Predicting Drug Target Proteins and Their Properties

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy (PhD) in the Faculty of Life Sciences

2014

Simon Bull

Table of Contents

Word Count: 52,348

List of Tables

List of Figures

Abstract

Predicting Drug Target Proteins and Their Properties
A thesis submitted to The University of Manchester for the degree of Doctor of Philosophy (PhD) in the Faculty of Life Sciences by Simon Bull, 2014.

The discovery of drug targets is a vital component in the development of therapeutic treatments, as it is only through the modulation of a target's activity that a drug can alleviate symptoms or cure. Accurate identification of drug targets is therefore an important part of any development program, and has an outsized impact on the program's success due to its position as the first step in the pipeline. This makes the stringent selection of potential targets all the more vital when attempting to control the increasing cost and time needed to successfully complete a development program, and in order to increase the throughput of the entire drug discovery pipeline.

In this work, a computational approach was taken to the investigation of protein drug targets. First, a new heuristic, Leaf, for the approximation of a maximum independent set was developed, and evaluated in terms of its ability to remove redundancy from protein datasets, the goal being to generate the largest possible non-redundant dataset. The ability of Leaf to remove redundancy was compared to that of pre-existing heuristics and an optimal algorithm, Cliquer. Not only did Leaf find unbiased non-redundant sets that were around 10% larger than the commonly used PISCES algorithm, it found ones that were no more than one protein smaller than the maximum possible found by Cliquer.

Following this, the human proteome was mined to discover properties of proteins that may be important in determining their suitability for pharmaceutical modulation. Data was gathered concerning each protein's sequence, post-translational modifications, secondary structure, germline variants, expression profile and target status. The data was then analysed to determine features for which the target and non-target proteins had significantly different values. This analysis was repeated for subsets of the proteome consisting of all GPCRs, ion channels, kinases and proteases, as well as for a subset consisting of all proteins that are implicated in cancer. Next, machine learning was used to quantify the proteins in each dataset in terms of their potential to serve as a drug target. For each dataset, this was accomplished by first inducing a random forest that could distinguish between its targets and non-targets, and then using the random forest to quantify the drug target likeness of the non-targets.

The properties that can best differentiate targets from non-targets were primarily found to be those that are directly related to a protein's sequence (e.g. secondary structure). Germline variants, expression levels and interactions between proteins had minimal discriminative power. Overall, the best indicators of drug target likeness were found to be the proteins' hydrophobicities, *in vivo* half-lives, propensity for being membrane bound and the fraction of non-polar amino acids in their sequences. In terms of predicting potential targets, datasets of proteases, ion channels and cancer proteins were able to induce random forests that were highly capable of distinguishing between targets and non-targets. The non-target proteins predicted to be targets by these random forests comprise the set of the most suitable potential future drug targets, and are therefore likely to produce the best results if used as the basis for building a drug development programme.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

List of Abbreviations

ADMET	absorption, distribution, metabolism, excretion and toxicity
BE	backwards elimination
CHC-GA	CHC genetic algorithm
DT	decision tree
FDA	Food and Drug Administration
FS	forward selection
GA	genetic algorithm
GPCR	G-protein-coupled receptor
Iid	independent and identically distributed
k-NN	k nearest neighbours
MIS	maximum independent set
NME	new molecular entity
OOB	out-of-bag
PPI	protein protein interaction
PS	Probability of superiority
PTCH1	protein patched homolog 1
RF	random forest
SVM	support vector machine
TTD	Therapeutic Target Database

Acknowledgements

I would like to thank my supervisor Andrew for helping to guide my work without prescribing its direction, and for giving me the freedom to work in the way that allowed me to be most productive. I would also like to thank my advisor Casey for his insightful questions and general advice. I always came out of our meetings with multiple new directions to consider. The Leaf project would not have been such a success without the contribution of Mark Muldoon. His insight into the problem and contribution to the running of the experiments was vital in obtaining the conclusive results that we did. The contributions of Penny Doig and Mark Kambites in designing the final version of the NeighbourCull algorithm were also greatly appreciated. Finally, I would like to thank Kieaibi and Mitra for their interdisciplinary sympathies.

In addition to the individuals who smoothed the progress of my PhD, I couldn't have completed the work without the funding provided to me by the BBSRC (UK). The work on the Leaf algorithm was made possible by running our Cliquer computations on the High Throughput Computing facility of the Faculty of Engineering and Physical Sciences at The University of Manchester. Similarly, the remainder of the work was made possible through the use of the Computational Shared Facility at The University of Manchester.

1 Introduction

Throughout history humanity has sought ways to alleviate the symptoms of the illnesses that afflict it. Over time, and through a process of trial and error, the medicinal and injurious properties of naturally occurring substances were discovered and knowledge of them passed down the generations. Eventually, the unstable and inconsistent nature of this serendipitous approach to medicine gave way to more rational and evidence based approaches. Today, through increased medical understanding, it is possible to develop synthetic cures to treat the specific cause of an illness.

1.1 Drugs and Their Targets

A *drug* may be considered to be a natural or synthetic substance, which is not food, and is intended to alleviate the symptoms of or cure a disease. In many countries these substances are regulated by a government agency in order to prevent the proliferation of unsafe medicines. In the United States, the Food and Drug Administration (FDA) controls the approval process for drugs, and defines them to be products that are "intended for use in the diagnosis, cure, mitigation, treatment, or prevention of disease" and "articles (other than food) intended to affect the structure or any function of the body of man or other animals" (FD&C Act, sec. 201(g)(1)). An *approved drug* can be considered to be a drug that has been through the approval process of an agency such as the FDA, and therefore been certified for sale by that agency. Approved drugs can fall into one of two broad categories: biologics, such as therapeutic proteins and monoclonal antibodies, and small molecules, which are low molecular weight organic compounds. Subsequent references to drugs will refer to small molecules meeting the FDA's definition of a drug, unless otherwise stated.

The activity of drugs can either be structurally specific or non-specific. Non-specific action, such as that found in general anaesthetics, osmotic diuretics and antacids, results from the accumulation of the drug in some part of the body, whereas specific action is dependent on the interaction of the drug with one or more molecular *targets*. These targets will have important regulatory functions within a desired organ or tissue, and are the sites within the body where the drug exerts its effect. In general, this effect is generated via the binding of the drug to a specific *binding site* on a target, which in turn influences the behaviour of the tissue within which the target resides (Negus 2006). However, it is possible for a drug to undergo unintentional interactions with other molecules, and thereby potentially cause undesirable side effects. The propensity of a drug to participate in these off-target interactions depends on the level of

selectivity that the drug has for its targets, with a more selective drug being less likely to take part in off-target interactions.

In order to attempt to understand the targets of approved drugs, it is first necessary to define what a drug target is. A good starting point is to consider the role of the target in the body under normal and diseased conditions, and the roles that would make a molecule desirable as a drug target. Firstly, a target should play a critical, and preferably unsubstitutable, role that is specific to a given disease process (Chen & Du 2007; Mandal et al. 2009; Zheng et al. 2006). Secondly, the target would preferably not be significantly involved in other processes in the body, so as to minimise the chance of unwanted side effects resulting from its modulation (Gashaw et al. 2011; Zheng et al. 2006). Finally, expression of the target should not be uniform throughout the body (Gashaw et al. 2011). Ideally the expression would be specific to the desired tissue, but could instead be constrained in the non-target tissues (Zheng et al. 2006).

While a drug target can belong to any one of a number of categories, such as macromolecules, biological pathways and key nodes in regulatory networks (Yang et al. 2009), it is usually a single macromolecule that is believed to be involved in the aetiology of a disease (Lang et al. 2006). However, due to the difficulty of developing potent, non-toxic, selective and effective small molecules that target non-protein macromolecules, the vast majority of the targets of approved drugs are proteins (Hopkins & Groom 2002; Xu et al. 2007). Knowledge of the proteins that are the targets of approved drugs enables the division of the human *proteome,* the set of all human proteins, into two classes: *approved drug targets* and *non-targets*. A protein is an approved drug target if it is the target of an approved drug, and a non-target otherwise. Any further references to drug targets will be referring solely to approved protein drug targets, unless otherwise specified.

The suitability of a protein to act as the target of a drug is dependent on more than just the role it plays within the body. In order for a protein to have any potential as a drug target it must be *druggable*. A druggable protein is one that possesses folds that favour interactions with small drug-like molecules, be they endogenous or exogenous, and therefore is one that contains a binding site (Chen & Du 2007; Hopkins & Groom 2002). These binding sites are expected to have certain characteristics that enable high affinity site-specific binding by the drug-like molecule. It is believed that “these characteristics probably define the sequence features, structural architectures, genomic signatures, and proteomic profiles of therapeutic targets and their roles at the pathway, cellular, and physiological levels” (Zheng et al. 2006). Although non-druggable proteins may have desirable properties in terms of their involvement in a disease process, the absence of a binding site means it is unlikely that the protein’s activity is capable of being modulated pharmaceutically (Russ & Lampel 2005). As with all drug targets, a potential protein

drug target must be linked to a disease process. In this case the activity of the protein must be involved in generating the undesired disease state, and modification of it must either cure the disease or provide relief from its symptoms (Chandra 2009; Mandal et al. 2009). A potential protein drug target should also be minimally similar to any of the proteomes of the gut flora microbes, in order to minimise the potential for undesirable side effects (Chandra 2009). It is important to note that a potential protein drug target must be both druggable and involved in a disease process. Neither condition is sufficient on its own.

Currently there is a lack of knowledge about both the number of proteins that modern pharmaceuticals act on and the number of potentially druggable proteins. Drews (1996) proposed one of the first counts of the number of human protein targets, and determined that there were only 417 protein drug targets (excluding anti-infectives acting on bacteria, viruses or parasites). More recent estimates for the number of protein drug targets have included 218 (Imming et al. 2006); a consensus number of 324 (Overington et al. 2006); 399, reduced to 120 when only approved drug targets are considered, (Hopkins & Groom 2002) and 435 (Rask-Andersen et al. 2011). In terms of potential drug targets, an analysis by Russ and Lampel (Russ & Lampel 2005) identified between 2000 and 3000 proteins that are druggable. Using a purely bioinformatics approach, Bakheet and Doig (2009) were able to identify 668 proteins that are not currently approved drug targets, but that have target-like properties. These latter estimates lend credence to the belief that, although the estimate of the number of currently targeted proteins is in the hundreds, the number of proteins that are druggable is substantially larger (Imming et al. 2006).

While knowledge of the number of proteins that may be amenable to pharmaceutical modulation is valuable, it is also useful to consider the families to which these proteins belong. Rask-Anderson *et al.* (2011) found that G-protein-coupled receptors (GPCRs) make up 44% of human drug targets, enzymes 29% and transporter proteins 15%; Overington *et al.* (2006) found that over 50% of drugs target GPCRs, nuclear receptors or ion channels; Hopkins and Groom (2002) found that enzymes comprise 47% of launched targets, while GPCRs account for 30%; and Zheng *et al*. (2006) found that enzymes make up 50% of approved targets. One very evident trend in these findings is the prominence of enzymes and GPCRs in the set of approved drug target proteins. Using the estimate of Fredriksson *et al*. (2003) that there are approximately 800 GPCRs coded for by the human genome, and the knowledge that there are just over 20,000 human proteins (The UniProt Consortium 2010), we can estimate that roughly 4% of human proteins are GPCRs. The fraction of GPCRs in the set of approved drug targets can therefore be seen to be vastly greater than would be expected if the set's composition was proportional to that of all human proteins. Potential reasons for this discrepancy include: the frequency with which proteins from specific families, such as GPCRs and ion channels, can be found to be involved in human

diseases, the nature of the diseases that affect developed countries and the potential difficulty of identifying and exploiting other families of proteins. The nature of the diseases that affect developed countries is of particular importance, as the majority of the pharmaceutical industry's customers have historically originated from them. This has led to neglected tropical diseases, such as trachoma and helminth infections, receiving minimal funding and research interest from the industry, despite their ubiquity and deadliness in developing countries. Although the importance of developing country markets to the pharmaceutical industry is increasing, the sustained capacity and willingness of the customers, insurance companies and governments of developed countries to pay exorbitant fees for medicines provides little incentive or financial benefit to the pharmaceutical firm that attempts to tackle diseases afflicting developing countries.

1.2 The Drug Development Process

Typically, drug development can be divided into three general stages: discovery, preclinical testing and clinical trials (Steinmetz & Spack 2009). The discovery stage consists of basic research to select suitable targets and drug-like compounds. Target selection involves identifying targets that can have their activity modulated by drugs in order to beneficially influence disease progression (Lindsay 2003; Yang et al. 2009), and then validating each identified target in order to determine whether a drug acting on it would work in the desired human disease state (Lindsay 2003; Szymkowski & Avenue 2003). Once a target has been selected, hit identification is performed in order to identify a hit, a potential drug-like compound that can modulate the target. The hit-to-lead step is used to refine these hits into higher quality lead compounds, which are then optimised during the lead optimisation step in order to improve favourable properties, such as effectiveness, potency and selectivity, and to minimise undesirable ones, such as toxicity (Hughes et al. 2011).

Following the discovery stage, preclinical testing is carried out. This involves *in vitro* and *in vivo*, but not in human, testing to determine whether a drug is likely to produce biological activity against the disease it is intended to treat (Peck et al. 1992), the optimal formulation and synthesis of the drug and what effect the drug will have on biological processes other than the one(s) it is intended to modulate (Rawlins 2004). This will typically involve absorption, distribution, metabolism, excretion and toxicity (ADMET) profiling. This enables investigators to evaluate the interactions of a drug with the body, and the changes in the drug's concentration in the body over time. This is done by monitoring the movement of the drug into the bloodstream from the site of administration (absorption), the intravascular to extravascular movement of the drug (distribution), the chemical conversion of the drug into metabolites (metabolism), the removal of the drug from the body (excretion) and the potential toxicity of the drug.

Results from preclinical trials are used to determine whether human clinical trials can go forward. Clinical trials are performed in order to obtain data such as: the safety of the drug in humans (and any adverse reactions it may cause), the dosing requirements of the drug, efficacy of the drug and ultimately whether the drug is more effective than either a placebo or a drug currently marketed for the same purpose as the one undergoing the trials. They are usually divided into phase I, II and III trials, with potentially an additional phase IV (or post-approval) trial being performed once the drug has been approved for sale. Phase I trials are carried out using healthy volunteers, and are primarily used to show that the drug is safe to use in humans. One exception to this general rule is antineoplastic drugs, which often undergo phase I trials in patients for whom existing therapies have failed, and are therefore often constructed to determine the efficacy of the drug as well. Phase II trials are typically used to determine the efficacy of the drug, and to determine the dosage required to observe the drug's therapeutic benefits in patients (Peck et al. 1992; Rawlins 2004). Phase III trials are used to confirm the therapeutic effect of the drug, at the dose(s) proposed for marketing, in a wider population than phase II trials (Peck et al. 2003; Rawlins 2004), and to compare the effect of the drug to a standard treatment, if one exists, and possibly a placebo (Rawlins 2004). Phase IV trials are used to determine long term safety of taking a drug, and may be used to determine other conditions for which the drug is suited (Lipsky & Sharp 2001). They also enable safety and effectiveness data to be collected from a broader range of patients than were used for phase III trials. For example, children, the elderly and pregnant women are highly unlikely to be included in phase III trials, and therefore safety and effectiveness data for them can only be collected once the drug is available to the general public (Lipsky & Sharp 2001; Suvarna 2010).

1.2.1 Difficulties Facing the Pharmaceutical Industry

The development of a drug is an expensive pursuit. In 2003, DiMasi *et al.* (2003) estimated the cost of developing a new molecular entity (NME) to be US $802 million, rising to US $900 million when post-approval studies are taken into account. More recent estimates of this cost have placed it at approximately US $1.2 billion (Adams & Brantner 2010) and US $1.8 billion (Paul et al. 2010). Munos (2009) estimated average NME development costs for individual companies, rather than using an industry-wide average, and found that in 2008 approximately 73% of companies had a cost per NME of US $1 billion or more. In addition to this high cost, the process, from discovery through approval, was estimated to take approximately 13.5 years in 2007 (Paul et al. 2010).

Recently, several studies have looked at the change in development costs over time, and have provided evidence to support the belief that drug development costs are increasing

unsustainably (Dickson & Gagnon 2004; Kola 2008; Schmid & Smith 2005). DiMasi *et al.* (2003) determined that the cost of developing an NME was increasing at 7.4% above the rate of general inflation during the 1990s, while Munos (2009) calculated that NME development costs have been increasing by 13.4% annually since the 1950s. This increasing cost of development has been hypothesised to dissuade pharmaceutical companies from developing drugs with expected peak sales of less than US $500 million (Rawlins 2004), and therefore acts to restrict the development of potentially useful NMEs.Currently new drug approvals by the FDA tend to be for derivatives or new indications of previously approved drugs, rather than for NMEs. Although this trend may make rising NME development costs seem less problematic, it actually serves to magnify the impact of the increase, as the approval of a derivative or new indication relies on the original discovery and approval of an NME. If rising development costs cause fewer NMEs to be approved, then pharmaceutical companies forego not only the sales of the undeveloped NMEs, but also those of the new indications and derivatives that could have potentially followed on from them.

The above inflation rise in cost is not bad in and of itself, provided it is accompanied by an equal or greater increase in the approval rate of new NMEs. However, the rate of new drug approvals has not increased (Kaitin & DiMasi 2011; Munos 2009), and the number of NMEs produced per dollar spent is decreasing (Booth & Zemmel 2004). In order to compensate for this increase in cost, companies will either have to introduce undesirable consumer price increases or attempt to reduce the costs incurred when developing an NME. Given that there is a limit to the price rises that consumers can afford or will accept, reducing development costs could enable pharmaceutical companies to maintain (or improve) their earnings per dollar spent developing an NME while limiting the cost to customers.

