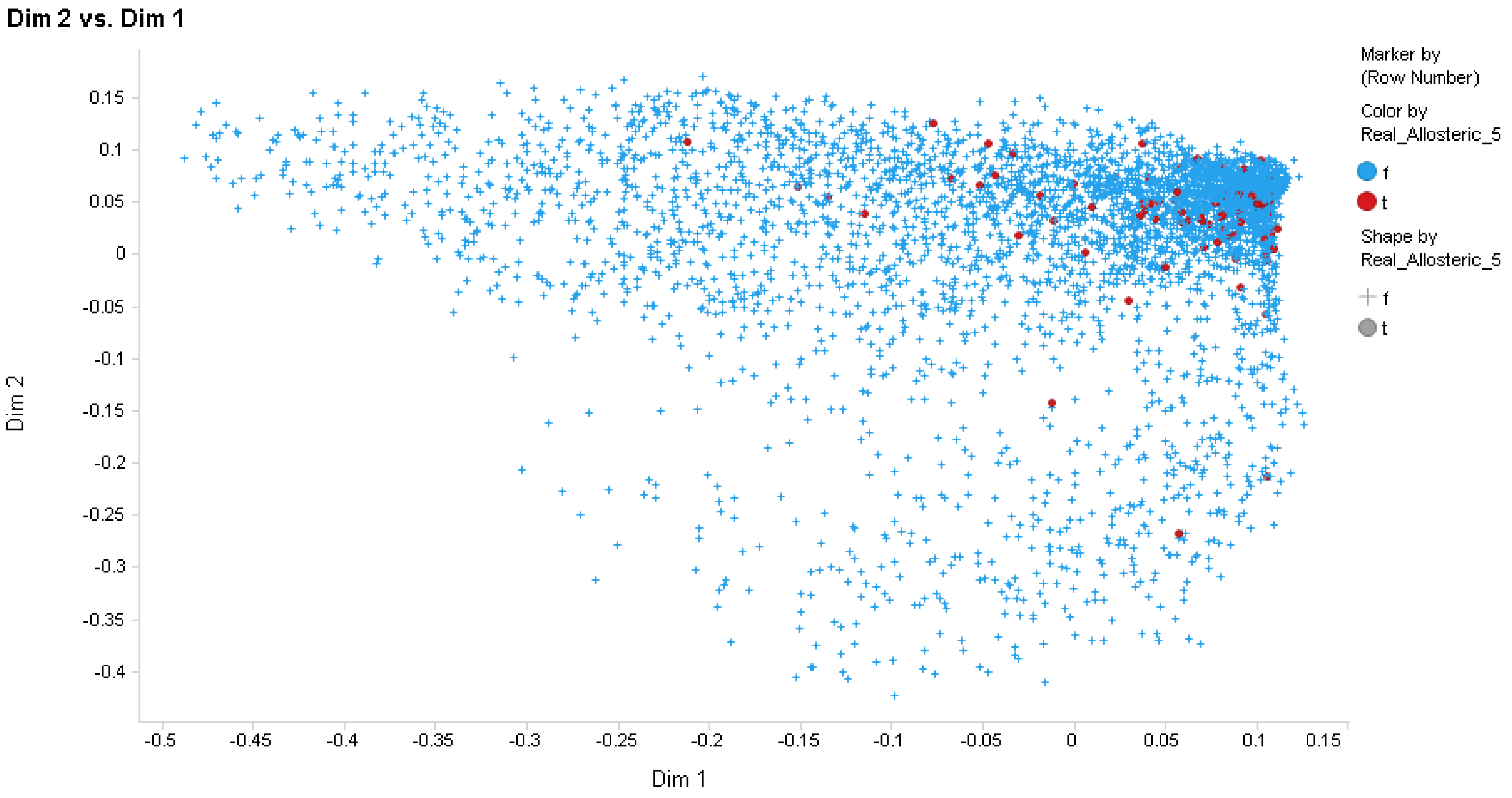


Figure 4.27: *A scatter plot of the first and second dimensions of the proximity measures in MDS space. Allosteric residues are rendered in red. A dense cluster of residues, including the vast majority of allosteric residues, can be seen in the upper right of the grid.*

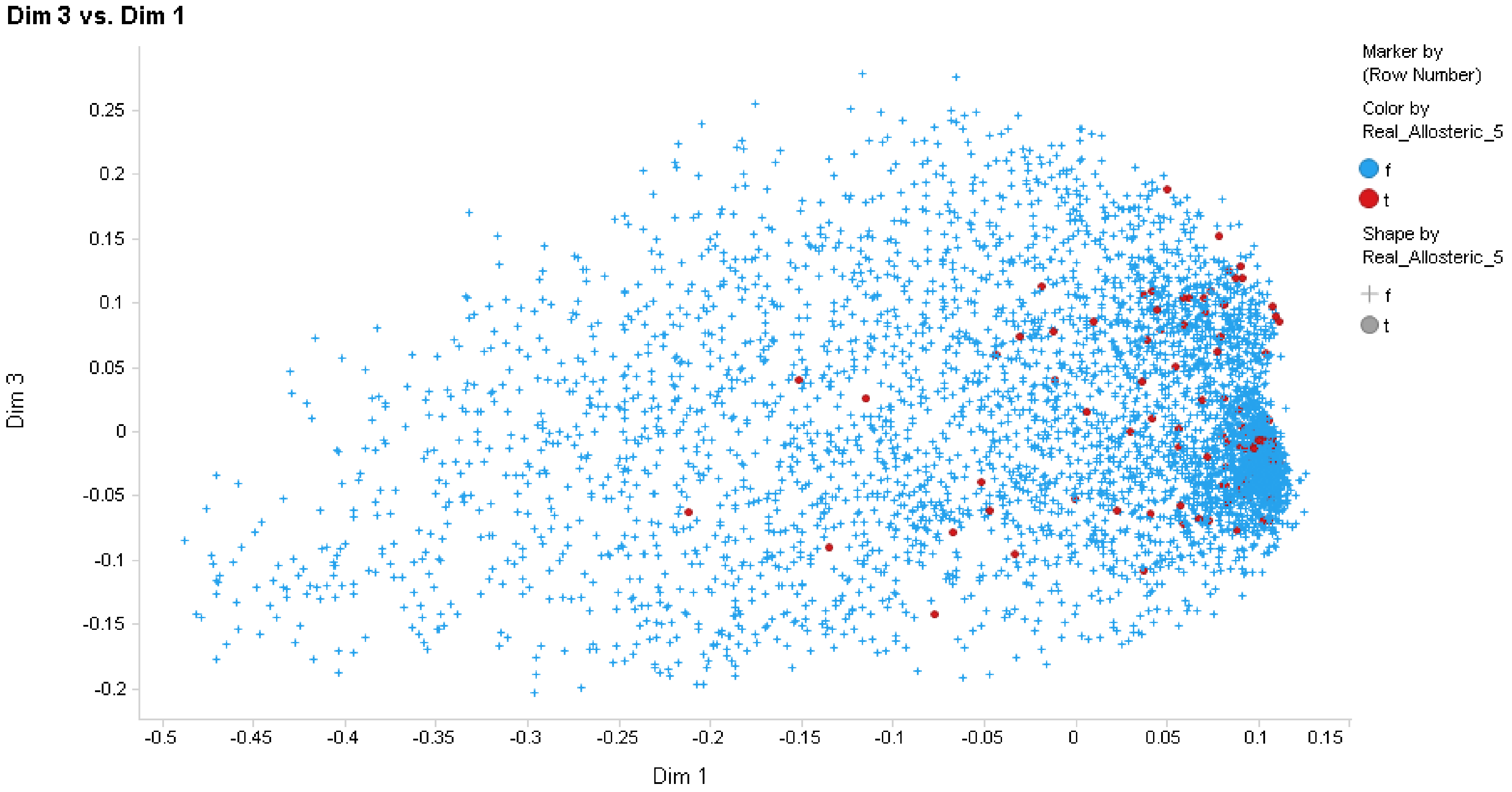


Figure 4.28: *A scatter plot of the first and third dimensions of the proximity measures in MDS space. Allosteric residues are rendered in red. A dense cluster of residues, including the vast majority of allosteric residues, can be seen at the lower right side of the grid.*

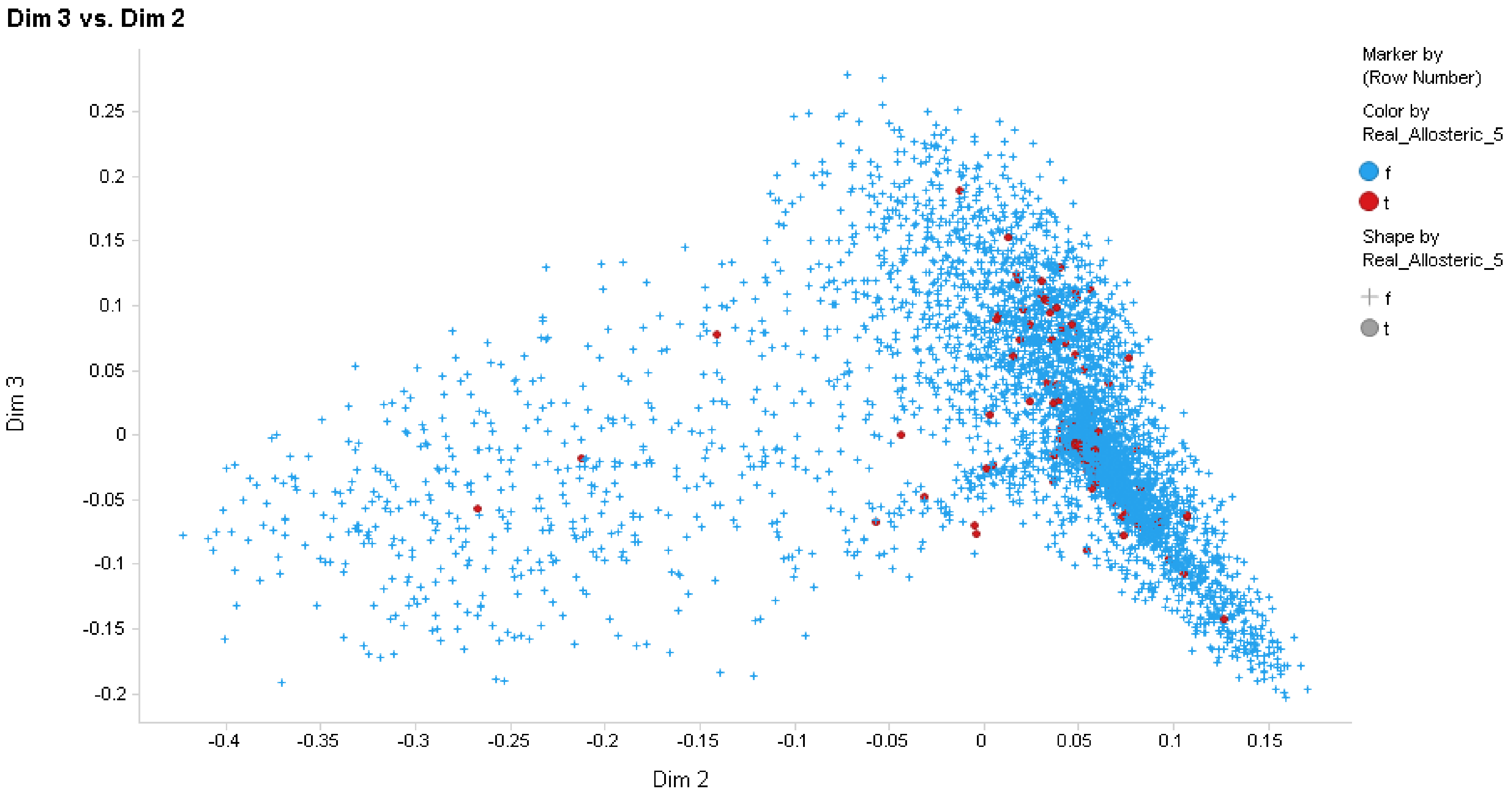


Figure 4.29: *A scatter plot of the second and third dimensions of the proximity measures in MDS space. Allosteric residues are rendered in red. A dense cluster of residues, including the vast majority of allosteric residues, can be seen down the right side of the grid.*

5. Controls and Validation

5.1 Overview

This chapter details a host of complementary experiments that reinforced the assertion that the final RF model was a successful step forward in the area of allosteric site prediction. To compare performance, equivalent models were trained solely on data from single structures of proteins. Significantly, a model trained from data based on crystallographic structures was produced to demonstrate the issue of ligand imprinting. The models are then compared and summarised together. As a basic control experiment, a y-randomisation was also performed; this establishes a baseline of model performance that could be expected when trained on noise alone. The validity of the MD trajectories, which were the basis for the majority of the final dataset, was also explored.

5.2 Model trained and tested on original crystal structures

All viable descriptors were calculated for the original protein conformations derived from crystal structures. For example, SID scores could be determined, but residue fluctuations could not. From the resulting dataset a RF model was trained in an equivalent manner to the final model. Using the scoring function described in section 4.5 the predictions were visualised: a correct prediction rate of 73% was found. Interestingly, this experiment was repeated using exclusively fpocket data and this rate remained unchanged at 73%.

Both of these results slightly outperformed the final model. The most significant difference between the final model and these two experiments was the use of crystallographic conformations of the proteins in the training data. Though the ligands were deleted *in silico*, the voids left in their stead remained, as did any effect on properties of the surrounding residues. In a ligand-bound state, residues in the binding site may adopt an artificially strained conformation, the energetic penalty for which is more than paid for by the accommodation of the ligand. Solvent-accessible surface areas, for example, would likely change with any reorientation of residues as well as, for similar reasons, the pocket volume. One can imagine the vast majority of residue properties skewed in one way or another by the presence of a ligand. Passing skewed data such as this to a machine learner would inevitably train it to identify these exaggerated or diminished properties.

It was, in all likelihood, these imprints of the removed ligands that were being detected by the models. A telling clue of this was betrayed by the performance holding, rather than decreasing, when all but the purely geometric fpocket variables were stripped from the dataset; these, of all variables in the dataset, would be most directly influenced by the artificial contortion of a protein cavity. A model built on a dataset comprising only these would therefore be the most responsive to these properties in terms of its predictions.

5.3 Model trained on original crystal structures and tested on minimised average structures

The fundamental problem with constructing predictive models using ligand-bound conformations as training data is that, when making a live prediction, the test protein will not be in its allosteric ligand-bound conformation. Proteins in the training set would contain an imprint of the allosteric ligand that would in turn be reflected in the altered values for any calculated properties. The resulting model would thus be trained to detect allosteric sites by separating these imprinted properties from the rest, rather than by picking up on any genuine, underlying signal in the data.