There are arguably three primary components to the cost of developing a drug: the length of the development time, the clinical trials used to meet regulatory requirements and the rate of attrition. Longer drug development times by themselves do not necessarily increase the out-of-pocket costs to the pharmaceutical company, as a drug may cost a certain amount to develop whether it takes five years or ten, but do increase the capitalised costs and decrease the length of time in which a company can profit from patent exclusivity. Therefore, lengthier development times can reduce the profitability of NME development. However, studies have shown that development times have not contributed substantially to the increasing development costs (DiMasi et al. 2010), and that the average time spent in development and seeking regulatory approval has decreased over time (Kaitin & DiMasi 2011; Kaitin & DiMasi 2000; Keyhani et al. 2006).

Following high profile withdrawals of drugs such as rofecoxib (Vioxx), regulatory authorities have demanded more data be supplied before approval is granted (Kola 2008). This

data must be gathered during clinical trials, and has caused clinical trial processes to become more complex, due to increased reporting requirements, and costlier (Collier 2009a). Costs are also increasing as a result of the fact that many newer drugs show only modest improvements over older ones. This means that larger datasets, and therefore larger trial sizes, are needed to demonstrate that the new drug is a statistically significant improvement (Collier 2009b).

The attrition rate describes how likely a drug is to fail to advance from one phase of the development process to another, and is ideally kept low. However, determining the best compound to modulate a target is a difficult process, and rightly results in many failed compounds for each success. For every drug that receives approval, approximately 5,000 to 10,000 compounds will enter preclinical testing (Klees & Joines 1997). An estimate of the overall clinical success rate for drugs entering development in the late 1980s was 21.5% (DiMasi et al. 2003). Estimates for more recent periods have placed the likelihood of a drug that enters clinical trials eventually receiving approval at 13% (DiMasi et al. 2010), 12% (Davis et al. 2011) and 11% (Kola & Landis 2004). These overall trends mask large differences in the attrition rate for each stage of the development process. While approximately 60% of drugs successfully complete phase I trials, 55% complete phase III and 75% succeed in obtaining approval after being registered for it, only 40% of drugs that enter phase II trials successfully complete them (Kola & Landis 2004). High rates of attrition, especially in phase II trials, are believed to be one of the most significant causes of rising drug development costs (Booth & Zemmel 2004; Paul et al. 2010; Schmid & Smith 2005).

Although increasing clinical trial size and complexity is a significant contributor to the rising cost of development, part of this increase is due to the unavoidable need for more data. Additionally, as failed development projects incur exactly the same costs as successful ones, it is likely to be more beneficial to reduce the number of development programs that fail during clinical trials, rather than solely attempt to reduce the costs of individual trials. A reduction in the attrition rate of clinical trials would decrease the overall cost of developing a drug by reducing the time and money spent on compounds that will ultimately not make it to market. However, it is not possible to achieve this reduction by only acting to improve the clinical trial process. Rather, it is necessary to be more stringent in the earlier stages of development, and thereby stop drugs that would not complete clinical trials from entering them. Additional evidence of the need to improve the preceding stages comes from Dimitri (2011), who demonstrated that much of the current productivity problems in drug development can be attributed to the discovery stage, and Morgan *et al.* (2012), who found that in a significant number of failed drug development programs understanding of the target and its interaction with the NME was insufficient to determine whether the failure was due to an unsuitable target or the NME failing to sufficiently modulate the target's activity.

Dimitri (2011) found that the steps within the discovery stage that would benefit most from productivity improvements were the hit-to-lead and lead optimisation steps, as they have very small marginal contributions to the overall probability of successfully launching a drug and provide the smallest increase in the probability of success per dollar spent (Dimitri 2011). The marginal contribution of these stages can be increased by improving the probability of a development program succeeding once it reaches them, or by increasing the stringency of the preceding stages. Increasing the stringency of the preceding stages will also have the added benefit of decreasing attrition in all subsequent stages, including hit-to-lead and lead optimisation. One solution to this problem is to be more stringent when selecting the targets to proceed forward with (Butcher 2003). Increased stringency when selecting targets would also help to alleviate the problem of pharmaceutical companies being able to identify novel targets, but have many of them be insufficiently validated for building a reliable development program on (Paul et al. 2010). Improving the identification and validation of protein targets should therefore reduce the attrition rate of subsequent stages in the drug development process by ensuring that only high quality targets are chosen to build a development program on. This will help to ensure that potential failures are identified before significant costs have been incurred, and that compounds making it through to the latter stages of development are more likely to succeed.

1.3 Computational Target Discovery

One approach to improving the identification and validation of targets, and therefore the drug development process as a whole, is *in silico* analysis of the suitability of non-target proteins to serve as drug targets. Ideally this would provide a quick and accurate method of cutting down the list of potential therapeutic targets, in order to reduce the need for more costly and time consuming experimental validation. Putative targets could then either be chosen from a list of proteins predicted to be suitable as drug targets, or could be determined experimentally and validated against the computational results. Either approach will enable resources to be utilised more productively, by devoting them to those proteins that appear to be most amenable to pharmaceutical modulation.

Prediction of the druggability of a putative target will often take one of three forms: family-structure- or feature-based. Family-based druggability prediction seeks to determine a protein's druggability based on the protein family it belongs to, and the historic druggability of proteins in the family (Cheng et al. 2007). However, this approach will only find druggable proteins in those families that already contain a target protein, and ignores the fact that not all members of a given family are equally druggable (Fauman et al. 2003).

Rather than simply matching protein families, structure-based druggability prediction consists of predicting ligand-binding sites that are complimentary to known drug-like properties (Fauman et al. 2011). These approaches often have three components (Fauman et al. 2011; Nayal & Honig 2006):

1. A method of identifying potential binding sites in the protein's structure.
2. A method of quantifying the physiochemical and geometric properties of the identified binding sites.
3. A method of assessing how the properties of a binding site fit a training set of proteins with a known outcome.

The main drawback in the usage of structure-based methods is the requirement that the structure of the proteins be known. As for most proteins this is not the case, methods relying on protein structure are effectively limited to a subset of all proteins, those with known structure. As many drug targets are transmembrane proteins, and the structures of membrane bound proteins are particularly difficult to determine, this is especially problematic for large-scale drug target identification.

In order to evaluate the druggability of a larger range of proteins, including the pharmacologically important membrane bound ones, an alternative to structure-based druggability prediction is required. One such potential alternative is feature-based druggability prediction, as it requires no *a priori* knowledge of a protein's properties, besides its sequence. Using features primarily derived from protein sequences, feature-based druggability prediction seeks to train a machine learning classifier to distinguish between target and non-target proteins. Once trained, the classifier is used to predict the druggability of non-target proteins, and therefore whether any of them appear to be suitable future drug targets. While the utility of simple sequence based properties in determining druggability may not be immediately obvious, previous work has shown combinations of them to be highly capable of discriminating between target and non-target proteins (Bakheet & Doig 2009; Huang et al. 2010; Li & Lai 2007).

Successful feature-based methods require reasoned decisions be made as to the set of features used and the method of constructing both the target and non-target sets. Li and Lai (2007), who carried out one of the first feature-based druggability prediction studies, derived from each protein's sequence 146 physiochemical properties, such as amino acid composition and hydrophobicity, that were believed to serve as an accurate representation of the sequence's properties. The dataset used consisted of 186 targets, those proteins found by Overington *et al.* (2006) to be the target of a currently approved drug, and a non-target set of 9,758 human proteins not contained in the families of any approved or research targets. Using these 146

features and approximately 10,000 proteins, Li and Lai (2007) were able to train a classifier to distinguish between targets and non-targets with 84% accuracy.

While an accuracy of 84% indicates that the trained classifier was able to distinguish between target and non-target proteins, the method used by Li and Lai (2007) suffers from both the small size and potential redundancy in the set of target proteins, and could possibly benefit from the use of features that describe more than just the properties of a protein's sequence. A more recent attempt that sought to address some of these limitations is that of Bakheet and Doig (2009). By using the set of all approved targets in DrugBank (Wishart et al. 2008) as the target set, they were able to start with a set of approximately 500 target proteins, which generated a set of 146 non-redundant targets following redundancy removal. Removing the approximately 500 targets, and then redundancy, from the set of all human proteins in UniProt (The UniProt Consortium 2010), generated a set of 3,573 non-redundant non-target proteins. While both the target and non-target sets are smaller than those used by Li and Lai, they are likely to be more representative of the sets of targets and non-targets due to the removal of redundancy. In addition to using more representative sets of proteins, Bakheet and Doig augmented the feature set by adding properties such as the number of glycosylation sites, phosphorylation sites and transmembrane regions of the proteins. Using their more representative sets of proteins and enhanced feature set, Bakheet and Doig were able to train a classifier to distinguish between target and non-target proteins with 89% accuracy. Applying this trained classifier to the set of non-targets removed as redundant, identified 668 non-targets as similar enough to the targets to be predicted as druggable.

Rather than attempt to predict the druggability of all human proteins, a constrained subset of the human proteome can instead be used, such as only those proteins in a particular family. This subdividing approach represents a promising direction for druggability prediction in general, as it removes the implicit assumption that all targets, irrespective of their protein family and method of modulation, are similar. The first feature-based attempt at this was by Huang *et al.* (2010), who sought to train a classifier to distinguish between target and non-target ion channels. Using 31 target ion channels, 16 non-target ion channels and a subset of the features used by Li and Lai (2007), they were able to train a classifier that could distinguish between ion channel targets and non-targets with an accuracy of 83%. Although this classifier would hopefully be more adept at discovering novel non-target ion channels with target-like characteristics, as there is less 'noise' from dissimilar proteins, the additional data required to identify specific proteins belonging to the subdivision, especially target proteins, proved problematic. Therefore, despite the promise of the results, a dataset of 47 proteins is unlikely to be sufficient to draw robust conclusions.

Despite these successes, feature-based prediction shares one main drawback with both other druggability prediction methods: their results depend heavily on the makeup of the training set used to develop them. As the classifiers will have only learned about druggability from the current targets, they are unlikely to be able to accurately predict the druggability of proteins that are dissimilar to them, or which are modulated via undiscovered mechanisms. These methods are therefore unlikely to be able to identify truly new classes of targets. However, they have been shown to be effective despite the limitations that the available data impose on them, and will continue to improve as more data is gathered and training sets become more comprehensive.

In light of this, the work presented here is an attempt to build on and extend the successes of the three previous feature-based prediction studies. One concern with all three previous approaches is the fact that the training sets used were relatively small, which makes both the relationships found by the classifier and its perceived accuracy questionable. In order to determine the potential for success now that greater data is available on both targets and proteins in general, predictions of druggability of the entire proteome will again be assessed. Furthermore, attempts to subdivide the proteome are likely to be most promising, and have only begun to be investigated for modes of action, and not at all for subdivisions based on other properties, such as targets involved in a specific disease. In order to remedy this, the approach pursued here is one of feature-based prediction using constrained subsets of the human proteome. However, rather than solely use simple sequence-based properties, the dataset is enriched with additional descriptive features, such as protein protein interactions, sequence variant information and expression profiles. Ideally this will enable the training of an improved classifier, while also giving insight into the importance of these properties in determining druggability.

2 Redundancy Removal – Development of New Methods

A dataset can be considered to be redundant if it contains duplicate or excessively similar observations. The two predominant drawbacks of redundancy are an increase in the size of the dataset and the potential introduction of bias into analyses of it. Compared to a non-redundant dataset generated from it, a redundant one will contain more observations, and therefore require more time and space to analyse. The computational disadvantage that this imposes could be partly offset if the greater number of observations enabled easier discovery of relationships contained in the data, but as the extra observations are effectively duplicates they provide no additional information.

Problems with analyses of a redundant dataset can be seen in the bias introduced, for example, when calculating averages or attempting to infer trends. Further to this, when training a machine learning classifier on a redundant dataset, the resultant perceived performance of the classifier will always be questionable. As it is customary in machine learning problems to split a dataset into training and testing portions, with the intention that no observation is used for both training and testing, a redundant dataset opens the trained classifier up to the possibility that duplicate observations occur both in the training and testing portions. In this case, the performance of the classifier, as determined by its performance on the testing portion of the dataset, will be positively biased. Additionally, if the training set itself contains duplicate observations, then the classifier will be biased to perform well on the overrepresented observations, and will therefore overfit the data.

In order to avoid the problems of redundancy, a pre-processing step is often used to generate a non-redundant dataset consisting solely of representative observations from the original redundant set. However, statistical analyses and machine learning methods benefit from being used with as large a dataset as possible. By treating the generation of this non-redundant datasets as an optimisation problem, it is possible to both remove redundancy and maximise the size of the resultant non-redundant dataset.

The datasets investigated here are all protein datasets, which can be considered to be redundant if they contain a pair of proteins for which some pre-determined similarity measure is greater than a given threshold. Removing redundancy from a protein dataset is usually achieved by removing proteins from the dataset until no two remaining proteins have a similarity greater than the threshold. In order to do this it is first necessary to define the similarity measure, so that the redundant relationships can be determined, and a method of selecting the proteins to remove. The work presented in this chapter seeks to define a standard similarity measure, and then develop and discuss potential methods for removing redundancy. Chapter 3 then contains the results of a comprehensive comparison of the different methods.

The content in this chapter and Chapter 3 has previously been published (Bull et al. 2013).

2.1 Measuring Redundancy

Algorithms for determining the similarity between two proteins are more useful if they compare the proteins' sequences rather than their structures, as structures are unavailable for many proteins and evolutionary relationships can be difficult to quantify. Alignment based approaches to calculating the sequence identity between two proteins can be either global or local. Global methods attempt to align every amino acid in both sequences, and consequently the alignment found may contain large stretches of low sequence identity (Needleman & Wunsch 1970). Instead of searching for one optimal global alignment, local alignment methods search for, potentially many, relatively conserved subsequences between the two sequences (Altschul et al. 1990). They are therefore more suitable than global alignment based methods when the sequences share only isolated regions of similarity, or when scanning a dataset with little to no *a priori* knowledge of the similarity between the query sequence and the sequences in the dataset (Barton 1996). Irrespective of the method used to calculate the alignment, the computational cost of calculating it necessitates the use of heuristics, the most prominent of which are BLAST (Altschul et al. 1990) and PSI-BLAST, which is more sensitive to weak sequence similarity in many cases (Altschul et al. 1997).

2.2 Current Methods of Redundancy Removal

One common approach to removing redundancy is a list based approach, such as that taken by the widely used PISCES algorithm(Wang & Dunbrack 2003), whereby the proteins in the dataset are sorted according to some property, descending sequence length in the case of PISCES, that they all possess and then processed using the following steps:

1. Find p, the protein highest up the list that is not marked as kept or removed.
2. Mark p as being kept.
3. Mark as removed all proteins that are considered to be too similar to p.
4. If the bottom of the list has been reached stop, else go to step 1.

Although the proteins marked as kept by this process will comprise a non-redundant set, the preference of the algorithm for keeping proteins higher up the list will bias the makeup of the set towards those proteins with a value of the sorting property that places them higher up the list. In the case of PISCES, the proteins are sorted in descending order by their sequence length, and the non-redundant set generated is therefore biased towards proteins with longer sequences.

One alternative to using list based approaches is to represent the similarity relationships between the proteins graphically. A protein similarity graph, $G(V, E)$, of a dataset is an undirected graph with vertices $V = \{1,2, \dots, n\}$ and edges $E = \{\{i, j\} | i, j \in V\}$, where n is the number of proteins in the dataset. Each protein, p, in the dataset is represented by a vertex, v_p, in G, and for every protein, s, that is too similar to p there is an edge between vertices v_p and v_s. As edges indicate redundancy, the goal of graph based redundancy removal algorithms is to select an *independent set*, $I \subseteq V$, of vertices such that $\forall i, j \in I: \{i, j\} \notin E$, meaning that no two vertices in I share an edge. Ideally a *maximum independent set* (MIS), the largest possible independent set in a graph, would be found. However, finding an MIS is an NP-complete problem, and therefore most methods will approximate it. This approximation will often take the form of a *maximal independent set*, an independent set, I, such that the addition of any vertex $v \in (V - I)$ to I would cause it to violate the properties of an independent set.

By not relying on an artificial ordering of the proteins, graph based redundancy removal methods can take a global view of the problem, and are therefore not constrained in their selection of the proteins to keep in the non-redundant set. Situations where this more global outlook leads to the generation of a larger non-redundant dataset are not difficult to construct. For example, if we consider the graphs in Figure 1, the two possible maximal independent sets are $\{A\}$ and $\{B, C\}$. Clearly $\{B, C\}$ is preferable from the point of maximising the size of the independent set. However, were the proteins in the graphs to be ranked according to their sequence length, as PISCES would do, the list would be:

1. Protein A (100 amino acids)
2. Protein C (75 amino acids)
3. Protein B (50 amino acids)

Therefore, using a list based approach, and sorting by descending sequence length, would lead to a non-redundant set consisting solely of protein A. This suboptimal choice can be avoided by using a graph based approach with an appropriate rule for deciding which of two connected vertices to keep in the non-redundant set.

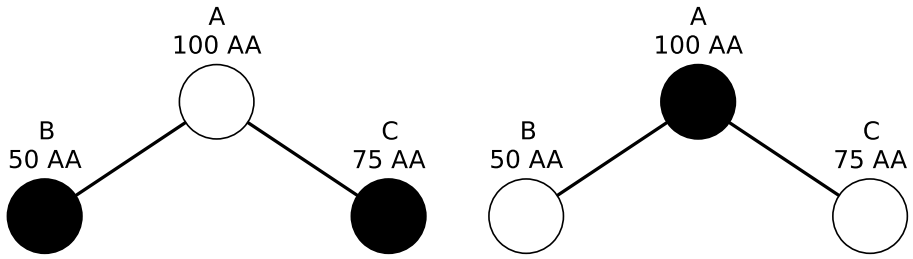

Figure 1: Alternative maximal independent sets. Graphs with this structure contain two different maximal independent sets. Black vertices are in the maximal independent set and white ones are not. In this example the sequence of the protein that vertex A represents contains 100 amino acids, protein B's sequence 50 amino acids and C's 75.

2.3 Development of New Removal Methods

2.3.1 Graph Definitions

In order to fully describe the graph based redundancy removal methods developed and tested here, some initial definitions of the properties of vertices and graphs are required (illustrated pictorially in Figure 2). The *neighbourhood* of a vertex, $v \in V$, in a graph, $G(V, E)$, can be defined as $N(v) = \{i | \{i, v\} \in E\}$, the set of all vertices that share an edge with v. The neighbourhood of a set of vertices, $s \subseteq V$, can be defined as the union of the neighbourhoods of the vertices in it, $N(s) = \bigcup_{v \in s} N(v)$, and will therefore contain all vertices that share an edge with at least one vertex in s. The $extended\ neighborhood$ of v, $EN(v) = N(v) \cup N\big(N(v)\big)$, is the set of all vertices that are reachable from v by traversing two edges or less. Using the definition of the neighbourhood of a vertex, $v \in V$, the *degree* and *support* of v can be defined as $degree(v) = \#N(v)$ and $support(v) = \sum_{i \in N(v)} degree(i)$ respectively.

A *clique*, $Q \subseteq V$, is a subset of the vertices in G such that $\forall i, j \in Q: \{i, j\} \in E$. A *maximum clique* of G is a clique, Q_m, such that $\#Q_m \geq \#Q_i$ for all other cliques, Q_i, in G. A vertex cover, $C \subseteq V$, of G is a set of vertices such $\forall \{i, j\} \in E: i \in C \vee j \in C$. A *minimum vertex cover* of G is a vertex cover, C_m, such that $\#C_m \leq \#C_i$ for all other vertex covers, C_i, of G. Finally, the complement of G is a graph $H(V, E')$ with the same vertex set, but with edges $E' = \{\{i, j\} | i \in V \wedge j \in V \wedge \{i, j\} \notin E\}$. Thus, two vertices share an edge in G only if they do not share one in H. Given a (minimum) vertex cover $C \subseteq V$, a (maximum) clique in G's complement is the vertices in $Q = V - C$. The vertices in Q are also a (maximum) independent set (Section 2.2) in G. Using these conversions it is possible to use algorithms for approximating a maximum clique or minimum vertex cover in order to approximate an MIS.

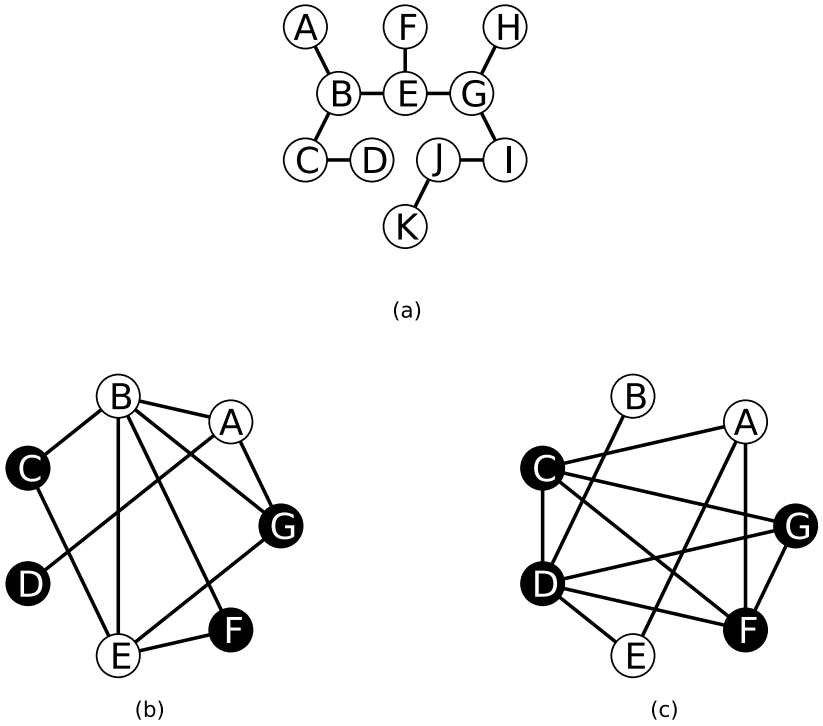

Figure 2: Illustrations of the graph theory definitions used. Taking vertex E in (a) to be the node of interest, the $N(E) = \{B, F, G\}$, the $N(N(E)) = \{A, C, E, H, I\}$, the $EN(E) = \{A, B, C, E, F, G, H, I\}$, the $degree(E) = 3$ and the $support(E) = degree(B) + degree(F) + degree(G) = 3 + 1 + 3 = 7$. The graphs (b) and (c) are each other's complement. The black vertices in (b) represent the MIS and the black ones in (c) the maximum clique, while the white vertices in (b) are the minimum vertex cover.

2.3.2 Redundancy Removal Methods Developed

Three heuristics for finding a maximal independent set (NeighbourCull, Leaf and FIS) were developed. These were then compared to existing approaches for finding maximal independent sets and generating non-redundant protein datasets. The algorithms described in this section were all implemented in order to carry out the comparisons in Chapter 3. These implementations can be found at https://github.com/SimonCB765/AlgorithmComparison.

2.3.2.1 NeighbourCull

The NeighbourCull algorithm is a graph based method that works by iteratively removing the vertex, v, with the most neighbours, along with all edges incident to v. After a number of iterations there will be no edges in the graph, and the vertices remaining will constitute a maximal independent set. Given a graph $G(V, E)$, pseudocode for the NeighbourCull algorithm can be seen in Figure 3. The first step in each iteration of the algorithm is to determine the set of vertices, $connected$, in the graph that have neighbours (line 2). If there are no vertices with neighbours, and therefore no edges remaining in the graph, a maximal independent set has been found and

the algorithm can exit (lines 3 and 4). However, if the graph still contains edges, then the set, max, of vertices that have the largest neighbourhood is found (line 6). If only one vertex has the largest neighbourhood, i.e. $\#max = 1$, then the single vertex in max, and its incident edges, is removed from the graph (lines 7 and 8). If there is more than one vertex with the largest neighbourhood, then the extended neighbourhood of each vertex in max is calculated, and the subset, $ext \subseteq max$, of vertices that have the smallest extended neighbourhood is determined (line 10). Following this, one of the vertices in ext, and its incident edges, is arbitrarily chosen to be removed from the graph (line 11).

```
1.  While True
2.    connected ≔ {i |i ∈ V ∧ degree(i) > 0}
3.    If connected = ∅
4.      Return V
5.    Else
6.      max = {i |i ∈ V ∧ (∀j ∈ V: degree(i) ≥ degree(j))}
7.      If #max = 1
8.        <Remove the vertex in max from the graph>
9.      Else
10.       ext = {i |i ∈ max ∧ (∀j ∈ max: #EN(i) ≤ #EN(j))}
11.       <Remove a vertex n ∈ ext from the graph>
```

Figure 3: Pseudocode for the NeighbourCull algorithm.

2.3.2.2 Leaf

The Leaf algorithm works through the iterative discovery of cliques satisfying the following 'Leaf criterion':

- Given a graph, $G(V, E)$, if there is a clique, $Q \subseteq V$, that contains a vertex, $v \in Q$, such that $N(v) = Q - \{v\}$, then v should be in the maximal independent set and the vertices in $Q - \{v\}$ should not.

Similar to NeighbourCull, each iteration of the Leaf algorithm will remove a set of vertices, those in $Q - \{v\}$, and all edges incident to them. After a number of iterations there will be no edges in the graph, and the vertices remaining will constitute a maximal independent set. Each iteration of the algorithm begins by searching for cliques of two vertices, and incrementing the size of the clique being searched for until either one satisfying the criterion is found or the required size of the clique precludes its existence.

Given a graph $G(V, E)$, pseudocode for the Leaf algorithm can be seen in Figure 4. The first step in each iteration of the algorithm is to determine whether there are any edges remaining in the graph, and to return the vertices in the graph if there are not (lines 2-4). If there are edges remaining, then the algorithm searches for a clique that satisfies the Leaf criterion. The first step in the search is to initialise the size of the clique being searched for (line 5). Next a loop (lines 6-10) is entered to search for sequentially larger cliques of a size that can be present in the graph. If a clique, Q, with a vertex, $v \in Q$, such that $N(v) = Q - \{v\}$, is found, then the vertices in $Q - \{v\}$, and all edges incident to them, are removed from the graph (lines 7 and 8). If no clique is found, then lines 6-11 of the NeighbourCull algorithm (Section 2.3.2.1) are used to determine a vertex, and incident edges, to remove from the graph (lines 11 and 12).

```
1.  While True
2.    <Set max to degree(i) such that i ∈ V and ∀j ∈ V: degree(i) ≥ degree(j)>
3.    If max = 0
4.      Return V
5.    nClique ≔ 2
6.    While nClique ≤ max + 1
7.      If there is a clique Q such that #Q = nClique ∧ ∃v ∈ Q: N(v) = Q − {v}
8.        <Remove the vertices in (Q − {v}) from the graph>
9.      Else
10.       nClique ≔ nClique + 1
11.   If nClique ≤ max + 1
12.     <Use lines 6-11 of NeighbourCull to update the graph>
```

Figure 4: Pseudocode for the Leaf algorithm.

2.3.2.2.1 Proof of the Optimality of the Leaf Algorithm Criterion

Given a graph $G(V, E)$, a clique, $Q \subseteq V$, and a function $Ind()$ that returns an MIS of a graph, the three options for the vertices in Q when generating an independent set are:

1. Include a vertex, $i \in Q$, such that $N(i) = Q - \{i\}$, in the independent set.
2. Include a vertex, $j \in Q$, such that $\exists v \in N(j): v \notin Q$, in the independent set.
3. Include no vertices from Q in the independent set.

Assuming $R = V - Q$, the possible independent sets that can be generated using the three options are:

1. $Ind(R) + \{i\}$
2. $Ind\big(R - N(j)\big) + \{j\}$
3. $Ind(R)$

Option 1 can be seen to be optimal as $\#(Ind(R) + \{i\}) \geq \#(Ind\big(R - N(j)\big) + \{j\}) > \#Ind(R)$. Therefore, including i in the independent set is always better than including no vertex from Q, and leads to an indpendnet set that is no smaller than that produced by including j.

2.3.2.3 FIS

The third new algorithm works by first initialising a maximal independent set in a greedy manner, and then permuting it in an attempt to increase its size. Given a graph $G(V, E)$, pseudocode for the FIS algorithm can be seen in Figure 5. The first step is to determine the initial vertex from which the maximal independent set will be generated. This is chosen to be the vertex with the fewest neighbours (lines 1 and 2), with ties broken arbitrarily. From this initial vertex, a maximal independent set is generated (line 3), and then permuted in an attempt to increase its size (line4). The initial maximal independent set is generated using the $addnodes$ sub-function. This takes as arguments an independent set, Ind, and the set of all vertices in the graph, V. The first step in generating a maximal independent set from Ind is to find the vertices, $add \subset V$, that share no edges with a vertex in Ind (line 6). If there are no vertices that meet this criterion, then Ind is returned (lines 7 and 8), otherwise a vertex from add is added to Ind (lines 9 and 10). This is done by finding the vertex, $v \in add$, such that $\#N(Ind + \{v\})$ is minimised. This process is repeated until all vertices are either in Ind or share an edge with a vertex in it.

Once $addnodes$ has been used to find a maximal independent set, the $swapnodes$ sub-function is used in an attempt to increase the size of the set by making small alterations to the vertices in it. $swapnodes$ takes as arguments an independent set, Ind, and the vertices in the graph, V. The vertices in $V - Ind$ are tested one at a time (lines 15-22) to see how many vertices in Ind they share an edge with (line 16). If there is a vertex, $i \in V - Ind$, that is adjacent to only one vertex, j, in Ind, then i can be added to Ind and j removed from it without invalidating the properties of an independent set (line 18). The new independent set resulting from this swap is passed to $addnodes$ to see if its size can be increased (line 19). If Ind contains fewer vertices than the independent set returned, $temp$, then Ind is set to $temp$ (line 21). This process is repeated until Ind's size cannot be increased after the call to $addnodes$ on line 18. Following the call to $swapnodes$ on line 4, the vertices in Ind comprise the maximal independent set.

1. <Select the vertex $i \in V$ such that $\forall j \in V: degree(i) \leq degree(j)$>
2. $Ind = \{i\}$
3. $Ind = addnodes(Ind, V)$
4. $Ind = swapnodes(Ind, V)$

$addnodes(Ind, V)$:

5. While $True$
6. $add = V - Ind - N(Ind)$
7. If $add = \emptyset$
8. Return Ind
9. <Select $i \in add$ such that $\forall j \in add : \#(add \cap N(i)) \leq \#(add \cap N(j))$>
10. $Ind \coloneqq Ind \cup \{i\}$
11. Return Ind

$swapnodes(Ind, V)$:

12. $changed \coloneqq True$
13. While $changed$
14. $changed \coloneqq False$
15. For i in $V - Ind$
16. $adj = N(i) \cap Ind$
17. If $\#adj = 1$
18. $test \coloneqq (Ind \cup \{i\}) - adj$
19. $temp \coloneqq addnodes(test, V)$
20. If $\#temp > \#Ind$
21. $Ind \coloneqq temp$
22. $changed \coloneqq True$
23. Return Ind

Figure 5: Psuedocode for the FIS algorithm.

2.3.2.4 Examples

The differences between the workings of the new algorithms can best be shown through the use of an example. The graph in Figure 6 will be used to demonstrate the differences in the execution and final independent set generated by the Leaf, NeighbourCull and FIS algorithms. When the execution of an algorithm calls for an arbitrary choice between vertices the lexicographical ordering of the vertex identifiers (A, B, C, etc.) will be used.

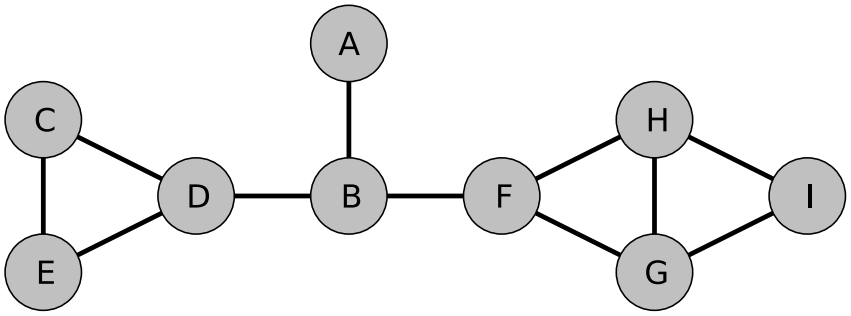

Figure 6: Graph for demonstrating the Leaf, NeighbourCull and FIS algorithms.

2.3.2.4.1 Leaf

The execution of the Leaf algorithm on the graph in Figure 6 is as follows (demonstrated pictorially in Figure 7):

1. Clique $\{A, B\}$ is the smallest clique with a vertex, A, that has no neighbours external to the clique. Vertex B is therefore removed from the graph.
2. There are three cliques ($\{C, D, E\}$, $\{F, G, H\}$ and $\{G, H, I\}$) in the graph that contain at least one vertex with no neighbours external to the clique. $\{C, D, E\}$ is arbitrarily chosen, and contains three vertices (C, D and E) that have no neighbours external to the clique. Therefore, vertices D and E are arbitrarily chosen to be removed from the graph.
3. There are two cliques ($\{F, G, H\}$ and $\{G, H, I\}$) remaining in the graph that contain at least one vertex with no neighbours external to the clique. $\{F, G, H\}$ is arbitrarily chosen as the clique to use, and as vertex F is the only vertex in the clique with no neighbours external to the clique, vertices G and H are removed from the graph.
4. No edges remain in the graph. The vertices remaining, $\{A, C, F, I\}$, are therefore the maximal independent set returned.

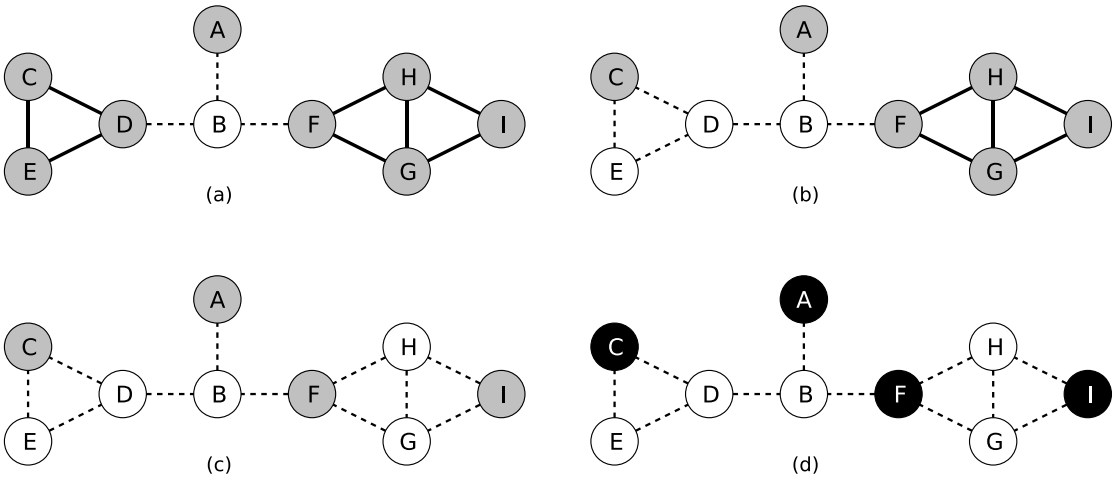

Figure 7: The progress of the execution of the Leaf algorithm on the graph in Figure 6. Black vertices represent the maximal independent set generated, white vertices are removed vertices and grey vertices are still under consideration for inclusion. Dashed edges indicate edges that are no longer in the graph, as they are incident to a removed vertex. Each graph corresponds to the state of the graph in Figure 6 following a step in the execution of the Leaf algorithm. Graph (a) is after step 1 of the algorithm, (b) after step 2, (c) after step 3 and (d) after step 4.