To test this argument, a hypothesis was posed. A model trained on data containing imprints of allosteric sites should perform well when tested on data that is equally imprinted (and it did). It should, however, suffer a drop in performance when presented with a protein conformation that is free of influence from a binding event. While true apo structures of all 60 proteins were not available, an alternative was: the energy-minimised average structures of the same set of proteins. Since these were free from bias due to co-crystallisation with allosteric ligands, they could be considered ‘pseudo-apo’ structures.

The described model was constructed, and performance was found to have dropped from 73% to 63% (6 fewer proteins). This demonstrates the significant penalty to predictive power incurred by using a model trained on ligand-imprinted data (from any methodology, not just RF).

5.4 Model trained and tested on minimised average structures

It was thought that at least some of the performance lost in section 5.3 could be regained by training the model, as well as testing it, on data derived from the minimised average structures of MD trajectories. This would confirm the importance of training data being as close as possible to testing data. Moreover, this model would not contain ligand-imprinted data, and so could not be invalidated in the same way as other single-conformation models.

The model achieved a performance of 70% (up from 63%) by manual inspection, confirming this further hypothesis. The apparent success of this model is notable, even though it was built on a relatively small set of descriptors that were based only on a single conformation per protein. It suggests that variables based on trajectories of MD data are not fundamentally necessary to begin to predict allosteric sites. Of course, in this case the 'pseudo-apo' structures originated from MD trajectories, but if true apo structures could be sourced from databases such as the PDB then this could be bypassed.

This proved to be another situation clouded by attempts to summarise models' performances numerically, requiring visual inspection in order to clarify it. This 'pseudo-apo' model, while achieving an impressive hit rate by standard criteria, did so with a large quantity of false positives. The scoring function was not enough to minimise these and, while the allosteric site was often correctly predicted, so was a large fraction of the entire protein. This is exemplified in Figure 5.1 and Figure 5.2, where the two models' predictions of the same protein are compared in each case. It should be noted that these figures were oriented to show as much of the relevant protein surface as possible, rather than optimise the view of the allosteric site; this is still visible by the superimposed ligand rendered in green. They were both rendered using the same scoring function.

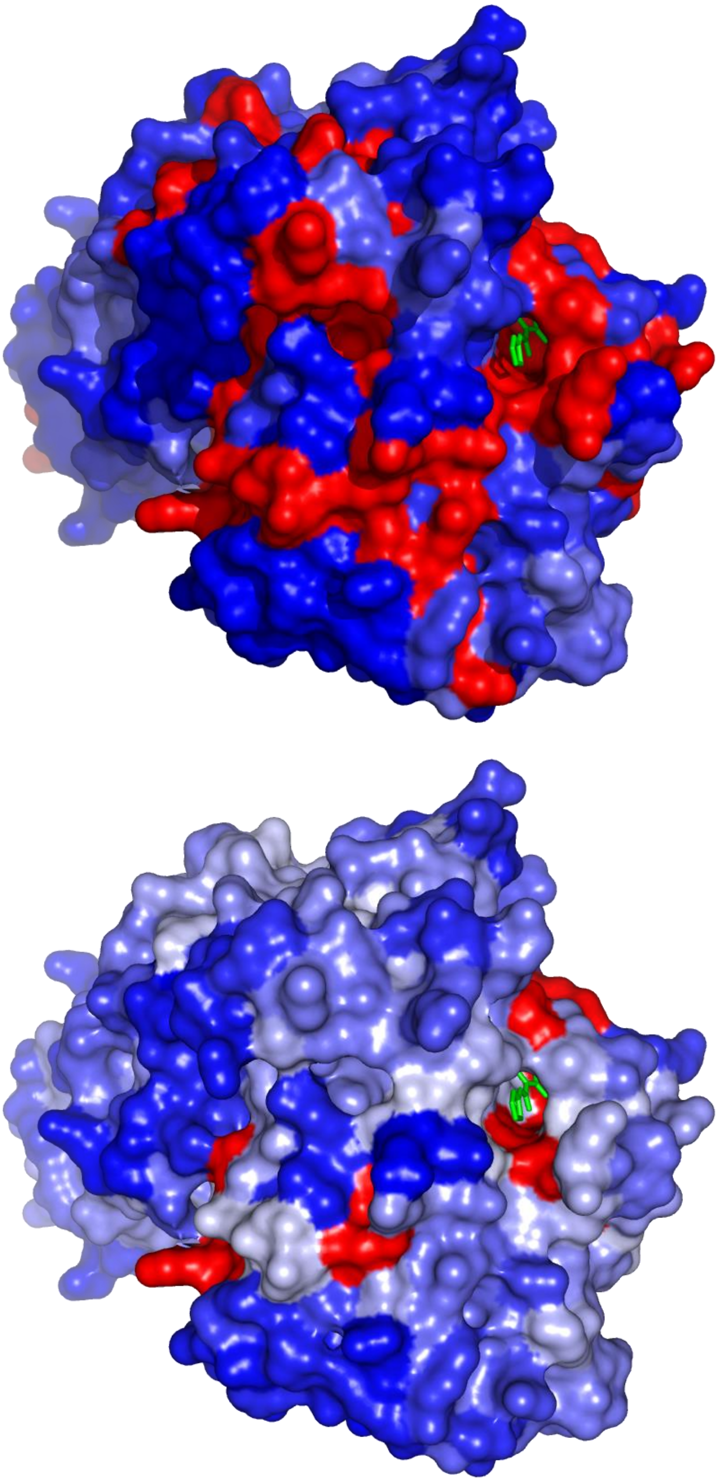


Figure 5.1: *Two models' predictions made on the same protein (human glucokinase, PDB code: 1V4T). Though both predicted the allosteric site correctly, the single-conformation 'pseudo-apo' model (top) also predicted a large fraction of the whole protein. The final model (bottom) was more precise, selecting far fewer residues not in the allosteric site.*

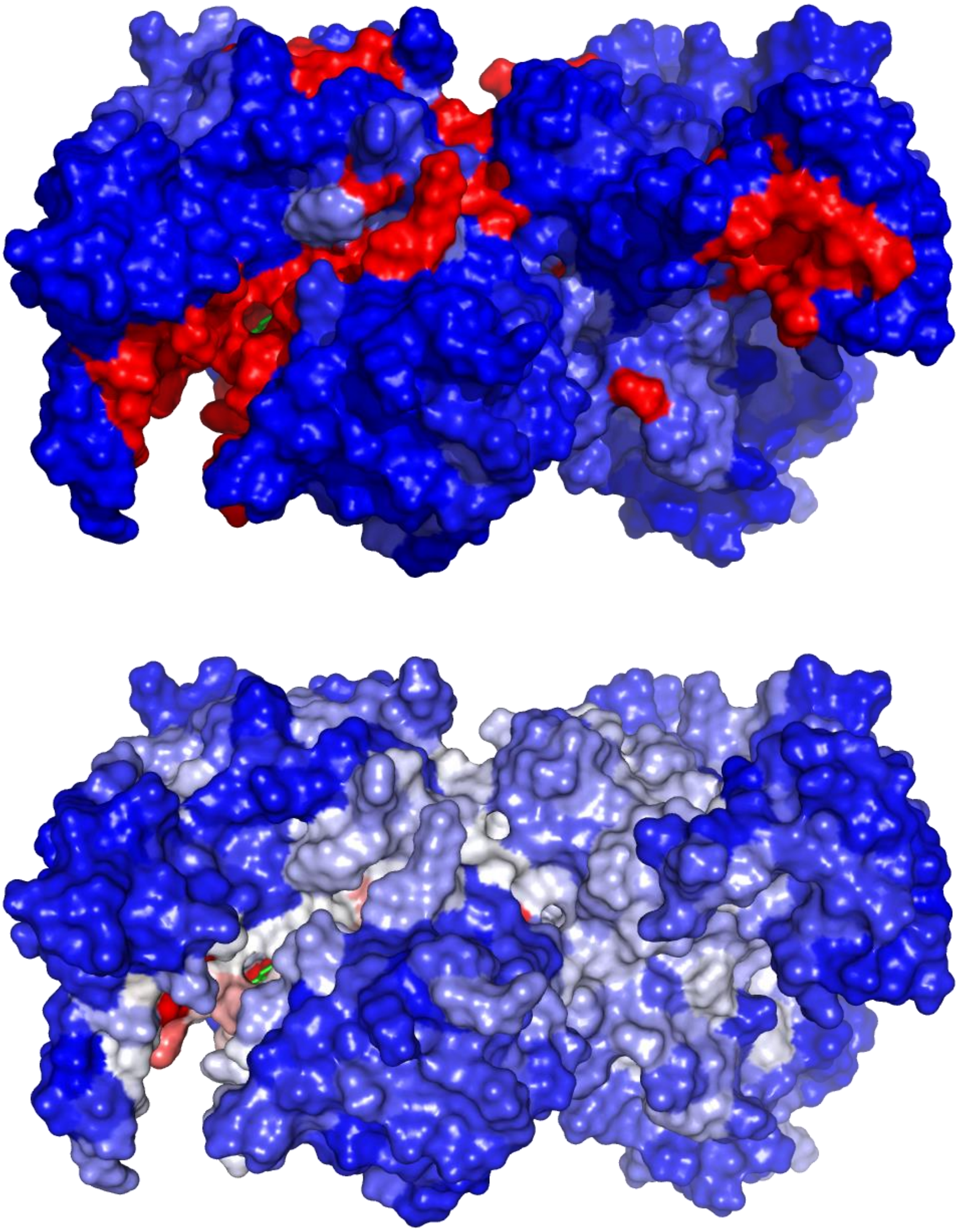


Figure 5.2: *Two models' predictions made on the same protein (myosin II heavy chain, PDB code: 2XO8). Similar to Figure 5.1, both predicted the allosteric site correctly. The 'pseudo-apo' model (top) also predicted a large fraction of the whole protein, while the final model (bottom) was more precise, selecting far fewer residues not in the allosteric site.*

5.5 Summary of Models

The performances of all models discussed so far in this chapter, as well as the final model utilising MD data, are summarised in Table 5.1.

Training data	Testing data	Performance by visual inspection (%)
MD	MD	72
crystal*	crystal*	73
crystal*	min-avg*	63
min-avg*	min-avg*	70

*dataset based on single conformation

Table 5.1: *Summary of models by data used in training, data used in testing and overall performance by visual inspection. Unless MD data was utilised, all variables were based on a single protein conformation.*

Since the single-conformation models were built purely from data derived from fpocket, it can be surmised that differences in performance are likely due to the fpocket scores in proteins that proved to be the "deciders." A Venn diagram of the three top performing models is presented in to succinctly capture which ones failed to predict the allosteric sites of which proteins. Proteins are listed by PDB code for brevity; the reader is referred to Table 4.1 for the proteins' identities and to Appendix 6 for their structures. An initial scan of the PDB codes did not reveal any significant trends: no type of protein was particularly poorly predicted in terms of class, isoform, size or number of chains.

Proceeding to look at individual groups, it could be seen that only 6 proteins were not predicted correctly by any model. These could be considered the most challenging proteins for RF to make a successful prediction on, regardless of the supplied training data. An examination of this set of structures confirmed that the allosteric sites indeed appeared "awkward" to the eye. They tended to be located in small and shallow cavities; this would have a significant impact on fpocket scores, which were known to be important descriptors across all models. Indeed, they were the *only* descriptors in the non-MD-based models. 2BKK contained another protein as its "ligand," and was included in datasets more as a curiosity to see how predictions would fare. The allosteric site as defined in this work covered a large face of the protein and was likely too poor and diffuse a definition for models to work with.

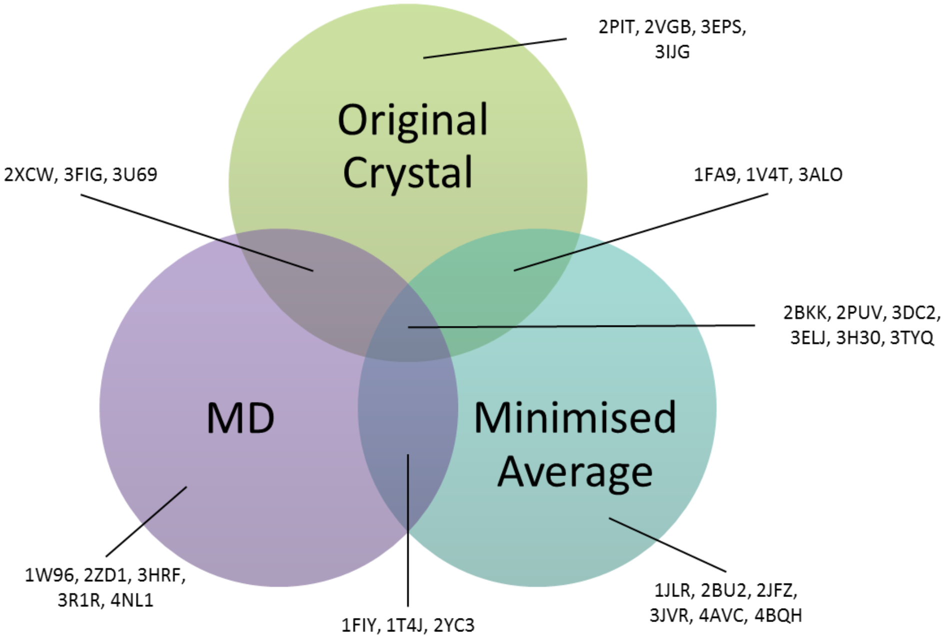


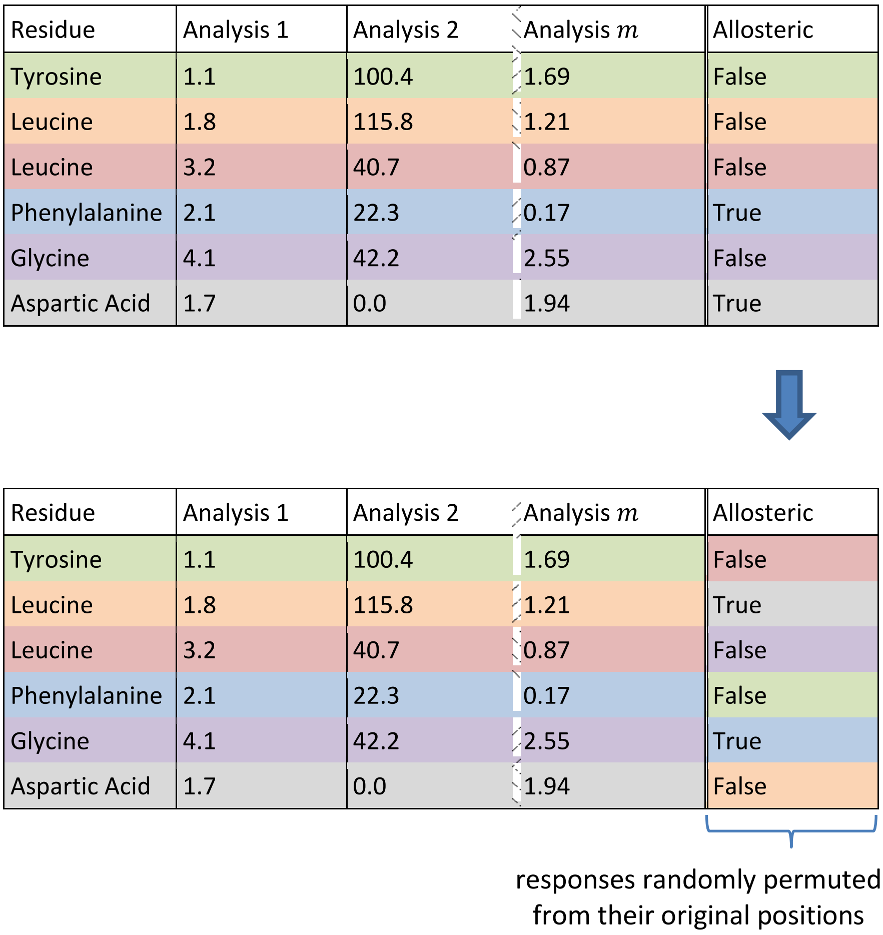
Figure 5.3: *A Venn diagram depicting the three different training sets from which RF models were constructed and which proteins were* <u>*incorrectly*</u> *predicted.*

A total of 9 proteins were only predicted correctly by one model: 3 per model. Once more, there was no particular trend in terms of the proteins' identities. However, it can be surmised that the fpocket scores varied significantly between conformations. For example, for those sites only predicted successfully by the final, MD-based model (1FA9, 1V4T, 3ALO), the fpocket scores of both the initial and average conformations must have been unusual and confounding to the models; only the remaining descriptors available exclusively to the MD-based model were able to "rescue" the predictions.

The converse must be true as well: the fpocket data must have been the "dead giveaway" descriptors for those predicted correctly only by single-conformation models (1W96, 2ZD1, 3HRF, 3R1R, 4NL1), and the MD-based descriptors served only to confound that model.

5.6 y-randomisation

A simple but effective method for testing a predictive model's ability to fit data to a genuine signal is y-scrambling or y-randomisation(183, 184). If a model's predictions are based upon real connections between the response, y, and predictors, x, it follows that severing this connection must cause a drastic drop in model performance, since there remains only noise upon which the model can be constructed. This is tested by randomly rearranging the values of y, scrambling any meaningful signal in the data. Figure 5.4 illustrates the process of y-randomisation on an example dataset.



Residue	Analysis 1	Analysis 2	Analysis m	Allosteric
Tyrosine	1.1	100.4	1.69	False
Leucine	1.8	115.8	1.21	False
Leucine	3.2	40.7	0.87	False
Phenylalanine	2.1	22.3	0.17	True
Glycine	4.1	42.2	2.55	False
Aspartic Acid	1.7	0.0	1.94	True

Residue	Analysis 1	Analysis 2	Analysis m	Allosteric
Tyrosine	1.1	100.4	1.69	False
Leucine	1.8	115.8	1.21	True
Leucine	3.2	40.7	0.87	False
Phenylalanine	2.1	22.3	0.17	False
Glycine	4.1	42.2	2.55	True
Aspartic Acid	1.7	0.0	1.94	False

responses randomly permuted from their original positions

Figure 5.4: *An example dataset where the responses match each case correctly, indicated by colour (upper table). In y-randomisation, the response values are permuted randomly, severing any overall link between them and the predictors (lower table).*

Multiple random permutations of the data are generally trialled in this way, aggregating the result. The genuine model must perform significantly better than the y-randomised permutations in order to have merit.

Akin to administering a placebo in clinical trials, y-randomisation serves to provide a baseline of performance that can be expected from a model. If the genuine model performs no better than the y-randomised models, it cannot be trusted. It is especially useful to have this benchmark when dealing with atypical datasets such as the highly imbalanced one in this project, since it reveals the performance attainable by predicting on noise alone. This is an even more important control in light of the proximity measures, which showed that there was little separation between the classes in terms of their properties in the final model.

The same dataset as the one used for the final model was used for this exercise. For each protein, a model was trained on the data of the remaining 59 proteins with the response randomly permuted (R contains a function to perform this). Testing was performed on protein kept aside, noting the confusion matrix. This procedure was repeated 10 times per protein, totalling 600 RF models. For each batch of 10 confusion matrices, the values of a, b, c, and d were averaged to the nearest integer. The results were conclusive: in all 60 cases, all residues were called *false* on average. Such an unambiguous result can be summarised with a single confusion matrix as shown in Table 5.2.

		Predicted	
		False	**True**
Observed	**False**	100%	0%
	True	100%	0%

Table 5.2: *A single confusion matrix summarises the y-randomisation procedure. For all proteins in the dataset, the aggregated results showed all residues classed as false.*

This result is precisely what one would expect with an imbalanced dataset containing no link between predictors and response. With the predictors effectively rendered meaningless, RF achieves maximum overall accuracy by classifying everything as the majority class. This issue was discussed in Section 1.4.5. In any case, a clear loss of signal was observed upon the scrambling of the response, indicating that the original RF model was fitting genuine trends in the data.

5.7 MD Stability

This section details a number of investigations into the MD trajectories generated as part of this project.

5.7.1 Use of MD to equilibrate initial structure

This project attempted to address a prevalent issue in the area of allosteric site prediction, namely the use of structures sourced from crystallographic databases without prior removal of ligand imprints. The proposed solution was to perform MD on the structures. Such simulations entail thorough minimisation and equilibration phases which could be exploited to handle ligand imprinting.

An imprinted protein conformation – that is, one that is unaltered save for the deletion of its co-crystallised ligand *in silico* – retains a structural deformation at an energetic cost without the stabilising interactions provided by the former presence of the ligand. In terms of a structural ensemble, it is unlikely to reside in a local minimum of an energy landscape. Thus, procedures that drive a molecular conformation towards an energy minimum, such as the standard phases of MD simulation, are suitable for removing the ligand imprint.

The results presented here are for Ser/Thr kinase CK2 (PDB code: 3H30), chosen arbitrarily from the proteins in the dataset as a representative case. To confirm that the preparatory phases of the MD simulation had sufficiently equilibrated the initial structure, the trajectory frames were aligned to the initial frame and the RMSD was tracked over the simulation. A plot of this result is shown in Figure 5.5, with a close-up of the key first portion following in Figure 5.6. The green line in these figures marks the point at which the simulation entered its production phase.

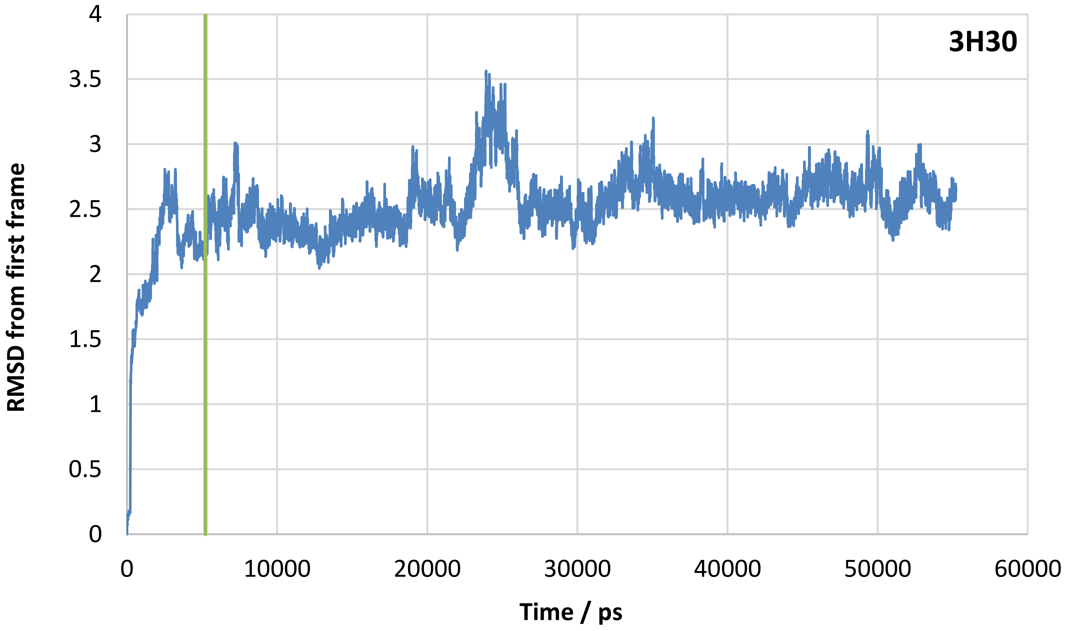


Figure 5.5: *A RSMD trace of 3H30's MD trajectory against its initial state. By the time the production phase commenced (marked by the green line) the protein's fluctuations had stabilised.*

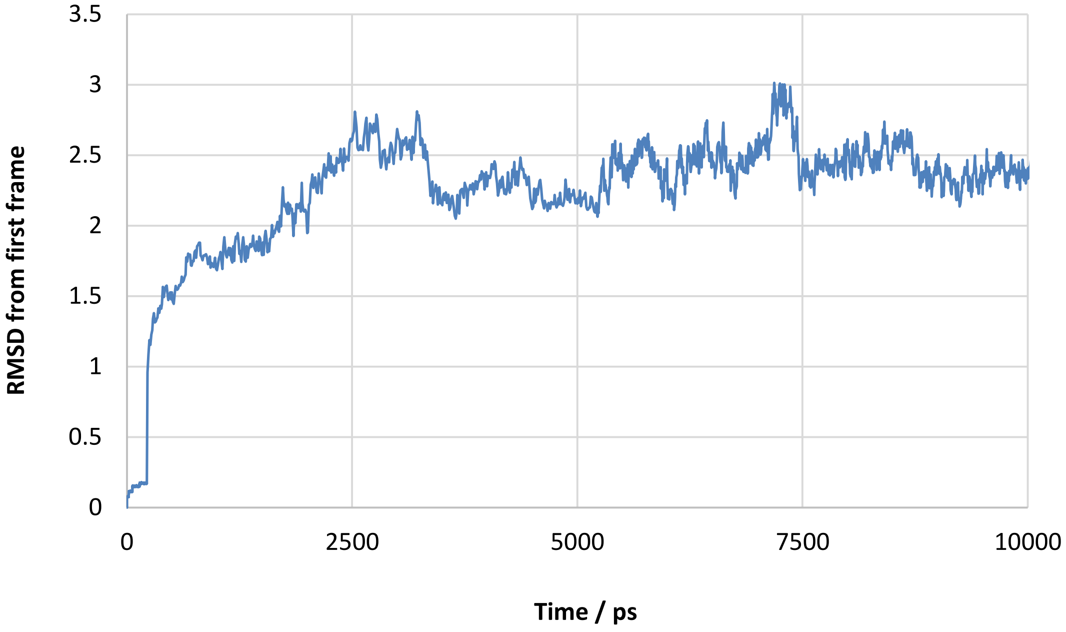


Figure 5.6: *A close-up of the plot in Figure 5.5. By the end of the heating phase alone (marked by the orange dashed line), the protein's fluctuations had not yet stabilised, indicating the importance of an intermediate equilibration phase.*

It can be seen from the RMSD trace that the fluctuations immediately before the production phase did not differ significantly from those observed after. This suggested that the protein had transitioned to a 'steady' dynamic state – that is, a state equilibrated with the ambient temperature, pressure and solvent – prior to entering the production phase. Interestingly, the close-up view in Figure 5.6 showed that the heating phase alone (marked with a dashed orange line) did *not* adequately prepare a protein for production MD, highlighting the importance of the intermediate equilibration phase.

5.7.2 Multiple simulations from a single starting structure

Before a MD simulation commences the atoms of the system are stationary. They have three-dimensional coordinates associated with them, *i.e.* a position in space, but no velocities. The software must artificially create initial velocities for the system, from which it can proceed to calculate new ones on application of the force field equation. Initial velocities are determined by a random seed, effectively giving each atom a 'push' in a random direction. It is probable that the magnitudes and directions of the initial velocities affected the regions of conformational space sampled by each trajectory. For the purposes of this project, this in itself was not an issue so long as the final prediction was not drastically affected. To investigate this, a small number of proteins were chosen arbitrarily. For each protein, a further two MD trajectories were simulated using the identical starting structure, totalling three. A different random seed was used each time, thus generating sets of three non-identical trajectories.

The trajectory sets were then compared directly by overlaying the RMSD traces from the first frame (which was identical for all three). The results presented in Figure 5.7 and subsequent figures are for one of the four proteins investigated, phosphoenolpyruvate carboxylase (PDB code: 1FIY). The findings for this protein are representative of the others'. It can be seen that broadly similar, though by no means identical, RMSD traces were produced for all three trajectories. It was perfectly acceptable to encounter variations of this scale since the protein was on different trajectory in each case. What was important was whether the three trajectories, which appeared similar by this simple analysis, produced widely differing descriptors, in turn resulting in different predictions when passed to a RF model.