2.3.2.4.2 NeighbourCull

The execution of the NeighbourCull algorithm on the graph in Figure 6 is as follows (demonstrated pictorially in Figure 8):

1. Vertices B, D, F, G and H all have the most neighbours, 3, so their extended neighbourhoods need to be calculated. The data for the neighbourhood and extended neighbourhood of each of the vertices is summarised in the table below:

Vertex	$N(v)$	$N(N(v))$	$EN(v)$	$\#EN(v)$
B	$\{A, D, F\}$	$\{B, C, E, G, H\}$	$\{A, B, C, D, E, F, G, H\}$	8
D	$\{B, C, E\}$	$\{A, C, D, E, F\}$	$\{A, B, C, D, E, F\}$	6
F	$\{B, G, H\}$	$\{A, D, F, G, H, I\}$	$\{A, B, D, F, G, H, I\}$	7
G	$\{F, H, I\}$	$\{B, F, G, H, I\}$	$\{B, F, G, H, I\}$	5
H	$\{F, G, I\}$	$\{B, F, G, H, I\}$	$\{B, F, G, H, I\}$	5

   Vertices G and H have the smallest extended neighbourhoods, and therefore vertex G is arbitrarily chosen as the vertex to remove from the graph.
2. Vertices B and D now have the most neighbours, 3, while their extended neighbourhoods contain 7 and 6 vertices respectively. Vertex D is therefore removed from the graph.
3. Vertices B, F and H now all have the most neighbours, 2, while their extended neighbourhoods contain 4, 5 and 4 vertices respectively. Vertex B is arbitrarily chosen to be removed from the graph.

4. Vertex H now has more neighbours than any other vertex, and is therefore removed from the graph.
5. Vertices C and E both have the most neighbours and the same extended neighbourhood. Vertex C is therefore arbitrarily chosen to be removed from the graph.
6. No edges remain in the graph. The vertices remaining, $\{A, E, F, I\}$, are therefore the maximal independent set returned.

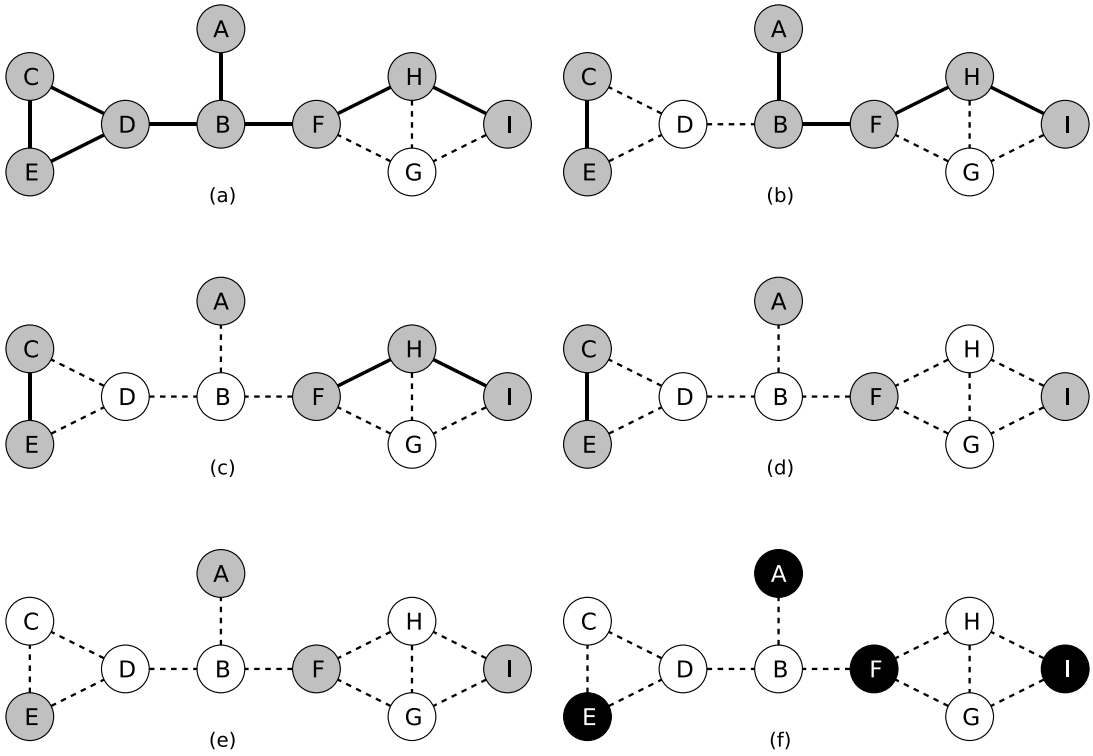

Figure 8: The progress of the execution of the NeighbourCull algorithm on the graph in Figure 6. Black vertices represent the maximal independent set generated, white vertices are removed vertices and grey vertices are still under consideration for inclusion. Dashed edges indicate edges that are no longer in the graph, as they are incident to a removed vertex. Each graph corresponds to the state of the graph in Figure 6 following a step in the execution of the NeighbourCull algorithm. Graph (a) is after step 1 of the algorithm, (b) after step 2, (c) after step 3, (d) after step 4, (e) after step 5 and (f) after step 6.

2.3.2.4.3 FIS

The execution of the FIS algorithm on the graph in Figure 6 is as follows (demonstrated pictorially in Figure 9):

1. The maximal independent set, Ind, is initialised to $\{A\}$, as vertex A has the fewest neighbours.
2. $addnodes$ is called with $Ind = \{A\}$ and $V = \{A, B, C, D, E, F, G, H, I\}$.

3. The addition of vertex C, D, E, F or I would cause the fewest vertices that are currently not adjacent to Ind to become adjacent to it. Vertex C is arbitrarly chosen to be added to Ind.
4. The addition of vertex F or I would cause the fewest vertices that are currently not adjacent to Ind to become adjacent to it. Vertex F is arbitrarly chosen to be added to Ind.
5. Vertex I is added to Ind as it causes the fewest additional vertices to become adjacent to it.
6. $Ind = \{A, C, F, I\}$ once $addnodes$ terminates.
7. $swapnodes$ is called with $Ind = \{A, C, F, I\}$ and $V = \{A, B, C, D, E, F, G, H, I\}$.
8. The vertices in $nonInd = V - Ind = \{B, D, E, G, H\}$ are evaluated to determine if they can be swapped with a vertex in Ind.
9. B is adjacent to two vertices in Ind, and therefore cannot be swapped with a vertex in it.
10. D is ony adjacent to one vertex, C, in Ind, and can therefore be swapped with it.
11. $addnodes$ is called with $Ind = \{A, D, F, I\}$ and $V = \{A, B, C, D, E, F, G, H, I\}$.
12. No vertices can be added to Ind.
13. As the size of Ind has not increased, Ind is reverted to being $\{A, C, F, I\}$ and the vertices in $nonInd$ continue to be evaluated.
14. E is ony adjacent to one vertex, C, in Ind, and can therefore be swapped with it.
15. $addnodes$ is called with $Ind = \{A, E, F, I\}$ and $V = \{A, B, C, D, E, F, G, H, I\}$.
16. No vertices can be added to Ind.
17. As the size of Ind has not increased, Ind is reverted to being $\{A, C, F, I\}$ and the vertices in $nonInd$ continue to be evaluated.
18. G is adjacent to two vertices in Ind, and therefore cannot be swapped with a vertex in it.
19. H is adjacent to two vertices in Ind, and therefore cannot be swapped with a vertex in it.
20. $nonInd$ has been fully evaluated without increasing the size of Ind, and therefore $swapnodes$ returns $\{A, C, F, I\}$.

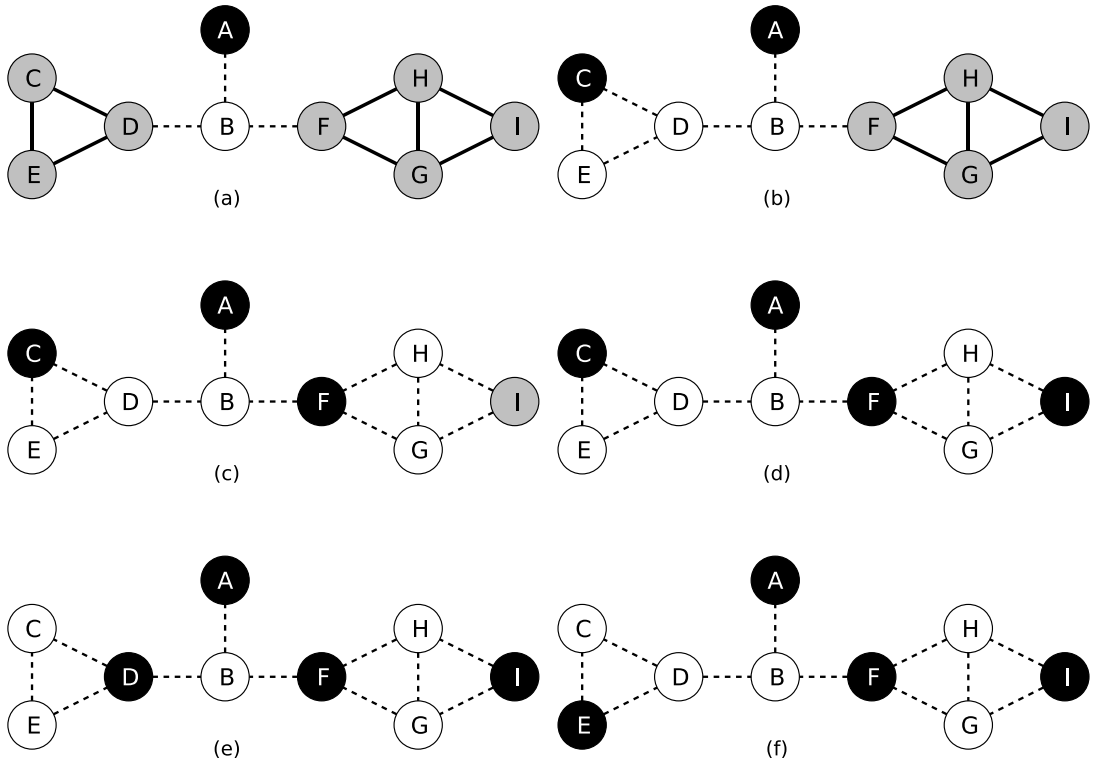

Figure 9: The progress of the execution of the FIS algorithm on the graph in Figure 6. Black vertices represent the maximal independent set being generated, white vertices are those vertices that are not in the maximal independent set and grey vertices are still under consideration for inclusion. Dashed edges indicate edges that no longer have any influence in determining the maximal independent set. Each graph corresponds to the state of the graph in Figure 6 following a step in the execution of the FIS algorithm. Graph (a) is after step 1 of the algorithm, (b) after step 3, (c) after step 4, (d) after step 5, (e) after step 10 and (f) after step 14.

2.3.2.5 Existing Algorithms Evaluated

The algorithms described in this section were all implemented in order to carry out the comparisons in Chapter 3. These implementations can be found at https://github.com/SimonCB765/AlgorithmComparison.

BlastCuller (Liu et al. 2009) is an independent set finding heuristic that works by building up a maximal independent set through the addition of isolated vertices. First, the maximal independent set, I, being generated is initialised as the empty set, and has all isolated vertices added to it. Following this, the vertex, $v \notin I$, with the most neighbours is removed from the graph, along with any edges incident to it, and any newly isolated vertices are added to I. This process is repeated until all vertices are either in I or have been removed from the graph.

The second heuristic used to find a maximal independent set was VSA (Balaji et al. 2010), which calculates an approximation to the minimum vertex cover in a graph. VSA works by starting with an empty set of vertices, C, and iteratively increasing the number of vertices in it by adding the vertex in $V - C$ with the largest support. If multiple vertices have the largest support, then the one with the most neighbours is added. When a vertex, v, is added to C, all edges incident to

v are removed from the graph. C is a vertex cover once there are no more edges remaining in the graph.

The final heuristic, abbreviated here as GLP, used to find a maximal independent set was a state of the art heuristic for approximating the maximum clique (Grosso et al. 2007). GLP works by first employing a greedy algorithm in order to find an initial maximal clique. It then uses random local search operations in order to search the space of all maximal cliques, and a random restart rule in order to resume the search in a potentially new location once a plateau has been reached. This combination of operators makes GLP's runtime potentially infinite, and for this reason a parameter that limits the number of vertex addition and swap operations that can be performed is built into the original algorithm. However, there are two problems with using the original GLP algorithm for finding maximal independent sets in arbitrary graphs. Firstly, the size of the MIS in the graphs being tested was known *a priori* in the GLP paper (Grosso et al. 2007), which meant that the algorithm could be stopped early if the clique being generated ever reached this size. With arbitrary graphs the size of the MIS is not known, and therefore the algorithm cannot be stopped early in the same way. Secondly, the value of the parameter limiting the number of operations cannot easily be set to control the run time of the algorithm. In order to use the GLP algorithm effectively, alterations are needed to either output the largest clique found at given checkpoints or to cap the time the algorithm can run for. The approach taken here was to implement GLP algorithm 1, along with restart rule 2, and incorporate a parameter that allows a time limit to be placed on the algorithm's total run time. The algorithm then returns the largest clique found before the time limit is breached.

Despite there being no known efficient algorithms for computing an MIS of an arbitrary graph, it is nonetheless possible to find one by using so-called branch-and-bound algorithms, which have a worst case running time that is exponential in the number of vertices. These methods typically combine a brute force approach with an upper bound that allows statements such as "any independent set that includes vertices 1, 25, 1548 and 21973 contains at most 53 other vertices", thereby eliminating whole families of subsets without having to enumerate and check each member. To obtain exact answers against which to check the results of the other algorithms, the Cliquer library (Niskanen & Östergård 2003; Östergård 2002) was used to find a maximum clique in the complement of the protein similarity graph. Cliquer works by successively computing the size, c_i, of the maximum clique in the subgraph containing the vertices in the set $V_i = \{v_1, v_2, \dots, v_i\}$. The size of the maximum clique in the set $V_{i+1} = V_i \cup \{v_{i+1}\}$ is then either c_i or $c_i + 1$ depending on whether there is a larger clique in V_{i+1} that includes the vertex v_{i+1}. This observation enables the upper bound that speeds Cliquer's search, and also means that the algorithm's runtime depends on the order in which the vertices are listed. Here, Cliquer's default

ordering was used. This proceeds by arranging the vertices in order of decreasing degree, and then using the greedy colouring algorithm to choose large sets of non-adjacent vertices. The final ordering lists the vertices in order of increasing colour-index (as assigned by the greedy colouring stage) and, within each colour group, in order of decreasing degree.

3 Redundancy Removal – Evaluation of Proposed Methods

3.1 Approach Taken

All seven of the methods described in 2.3.2 were evaluated against PISCES (Section 2.2) and one another in terms of the size of the non-redundant dataset generated and the time taken to generate it. The algorithms were first tested on subsets of sequences from the entire human proteome, obtained from UniProt release 2010_12 (The UniProt Consortium 2010). The set, S, of pairwise sequence identities between all pairs of protein sequences in the human proteome was generated using PSI-BLAST version 2.2.25, from the NCBI BLAST+ package (Camacho et al. 2009), with the default scoring matrix. Sequence similarities were calculated once and used in all tests in order to ensure that the similarity calculation had no undue influence on the results. Fifty datasets of 500, 1000, 2000 and 5000 proteins were randomly generated by sampling without replacement from the set of all human proteins. The process for each dataset, d, was as follows:

1. Extract the subset, $S_d \subseteq S$, of pairwise sequence identities between all pairs of proteins in d.
2. Run PISCES on d with the sequence idenitity threshold set to t. The sequence identities in S_d are used rather than having PISCES re-compute them for itself.
3. Generate the protein similarity graph for d with threshold t, i.e. generate the graph of the proteins in d with edges between all pairs of proteins with a pairwise sequence identity greater than t.
4. Run the seven graph based algorithms (three new heuristics, three existing heuristics and Cliquer) on the protein similarity graph generated in step 3.

The sequence identity thresholds, t, used were 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90%. In order to ensure that the sequence similarities calculated previously were used by PISCES, it had to be altered to accept a file of similarities as input. The time limit given to the GLP algorithm was set to be the longer of two minutes or ten times the running time of the Leaf algorithm. The run time for each algorithm was measured as the time taken to compute the maximal independent set, ignoring any set up required before the computation was performed and any steps to produce the output afterwards. This enabled the algorithms to be compared solely on their ability to produce a maximal independent set.

The algorithms were also compared using the same sequence identity thresholds on five full proteomes. The set of all reviewed proteins in the complete proteome were extracted from UniProt release 2011_10 using taxonomy identifiers: 9606 for *H. sapiens*, 10090 for *M. musculus*, 83333 for *E. coli*, 559292 for *S. cerevisiae* and 3702 for *A. thaliana*. By testing the algorithms on

entire proteomes it is possible to determine whether they are practical for redundancy removal in larger datasets.

The final test of the algorithms was the BHOSLIB suite of benchmark graphs (http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/graph-benchmarks.htm). By benchmarking the performance of the algorithms on a set of graphs where an MIS is known to be difficult to find, it is possible to test both the performance of the algorithms on difficult problems and the ability of the algorithms to find an MIS in general. The BHOSLIB benchmarking also served to test how the methods designed for sparser and simpler protein similarity graphs fare on denser more complex graphs. Cliquer was not tested due to the excessive run time that would be needed to find an MIS.

3.2 Results

3.2.1 Subsets of the Human Proteome

The results of the comparison of the algorithms on the subsets of the human proteome can be seen in Table 1. The quality of the algorithms was measured in terms of the number of proteins in the non-redundant datasets generated, with algorithms that generated larger datasets being considered better. All graph theory based algorithms, except GLP, perform as well as or better than PISCES for every threshold and dataset size. Differences in the performances of the algorithms are less evident when using smaller datasets, but become increasingly obvious as the dataset size increases and/or the threshold decreases. For all dataset size/threshold combinations, the ordering of the algorithms from best to worst, excluding GLP, was: Leaf, FIS, NeighbourCull, BlastCuller, VSA and finally PISCES.

The time taken by the different algorithms using a sequence identity threshold of 20% can be seen in Table 2. For all algorithms, except GLP, the time taken increases with the dataset size. Despite the greater complexity of the graph theory based algorithms compared to PISCES, the absolute difference in execution time of the non-GLP graph theory algorithms and PISCES is minimal. The time taken by GLP, 120 seconds, indicates that it is hitting its execution time limit of the longer of two minutes or ten times the execution time of Leaf.

Subset Size	Threshold	PISCES	Leaf	FIS	NC	VSA	BC	GLP
500 Proteins	20%	371.9	384.3	384.1	384.0	380.4	383.4	380.9
	30%	442.2	445.1	445.0	445.0	444.5	444.8	442.6
	40%	472.3	474.5	474.5	474.5	474.1	474.4	473.7
	50%	488.5	489.0	489.0	489.0	488.9	489.0	488.5
	60%	493.4	493.4	493.4	493.4	493.4	493.4	493.3
	70%	496.0	496.0	496.0	496.0	496.0	496.0	495.9
	80%	497.2	497.2	497.2	497.2	497.2	497.2	497.1
	90%	498.4	498.4	498.4	498.4	498.4	498.4	498.4
1000 Proteins	20%	668.0	699.6	698.7	698.0	688.2	695.8	694.9
	30%	840.2	849.2	849.0	849.0	846.9	848.6	842.2
	40%	917.4	922.0	921.9	921.9	920.6	921.6	919.7
	50%	958.4	960.3	960.3	960.3	960.2	960.3	958.5
	60%	976.6	977.0	977.0	977.0	977.0	977.0	976.2
	70%	985.5	985.6	985.6	985.6	985.6	985.6	985.4
	80%	990.4	990.6	990.6	990.6	990.6	990.6	990.4
	90%	994.8	994.9	994.9	994.9	994.9	994.9	994.8
2000 Proteins	20%	1158.2	1240.1	1236.2	1234.9	1210.1	1229.9	1224.7
	30%	1544.7	1575.5	1574.8	1574.6	1566.2	1573.0	1559.4
	40%	1754.6	1768.7	1768.2	1768.4	1765.6	1767.9	1758.8
	50%	1864.1	1870.1	1870.1	1870.1	1869.3	1870.1	1865.7
	60%	1920.1	1922.1	1922.1	1922.1	1922.0	1922.1	1918.6
	70%	1947.9	1948.9	1948.9	1948.9	1948.8	1948.9	1947.6
	80%	1980.6	1980.9	1980.9	1980.9	1980.9	1980.9	1967.0
	90%	1980.6	1981.0	1981.0	1981.0	1981.0	1981.0	1980.5
5000 Proteins	20%	2284.0	2520.0	2504.7	2503.3	2439.7	2491.8	1592.8
	30%	3293.0	3423.1	3419.2	3417.7	3380.8	3410.7	3189.4
	40%	3997.3	4052.1	4050.5	4050.6	4040.9	4048.5	4012.4
	50%	4385.4	4417.8	4417.8	4417.5	4413.0	4416.9	4397.9
	60%	4622.7	4634.1	4634.0	4634.0	4632.7	4633.7	4619.4
	70%	4756.5	4762.5	4762.5	4762.5	4762.2	4762.4	4754.2
	80%	4905.6	4908.5	4908.5	4908.5	4908.4	4908.5	4840.0
	90%	4905.8	4908.7	4908.7	4908.6	4908.6	4908.7	4904.6

Table 1: Human proteome subset results (nonredundant dataset sizes). Mean number of proteins in the non-redundant datasets generated by each algorithm from fifty datasets of 500, 1000, 2000 and 5000 proteins. NC is NeighbourCull and BC is BlastCuller.

Dataset Size	PISCES	Leaf	FIS	NeighbourCull	GLP	VSA	BlastCuller
500	0.06	0.00	0.01	0.01	120.00	0.01	0.00
1000	0.15	0.01	0.08	0.04	120.00	0.07	0.03
2000	0.51	0.07	0.60	0.25	120.00	0.49	0.19
5000	2.86	1.82	10.19	3.03	120.00	7.18	2.66

Table 2: Human proteome subset results (execution time). Mean execution time (in seconds) for datasets of 500, 1000, 2000 and 5000 proteins using a threshold of 20%.

3.2.1.1 GLP

For datasets of 500, 1000 and 2000 proteins, GLP performs worse than all alternatives to PISCES, except VSA at a 20% threshold, and its improvement over PISCES deteriorates more rapidly than that of the other algorithms as the threshold increases. Once the threshold reaches 60%, PISCES performs at least as well as, and often better than, GLP. Datasets of 5000 proteins show a slight difference, as GLP performs worse than all algorithms, including PISCES, except at 40% and 50% thresholds where it performs marginally better than PISCES, but still worse than the other alternatives.

The GLP execution time limit of the longer of two minutes or ten times the execution time of Leaf clearly curtailed its capabilities. In order to determine the size of the effect that the time limit had on GLP's performance, ten datasets of 5000 proteins were randomly chosen to be tested with an extended time limit of 500 times the execution time of Leaf. The results of this can be seen in Table 3. For each threshold, GLP took approximately 2000 seconds, again hitting its execution time limit of 500 times the Leaf algorithm. The size of the non-redundant datasets found by GLP at thresholds of 20% and 30% were substantially improved by the increased time limit, by 355.1 and 117.3 proteins respectively. However, GLP was still only able to improve on the size of the non-redundant dataset found by PISCES at thresholds of 30%, 40% and 50%, despite taking roughly 500 times as long as it.

Threshold	PISCES	Leaf	GLP
20%	2271.3	2510.8	1947.9
30%	3276.4	3412.0	3306.7
40%	3999.0	4051.2	4010.8
50%	4387.9	4416.3	4394.0
60%	4625.0	4635.0	4621.4
70%	4757.1	4762.4	4754.9
80%	4841.7	4844.7	4839.7
90%	4906.5	4909.2	4904.8

Table 3: Human proteome subset results (GLP evaluation). Mean number of proteins in the non-redundant dataset generated by PISCES, Leaf and GLP from ten datasets of 5000 proteins. GLP was given an execution time limit of 500 times the execution time of Leaf.

3.2.2 Comparisons to the Maximum Independent Set

Despite running Cliquer using a Condor (Thain et al. 2005) distributed computing pool, it was only possible to find the MIS for datasets of 500 and 1000 proteins. Jobs submitted to the pool run mainly on inactive, recent-model desktop machines in student computing clusters and, during the academic term, get around 8–10 hours of uninterrupted processor time per day. Even with the added computing power of the Condor pool, the MIS could not be found for any dataset

of 5000 proteins six months after submission to the pool. Finding a known MIS is therefore an impractical approach for removing redundancy from protein datasets, and heuristics must be used instead.

The results of the comparisons between the heuristics and the exact MIS, as found by Cliquer, can be seen in Table 4, for datasets of 500 proteins, and Table 5, for datasets of 1000 proteins. The most notable results are those of the new algorithms (Leaf, FIS and NeighbourCull), for which the mean size of the non-redundant datasets generated is never more than two proteins less than that of the exact MIS, and for all thresholds except 20% on the datasets of 1000 proteins, never more than one protein smaller. In addition to this, all non-redundant datasets generated by Leaf are the same size as or one protein smaller than the MIS, with the mean difference never being greater than 0.2 proteins.

Threshold	Cliquer	PISCES	Leaf	FIS	NeighbourCull	VSA	BlastCuller	GLP
20%	384.3	12.6	0.0	0.2	0.3	3.9	1.0	3.4
30%	445.1	2.8	0.0	0.1	0.1	0.6	0.3	2.4
40%	474.5	2.2	0.0	0.0	0.0	0.5	0.1	0.8
50%	489.0	0.4	0.0	0.0	0.0	0.1	0.0	0.5
60%	493.4	0.0	0.0	0.0	0.0	0.0	0.0	0.1
70%	496.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
80%	497.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
90%	498.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4: Exact MIS comparison for fifty datasets of 500 proteins. The mean number of proteins in the non-redundant datasets generated by an exact MIS finding algorithm, Cliquer, is given. For the other algorithms, the numbers given are the mean number of proteins by which the non-redundant datasets generated are smaller than those generated by Cliquer.

Threshold	Cliquer	PISCES	Leaf	FIS	NeighbourCull	VSA	BlastCuller	GLP
20%	699.7	31.8	0.2	1.1	1.7	11.6	3.9	40.3
30%	849.2	9.1	0.0	0.2	0.2	2.3	0.6	1.7
40%	922.0	4.7	0.1	0.2	0.2	1.5	0.5	2.4
50%	960.3	2.0	0.0	0.0	0.0	0.2	0.0	1.2
60%	977.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0
70%	985.6	0.2	0.0	0.0	0.0	0.0	0.0	0.0
80%	990.6	0.1	0.0	0.0	0.0	0.0	0.0	0.0
90%	994.9	0.1	0.0	0.0	0.0	0.0	0.0	0.0

Table 5: Exact MIS comparison for fifty datasets of 1000 proteins. The mean number of proteins in the non-redundant datasets generated by an exact MIS finding algorithm, Cliquer, is given. For the other algorithms, the numbers given are the mean number of proteins by which the non-redundant datasets generated are smaller than those generated by Cliquer.

3.2.3 BHOSLIB Benchmark

The performance of the algorithms on the BHOSLIB benchmark graphs can be seen in Table 6. Unlike the protein datasets, GLP consistently outperformed the other algorithms on the BHOSLIB graphs, with FIS finding the second largest independent set for each dataset. The two best performing algorithms were therefore the approaches that used permutations in order to perform a local search. This discrepancy between the performance of the algorithms that utilise permutations and those that do not indicates that, while effective on sparse protein similarity graphs, the simple heuristics used by Leaf, NeighbourCull, VSA and BlastCuller are less effective on the dense highly connected BHOSLIB ones. It is therefore likely that either a more complex heuristic or some form of less directed search is required to find larger independent sets on denser graphs.

Dataset	Leaf	FIS	NeighbourCull	GLP	VSA	BlastCuller
frb30-15	21.8	24.2	21.2	28.0	18.4	20.4
frb35-17	26.6	29.8	25.0	32.4	21.6	24.6
frb40-19	29.0	33.4	28.2	36.8	24.8	26.6
frb45-21	31.2	36.8	29.8	40.6	26.6	29.8
frb50-23	34.8	42.0	33.6	44.2	30.0	32.0
frb53-24	34.8	44.0	33.0	47.2	30.0	33.8
frb56-25	36.8	45.2	36.2	49.8	33.0	35.2
frb59-26	39.6	47.8	37.6	52.0	33.6	37.8
frb100-40	59.0	81.0	54.0	88.0	51.0	55.0

Table 6: Comparison of the algorithms on the BHOSLIB benchmark graphs. BHOSLIB datasets are named as frb$X$-Y, indicating that the graph consists of X cliques of Y nodes each ($X * Y$ nodes in total), and the MIS consists of X nodes. The numbers for each algorithm indicate the mean number of nodes in the independent sets found.

3.2.4 Model Organisms

The number of proteins in the non-redundant datasets generated from the entire proteomes of *H. sapiens*, *M. musculus*, *E. coli*, *A. thaliana* and *S. cerevisae* can be seen in Table 7. These show the same trends as the results from the subsets of the human proteome (Section 3.2.1), with all algorithms tested outperforming PISCES for every combination of dataset and threshold. Leaf was again the best algorithm in terms of the size of the non-redundant datasets returned. GLP was not tested due to the problems highlighted by the results in Section 3.2.1.

Organism	Threshold	PISCES	Leaf	FIS	NC	VSA	BC
H. sapiens	20%	5700	6643	6572	6580	6365	6541
	30%	9007	9856	9796	9800	9594	9762
	40%	12422	12843	12832	12829	12746	12811
	50%	14927	15169	15167	15164	15129	15154
	60%	16771	16887	16884	16886	16874	16884
	70%	17969	18036	18036	18036	18030	18033
	80%	18763	18801	18801	18801	18798	18801
	90%	19366	19389	19388	19389	19388	19388
M. musculus	20%	4875	5677	5622	5617	5393	5572
	30%	8001	8618	8587	8580	8421	8552
	40%	10988	11256	11250	11252	11185	11237
	50%	12997	13097	13096	13094	13079	13089
	60%	14325	14379	14379	14379	14373	14378
	70%	15196	15216	15216	15216	15214	15216
	80%	15737	15743	15743	15743	15742	15742
	90%	16066	16068	16068	16068	16068	16068
E. coli	20%	2610	2712	2708	2707	2686	2703
	30%	3364	3416	3415	3414	3403	3410
	40%	3863	3878	3878	3878	3878	3878
	50%	4047	4049	4049	4049	4049	4049
	60%	4128	4129	4129	4129	4129	4129
	70%	4175	4178	4178	4178	4178	4178
	80%	4213	4214	4214	4214	4214	4214
	90%	4233	4234	4234	4234	4234	4234
A. thaliana	20%	2181	2417	2394	2397	2333	2382
	30%	3386	3649	3634	3635	3563	3619
	40%	4910	5105	5099	5093	5042	5082
	50%	6288	6431	6424	6428	6409	6421
	60%	7547	7634	7634	7633	7621	7631
	70%	8587	8629	8628	8629	8629	8629
	80%	9569	9591	9591	9591	9589	9591
	90%	10359	10363	10363	10363	10363	10363
S. cerevisiae	20%	3913	4194	4188	4177	4126	4163
	30%	4959	5062	5062	5061	5043	5057
	40%	5559	5603	5603	5603	5603	5602
	50%	5837	5863	5863	5863	5863	5863
	60%	6005	6026	6026	6026	6026	6026
	70%	6121	6136	6135	6136	6136	6136
	80%	6199	6210	6210	6210	6210	6210
	90%	6271	6278	6277	6278	6278	6278

Table 7: Number of proteins in the non-redundant datasets generated from entire proteomes. NC is NeighbourCull and BC is BlastCuller.

4 Feature-Based Druggability Prediction - Target Subdivisions and Machine Learning

4.1 Target Types Investigated

In order to be useful for a druggability study, a subdivision of the human proteome needs to be pharmacologically interesting, and contain a sufficient number of proteins with which to train a machine learning classifier. Subdivisions based on protein family membership can meet both these criteria, provided that the family is sufficiently large and contains a significant number of drug targets. The four families investigated here, GPCRs, ion channels, kinases and proteases, are well represented in the set of all drug targets, and contain a large enough number of proteins to provide a suitably sized non-target set. While protein family based subdivisions are straightforward, potentially more interesting subdivision are those based on involvement in disease. However, as both a set of targets and non-targets is needed, it must be possible to causatively link proteins with the disease, without them being used as a target. For most diseases, the data needed for this is unavailable or insufficient in quantity and/or quality. One prominent exception is that of cancer. Thanks to the large body of work on the causes of cancer, it is possible to causatively associate proteins with cancer, irrespective of whether they are the target of an approved drug. Additionally, the large number of approved antineoplastic drugs means that there are a large number of proteins known to be the target of a drug intended to treat cancer. The dataset will therefore contain sufficient numbers of proteins, both targets and non-targets, to train a classifier. The remainder of this section investigates the subdivisions used, the characteristics of the proteins in them and methods by which drugs target them.

4.1.1 Antineoplastic

The term cancer refers to a diverse collection of malignant genetic diseases that, through a series of genetic and epigenetic changes, enables populations of somatic cells to escape the normal homeostatic restraints preventing aberrant growth and replication (Harley 2008; Lowe et al. 2004). The resultant phenotypic changes are predominantly brought about through mutations that activate/amplify the expression of oncogenes or inhibit/reduce the expression of tumour suppressors, thereby removing many of the shackles placed on cellular replication following an organism's development (Bertino et al. 2002; Luo et al. 2009). The deregulation of signal transduction pathways caused by these mutations underlies the four distinguishing features of cancer: unregulated growth, immortalisation, sustained angiogenesis and invasion/metastasis (Blume-Jensen & Hunter 2001; Hanahan & Weinberg 2000).