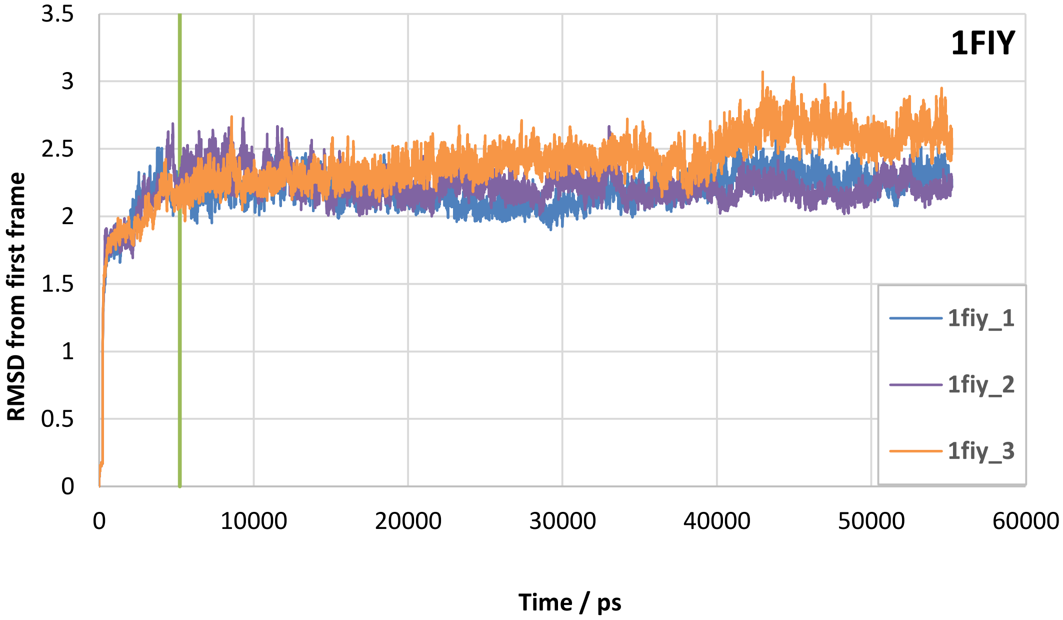


Figure 5.7: *The RMSD traces of three MD trajectories of 1FIY. The starting structure was identical for all three, but a different random seed was chosen in each case, leading to non-identical trajectories.*

The simplest way to proceed was to recalculate all descriptors for this protein using the alternative trajectories, make the predictions and view the results. It should be noted that this exercise was not performed on the final model but an intermediate one. However, this was acceptable since it was the model's consistency, rather than its performance, that was being tested here.

The three predictions for 1FIY were rendered using the scoring function described in Chapter 4 and are shown below. Figure 5.8 shows the prediction for the original trajectory ('1fiy_1' in Figure 5.7), with Figure 5.9 and Figure 5.10 showing the other two predictions ('1fiy_2' and '1fiy_3' respectively in Figure 5.7). Some minor variances can be seen among them, more so between the first and the other two. However, the overall 'hot' and 'cold' regions of the protein appeared broadly consistent across the three predictions.

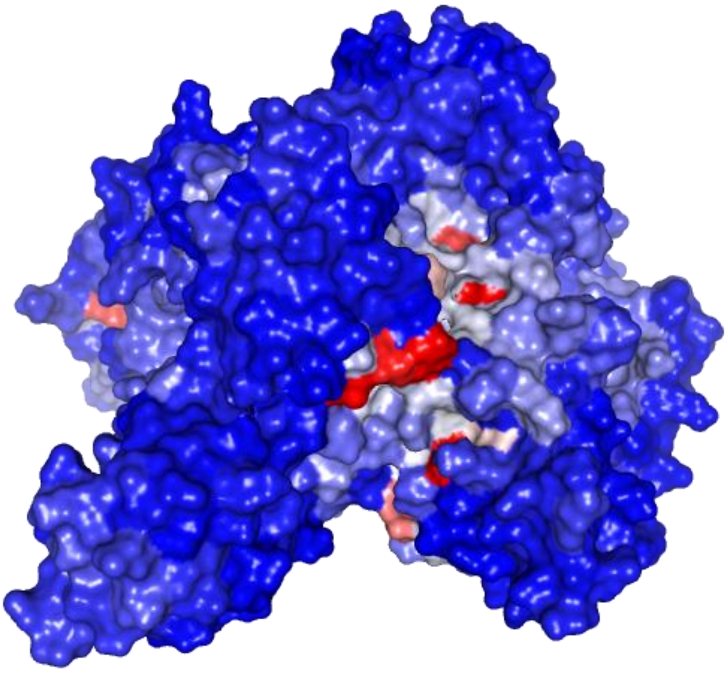


Figure 5.8: *The prediction produced with descriptors derived from the first trajectory of 1FIY.*

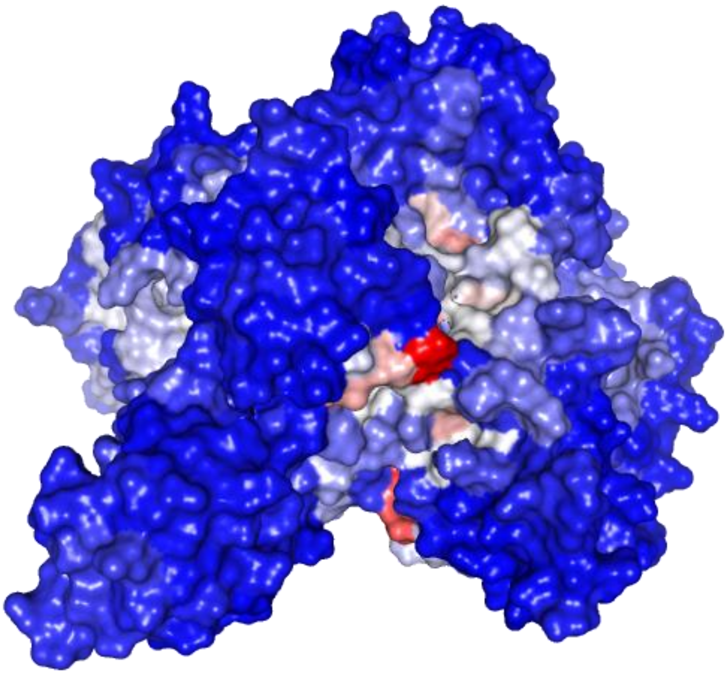


Figure 5.9: *The prediction produced with descriptors derived from the second trajectory of 1FIY.*

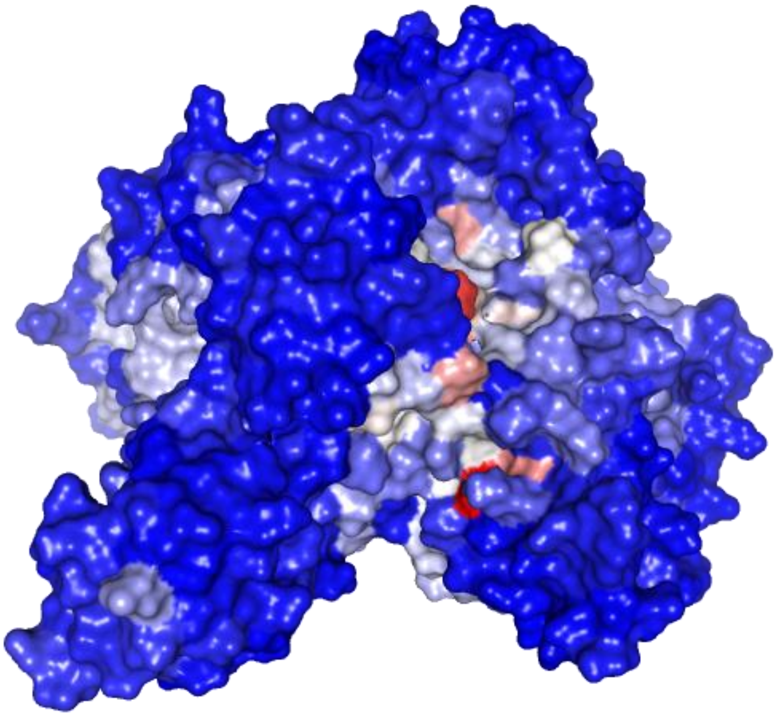


Figure 5.10: *The prediction produced with descriptors derived from the third trajectory of 1FIY.*

This investigation showed that the model was fairly stable to the inevitable random chance involved in initiating a MD simulation. Minor variations could be discerned, but nothing major enough to alter the guidance offered by the prediction. These small differences did not necessarily betray a weakness in the model; in fact, quite the opposite, they indicated that the descriptors used by the model are sensitive enough to the movements made by the protein over the simulation. However, this conclusion inevitably implied that there was a potential vulnerability in the model due to insufficient sampling.

This vulnerability was expected from the outset. Ideally, far longer trajectories (at least an order of magnitude longer) would be generated for all proteins included in the dataset. This would have dampened the impact that minor fluctuations in a trajectory had on the final descriptors for a protein. It was out of necessity rather than choice that the trajectories in this project were limited to 50 ns of production MD. An immediate lesson to be learned from this investigation was that, when making a live prediction, several MD trajectories ought to be prepared, as they were here. The predictions could then be compared and cross-examined with their RMSD traces to determine the causes of any variations in predictions before making a manual judgement.

5.7.3 Different starting structures

A similar investigation to the above was performed, this time examining the variation in predictions obtained when different starting structures were used. A second, alternative starting structure was identified for two proteins in the dataset. The proteins were CDK2 (PDB code: 3PY1) and NS5B RNA polymerase (PDB code: 4NLD); in both cases the alternative structure was the crystallised apo form of the protein (PDB codes: 3PXR and 3TYQ, respectively). The alternative structures were treated as distinct proteins: MD trajectories of each were simulated, and all required descriptors were calculated in order to allow predictions to be made. The results are shown below, with Figure 5.11 comparing 3PY1 and 3PXR and Figure 5.12 comparing 4NLD and 3TYQ.

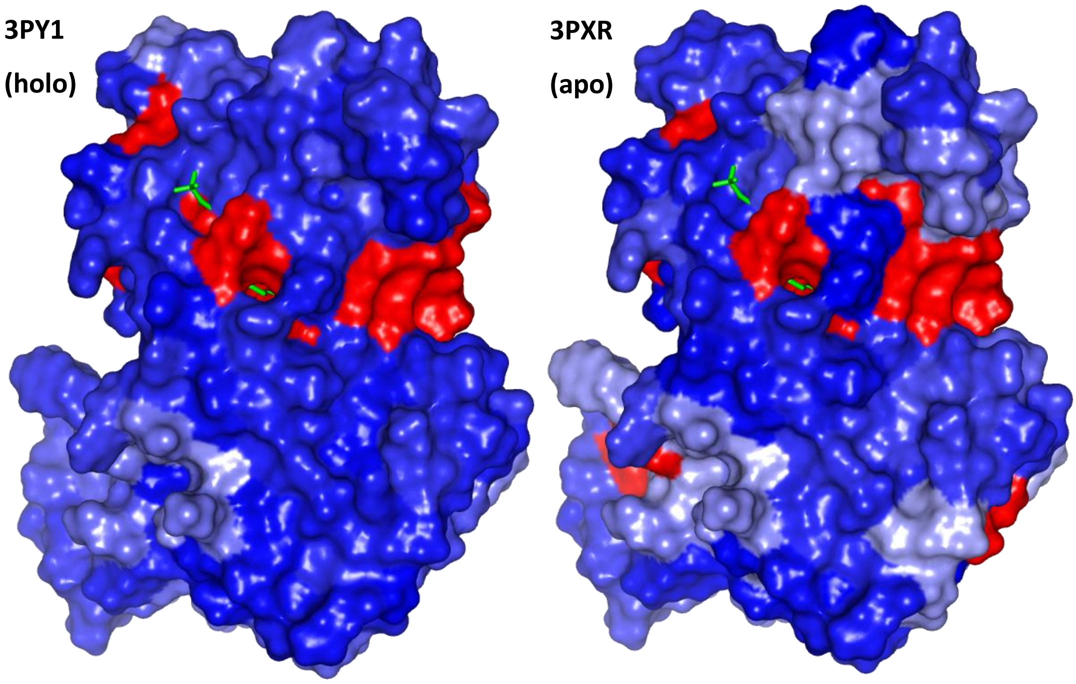


Figure 5.11: *The predictions for 3PY1 (left) and 3PXR (right). Both are conformations of CDK2, with 3PXR being the apo form. The ligand from 3PY1 was superimposed onto the structure for reference. A fair similarity in predictions can be seen between the two structures, despite them being treated independently.*

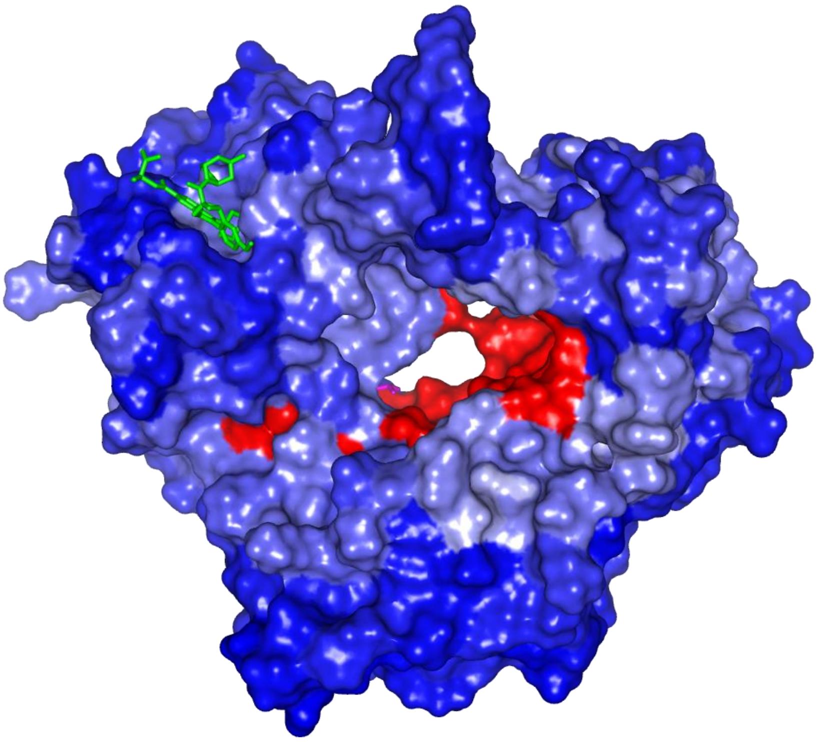
In Figure 5.11, the predictions made for 3PY1 and 3PXR appeared highly similar. A few extra residues were predicted to be allosteric in 3PXR (shaded red by the scoring function), but the major features of the predictions – the residues in the vicinity of the allosteric site, and the large patch of red residues directly to the right of it – were consistent. Considering that these predictions were made entirely independently from different starting structures, this was an encouraging sign. It suggested that influence on the descriptors due to the starting structure was being successfully removed by MD.

Figure 5.12 shows the predictions for 4NLD and 3TYQ. These were undoubtedly unsuccessful predictions: no residues in the vicinity of the allosteric site were predicted correctly. However, this was an exercise in consistency rather than accuracy. So long as the prediction remained approximately the same when an alternative structure was used, the analysis could be considered a success. Indeed, this was the case: both predictions highlighted little beyond the central cavity of the protein, which is the orthosteric site.

It should be noted that, originally, the allosteric ligand in 3TYQ was superimposed from 4NLD, but this did not align well and resulted in major overlap with the protein structure. Instead, another allosteric ligand was taken from a third structure of NS5B RNA polymerase (PDB code: 1GX5). 1GX5 could have been used for this analysis just as well, but 3TYQ was chosen over it because the crystal structure was solved at a higher resolution and it contained fewer missing loops.

4NLD

(holo)



3TYQ

(apo form with superimposed ligands from alternative structure)

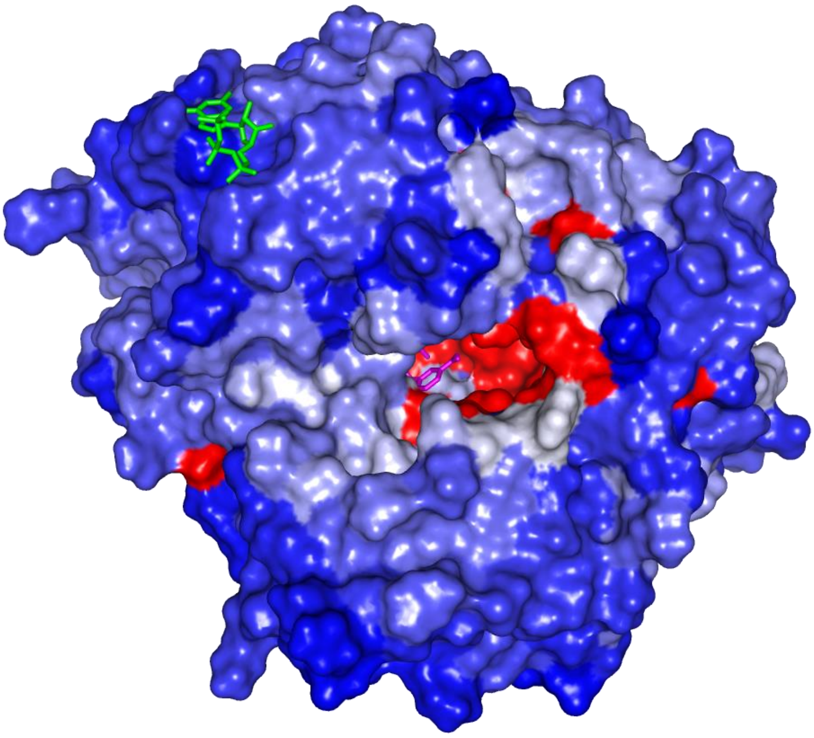


Figure 5.12: *The predictions for 4NLD (top) and 3TYQ (bottom). Both are conformations of NS5B RNA polymerase, with 3TYQ being the apo form (with ligands superimposed to highlight the binding sites). Though both predictions were unsuccessful, they were nevertheless consistent enough to demonstrate the model's stability to the precise conformation of the tested protein.*

5.8 Comparison to Allosite

As detailed in 1.5.3, Allosite is a web-based, automated workflow that first calculates fpocket scores on an inputted protein structure. The scores for each cavity are then passed to a pre-trained SVM-based predictive model to detect pockets containing allosteric ligands. There are key similarities between Allosite and the crystal structure-based RF model described in section 5.2. Firstly, descriptors are calculated from original crystallographic data. The descriptors used are the same, namely the outputs of the fpocket program. Finally, machine learning is utilised to detect trends in the data and predict allosteric binding sites. Aside from the choice of machine-learning algorithm the underlying concept of Allosite is similar to that of the project.

With its fast, web-based implementation, Allosite offered an interesting opportunity to compare methodologies. The main difference is that Allosite works purely on the output of fpocket, which identifies and subsequently determines properties of sites rather than individual residues. Thus Allosite operates in terms of sites, rather than individual residues, and so deals with a much smaller number of cases. In this project's dataset of 60 proteins, the number of sites per protein, as determined by fpocket, varied from 8 for the smallest to approximately 500 for the largest (*cf.* approximately 250 to >2000 residues per protein). If it was classified as allosteric by the SVM model, a site was filled with dummy atoms (originating from the fpocket analysis) to mark its contours on the protein and returned in .pdb and .pml formats. An example output from Allosite, which is in essence the same as the output of fpocket, is shown in Figure 5.13.

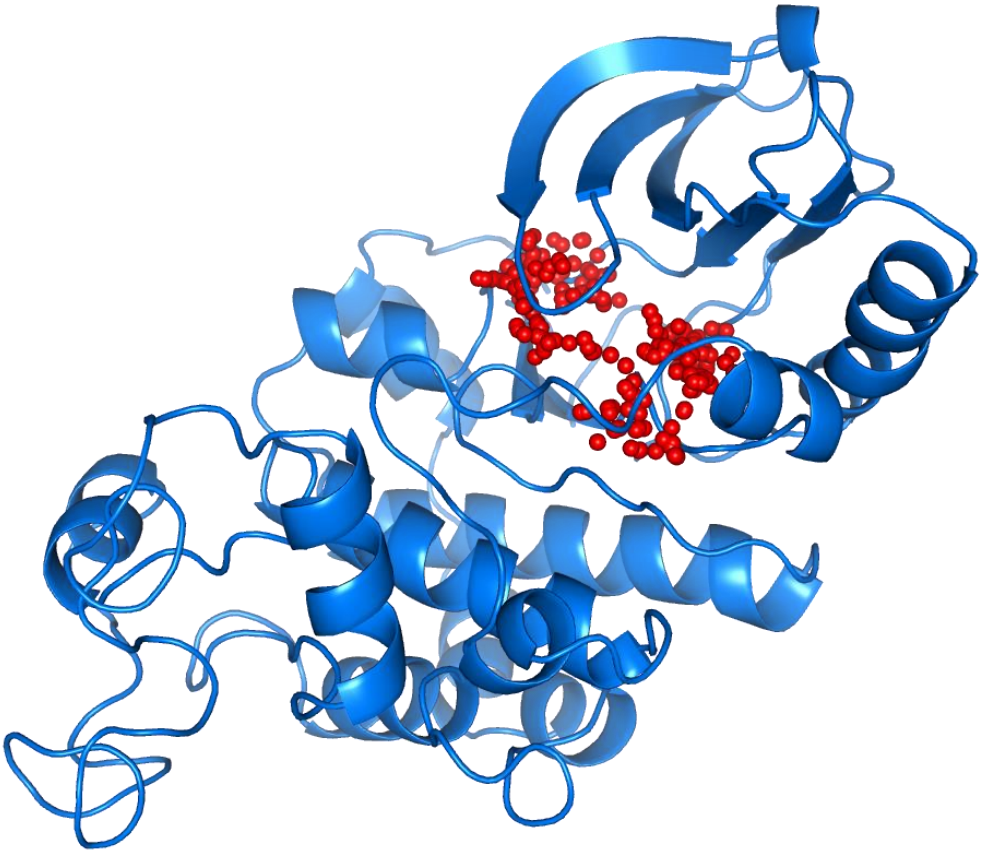


Figure 5.13: *An example output of the Allosite web server, with the protein structure rendered as a ribbon. The predicted site is filled with dummy atoms originating from the fpocket analysis that mark its contours.*

The authors do not enter into any lengthy discussion on their criteria for a successful prediction, but they claim to an accuracy of approximately 95%. However, accuracy can be a misleading measure when dealing with an imbalanced dataset. The authors do cede this, reporting sensitivities and specificities of >83% as well. Together, these two measures are the basic of the ROC curve, which indeed accounts for class imbalance. Unfortunately, this level of performance was not reflected when the 60 proteins forming this project's dataset were passed to the Allosite server for a prediction – both the original crystal structures and the 'pseudo-apo' minimised-average structures of MD simulations.

Applying a similar criterion to that imposed on the RF-based models constructed in this project, an Allosite prediction was deemed to be successful if the selected pocket(s) at least partially overlapped with the allosteric ligand. The example prediction shown above was indeed successful, as shown below in Figure 5.14 with the ligand superimposed.

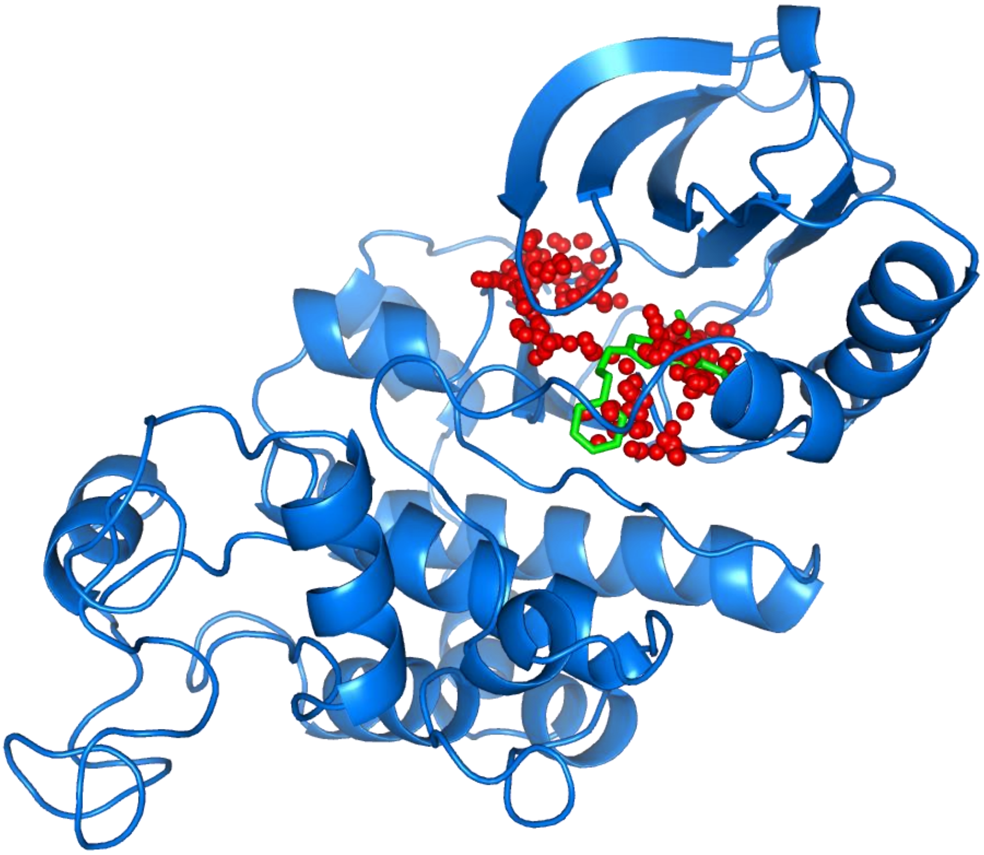


Figure 5.14: *The allosteric ligand is superimposed onto the Allosite prediction. In this case, the prediction was successful.*

By visual inspection, the Allosite server returned a 46% success rate for the original crystals structures and a 48% success rate for the 'pseudo-apo' minimised average structures. These results were comparable to the results of the crystal structure-based RF model trained for this project: the former against the 73% success rate detailed in section 5.2 and the latter against the 'pseudo-apo' data tested in section 5.3 – this achieved a 63% success rate. Like Allosite, the RF model only utilised fpocket data but achieved a significantly greater success rate for both sets of data.

It must be noted that some of the Allosite prediction requests failed. This was likely for benign reasons such as an input file not being formatted precisely as required, but with no debugging information returned by the server, the problem could not be corrected. Consequently, the figure of 46% success rate for the original crystal structure set is based on 25 correct predictions out of 54, with 6 cases failing to return. It was therefore theoretically possible for Allosite to attain 52% (31 out of 60) if the failed cases, having been debugged, were predicted correctly. Equally, the figure of 48% for the 'pseudo-apo' set is based on 25 correct predictions out of 52, with 8 cases failing to return. It was therefore theoretically possible for Allosite to attain 55% (33 out of 60). The results described in this section are summarised in Table 5.3.

Test Set	**Allosite prediction rate**	**Potential Allosite prediction rate**	**Equivalent prediction rate of RF model**
crystal	25/54 = **46%**	31/60 = **52%**	44/60 = **73%**
min-avg	25/52 = **48%**	33/60 = **55%**	38/60 = **63%**

Table 5.3: *Summary of Allosite performance against the original crystal structures and 'pseudo-apo' structures of the 60 protein dataset. Since some predictions failed to run, the maximum potential performance is calculated, assuming that all of these would have been returned as correct predictions. These are compared to the appropriate RF-based model constructed for this project.*

It has already been discussed that crystal structures retain significant imprints of ligands even after they have been removed *in silico*. This would invalidate the results of both Allosite and the equivalent, RF-based version of it constructed as part of this project, since these were trained from data based on crystal structures. At the very least, this comparison offers strong evidence for the superiority of RF over SVM for this type of data.

6. Conclusions and Future Work

6.1 Final Model Performance

The main result of the project was the production of a RF model that predicted the location of allosteric sites in proteins with, by what is proposed to be fair criteria, a 72% success rate. The model was trained on data derived from MD trajectories, granting access to dynamic information from the proteins. Success was achieved with no prior knowledge of the sites' locations, utilising instead only data that could be determined equally well from an apo structure. This is a critical detail. The use of data obtained from proteins in their bound conformation is a trap that most predictive methods to date have fallen into, and one that a significant portion of the work in this project was devoted to avoiding.

The conformations of crystallised, ligand-bound proteins are generally highly contorted in order to accommodate the interacting ligand, particularly in the close vicinity of it. The ligand's presence provides an energetic ballast that stabilises the conformation and makes it viable. Without it, the protein would generally 'exhale,' at least partially collapsing the cavity left by the ligand and reorienting to a lower energy state. It is not sufficient to delete the ligand from a ligand-protein complex *in silico* and treat the remaining protein as though its binding site has been concealed. A small selection of metrics from the literature can be enough to detect the abnormalities left behind by such crude treatment. Passing the data to a machine learning algorithm such as RF can quickly lead to spectacular results: in this project, 73% success rate was achieved using only data from fpocket.

Unfortunately, models trained on such data do not match their tested performance when faced with a protein that is not in a similarly contorted conformation. This was demonstrated by testing the same model with the minimised average structures of each protein's MD trajectory rather than the original crystal structures. The result was a substantial drop in performance from 73% to 63%.

Encouragingly, the issues of ligand imprinting and loss of performance were both resolved by constructing an equivalent model based on minimised average structures of MD trajectories. This model achieved a success rate of 70% when making predictions on non-imprinted structures, an increase from the 63% attained by the model based on crystal structures. However, this success rate came with a large number of false positive predictions and, crucially, required MD to produce the minimised average structures. If one has gone to the trouble of generating MD trajectories, there is little merit in opting to use this model over the final model developed in this project. The final model made more extensive use of the data: this was reflected in the slightly improved success rate and vastly improved incidence of false positives.

Nevertheless, a simple model could potentially be developed with further investigation. It would be of some value to the community to produce a model that could proceed from initial input to final prediction with a quick turnaround. Such a model could be implemented as a web-based service akin to PARS, SPACER or Allosite. However, in order for this to be worthwhile endeavour a more efficient method for removing a ligand imprint from a protein would be required.