The proliferation of normal somatic cells is controlled by a network of counterbalancing extracellular positive and negative growth signals, the presence of which activates internal proliferative or apoptotic pathways by binding to cell surface receptors. However, the genetic changes in cancerous cells interfere with this balance, by causing the cells to become capable of sustained proliferative signalling and/or resisting apoptosis, thereby tipping the balance in favour of proliferation and away from apoptosis (Blume-Jensen & Hunter 2001; Hanahan & Weinberg 2000; Luo et al. 2009). One possible method by which a cancerous cell can achieve this is autocrine signalling, whereby the cell produces growth factors to which it can respond via expression of associated cell surface receptors, e.g. the production of platelet-derived growth factor by glioblastomas (Nazarenko et al. 2012). Alternatively, the cell can alter its expression of the growth factor receptors, either by overexpressing them, thus rendering the cell hyper-responsive, or by making them capable of ligand-independent activation through structural alterations (Evan & Vousden 2001; Hanahan & Weinberg 2011; Leber & Efferth 2009). Growth factor independent proliferation can also be induced via the activation/inactivation of signalling network components downstream of the growth factor receptors, thereby obviating the need for growth factor mediated activation of the receptors (Hanahan & Weinberg 2011). Due to these growth stimulatory alterations, collections of cancerous cells will naturally encounter pro-apoptotic conditions such as hypoxia, telomere shortening and DNA damage as they proliferate (Lowe 2000). In order to circumvent the activation of apoptosis, and subsequent reduction in proliferative potential, cancerous cells develop mechanisms that enable them to resist the pro-apoptotic nature of these conditions (Hanahan & Weinberg 2000; Leber & Efferth 2009), the most common of which is the loss of the pro-apoptotic p53 tumour suppressor gene through mutation (Evan & Vousden 2001; Lowe 2000).

The decoupling of growth from environmental signals enables a cancerous cell to proliferate at will, but is insufficient for the development of a macroscopic tumour (Hanahan & Weinberg 2011). This is because the progressive shortening of telomeres during each cell cycle introduces a finite capacity for division, the Hayflick limit, that constrains the number of mitoses an individual cell can undergo. Sufficiently shortened telomeres will lead to the cell entering a state of replicative senescence, resulting in permanent cell cycle arrest and prevention of proliferation (Artandi & DePinho 2010). The cell can be rescued from this state, for example by inactivating p53 and the Retinoblastoma tumour suppressor protein (Shay et al. 1991), thereby enabling it to continue dividing until it reaches a stage known as crisis, characterised by severe chromosomal instability, apoptosis and the rare emergence of an immortalised cell with unlimited proliferative potential (Hahn & Meyerson 2001). Achieving and maintaining this immortalised state requires that a cell be able to lengthen its telomeres, in order to ensure that they remain at a length above that at which crisis would occur (Hanahan & Weinberg 2000). In the vast majority

of immortalised cells this is achieved by re-acquiring the ability to express telomerase, the specialised reverse transcriptase that adds DNA sequence repeats to the ends of telomeric DNA, or through an alternative recombinant based mechanism (Hanahan & Weinberg 2011).

The combination of deregulated growth and immortalisation serves to enable an individual cell to proliferate independently and indefinitely, provided that a steady source of oxygen and nutrients, along with the removal of metabolic waste products, can be maintained. This necessitates that the cell, like all cells in a tissue, reside within a couple hundred micrometres of the existing microvasculature, as this is the maximum distance over which effusion can occur (Carmeliet & Jain 2000; Chang & Werb 2001). However, unlike normal tissues, the proliferative capacity of a mass of cancerous cells enables it to grow beyond this point, thereby leading to portions of it becoming hypoxic and nutrient deprived. In order to prevent the apoptosis and necrosis to which this can lead, a tumour will induce the formation of new blood vessels, thereby enabling it to circumvent the vascular restrictions placed on its growth (Carmeliet & Jain 2000; Nussenbaum & Herman 2010). In order to achieve this, the tumour must tilt the natural balance of pro- and anti-angiogenic molecules in favour of pro-angiogenesis, through changing physiological stimuli (e.g. hypoxia and nutrient deprivation), genetic alterations concomitant with cancer or the secretion of growth factors and cytokines (Baeriswyl & Christofori 2009; Bergers & Benjamin 2003; Faivre et al. 2007). The resultant angiogenesis will lead to the recruitment of new blood vessels to the tumour, either by sprouting from existing vessels or intussusception, the splitting of an existing blood vessel in two (Carmeliet & Jain 2000; Chang & Werb 2001).

Having co-opted its host's vascular system to its own ends, the tumour is now able to grow to macroscopic size. However, before it can be considered cancerous, it must first be capable of establishing a secondary tumour in a tissue unconnected to that of the primary one, a process known as metastasis (Leber & Efferth 2009). In order to successfully metastasise, a tumour must spawn a pioneer cell that can enter the vascular system of its host (intravasation), circulate through the bloodstream to a distant site (circulation), exit the vascular system (extravasation) and proliferate in the foreign microenvironment (colonisation) (Chaffer & Weinberg 2011; Leber & Efferth 2009). Depending on the distance between the pioneer cell and nearby blood vessels, the cell may have to gain the ability to invade and migrate through surrounding tissue before it can reach the vessels and intravasate. This process is facilitated by the activation and secretion of lytic enzymes, such as matrix metalloproteases, that degrade the extracellular matrix, along with the tumour's recruitment of new blood vessels, which decreases the distance over which the pioneer cell needs to migrate (Baeriswyl & Christofori 2009; Chaffer & Weinberg 2011; Leber & Efferth 2009). Upon reaching a vessel, the pioneer cell will proteolytically degrade the vessel's endothelium, thereby gaining access to its lumen, following which it can

enter the vessel and circulate around the body in the bloodstream (Leber & Efferth 2009). Eventually the cell will extravasate, either through a similar process as intravasation, or by proliferating in the lumen of the vessel until the mass of cells causes the vessel's wall to break (Fidler 2003; Leber & Efferth 2009). This is followed by colonisation and proliferation in the new tissue, and eventually the growth of a secondary tumour and spawning of additional metastases.

Classical pharmacological treatments for cancer involve the administration of cytotoxic drugs that seek to inhibit the proliferation of cancerous cells and reduce tumour size by interfering with DNA replication (Evan & Vousden 2001). These drugs are therefore most effective against rapidly proliferating cells, such as cancerous ones, due to their greater rate of DNA replication. However, they are also harmful to cells that divide rapidly under normal circumstances, for example those in the bone marrow and digestive tract, resulting in side effects such as alopecia and vomiting (Gordon et al. 2008). The two most common classes of cytotoxic drugs are alkylating agents and antimetabolites. Alkylating agents can be used to prevent cell division by cross-linking DNA strands, leading to their breakage and ultimately cell death. In contrast, most anti-metabolites are structural analogues of naturally occurring metabolites of DNA/RNA synthesis, and interfere with this process by either competing with the normal metabolites for a catalytic or regulatory site of a key enzyme, or by becoming incorporated into the DNA/RNA molecule and preventing its normal use. Additional methods by which cytotoxic drugs hinder DNA replication include: directly interfering with mitosis, damaging DNA through the production of reactive oxygen species and inhibiting topoisomerases.

As an alternative to the cytotoxic approach to antineoplastic pharmaceuticals, targeted therapies seek to modulate the activity of specific molecular targets that are believed to have a critical role in tumour growth and/or cancer progression. While these targets may be present in non-cancerous cells, they are often overexpressed or altered in cancerous cells, thereby giving targeted therapies increased selectivity and reduced toxicity over conventional cytotoxic treatments (Gerber 2008; Shawver et al. 2002). By targeting specific proteins, rather than indiscriminately killing proliferating cells, targeted therapies can be used to interfere with specific aspects of cancer progression. For example, the immortalisation of cancer cells could be attacked via the targeting of telomerase, as it is both specific to cancerous cells and necessary for their survival (Harley 2008); molecular alterations that deregulate growth can be corrected, as with Imatinib's targeting of the BCR-ABL protein in chronic myelogenous leukaemia; or the tumour's blood supply can be cut off by preventing angiogenesis, as done by the drug Bevacizumab's inhibition of vascular endothelial growth factor A (Ferrara & Kerbel 2005). Due to their importance in modulating growth factors, tyrosine kinases are an especially useful group of targets (Arora & Scholar 2005), with drugs such as Imatinib, Gefitinib, Erlotinib and Sunitinib

targeting them. Other important targets include growth factors and proteasomes, inhibition of which can potentially slow a tumour's proliferation by inhibiting growth/angiogenesis or increasing apoptosis respectively.

4.1.2 GPCRs

The GPCRs are a family of eukaryotic transmembrane receptors relied on by cells to transduce various extracellular stimuli into intracellular secondary messengers and, ultimately, cellular responses (Ferguson 2001). They convey the majority of signals across the cell membrane, and are activated by a wide range of stimuli, including ions, peptides, lipids, odorants and light (Filmore 2004; Fredriksson et al. 2003; Millar & Newton 2010). Due to this, GPCRs are involved in most physiological processes (e.g. smelling, tasting, muscle contraction and immune responses), and their dysfunction has been implicated in multiple human diseases (Dorsam & Gutkind 2007).

GPCRs are membrane-bound proteins consisting of an extracellular N-terminus, seven α-helical transmembrane domains (connected by three extracellular and three intracellular loops) and an intracellular C-terminus (Figure 10). The transmembrane regions arrange themselves into a tertiary barrel-like structure surrounding a ligand binding site, although ligands may also bind elsewhere, such as the extracellular loops (Congreve & Marshall 2010). Each GPCR is also associated with a heterotrimeric G protein, consisting of a G_α, G_β and G_γ subunit, that is responsible for much of its signalling capability. As each G protein subunit is encoded by multiple genes in the human genome, the diversity in GPCR signalling can partly be explained by the multiple permutations of different G_α, G_β and G_γ subunits combining to produce a variety of G proteins, although not all interactions between subunits are favourable (Kristiansen 2004).

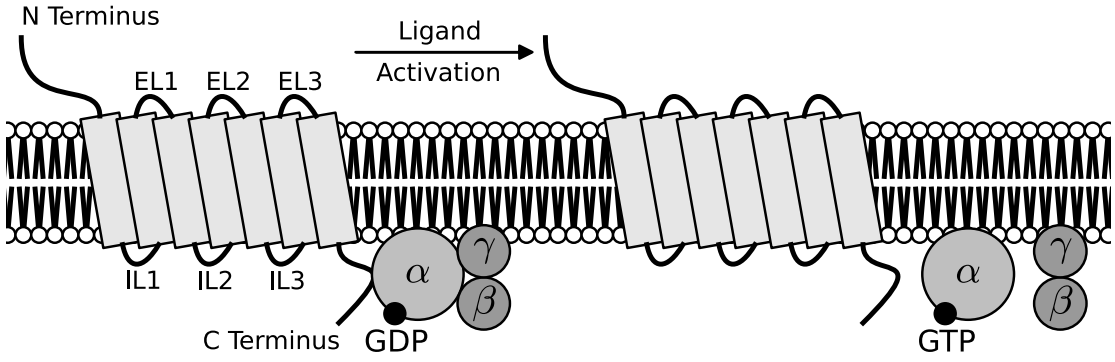

Figure 10: Diagrammatic representation of the activation of a GPCR. The extracellular N-terminus, seven α-helical transmembrane domains, three extracellular loops (EL1-3), three intracellular loops (IL1-3) and the intracellular C-terminus that make up the GPCR can be seen. On the left, a G protein, consisting of G_α, G_β and G_γ subunits, can be seen bound to an inactive GPCR. Following the activation of the GPCR, GDP dissociates from the G_α subunit, and is quickly replaced by GTP. This activates the G_α subunit and causes it to dissociate from both the $G_{\beta\gamma}$ subunit and the GPCR.

Prior to activation, the G_α subunit binds to GDP, a membrane bound $G_{\beta\gamma}$ complex and an inactive GPCR (Kristiansen 2004). In this state stimulation by an agonist will activate the GPCR, causing it to undergo a conformation change that enables it to act as a guanine nucleotide exchange factor (Dorsam & Gutkind 2007; Pitcher et al. 1998). The activated GPCR can then interact with its bound G protein, promoting the dissociation of GDP from the G_α subunit (Pierce et al. 2002). Under normal physiological conditions, GTP will immediately replace the dissociated GDP, thereby activating the G_α subunit and causing it to dissociate from both the $G_{\beta\gamma}$ subunit and the GPCR (Dorsam & Gutkind 2007; Kristiansen 2004; Pierce et al. 2002). Although both the G_α subunit bound to GTP and the $G_{\beta\gamma}$ complex remain tethered to the cell membrane, they can diffuse laterally in order to stimulate various downstream effector molecules and membrane proteins (Dorsam & Gutkind 2007; Ferguson 2001). This active signalling state of the G protein can persist until the G_α's bound GTP is hydrolysed to GDP, at which point the G protein can reassemble and associate with an inactive GPCR (Kristiansen 2004).

In addition to activating G protein mediated signalling cascades, agonist binding can also activate regulatory feedback mechanisms for modulating a GPCR's sensitivity to further stimulation (Ferguson 2001), often resulting in a rapid attenuation of its responsiveness to the agonist (Pitcher et al. 1998). This dampening of the GPCR's signalling capabilities can be attributed to the interplay of four distinct processes: receptor desensitisation, sequestration, downregulation and resensitisation. Desensitisation inhibits GPCR-mediated signal transduction by changing the functional activity of GPCRs, usually via the phosphorylation of the GPCR and subsequent decoupling of it from its G protein (Ferguson 2001; Krupnick & Benovic 1998). This process can occur as a result of the GPCR's own activation (homologous desensitisation) or the activation of another GPCR (heterologous desensitisation) (Luttrell & Lefkowitz 2002). For the majority of GPCRs, homologous desensitisation is a two-step process involving selective phosphorylation by a G protein-coupled receptor kinase, which on its own has little desensitising effect for many GPCRs (Magalhaes et al. 2012), followed by arrestin binding, which promotes both the physical uncoupling of the GPCR from the G protein and GPCR sequestration (Ferguson 2001; Krupnick & Benovic 1998; Luttrell & Lefkowitz 2002). In contrast to homologous desensitisation, phosphorylation during heterologous desensitisation directly alters the conformation of the GPCR, and is therefore capable of decoupling the GPCR and G protein in the absence of arrestins (Lefkowitz 1998; Luttrell & Lefkowitz 2002).

A further level of GPCR regulation stems from their agonist-mediated sequestration in intracellular compartments (Ferguson 2001; Krupnick & Benovic 1998; Luttrell & Lefkowitz 2002). Following homologous desensitisation, the arrestin bound to the GPCR can target the receptor to clathrin coated pits in the cell membrane, enabling the receptor and part of the membrane it is

embedded in to be brought inside the cell (Lane et al. 2013; Luttrell & Lefkowitz 2002; Magalhaes et al. 2012). This sequestration of GPCRs is necessary for receptor downregulation in response to long term, hours or days, agonist exposure, and for maintaining the cell's ability to respond to external stimuli through resensitisation (Ferguson 2001; Luttrell & Lefkowitz 2002). During long term agonist exposure, the number of GPCRs present on the surface of a cell decreases through a process known as downregulation, resulting in a marked decrease in the cell's ability to respond to specific stimuli. This process is at least partially controlled via transcriptional and post-transcriptional means, but also through degradation of sequestered receptors (Luttrell & Lefkowitz 2002; Pitcher et al. 1998). Recovery of full sensitivity following downregulation is a slow process involving the biosynthesis of new receptors, as both the degradation and decreased biosynthesis of receptors during downregulation must be compensated for (Pitcher et al. 1998).

Degradation as part of the downregulation process is not the only possible fate of a sequestered GPCR. Rather, some receptors undergo a process of resensitisation, whereby the GPCR is made active, receptive to agonist stimulation, and is recycled back to the cell membrane (Ferguson 2001; Lane et al. 2013). This requires that the GPCR be placed back in its pre-ligand-bound state by releasing the bound arrestin, dissociating from its ligand and being dephosphorylated (Ferguson 2001; Luttrell & Lefkowitz 2002). The resensitisation pathway is essential for maintaining GPCR responsiveness to external stimuli, as without it desensitisation and downregulation would leave the cell unable to transduce external signals (Ferguson 2001).

The prominent role that GPCRs play in many physiological processes, in addition to the role that their dysfunction plays in many diseases, means that GPCRs make up a significant fraction of the targets of approved drugs (Dorsam & Gutkind 2007; Lane et al. 2013). One approach to modulating the activity of a GPCR pharmacologically is to develop a drug that competes with the receptor's endogenous ligand for access to its orthosteric site. However, in order for a drug to effectively modulate a GPCR's activity in this manner it must out-compete its endogenous ligand, which necessitates that the drug have a high affinity for the specific GPCR and be maintained at a sufficiently high concentration (May et al. 2007). Consequently, orthosteric drugs may have problems with toxicity, desensitisation and downregulation, in addition to selectivity issues caused by the highly conserved nature of the endogenous ligand binding site amongst some GPCR subfamilies (Conn et al. 2009; May et al. 2007). Rather than compete for the orthosteric site, a drug can modulate the GPCR's activity allosterically by binding to a location topographically distinct from the endogenous ligand's binding site. These allosteric modulators can benefit not only from the increased selectivity due to the often less conserved nature of their binding sites, but also from the fact that the endogenous ligand can still bind to the orthosteric site (Cavanaugh et al. 2012; Christopoulos 2002). This is beneficial, as the conformational change

induced in the GPCR by the allosteric modulator represents a distinct new structure that can react to the presence of the endogenous ligand in a different manner to the unmodulated receptor (May et al. 2007). Therefore, in addition to possessing its own intrinsic efficacy, an allosteric modulator can alter the affinity, by changing association/dissociation rates, and/or the efficacy, by altering intracellular responses, of the endogenous ligand (Conn et al. 2009; May et al. 2007). Its physiological effects can therefore be restricted both spatially and temporally to instances where the endogenous ligand is also present, whereas orthosteric drugs will invariably modify a receptor's activity whenever they are present (Christopoulos 2002).