6.2 Dataset

The results showed that the variables used to train this model were indeed effective predictors of the residues in the vicinity of allosteric sites. Each variable quantified a directly measureable structural, topological or energetic property that required no prior knowledge of the location of an allosteric site. These could therefore be calculated in an identical manner for proteins with no known allosteric site.

The array of variables used was indeed diverse; in particular, the dihedral angle analyses were novel, having been developed specially for this project. However, many other analyses that could produce potential descriptors were inevitably excluded. There was no reason for this beyond there being too little time or too little expertise available to implement them. A specific example is the COREX/BEST algorithm(185, 186), a methodology that has received much attention in the field. This was identified at an early stage of the project as a method that yielded the correct type of variable, *i.e.* a single value per residue, and so could be utilised. However, it was not successfully implemented; if this were revisited in the future, it would likely be a valuable addition to the dataset.

The variables produced by SPACER, named local closeness and binding leverage, are again of the correct format for immediate addition to the dataset. In fact, many of the project's 60 proteins were submitted to the SPACER server for analysis, though the results were not presented here due to the large number of cases returning with errors. This was likely due to bugs concerning the format of the .pdb files used as input. Solving this problem would not only allow for comparison of the SPACER model's performance to the final proposed model in this work, but also provide two new variables for inclusion in the dataset.

Provided the dataset remains filtered for correlated or noisy variables, a RF model trained from it can only improve as it grows. The dataset can be grown either by increasing the number of cases or the number of variables. There is only one way to increase the number of cases in the dataset, and that is to identify more proteins with known allosteric sites. It is recommended that this search is performed manually, or at least that any automated search is followed by a manual filter, since many subtly unsuitable (or potentially misleading) candidate protein structures exist in the literature.

The addition of more proteins to the dataset steadily improved model performance in the project. There may be a ceiling to this improvement; this situation was encountered in the initial stages of the project when only a small number of descriptors were available. However, when more descriptors were implemented the performance continued to rise as the number of cases did.

Further analyses that generate descriptors for the dataset must also be sought from the literature. If this work were to be continued, as much focus ought to be devoted to identifying these as to identifying more proteins. It was upon incorporating a powerful new set of variables, such as those from fpocket, that the greatest boosts to performance were observed.

Of course, each time a new protein was added to the dataset, all descriptors had to be calculated for its residues, and when a new descriptor was added, it had to be calculated for all residues. The logistics of growing the dataset became increasingly challenging as it grew. The next section reflects on these aspects of the project.

6.3 Workflow Automation

A vast quantity of programming was required to make the volume of data calculated for this project feasible. Most stages in a protein's journey from initial download as a .pdb file to a full list of calculated variables in a formatted dataset required some support from custom scripts. These scripts varied from performing minor editing of file formats in order to fit the requirements of the next piece of software using them to performing entire analyses wrapped into functions. The preparation of proteins for MD was, out of necessity, entirely automated in the latter stages of the project. The '1-in-1-out' RF cycles that became the standard method to produce predictions on all 60 proteins were automated.

A further necessary development of this project was a script to automatically compile a formatted dataset that was ready for use with R. Over the course of the project a great number of datasets, each containing different combinations of residues and variables, was required; the time invested in coding this script was likely the most valuable use of any time on the project. Its structure was fully modular, such that it was trivial to update as new variables or proteins were introduced.

It was for more than merely pragmatic reasons that much time was devoted to automating the project workflow. In doing so, all written code remained in place after producing the final dataset. As a result, aside from being able to quickly pick up the project from where it was left, the successor would also be able to make any desired alterations and additions to the dataset by modifying the appropriate code in the workflow.

To that end, it would be interesting to modify the model to operate on sites rather than individual residues. The majority of alternative models in the literature, such as PARS and Allosite, are site-centric. Throughout this work, residues were chosen as the cases of the RF dataset. This is because no ambiguity is associated with the identity of a protein residue: each is a rigidly defined chemical entity. This provided a stable anchor point from which to embark on the gathering of data. However, provided a suitable definition for them is reached, approaching the problem in a site-centric manner could allow access to a host of properties that, for all is known at the time of writing, could be more relevant to the prediction of allosteric sites.

This suggestion is indeed a worthwhile avenue of future work, though it cannot be denied that it would require a thorough overhaul of the present workflow. However, it also cannot be denied that this task is far more preferable to beginning again from nothing.

6.4 Random Forest

The initial optimisation procedure suggested that a radius of 7 Å was the best choice of those suggested. This indeed seemed appropriate, since it had already been found to be the approximate sphere of influence (from the alpha carbon) of an amino acid in a protein(169, 170), and also agreed with observations made in section 4.2.3 that the amino acid composition of an allosteric site deviated most from the standard for proteins up to a distance of approximately 7 Å. A later round of optimisation found that a 5 Å radius yielded better results. As this was found empirically the underlying reason for it was not known, but the sphere of influence of an amino acid could perhaps be split into inner and outer layers, with 5 Å roughly marking the inner layer where frequent and major interactions occur.

The initial optimisation also showed that a downsampling *false*/*true* class ratio of 1.75:1 was optimal. There was no artificial reason for this value to appear statistically optimal, lending some credence to its selection. The later round of optimisation did not reveal a superior ratio. Once again, this is a purely empirical finding: with no rationale for this specific ratio to be optimal, it should be monitored closely if the dataset is extended, since it could drift.

It was mentioned in earlier sections that there was a large subjective component to quantifying the true effectiveness of the model. Extreme care was required in the interpretation of evaluation measures, and even then they provided only rough guidance. However, a few conclusions could safely be made, the first being that the model was significantly better than random chance. The evaluation measures confirmed this statistically (and were capable of doing so). In particular, the mean ROC AUC of the model iterations showed a clear upward trend in ROC AUC, *i.e.* away from the baseline of random chance, as the dataset expanded. The y-randomisation procedure, which tested precisely the hypothesis that predictive power is due to random chance, provided a final confirmation.

However, any statistical measure based solely on the counts of *true* and *false* cases came with the important caveat that the 'real' *true*/*false* classifications of the protein residues were the result of a fairly crude definition based on the proximity to a ligand in the original crystal structure. It is unlikely that a simple radius around the binding site accurately captures all residues participating in an allosteric event. Many residues relevant to the allosteric phenomenon may reside outside this radius and, conversely, residues within the radius may in fact be of little relevance at all. This is discussed further in the next section.

6.5 Site Definition

Conclusions must be drawn very carefully when dealing with predictive models, and doubly so when the response has been artificially generated and bears an element of ambiguity. The reason there is a need for work such as this is that allosteric sites have not been fully characterised. They can only be defined empirically: in relation to ligands they bind with and the orthosteric sites they are coupled with. If a ligand binds to a protein and modulates activity at the orthosteric site, its location is considered allosteric. Little about a site itself tangibly defines it as allosteric, making it problematic to train a model to identify one.

The site definitions used in this project were based on a very simple, distance-based algorithm, though allosteric effects were already known to be more complex. Residues far from the allosteric site can play a pivotal role in relaying an allosteric signal, and some proximal residues can be irrelevant, though there is no known way to determine this from a crystal structure of the allosteric-bound complex.

It was for this reason that the optimisation protocol described was necessary. Iteration over downsampling class ratios in training sets was performed to determine which would yield the best predictions of response classes; however, the response classes *themselves* were also varied according to different site selection radii. Optimum predictive power was desired, of course, but it was important to consider the implications of using each radius as well as the ability of the model to classify sites according to it. For example, a model well able to classify residues at 50 Å site selection radius should not be considered a good model for the prediction of allosteric sites, since 50 Å is a very poor radius for site definition that likely encompasses the majority of a protein. Such a model would be predicting nothing of value even if it predicted it well and was, by all evaluation measures, powerful and high-performing.

Put differently, an answer can only be as good as the question. The question asked of a constructed model – in this case, "which residues in the dataset are allosteric?" – had to be as carefully monitored as the answer it produced. After all, what exactly is an "allosteric" residue?

The requirement that the allosteric definition be reduced to rigid, binary states, *i.e.* to “allosteric” and “non-allosteric” residues, for the purposes of performing RF potentially exacerbated the problem, though it is an unavoidable consequence of using any classification-based method that data be put into discrete categories.

In summary, there was most likely a significant, though unknowable and unavoidable, level of noise in the response that the RF models were being trained to match data to. While the precise definition of an allosteric site is likely the most capricious area of this project and the most challenging to improve upon, it is also one of the most important. The y-randomisation procedure in section 5.5 emphatically demonstrated the destructive effect that noise in the response can have on model performance. Thus, eliminating noise in the response would surely lead to a great leap in model performance.

6.6 MD Trajectories

50 ns trajectories of production MD with 5 ns of equilibration time were the greatest that could be produced at the outset of this project with the computational power available. Though it can be said of virtually all MD-based research, it is worth stating that longer simulations would improve the reliability of the model. The danger of retaining the imprint of a removed ligand on a protein structure has been discussed at length, and the most certain method to minimise this danger (through MD simulation) is to give the protein as much time as possible to move away from its initial conformation. The longer the overall simulation is, the longer the initial portion that can be discarded. Furthermore, a protein will access more conformational space with a longer simulation, which can only enhance the quality of data derived from the trajectory.

Most investigations use upwards of 1 µs of simulation time to characterise a single allosteric site. Since the dataset built for this project was based on 60 proteins, it was not feasible to devote this much simulation time to each. Considering the aggressive rate at which GPU cards are improving, as well as the MD software itself – the release of AMBER 14 boasted a 30% increase in simulation speeds simply through code optimisation(187) – this quantity of simulation time per protein could soon be attainable with an affordable setup.

Though these improvements are constantly increasing the reach of 'brute force' MD of the type performed in this project – that is, constant volume and temperature simulations performed for as long a time period as possible – it would be wise to investigate alternative methods for enhanced sampling of the conformational landscape of a protein. Examples of such methods include accelerated MD(188), replica exchange(189) and Markov state models(190). While methods such as these will most likely not eliminate the issue of poor sampling entirely, they could certainly help to alleviate it. The latter two can also make better use of parallelised computational resources than conventional MD, which may suit individual research groups.

6.7 Summary

Much of the work described in this report constituted the construction of a robust dataset of residue features in proteins with known allosteric sites. The project culminated in the development of a RF-based model that demonstrated a 72% success rate in the prediction of allosteric binding sites in proteins.

The main result demonstrated the feasibility of using RF as a method for predicting the location of allosteric sites from a set of residue properties. The correct prediction rate is competitive with existing techniques. Moreover, unlike many existing techniques, this method does not inadvertently abuse skewed properties derived from the allosteric ligand-bound complex crystal structure to achieve its predictions. In a live situation where an allosteric site is genuinely unknown, this method should retain its level of performance.

It is perhaps a testament to the sensitivity of available metrics that ligand-imprinting is even an issue. It means that current methods are capable of homing in on these perturbed regions of protein structure. This can serve as a platform from which new methods such as the one presented here can be fine-tuned.

While allostery has been historically difficult to characterise, many successful strides have been made. Individual groups of researchers have approached the area from a great variety of angles, and many of their methodologies were utilised in this project to construct the dataset. The overarching goal was to demonstrate that a data-driven approach to characterising allostery was feasible, and this was certainly achieved with the use of RF.

7. Appendices

All appendices are deposited in Pure, the University's research information management system. These can be accessed electronically at http://pure.strath.ac.uk/portal/ by searching for the author (Antony Vassileiou), or directly at http://dx.doi.org/10.15129/d40aa95f-cd9a-47e2-abd4-a08381668b47. The contents of each Appendix are summarised below.

- **Appendix 1**

Raw data downloaded from the PGD

- **Appendix 2**

Percent-identity matrix including a numerical version

- **Appendix 3**

Graphs of deviations in site residue abundances against site selection radius

- **Appendix 4**

Tab-delimited text file containing all descriptors generated for the final 60-protein dataset

- **Appendix 5**

correlation analysis showing eliminated variables

- **Appendix 6**

a set of 60 PyMOL sessions with each protein’s rendered prediction from the final model

8. Bibliography

1. Good MC, Zalatan JG, Lim WA (2011) Scaffold proteins: hubs for controlling the flow of cellular information. *Science* 332(6030):680–686.

2. Nussinov R, Tsai C-J (2013) Allostery in disease and in drug discovery. *Cell* 153(2):293–305.

3. Fenton AW (2008) Allostery: an illustrated definition for the “second secret of life”. *Trends Biochem Sci* 33(9):420–425.

4. Monod J, Jacob F (1961) General Conclusions: Teleonomic Mechanisms in Cellular Metabolism, Growth, and Differentiation. *Cold Spring Harb Symp Quant Biol* 26(0):389–401.

5. Monod J, Wyman J, Changeux J-P (1965) On the nature of allosteric transitions: A plausible model. *J Mol Biol* 12(1):88–118.

6. Koshland DE, Némethy G, Filmer D (1966) Comparison of Experimental Binding Data and Theoretical Models in Proteins Containing Subunits. *Biochemistry* 5(1):365–385.

7. Gunasekaran K, Ma B, Nussinov R (2004) Is allostery an intrinsic property of all dynamic proteins? *Proteins* 57(3):433–443.

8. Gerek ZN, Ozkan SB (2011) Change in allosteric network affects binding affinities of PDZ domains: analysis through perturbation response scanning. *PLoS Comput Biol* 7(10):e1002154.

9. Mittermaier AK, Kay LE (2009) Observing biological dynamics at atomic resolution using NMR. *Trends Biochem Sci* 34(12):601–11.

10. Davis IW, Arendall WB, Richardson DC, Richardson JS (2006) The backrub motion: how protein backbone shrugs when a sidechain dances. *Structure* 14(2):265–274.

11. Frederick KK, Marlow MS, Valentine KG, Wand AJ (2007) Conformational entropy in molecular recognition by proteins. *Nature* 448(7151):325–329.

12. Henzler-Wildman KA, et al. (2007) A hierarchy of timescales in protein dynamics is linked to enzyme catalysis. *Nature* 450(7171):913–916.

13. Rasmussen BF, Stock AM, Ringe D, Petsko GA (1992) Crystalline ribonuclease A loses function below the dynamical transition at 220 K. *Nature* 357(6377):423–4.

14. Cooper A, Dryden DTF (1984) Allostery without conformational change. *Eur Biophys J* 11(2):103–109.

15. Formaneck MS, Ma L, Cui Q (2006) Reconciling the “old” and “new” views of protein allostery: a molecular simulation study of chemotaxis Y protein (CheY). *Proteins* 63(4):846–867.