4.1.3 Ion Channels

Ion channels are intrinsic membrane proteins that regulate the passage of ions across the membranes of cells and their organelles. They function as gated pores that open and close in response to various stimuli, thereby enabling the diffusion of ions down an electrochemical potential gradient. In order to facilitate this, ion channels form an aqueous pore lined with hydrophilic amino acids, which enable an ion's charge to be stabilised as it passes through the channel. This transport of ions across cell and organelle membranes is essential for many of life's processes, such as regulating cell volume and electrical signalling, making ion channels a vital component in many fundamental physiological processes (Hille 2001).

Ion channels are typically oligomeric complexes consisting of several tightly packed protein subunits embedded in the membrane and arranged such that they enclose a water filled transmembrane pore (Figure 11). In terms of facilitating the passage of ions through this pore, the most important aspects of an ion channel are its gating and its selective permeability (Jiang et al. 2002; Perozo 2002). In order to prevent the uncontrolled influx/efflux of ions, and thereby maintain the electrical potential across the membrane, ion channels need a method of selectively blocking the flow of ions through them. This is achieved through the use of a gate that physically blocks the passage of ions, and a sensor that controls the opening and closing of the gate in response to a specific stimulus (Gadsby 2009). Gating then refers to an ion channels ability to transfer between open (conducting) and closed (non-conducting) states in response to the gating stimulus. In the absence of the stimulus, the channel will be in a closed, or resting, state and impermeable to ions. Upon recognition of a specific gating stimulus, the channel will transition to an open state by undergoing conformation changes that enable ions to pass through it (Perozo 2002). If the gating stimulus then persists for too long, the channel will become desensitised and unresponsive to further stimuli until it is reactivated (Sun et al. 2002). In order to enable channels to respond to a range of stimuli, such as changes in transmembrane potential, ligand binding and

pH (Gabashvili et al. 2007), different channels contain different sensors, with ligand and voltage-gated channels the most common.

The selective permeability of an ion channel arises from interactions between the permeating ions and the channel's selectivity filter, the diameter of which is narrow enough to require that the ions passing through it be partially dehydrated (Hille 1978). Selective permeability therefore refers to the fact that certain ionic species have less difficulty passing through a channel than others (Keramidas et al. 2004). Part of this selectivity arises from the diameter of the pore itself, as for an ion to pass through a pore its diameter must be smaller than that of the pore. However, if this were the only method of determining the permeability, then potassium channels would also be permeable to sodium ions, as well as to anions smaller than a potassium ion. Therefore, in addition to the pore size, the solvation energy of ions in water and channel binding energies are also exploited (Armstrong & Hille 1998; Gouaux & Mackinnon 2005). As mentioned previously, the narrowness of the selectivity filter requires that an ion be dehydrated in order to pass through. As this is energetically unfavourable, the pores of ion channels are lined with sites to which a diffusing ion can bind in order to compensate for the energetic cost of dehydration (Gouaux & Mackinnon 2005). By requiring ions to be dehydrated before they can pass through, an ion channel can prevent the passage of smaller ions, as the channel binding energy will not be sufficient to overcome the solvation energy (Gouaux & Mackinnon 2005). Therefore, the particular type of ion that a channel selects for is determined by both the physical width of the pore and the amino acids lining its interior.

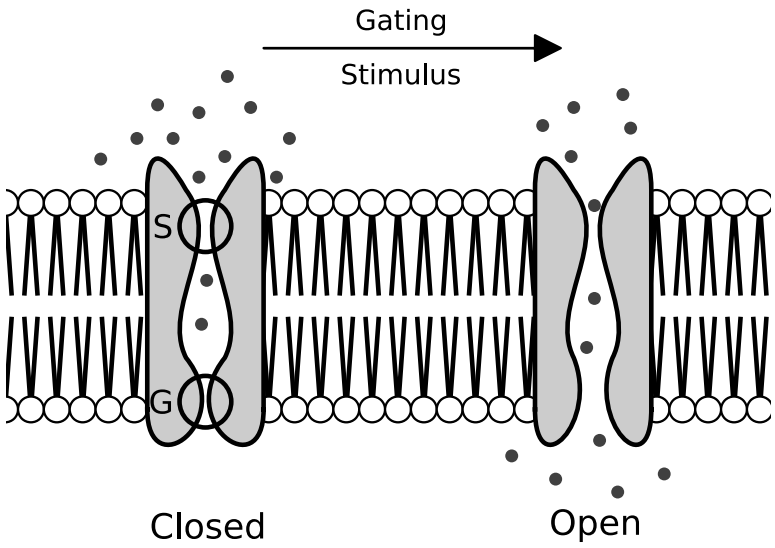

Figure 11: Diagrammatic representation of the activation of an ion channel. When in a closed conformation, an ion channel prevents the passage of ions. Upon detecting a gating stimulus, the channel undergoes a conformational change that makes it selectively permeable to a specific set of ions. The selectivity filter (S in the diagram) controls the selectivity of the channel for specific ions, while the gate (G in the diagram) controls the passage of the selected ions.

Ion channels are popular targets for pharmacological intervention due to their key roles in human physiology, localisation in the membrane and pattern of distribution throughout the body (Ashcroft 2006; Bagal et al. 2013). The drugs that target them alter their permeability by changing the probability that the channel will be in a given state, often by preferentially binding to and stabilising a particular channel conformation (Andersen 2008). The resultant modulation then depends on the ratios of a drug's affinities for, and the temporal distribution of, the different channel conformations.

Pharmacological modulation of ion channels is generally achieved by interacting with the channel's pore or altering its gating (Dilly et al. 2011). Pore modulators are primarily inhibitors that exert their effect by binding to the pore and physically or electrostatically blocking the flow of ions (Dilly et al. 2011; McNulty et al. 2007), predominantly by occluding the pore or stabilising a closed or inactive state of the channel. However, occlusion of the pore or stabilisation of a closed state will prevent normal channel function, as the drug's presence prevents the passage of ions through the channel under all circumstances. In contrast, stabilising an inactive state prevents conductance indirectly by delaying the reactivation of the channel, thereby providing selective inhibition during periods of continuous or repetitious stimulus without affecting normal channel function (Clare et al. 2000). Rather than interact with the pore, gating modulators bind to the channel and change the kinetics of the gating process (Dilly et al. 2011). They are therefore allosteric in nature, and can be designed to enhance the normal conductance of a channel, either positively or negatively, or exert their effect independently of the channel's gating stimulus (Börjesson et al. 2010). However, by far the majority of approved ion channel modulators interact with a channel's pore, rather than through its gating.

4.1.4 Kinases

A kinase is an enzyme that catalyses the transfer of a phosphate group from a high energy donor molecule, such as ATP, to specific substrate molecules, in a process known as phosphorylation. Although the phosphate group can be transferred to a range of organic molecules, the vast majority of eukaryotic kinases are protein kinases, and are therefore responsible for the phosphorylation of another protein. As this usually results in a functional change in the phosphorylated protein (e.g. alteration of its activity, localisation or interactions with other proteins), the control of the spatial and temporal phosphorylation of substrate proteins means that kinase activity plays a key role in many cellular processes, such as cell cycle progression, apoptosis, differentiation and signal transduction (Manning et al. 2002).

Eukaryotic protein kinases are related by a homologous catalytic domain of approximately 250-300 amino acids (Hanks & Hunter 1995), and can be grouped into the serine/threonine and tyrosine kinases, which are responsible for phosphorylating the hydroxyl oxygen of their respective amino acids (Shugar et al. 2005). The conserved core of all protein kinases is composed of two lobes, a smaller N-terminal and larger C-terminal one, linked by a hinge segment (Figure 12) (Ubersax & Ferrell 2007). ATP is bound in a cleft formed by the two lobes, with its adenine ring sequestered in a hydrophobic pocket formed by the N-terminal lobe and its triphosphate group exposed to the substrate (Kornev et al. 2006; Ubersax & Ferrell 2007). The protein substrate binds along the cleft, and the phosphorylation of its serine, threonine or tyrosine residue is catalysed by a set of conserved catalytic domain residues (Ubersax & Ferrell 2007). Following phosphorylation, the substrate and ADP are released from the kinase. Despite the conserved nature of the catalytic domain, kinases are capable of binding to and phosphorylating different protein substrates. This selectivity for specific substrates, along with mechanisms for controlling kinase activity, is vital for ensuring that only a desired subset of proteins is phosphorylated at any given time.

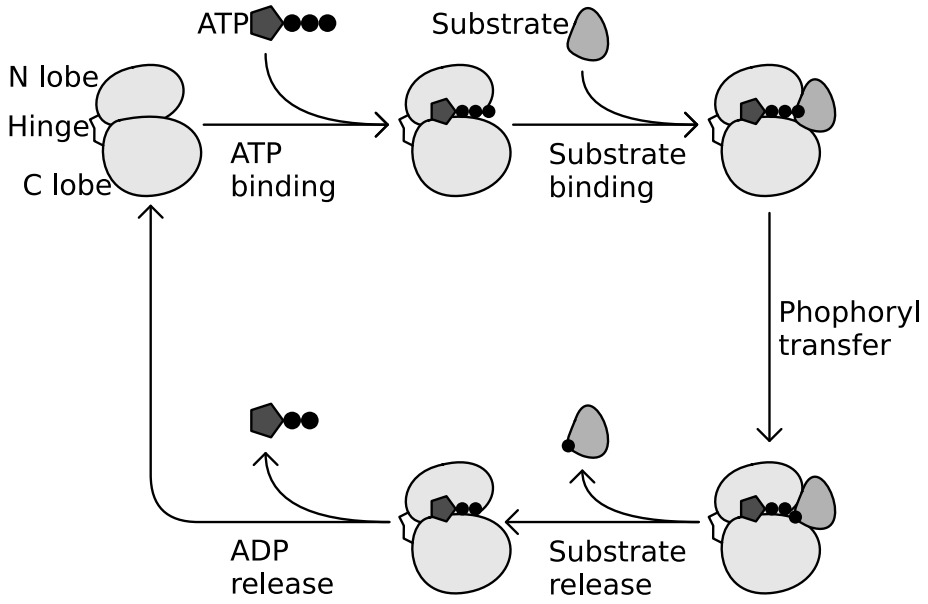

Figure 12: Diagrammatic representation of a kinase phosphorylating a substrate. Eukaryotic protein kinases consist of a smaller N-terminal and larger C-terminal lobe, linked by a hinge segment. ATP binds in a cleft formed by the two lobes, and the protein substrate binds along this cleft. Following the phosphorylation of the substrate, both the substrate and ADP are released from the kinase.

Individual kinases are responsible for the phosphorylation of anywhere between one and a few hundred residues, potentially spread across multiple protein substrates, and the majority must therefore be able to recognise general structural features of their substrates (Pinna & Ruzzene 1996; Ubersax & Ferrell 2007). Multiple mechanisms have evolved to provide the

selectivity needed to enable this promiscuity, including the recognition of a consensus sequence around the phosphorylation target, distal interactions between substrate and kinase and the controlled localisation of the kinase (Ubersax & Ferrell 2007). The active site of a kinase interacts with its substrate via a consensus sequence of amino acids situated approximately four residues or less, in both the N- and C-terminal directions, from the site of phosphorylation (Adams 2001; Ubersax & Ferrell 2007). The importance of this consensus sequence in kinase-substrate recognition is evident from the fact that the minimal substrate for most kinases is a peptide, rather than the single amino acid being phosphorylated (Ubersax & Ferrell 2007). In order to enhance selectivity beyond what is capable using purely active site sequence preferences, many kinases undergo distal interactions with their substrates, thereby separating catalytic function from substrate recognition. This is often achieved by the kinase possessing a dedicated docking groove and/or modular protein-protein interaction domain, and the substrate containing one or more complementary motifs for interacting with them (Reményi et al. 2006). In addition to the selectivity enforced by the interactions between kinase and substrate, the localisation of the kinase can also be used to enhance its selectivity by limiting the number of different substrates to those with which the kinase can come into contact (Ubersax & Ferrell 2007).

Due to the pivotal role of kinases in the regulation of many cellular processes, aberrant kinase activity has been associated with a variety of diseases and the majority of human cancers (Bertino et al. 2002; Engh & Bossemeyer 2002). Pharmacological interventions targeting kinases have historically been focussed on the inhibition of malfunctioning kinases, and therefore on preventing irregular kinase activity rather than promoting or enhancing normal activity (Engh & Bossemeyer 2002). These inhibitors can be classified based on the state of the kinase they target, active or inactive, and whether they bind to the active site, an allosteric site or both. The majority of kinase inhibitors developed to date compete directly with ATP for its binding pocket (Garuti et al. 2010; Liu & Gray 2006). These type I inhibitors form reversible non-covalent bonds with the hinge region of the (usually) active form of the kinase, mimicking those formed by the adenine ring of ATP (Gavrin & Saiah 2013). Although the successful development of type I inhibitors indicates that sufficient selectivity can be achieved in this manner, the conservation of the ATP binding pocket amongst the protein kinases can make this difficult (Garuti et al. 2010), as does the level of potency required in order to compete with the high intracellular levels of ATP present *in vivo* (Cohen 2002).

In order to avoid the aberrant effects of uncontrolled phosphorylation, most protein kinases are kept in an inactive state, and must be activated in order to perform their enzymatic function (Engh & Bossemeyer 2002). Commonly this inactivation is achieved through autoinhibition, whereby a domain of the kinase prevents the binding of ATP and/or the substrate

(Engh & Bossemeyer 2002; Lew 2003). Autoinhibition usually involves an activation loop that undergoes conformational changes upon being phosphorylated, thereby activating the kinase through the removal of the obstacle to ATP/substrate binding (Kornev et al. 2006; Nolen et al. 2004). This need for inactive and active states provides a second approach for kinase inhibitor design. While type I inhibitors rely on the availability of a kinase's active site, and therefore its active state, type II inhibitors target the inactive form of the kinase, which can display more structural variation as it is not constrained by the need to catalyse the phosphorylation reaction (Levinson et al. 2006; Noble et al. 2004). This allows type II inhibitors to overcome the selectivity and potency issues caused by type I inhibitors needing to out-compete ATP for the kinase's active site. In addition to the hydrophobic pocket in which ATP binds, type II inhibitors also make use of an adjacent pocket that is only made available by the DFG motif at the N-terminus of the activation loop being in an 'out' conformation unique to the inactive state (Liu & Gray 2006). The specificity of drugs, such as Imatinib, that target this DFG-out conformation can be partly explained by the amino acids surrounding this allosteric pocket being less conserved, and therefore more specific to individual kinases (Levinson et al. 2006; Liu & Gray 2006).

Three further types of kinase inhibitor can be identified. Type III inhibitors bind to a pocket adjacent to the ATP binding site without interacting with the hinge region or displacing ATP (Gavrin & Saiah 2013). Type IV inhibitors bind to a distinct allosteric site several Ångstroms from the ATP binding site, and induce a conformational change that inactivates the protein (Gavrin & Saiah 2013). Finally, type V inhibitors cover both bisubstrate and bivalent inhibitors, and target two different sites simultaneously, for example by targeting both the ATP and substrate binding regions (Gavrin & Saiah 2013). However, no type III, IV or V drugs have been approved yet.

4.1.5 Proteases

Proteases are a group of structurally and functionally diverse enzymes that serve to catalyse the hydrolysis of peptide bonds, and are essential for controlling biological processes including DNA replication, cell proliferation, differentiation and immunological reactions (Puente & López-Otín 2004). Their activity constitutes a form of irreversible post-translational modification, and can be considered to be either degradatory or regulatory, depending on whether it is intended to break the substrate down or alter its activity (Bird et al. 2009). In order to fulfil their diverse roles, the substrate specificity of proteases varies widely, with some interacting with only a single substrate, and others showing significantly more promiscuity. The specific substrates that a protease will interact with are determined by structural variations in its active site, along with the residues that flank the substrate's target scissile bond. These flanking residues will have a

one-to-one correspondence with the subsites in the protease's active site, thereby conferring on the protease a specificity for a specific amino acid sequence (Schechter & Berger 1967).

Eukaryotic proteases can be divided into ones that perform non-covalent (aspartic and metallo proteases) or covalent (cysteine, serine and threonine proteases) catalysis. In both cases, substrate cleavage uses the same general mechanism of a nucleophilic attack on the carbonyl-carbon of a peptide bond, with the nucleophile being an activated water molecule in aspartic and metallo proteases and a catalytic amino acid residue in cysteine, serine and threonine proteases (Puente et al. 2003). A further similarity between the protease types is that catalysis involves the acceleration of the formation of a transition state, the tetrahedral intermediate, which is a prerequisite for the proteolytic scission of a peptide bond (Drag & Salvesen 2010). Aspartic and metallo proteases form this intermediate using a non-covalent acid-base mechanism, in contrast to the covalent bond formed between the substrate and the nucleophilic residue in cysteine, serine and threonine proteases (Drag & Salvesen 2010). Unlike non-covalent catalysis, where a water molecule is activated and performs the nucleophilic attack directly, covalent catalysis is a two-step process. First, the protease is covalently bonded to the N-terminal half of the substrate, forming an acyl-enzyme intermediate, and the C-terminal half of the substrate is released (Turk 2006). Following this, the acyl-enzyme intermediate is hydrolysed by an activated water molecule, and the second half of the product is released.

Due to the irreversible nature of proteolysis, multiple mechanisms exist for controlling the activity of proteases, the two main ones being the synthesis of proteases in an inactive form, known as a zymogen, and the suppression of protease activity via the binding of inhibitors (Bird et al. 2009; Farady & Craik 2010). The synthesis of an inactive precursor zymogen enables proteases' catalytic activities to be spatially and temporally controlled until a specific signal is received, at which point the zymogen is irreversibly converted into an active enzyme (Khan & James 1998). Once a protease has been activated, its activity can be controlled further via the binding of endogenous or exogenous inhibitors. These inhibitors predominantly exert their effect by binding to the protease's active site, thereby preventing any interaction between the protease and its substrate (Farady & Craik 2010).