16. Okazaki K-I, Takada S (2008) Dynamic energy landscape view of coupled binding and protein conformational change: induced-fit versus population-shift mechanisms. *Proc Natl Acad Sci U S A* 105(32):11182–7.

17. Weber G (1972) Ligand binding and internal equilibiums in proteins. *Biochemistry* 11(5):864–878.

18. Zhuravlev PI, Papoian GA (2010) Protein functional landscapes, dynamics, allostery: a tortuous path towards a universal theoretical framework. *Q Rev Biophys* 43(3):295–332.

19. Motlagh HN, Wrabl JO, Li J, Hilser VJ (2014) The ensemble nature of allostery. *Nature* 508(7496):331–9.

20. Cui Q, Karplus M (2008) Allostery and cooperativity revisited. *Protein Sci* 17(8):1295–1307.

21. Hilser VJ, Wrabl JO, Motlagh HN (2012) Structural and energetic basis of allostery. *Annu Rev Biophys* 41:585–609.

22. Tsai C-J, Nussinov R (2014) A unified view of “how allostery works”. *PLoS Comput Biol* 10(2):e1003394.

23. Monod J, Changeux JP, Jacob F (1963) Allosteric proteins and cellular control systems. *J Mol Biol* 6:306–329.

24. Berg JM, Tymoczko JL, Stryer L (2002) *Biochemistry, 5th edition* (W H Freeman, New York).

25. Marr JJ, Weber MM (1969) Feedback Inhibition of an Allosteric Triphosphopyridine Nucleotide-specific Isocitrate Dehydrogenase. *J Biol Chem* 244(20):5709–5712.

26. Chen Z, Bommareddy RR, Frank D, Rappert S, Zeng A-P (2014) Deregulation of feedback inhibition of phosphoenolpyruvate carboxylase for improved lysine production in Corynebacterium glutamicum. *Appl Environ Microbiol* 80(4):1388–1393.

27. Gerhart J (2014) From feedback inhibition to allostery: the enduring example of aspartate transcarbamoylase. *FEBS J* 281(2):612–620.

28. Tran KL, et al. (2005) Lipid sulfates and sulfonates are allosteric competitive inhibitors of the N-terminal phosphatase activity of the mammalian soluble epoxide hydrolase. *Biochemistry* 44(36):12179–87.

29. Luo L, et al. (2007) ATP-competitive inhibitors of the mitotic kinesin KSP that function via an allosteric mechanism. *Nat Chem Biol* 3(11):722–6.

30. Urbaniak MD, et al. (2013) A novel allosteric inhibitor of the uridine diphosphate N-acetylglucosamine pyrophosphorylase from Trypanosoma brucei. *ACS Chem Biol* 8(9):1981–7.

31. Perutz MF, Fermi G, Abraham DJ, Poyart C, Bursaux E (1986) Hemoglobin as a receptor of drugs and peptides: x-ray studies of the stereochemistry of binding. *J Am Chem Soc* 108(5):1064–1078.

32. Abraham DJ (1974) The potential role of single crystal X-ray diffraction in medicinal chemistry. *Int Sci Chem Rep* 8:1–9.

33. Peracchi A, Mozzarelli A (2011) Exploring and exploiting allostery: Models, evolution, and drug targeting. *Biochim Biophys Acta* 1814(8):922–33.

34. Bruno S, Ronda L, Faggiano S, Bettati S, Mozzarelli A (2003) *Burger’s Medicinal Chemistry and Drug Discovery* eds Abraham DJ, Rotella DP (John Wiley & Sons, Inc., Hoboken, NJ, USA). 7th Ed.

35. Yang J-S, Seo SW, Jang S, Jung GY, Kim S (2012) Rational engineering of enzyme allosteric regulation through sequence evolution analysis. *PLoS Comput Biol* 8(7):e1002612.

36. Kenakin TP (2012) Biased signalling and allosteric machines: new vistas and challenges for drug discovery. *Br J Pharmacol* 165(6):1659–1669.

37. May LT, Leach K, Sexton PM, Christopoulos A (2007) Allosteric modulation of G protein-coupled receptors. *Annu Rev Pharmacol Toxicol* 47:1–51.

38. Lazareno S, Dolezal V, Popham A, Birdsall NJM (2004) Thiochrome enhances acetylcholine affinity at muscarinic M4 receptors: receptor subtype selectivity via cooperativity rather than affinity. *Mol Pharmacol* 65(1):257–266.

39. Grover AK (2013) Use of allosteric targets in the discovery of safer drugs. *Med Princ Pract* 22(5):418–426.

40. Kenakin T (2007) Collateral efficacy in drug discovery: taking advantage of the good (allosteric) nature of 7TM receptors. *Trends Pharmacol Sci* 28(8):407–415.

41. Christopoulos A, Kenakin T (2002) G Protein-Coupled Receptor Allosterism and Complexing. *Pharmacol Rev* 54(2):323–374.

42. Groebe DR (2006) Screening for positive allosteric modulators of biological targets. *Drug Discov Today* 11(13–14):632–639.

43. Mohr K, Schmitz J, Schrage R, Tränkle C, Holzgrabe U (2013) Molecular alliance-from orthosteric and allosteric ligands to dualsteric/bitopic agonists at G protein coupled receptors. *Angew Chem Int Ed Engl* 52(2):508–16.

44. Kenakin T (2005) New concepts in drug discovery: collateral efficacy and permissive antagonism. *Nat Rev Drug Discov* 4(11):919–27.

45. Jakubik J, Bacakova L, El-Fakahany EE, Tucek S (1997) Positive Cooperativity of Acetylcholine and Other Agonists with Allosteric Ligands on Muscarinic Acetylcholine Receptors. *Mol Pharmacol* 52(1):172–179.

46. Wootten D, et al. (2012) Allosteric modulation of endogenous metabolites as an avenue for drug discovery. *Mol Pharmacol* 82(2):281–290.

47. Groebe DR (2009) In search of negative allosteric modulators of biological targets. *Drug Discov Today* 14(1–2):41–9.

48. Hardy J a, Wells JA (2004) Searching for new allosteric sites in enzymes. *Curr Opin Struct Biol* 14(6):706–715.

49. Wootten D, Christopoulos A, Sexton PM (2013) Emerging paradigms in GPCR allostery: implications for drug discovery. *Nat Rev Drug Discov* 12(8):630–644.

50. Sadowsky JD, et al. (2011) Turning a protein kinase on or off from a single allosteric site via disulfide trapping. *Proc Natl Acad Sci U S A* 108(15):6056–61.

51. Turlington M, et al. (2014) Tetrahydronaphthyridine and dihydronaphthyridinone ethers as positive allosteric modulators of the metabotropic glutamate receptor 5 (mGlu$_5$). *J Med Chem* 57(13):5620–37.

52. Sliwoski G, Kothiwale S, Meiler J, Lowe EW (2014) Computational methods in drug discovery. *Pharmacol Rev* 66(1):334–95.

53. Lipinski CA, Lombardo F, Dominy BW, Feeney PJ (1997) Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv Drug Deliv Rev* 23(1–3):3–25.

54. van Westen GJP, Gaulton A, Overington JP (2014) Chemical, target, and bioactive properties of allosteric modulation. *PLoS Comput Biol* 10(4):e1003559.

55. Zheng H, Hou J, Zimmerman MD, Wlodawer A, Minor W (2014) The future of crystallography in drug discovery. *Expert Opin Drug Discov* 9(2):125–37.

56. Davis AM, St-Gallay SA, Kleywegt GJ (2008) Limitations and lessons in the use of X-ray structural information in drug design. *Drug Discov Today* 13(19):831–841.

57. Durrant JD, McCammon JA (2011) Molecular dynamics simulations and drug discovery. *BMC Biol* 9:71.

58. Bourne Y, Talley TT, Hansen SB, Taylor P, Marchot P (2005) Crystal structure of a Cbtx-AChBP complex reveals essential interactions between snake alpha-neurotoxins and nicotinic receptors. *EMBO J* 24(8):1512–1522.

59. Chiappori F, Merelli I, Colombo G, Milanesi L, Morra G (2012) Molecular mechanism of allosteric communication in Hsp70 revealed by molecular dynamics simulations. *PLoS Comput Biol* 8(12):e1002844.

60. Sun X, Agren H, Tu Y (2014) Microsecond Molecular Dynamics Simulations Provide Insight into the Allosteric Mechanism of the Gs Protein Uncoupling from the β2 Adrenergic Receptor. *J Phys Chem B* 118(51):14737–14744.

61. Gkeka P, Papafotika A, Christoforidis S, Cournia Z (2015) Exploring a non-ATP pocket for potential allosteric modulation of PI3Kα. *J Phys Chem B* 119(3):1002–16.

62. Marino KA, Sutto L, Gervasio FL (2015) The effect of a widespread cancer-causing mutation on the inactive to active dynamics of the B-Raf kinase. *J Am Chem Soc* 137(16):5280–3.

63. Goodman JM (1998) *Chemical applications of molecular modelling* (Royal Society of Chemistry, Cambridge).

64. Leach AR (1996) *Molecular Modelling Principles and Applications* (Ashley Wesley Longman Limited, Harlow).

65. Šponer J, et al. (2013) How to understand quantum chemical computations on DNA and RNA systems? A practical guide for non-specialists. *Methods* 64(1):3–11.

66. Brooks BR, et al. (1983) CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J Comput Chem* 4(2):187–217.

67. Scott WRP, et al. (1999) The GROMOS Biomolecular Simulation Program Package. *J Phys Chem A* 103(19):3596–3607.

68. Cornell WD, et al. (1995) A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J Am Chem Soc* 117(19):5179–5197.

69. Pronk S, et al. (2013) GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* 29(7):845–854.

70. Lindorff-Larsen K, et al. (2010) Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins* 78(8):1950–1958.

71. Best RB, et al. (2012) Optimization of the Additive CHARMM All-Atom Protein Force Field Targeting Improved Sampling of the Backbone φ, ψ and Side-Chain χ_1 and χ_2 Dihedral Angles. *J Chem Theory Comput* 8(9):3257–3273.

72. Case DA, et al. (2012) AMBER 12.

73. Vanommeslaeghe K, Guvench O, MacKerell AD (2014) Molecular mechanics. *Curr Pharm Des* 20(20):3281–92.

74. Chipot C, Pearlman DA (2010) Free Energy Calculations. The Long and Winding Gilded Road. *Mol Simul* 28(1):1–12.

75. van der Kamp MW, Mulholland AJ (2013) Combined Quantum Mechanics/Molecular Mechanics (QM/MM) Methods in Computational Enzymology. *Biochemistry* 52(16):2708–2728.

76. Senn HM, Thiel W (2009) QM/MM methods for biomolecular systems. *Angew Chem Int Ed Engl* 48(7):1198–229.

77. Lin H, Truhlar DG (2006) QM/MM: what have we learned, where are we, and where do we go from here? *Theor Chem Acc* 117(2):185–199.

78. Kleinjung J, Fraternali F (2014) Design and application of implicit solvent models in biomolecular simulations. *Curr Opin Struct Biol* 25:126–34.

79. Fogolari F, Brigo A, Molinari H (2002) The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology. *J Mol Recognit* 15(6):377–392.

80. Darden T, York D, Pedersen L (1993) Particle mesh Ewald: An N· log(N) method for Ewald sums in large systems. *J Chem Phys* 98(12):10089.

81. Cheatham TEI, Miller JL, Fox T, Darden TA, Kollman PA (1995) Molecular Dynamics Simulations on Solvated Biomolecular Systems: The Particle Mesh Ewald Method Leads to Stable Trajectories of DNA, RNA, and Proteins. *J Am Chem Soc* 117(14):4193–4194.

82. Arnold GE, Ornstein RL (1994) An evaluation of implicit and explicit solvent model systems for the molecular dynamics simulation of bacteriophage T4 lysozyme. *Proteins* 18(1):19–33.

83. Adcock SA, McCammon JA (2006) Molecular dynamics: survey of methods for simulating the activity of proteins. *Chem Rev* 106(5):1589–615.

84. Izaguirre JA, Catarello DP, Wozniak JM, Skeel RD (2001) Langevin stabilization of molecular dynamics. *J Chem Phys* 114(5):2090.

85. Berendsen HJC, Postma JPM, van Gunsteren WF, DiNola A, Haak JR (1984) Molecular dynamics with coupling to an external bath. *J Chem Phys* 81(8):3684.

86. Cerutti DS, Duke R, Freddolino PL, Fan H, Lybrand TP (2008) A Vulnerability in Popular Molecular Dynamics Packages Concerning Langevin and Andersen Dynamics. 1669–1680.

87. Fuxreiter M (2014) *Computational Approaches to Protein Dynamics: From Quantum to Coarse-Grained Methods* ed Fuxreiter M (CRC Press, Boca Raton, FL).

88. Ryckaert J-P, Ciccotti G, Berendsen HJ. (1977) Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J Comput Phys* 23(3):327–341.

89. Hopkins CW, Le Grand S, Walker RC, Roitberg AE (2015) Long-Time-Step Molecular Dynamics through Hydrogen Mass Repartitioning. *J Chem Theory Comput* 11(4):1864–74.

90. Liu W, Schmidt B, Voss G, Müller-Wittig W (2008) Accelerating molecular dynamics simulations using Graphics Processing Units with CUDA. *Comput Phys Commun* 179(9):634–641.

91. Friedrichs MS, et al. (2009) Accelerating molecular dynamic simulation on graphics processing units. *J Comput Chem* 30(6):864–72.

92. Götz AW, et al. (2012) Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born. *J Chem Theory Comput* 8(5):1542–1555.

93. Cunningham SJ (1995) Machine Learning and Statistics: A matter of perspective. 1–8.

94. Srivastava T (2015) Difference Between Machine Learning And Statistical Modeling. Available at: https://www.analyticsvidhya.com/blog/2015/07/difference-machine-learning-statistical-modeling/ [Accessed January 22, 2017].

95. Luts J, et al. (2010) A tutorial on support vector machine-based methods for classification problems in chemometrics. *Anal Chim Acta* 665(2):129–145.

96. Krogh A (2008) What are artificial neural networks? *Nat Biotechnol* 26(2):195–197.

97. Sivanandam SN, Deepa SN (2008) *Introduction to Genetic Algorithms* (Springer-Verlag Berlin Heidelberg, New York).

98. Ross TJ (2010) *Fuzzy Logic with Engineering Applications* (John Wiley and Sons Ltd., Chichester). 3rd Ed.

99. Sun T, et al. (2013) Comparative evaluation of support vector machines for computer aided diagnosis of lung cancer in CT based on a multi-dimensional data set. *Comput Methods Programs Biomed* 111(2):519–524.

100. Rodriguez-Galiano V, Sanchez-Castillo M, Chica-Olmo M, Chica-Rivas M (2015) Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geol Rev* 71:804–818.

101. Palmer DS, O'Boyle NM, Glen RC, Mitchell JBO (2006) Random Forest Models To Predict Aqueous Solubility. *J Chem Inf Model* 47:150–158.

102. Sesnie SE, et al. (2010) The multispectral separability of Costa Rican rainforest types with support vector machines and Random Forest decision trees. *Int J Remote Sens* 31(11):2885–2909.

103. Hsieh C-H, et al. (2011) Novel solutions for an old disease: Diagnosis of acute appendicitis with random forest, support vector machines, and artificial neural networks. *Surgery* 149(1):87–93.

104. Liu M, Wang M, Wang J, Li D (2013) Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and Chinese vinegar. *Sensors Actuators B Chem* 177:970–980.

105. Statnikov A, Wang L, Aliferis CF (2008) A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics* 9:319.

106. Johnston A, Johnston BF, Kennedy AR, Florence AJ (2008) Targeted crystallisation of novel carbamazepine solvates based on a retrospective Random Forest classification. *CrystEngComm* 10(1):23–25.

107. Zheng L, et al. (2009) A chemometric study of chromatograms of tea extracts by correlation optimization warping in conjunction with PCA, support vector machines and random forest data modeling. *Anal Chim Acta* 642(1):257–265.

108. Bhardwaj RM, Johnston A, Johnston BF, Florence AJ (2015) A random forest model for predicting the crystallisability of organic molecules. *CrystEngComm* 17(23):4272–4275.

109. Breiman L (2001) Random Forests. *J Mach Learn Res* 45:5–32.

110. Fisher RA (1936) The Use of Multiple Measurements in Taxonomic Problems. *Ann Eugen* 7(2):179–188.

111. R Core Team. R: A Language and Environment for Statistical Computing (2013).

112. Ceriani L, Verme P (2011) The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini. *J Econ Inequal* 10(3):421–443.

113. Díaz-Uriarte R, Alvarez de Andrés S (2006) Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 7(1):3.

114. Strobl C, Boulesteix A-L, Kneib T, Augustin T, Zeileis A (2008) Conditional variable importance for random forests. *BMC Bioinformatics* 9(1):307.

115. Genuer R, Poggi J-M, Tuleau C (2008) Random Forests : some methodological insights.

116. Blagus R, Lusa L (2010) Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics* 11(1):523.

117. Chen C, Liaw A, Breiman L (1999) Using Random Forest to Learn Imbalanced Data. 1–12.

118. Do T, Lenca P, Lallich S, Pham N (2010) Classifying Very-High-Dimensional Data with. 39–55.

119. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Inf Process Manag* 45(4):427–437.

120. Powers DMW (2011) Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *J Mach Learn Technol* 2(1):37–63.

121. Ben-David A (2007) A lot of randomness is hiding in accuracy. *Eng Appl Artif Intell* 20(7):875–885.

122. Kubat M, Matwin S (1997) Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. *Proceedings of the Fourteenth International Conference on Machine Learning*, pp 179–186.

123. Cohen J (1960) A Coefficient of Agreement for Nominal Scales. *Educ Psychol Meas* 20:37–46.

124. Nelson KP, Edwards D (2008) On population-based measures of agreement for binary classifications. *Can J Stat* 36(3):411–426.

125. Vach W (2005) The dependence of Cohen's kappa on the prevalence does not matter. *J Clin Epidemiol* 58(7):655–61.

126. Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta - Protein Struct* 405(2):442–451.

127. Lee M-L, Schneider G (2001) Scaffold Architecture and Pharmacophoric Properties of Natural Products and Trade Drugs: Application in the Design of Natural Product-Based Combinatorial Libraries. *J Comb Chem* 3(3):284–289.

128. Jones DT, Ward JJ (2003) Prediction of disordered regions in proteins from position specific score matrices. *Proteins* 53 Suppl 6:573–578.

129. Petersen TN, Brunak S, von Heijne G, Nielsen H (2011) SignalP 4.0: discriminating signal peptides from transmembrane regions. *Nat Methods* 8(10):785–6.

130. Triballeau N, Acher F, Brabet I, Pin J-P, Bertrand H-O (2005) Virtual screening workflow development guided by the "receiver operating characteristic" curve

approach. Application to high-throughput docking on metabotropic glutamate receptor subtype 4. *J Med Chem* 48(7):2534–47.

131. Li X, et al. (2013) Toward an understanding of the sequence and structural basis of allosteric proteins. *J Mol Graph Model* 40:30–39.

132. Fernandes JA, Irigoien X, Boyra G, Lozano JA, Inza I (2008) Optimizing the number of classes in automated zooplankton classification. *J Plankton Res* 31(1):19–29.

133. Lichtarge O, Bourne HR, Cohen FE (1996) An Evolutionary Trace Method Defines Binding Surfaces Common to Protein Families. *J Mol Biol* 257(2):342–358.

134. Johansson F, Toh H (2010) Relative Von Neumann Entropy for Evaluating Amino Acid Conservation. *J Bioinform Comput Biol* 8(5):809–823.

135. Yang L, Song G, Jernigan RL (2009) Protein elastic network models and the ranges of cooperativity. *Proc Natl Acad Sci* 106(30):12347–12352.

136. Bahar I, Rader A (2005) Coarse-grained normal mode analysis in structural biology. *Curr Opin Struct Biol* 15(5):586–592.

137. Wagner JR, et al. (2016) Emerging Computational Methods for the Rational Discovery of Allosteric Drugs. *Chem Rev* 116(11):6370–6390.

138. Panjkovich A, Daura X (2012) Exploiting protein flexibility to predict the location of allosteric sites. *BMC Bioinformatics* 13(1):273.

139. Panjkovich A, Daura X (2014) PARS: a web server for the prediction of Protein Allosteric and Regulatory Sites. *Bioinformatics* 30(9):1314–5.

140. Huang B, Schroeder M (2006) LIGSITEcsc: predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct Biol* 6(1):19.

141. Panjkovich A, Daura X (2010) Assessing the structural conservation of protein pockets to study functional and allosteric sites: implications for drug discovery. *BMC Struct Biol* 10(1):9.

142. Mitternacht S, Berezovsky IN (2011) Coherent conformational degrees of freedom as a structural basis for allosteric communication. *PLoS Comput Biol* 7(12):e1002301.

143. Mitternacht S, Berezovsky IN (2011) Binding leverage as a molecular basis for allosteric regulation. *PLoS Comput Biol* 7(9):e1002148.

144. Goncearenco A, et al. (2013) SPACER: Server for predicting allosteric communication and effects of regulation. *Nucleic Acids Res* 41(Web Server issue):266–272.

145. Mitternacht S, Berezovsky IN (2011) A geometry-based generic predictor for catalytic and allosteric sites. *Protein Eng Des Sel* 24(4):405–9.

146. Zheng W, Brooks BR, Thirumalai D (2006) Low-frequency normal modes that describe allosteric transitions in biological nanomachines are robust to sequence variations. *Proc Natl Acad Sci* 103(20):7664–7669.

147. Townsend PD, et al. (2015) Global low-frequency motions in protein allostery: CAP as a model system. *Biophys Rev* 7(2):175–182.

148. Ma J (2005) Usefulness and Limitations of Normal Mode Analysis in Modeling Dynamics of Biomolecular Complexes. *Structure* 13(3):373–380.

149. Huang W, et al. (2013) Allosite: a method for predicting allosteric sites. *Bioinformatics* 29(18):2357–2359.

150. Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: an open source platform for ligand pocket detection. *BMC Bioinformatics* 10(1):168.

151. Warmuth MK, et al. (2003) Active Learning with Support Vector Machines in the Drug Discovery Process. *J Chem Inf Model* 43(2):667–673.

152. Huang Z, et al. (2011) ASD: a comprehensive database of allosteric proteins and modulators. *Nucleic Acids Res* 39(Database issue):D663-9.

153. Andreassi JL, Bilder PW, Vetting MW, Roderick SL, Leyh TS (2007) Crystal structure of the Streptococcus pneumoniae mevalonate kinase in complex with diphosphomevalonate. *Protein Sci* 16(5):983–989.

154. Hardie DG, Ross FA, Hawley SA (2012) AMPK: a nutrient and energy sensor that maintains energy homeostasis. *Nat Rev Mol Cell Biol* 13(4):251–262.

155. Xiao B, et al. (2007) Structural basis for AMP binding to mammalian AMP-activated protein kinase. *Nature* 449(7161):496–500.

156. Ceccarelli DF, et al. (2011) An allosteric inhibitor of the human Cdc34 ubiquitin-conjugating enzyme. *Cell* 145(7):1075–1087.

157. Huang H, et al. (2014) E2 enzyme inhibition by stabilization of a low-affinity interface with ubiquitin. *Nat Chem Biol* 10(2):156–163.

158. Kissinger CR, et al. (2004) Crystal structure of human ABAD/HSD10 with a bound inhibitor: implications for design of Alzheimer's disease therapeutics. *J Mol Biol* 342(3):943–952.

159. Francotte P, et al. (2014) Positive allosteric modulators of 2-amino-3-(3-hydroxy-5-methylisoxazol-4-yl)propionic acid receptors belonging to 4-cyclopropyl-3,4-dihydro-2h-1,2,4-pyridothiadiazine dioxides and diversely chloro-substituted 4-cyclopropyl-3,4-dihydro-2H-1,2,4-benzothiad. *J Med Chem* 57(22):9539–53.

160. Rath VL, et al. (2000) Activation of Human Liver Glycogen Phosphorylase by Alteration of the Secondary Structure and Packing of the Catalytic Core. *Mol Cell* 6(1):139–148.

161. Larkin MA, et al. (2007) Clustal W and Clustal X version 2.0. *Bioinformatics* 23(21):2947–2948.

162. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML (1983) Comparison of simple potential functions for simulating liquid water. *J Chem Phys* 79(2):926.

163. Lodish H, et al. (2000) *Intracellular Ion Environment and Membrane Electric Potential* (W. H. Freeman, New York). 4th Ed.

164. Ewald PP (1921) Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann Phys* 369(3):253–287.

165. Kyte J, Doolittle RF (1982) A simple method for displaying the hydropathic character of a protein. *J Mol Biol* 157(1):105–132.

166. Hubbard S, Thornton J (1993) "NACCESS" Computer Program.

167. Doyle PG, Snell JL (1984) Random Walks and Electric Networks. *Math Assoc Am*.

168. Chennubhotla C, Bahar I (2007) Signal propagation in proteins and relation to equilibrium fluctuations. *PLoS Comput Biol* 3(9):1716–1726.

169. Erman B (2011) Relationships between ligand binding sites, protein architecture and

correlated paths of energy and conformational fluctuations. *Phys Biol* 8(5):56003.

170. Pritchard L, Cardle L, Quinn S, Dufton M (2003) Simple intrasequence difference (SID) analysis: an original method to highlight and rank sub-structural interfaces in protein folds. Application to the folds of bovine pancreatic trypsin inhibitor, phospholipase A2, chymotrypsin and carboxypeptidase A. *Protein Eng Des Sel* 16(2):87–101.

171. Gilson MK, Honig B (1988) Calculation of the total electrostatic energy of a macromolecular system: Solvation energies, binding energies, and conformational analysis. *Proteins Struct Funct Genet* 4(1):7–18.

172. Rocchia W, et al. (2002) Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: Applications to the molecular systems and geometric objects. *J Comput Chem* 23(1):128–137.

173. Liaw A, Wiener M (2003) Classification and Regression by randomForest. *R News 2* 3:18–22.

174. Hill J, et al. (2008) SPRINT: a new parallel framework for R. *BMC Bioinformatics* 9(1):558.

175. Ramachandran GN, Ramakrishnan C, Sasisekharan V (1963) Stereochemistry of Polypeptide Chain Configurations. *J Mol Biol* 7:95–99.

176. Hollingsworth SA, Karplus PA (2011) standard structures in proteins. 1(Figure 1):271–283.

177. Gunasekaran K, Ramakrishnan C, Balaram P (1996) Disallowed Ramachandran conformations of amino acid residues in protein structures. *J Mol Biol* 264(1):191–8.

178. Horn JR, Shoichet BK (2004) Allosteric inhibition through core disruption. *J Mol Biol* 336(5):1283–1291.

179. Minasov G, Wang X, Shoichet BK (2002) An Ultrahigh Resolution Structure of TEM-1 β-Lactamase Suggests a Role for Glu166 as the General Base in Acylation. *J Am Chem Soc* 124(19):5333–5340.

180. Berkholz DS, Krenesky PB, Davidson JR, Karplus PA (2010) Protein Geometry Database: a flexible engine to explore backbone conformations and their relationships to covalent geometry. *Nucleic Acids Res* 38(Database issue):D320-5.

181. Chenna R, et al. (2003) Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Res* 31(13):3497–3500.

182. Leinonen R, et al. (2004) UniProt archive. *Bioinformatics* 20(17):3236–3237.

183. Tropsha A, Gramatica P, Gombar V (2003) The Importance of Being Earnest: Validation is the Absolute Essential for Successful Application and Interpretation of QSPR Models. *QSAR Comb Sci* 22(1):69–77.

184. Rücker C, Rücker G, Meringer M (2007) y-Randomization and its variants in QSPR/QSAR. *J Chem Inf Model* 47(6):2345–57.

185. Hilser VJ, Freire E (1996) Structure-based Calculation of the Equilibrium Folding Pathway of Proteins. Correlation with Hydrogen Exchange Protection Factors. *J Mol Biol* 262(5):756–772.

186. Vertrees J, Barritt P, Whitten S, Hilser VJ (2005) COREX/BEST server: a web browser-based program that calculates regional stability variations within protein structures.

Bioinformatics 21(15):3318–3319.

187. Le Grand S, Walker RC AMBER 14 NVIDIA GPU Acceleration Support. *http://ambermd.org/gpus14/*. Available at: http://ambermd.org/gpus14/ [Accessed September 22, 2016].

188. Pierce LCT, Salomon-Ferrer R, De Oliveira CAF, McCammon JA, Walker RC (2012) Routine Access to Millisecond Time Scale Events with Accelerated Molecular Dynamics. *J Chem Theory Comput* 8(9):2997–3002.

189. Ostermeir K, Zacharias M (2013) Advanced replica-exchange sampling to study the flexibility and plasticity of peptides and proteins. *Biochim Biophys Acta - Proteins Proteomics* 1834(5):847–853.

190. Chodera JD, Noé F (2014) Markov state models of biomolecular conformational dynamics. *Curr Opin Struct Biol* 25:135–144.