Commensurate with their biological importance, deficient or abnormal protease function is present in many pathological conditions (Drag & Salvesen 2010). Therefore, pharmacological modulation of their activity is a potentially important therapeutic option for treating disease, with an estimated 5-10% of all drugs under development targeting proteases (Drag & Salvesen 2010). The therapeutic modulation of protease activity is generally achieved using small molecule inhibitors, with the most common approach being to develop a drug that mimics the structure of a protease's substrate and competes with it for the protease's active site (Drag & Salvesen 2010;

Turk 2006). Although non-competitive inhibition of protease activity is possible, no non-competitive inhibitors have been approved for sale nor reached the advanced stages of development (Drag & Salvesen 2010; Shen 2010).

Two possibilities, irreversible and reversible, exist for competitive inhibition of protease function by small molecule inhibitors. Irreversible ones work by trapping the protease covalently, so that neither the protease nor the inhibitor can participate in further reactions (Fear et al. 2007). However, the irreversible nature of these inhibitors means that they risk potentially toxic side effects by binding to unintended nucleophiles, and must be highly selective for their target to avoid unintended permanent inactivation of proteases with similar active sites (Leung et al. 2000). Unlike irreversible inhibitors, reversible ones are predominantly transition state analogues, and therefore exploit the capability of a protease's active site to stabilise a high energy transition state intermediate by mimicking its properties (e.g. polarity and charge) and/or three dimensional structure (Schramm 2011; Smyth 2004; Turk 2006). Although the analogue has similar properties to the original transition state, its differences mean that it will either not undergo a catalysed reaction or the reaction will produce an inconsequential product. These analogues are designed to be more stable than the transition-state itself, thereby binding to the protease with greater affinity than the normal substrate, and consequently preventing its proteolysis (Smyth 2004).

4.2 Machine Learning

Machine learning involves developing and analysing systems that can learn from data. Rather than giving a system explicit instructions for completing a task, it must instead learn from experience (data). The experience consists of a set of observations that are pertinent to the task, and may be provided to the system or gained by it as it tries to perform the task. Following the learning process, the system can be used to evaluate new data and reason about the task.

The tasks that a system can be trained to perform can be divided into two broad categories depending on the desired output. In regression, the system is seeking to learn a relationship between the data and a continuous valued output (or outputs). For example, regression analysis may be used when attempting to determine how the characteristics of a car affect the sales volume, and then to predict the volume of sales for a new car based on its characteristics. The alternative to regression is classification. Rather than determining a continuous valued output, classification seeks to determine group membership based on the properties of an individual. For example, a system can be trained to classify loan applicants, and then be used to predict whether new customers should be approved for a loan. In addition, the

system could help to determine the characteristics of an applicant that make them more or less likely to default on a loan.

As we are concerned with learning the properties of proteins that make them suitable drug targets, and predicting new targets, the problem naturally lends itself to classification. Additionally, many classification systems are capable of providing a measure of the certainty of the classification they produce. This should enable us to not only determine those proteins that are most target-like, but to provide a hierarchical ranking of target-likeness. The remainder of this section considers the requirements for using machine learning, and the options available when performing a classification task.

4.2.1 Definitions

The input to a classification problem consists of a dataset, $\mathcal{D}$, of $N = |\mathcal{D}|$ *observations*; a set, $\mathcal{X} = \{X_1, X_2, \dots, X_F\}$, of $F = |\mathcal{X}|$ *features* and a set, $\mathcal{Y} = \{C_1, C_2, \dots, C_K\}$, of $K = |\mathcal{Y}|$ possible classes. $\mathcal{D}$ is assumed to be independent and identically distributed (iid) with respect to some underlying distribution $\mathcal{P}$, and each observation in it, $d_i, i = 1, \dots, N$, is composed of a pair, (x_i, y_i), where $x_i = \{x_{i1}, x_{i2}, \dots, x_{iF}\}$ is one possible instantiation of $\mathcal{X}$ and $y_i \in \mathcal{Y}$. Each feature in $\mathcal{X}$ can be of any type (discrete or continuous), and for any instantiation, x_i, of $\mathcal{X}$ the value for feature x_{ij} may be corrupted by noise or missing.

4.2.2 Feature Reduction

The number of features used when training a system can have a significant impact on its performance. Intuitively, using more features means that more information is available about a task, and should therefore lead to an increase in the classification and descriptive capabilities of the system. However, many methods of data analysis and exploration benefit from having a simple, compact and non-redundant representation of the data (Eugene Tuv 2009). This is understandable once you consider the fact that $\mathcal{D}$ is often only a subset of the observations that can be drawn from $\mathcal{P}$. As $\mathcal{D}$ does not cover all possible observations, the ability of a system that uses $\mathcal{D}$ to learn about $\mathcal{P}$ must be evaluated in terms of its ability to generalise. This is a measure of how applicable any conclusions reached using D are to the set of all observations that can be drawn from P. In classification tasks this is often expressed as the accuracy with which a system trained on $\mathcal{D}$ can predict the class of an observation $p \sim \mathcal{P}, p \notin \mathcal{D}$. If F grows too large, then the system is able to learn the intricacies of $\mathcal{D}$, rather than general properties of $\mathcal{P}$. A classification system is then deemed to have overfit the dataset, as it is increasing its predictive accuracy on $\mathcal{D}$ at the expense of its generalisation accuracy.

Problems with using an increasing number of features can also manifest themselves in other ways. For example, using more features requires more space for storage, slows down analysis and can lead to what is known as the curse of dimensionality or the Hughes phenomenon. This states that for a given learning algorithm, $\mathcal{A}$, and fixed value of N there is an optimal feature set size F^*, such that while $1 \leq F \leq F^*$ holds the generalisation accuracy of $\mathcal{A}$ trained using F features and N observations will improve. However, once $F > F^*$ the generalisation accuracy will decline. Additionally, F^* increases as N increases, and $\lim_{N\to\infty} F^* = \infty$. This can be seen graphically in Figure 13.

One reason for the decline in generalisation accuracy once $F > F^*$ is that the observations in $\mathcal{D}$ become increasingly sparse in the feature space as F increases (Janecek et al. 2008). This can be avoided if N is able to grow with F. However, in real world situations N is fixed, and potentially at a value such that $F > F^*$. This presents a problem, because if F is too large relative to N it becomes increasingly simple to form even a linear separation of the classes. This means that any given separation of the observations in $\mathcal{D}$ is less likely to have any meaningful ability to generalise to $\mathcal{P}$. For example, methods that rely on forming clusters of similar observations have difficulty forming clusters of meaningful density as distances between observations become more uniform, which is one effect of increasing F (Janecek et al. 2008).

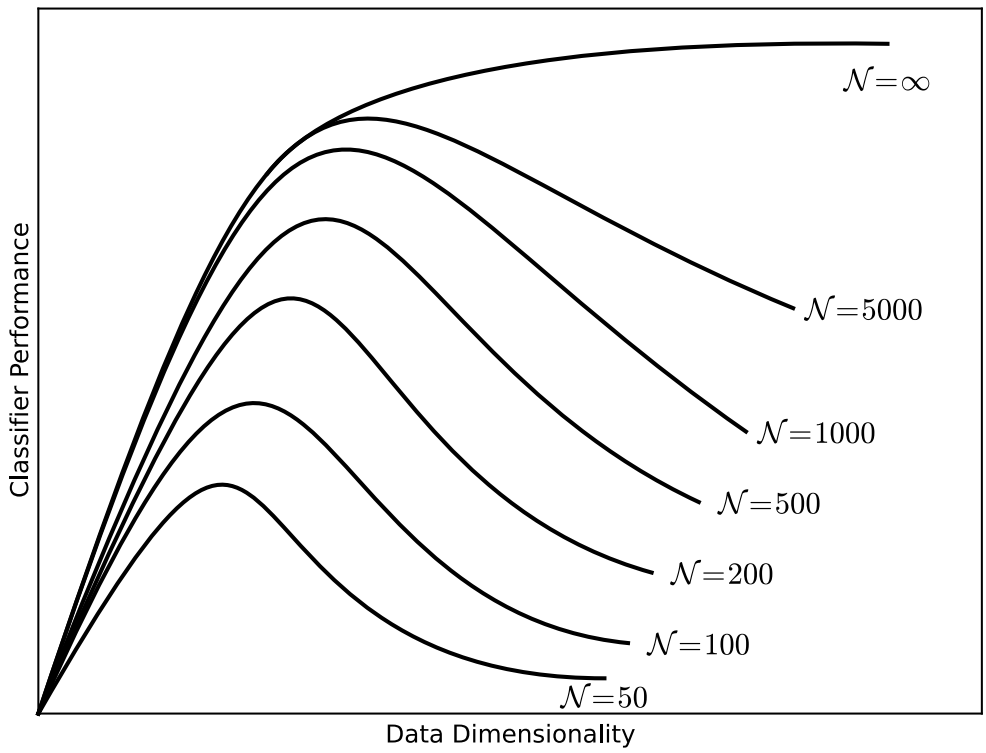

Figure 13: Illustration of the Hughes phenomenon. The inflexion point where $F > F^*$ becomes true can be seen for all dataset sizes except $N = \infty$. As N increases, the value of F^*, and therefore the value of F needed to reach the inflexion point, also increases.

In order to avoid overfitting and the Hughes phenomenon, F should be as close as possible to F^* (Hua et al. 2005). If $F > F^*$, this will require a reduction in the number of features being used. However, as the choice of which features to use is just as important as the choice of the number, this reduction should not be indiscriminate. Ideally the number of features will be reduced to F^* by removing the features that are least useful. The two primary methods for performing this are dimensionality reduction and feature selection. Dimensionality reduction seeks to reduce the original F dimensional feature space, by transforming the original features into $M < F$ new ones. This involves the generation of new 'virtual' features that are composed of combinations of the original features. Rather than create new features, feature selection seeks to select a subset of the original features that optimises some pre-determined criterion. Often this will be the subset that enables the learning of the classifier with the greatest predictive accuracy, or that provides the most discriminative information about the observations in $\mathcal{D}$. While both methods can decrease the feature space used, feature selection is more likely to lead to a loss of information, especially if the original features are diverse and each provides some information (Janecek et al. 2008). However, dimensionality reduction produces virtual features that, while capturing the information of the original feature space, generally do not provide an explicit indication of the contribution of the original features in training the system (Janecek et al. 2008). As we are concerned with the importance of the chosen features in discriminating between classes, rather than only being concerned with predictive ability, we need to ensure that the original meaning of each feature is preserved in our representation. For this reason feature selection is more applicable than dimensionality reduction for our purpose.

4.2.2.1 Subset Selection

Subset selection can be approached in one of two ways. You can either attempt to discover all the relevant features in the dataset, or find a minimal subset of features that optimises the performance of a classifier (Nilsson et al. 2007). Here we define the relevance of a feature along the lines proposed by Kohavi and John (1997). A feature, $f \in \mathcal{X}$, is strongly relevant if removing only f from $\mathcal{D}$ always decreases the accuracy of the optimal Bayes classifier trained on $\mathcal{D}$; weakly relevant if it is not strongly relevant, and there is a subset of features, $f \notin S \subset \mathcal{X}$, such that the accuracy of the optimal Bayes classifier is greater on $S \cup \{f\}$ than on S and irrelevant if it is neither strongly nor weakly relevant (Kohavi & John 1997). While the strongly relevant features are always useful to keep in the selected subset, the usefulness of weakly relevant ones depends on the combination of other features selected. However, being relevant does not necessarily mean a feature will be useful in training a classifier, nor does the fact that a feature is useful in training a classifier mean that it is relevant (Blum & Langley 1997). The

optimality of a feature subset, in terms of training a classifier, is therefore independent of the relevance of the individual features in it. In fact, it is often suboptimal to use all the relevant features for training a classifier, and a subset of useful features can often exclude many redundant, but relevant, features (Guyon & Elisseeff 2003).

Methods for selecting a subset can be divided into filter, wrapper and embedded approaches. Filters work independently of the chosen learning algorithm (Guyon & Elisseeff 2003). They rely solely on general characteristics of the observations in the dataset to determine which features to filter out (Blum & Langley 1997; Eugene Tuv 2009; Yu & Liu 2003). However, this invariance to the choice of learning algorithm means that filters cannot not take into account its biases (Yu & Liu 2003). This can prove to be detrimental, as different feature subsets will work better with different algorithms. Another disadvantage is that filter methods generally produce ranked lists of features based on a univariate analysis (Eugene Tuv 2009). This means that a decision of how many features to use is still required, and that the method cannot take interactions between features into account.

Whereas filters rely only on characteristics of the dataset, wrappers and embedded methods evaluate feature subsets in the context of a specific learning algorithm. They can therefore provide superior performance when the learning algorithm is predetermined (Blum & Langley 1997). Wrappers treat the learning algorithm as a black box, and score individual feature subsets based on the quality of the classifier that they induce. Rather than scoring individual subsets, embedded methods train a classifier on the entire feature set, and then determine the importance of a feature with respect to its usefulness in training that classifier (Eugene Tuv 2009). In this way embedded methods link the importance of a feature to its usefulness to the specific learning algorithm. However, in order for this approach to work the learning algorithm must have a built in method of assessing the importance of individual features. Unfortunately, both wrappers and embedded methods are more prone to overfitting than filters. The propensity of wrappers for overfitting can be understood in a similar vein to multiple comparisons in statistical significance testing. As wrappers function by testing the quality of many different feature subsets, it is possible that a subset, F_o, will be found that induces the best classifier purely by chance. In this case it is possible that F_o will perform substantially worse when the classifier it induces is used to generalise from $\mathcal{D}$ to $\mathcal{P}$. Embedded methods avoid this problem by only evaluating the features in relation to a particular classifier, C, rather than repeatedly testing different subsets. However, C may be overfit with respect to $\mathcal{D}$, and therefore the usefulness of a feature in training C may not be representative of the usefulness of the feature in training a classifier on a truly representative dataset drawn from $\mathcal{P}$.

Unlike embedded methods, where the feature importance is built into the classifier, wrappers require a framework to be built around the chosen classifier. In order to implement a wrapper approach, it is necessary to define the state space through which the wrapper will search, the initial state of the search, the method of searching and a condition for terminating the search. For feature selection the state space is the powerset of $\mathcal{X}$, and therefore contains all 2^F possible subsets of features. An example of the state space for a dataset containing four features can be seen graphically in Figure 14.

As the size of the state space is exponential in F, an exhaustive search of all possible feature subsets, though ideal, is intractable for most $\mathcal{X}$. However, results equivalent to an exhaustive search can be achieved using a branch and bound method. These function similarity to exhaustive searches, but avoid searching through every subset by identifying sections of the feature space that cannot contain the optimal subset. While quicker than exhaustive searches, branch and bound based methods are still impractical for problems where the dataset contains over thirty features (Pudil et al. 1994).

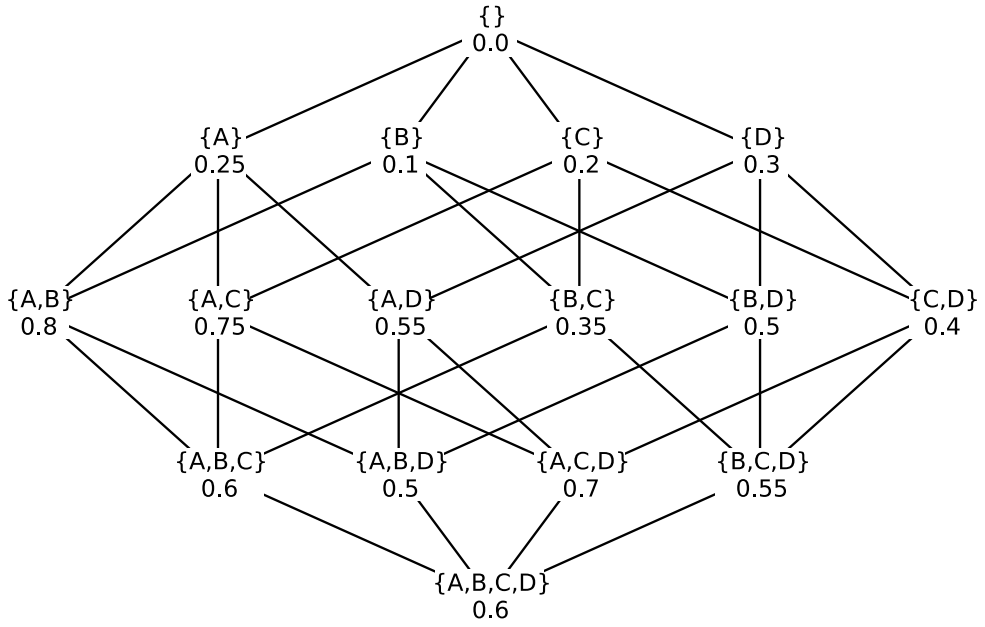

Figure 14: A representation of the feature space to be searched when $\mathcal{X} = \{A, B, C, D\}$. The number below each subset indicates the quality of an arbitrary classifier induced using the feature subset. In this case 1 would represent the best possible classifier and 0 the worst. Each feature subset f_i shares an edge with all subsets, f_j, such that f_j can be generated by the addition or removal of one feature from f_i.

Due to the limitations of exhaustive search methods, heuristics are often used to guide the search. Common heuristics for searching the state space are forward selection (FS) and backwards elimination (BE). Pseudocode for the FS algorithm can be seen in Figure 15.
The first step is to initialise the current feature set to the empty set (line 1). Then the body of the

algorithm is repeated until the current feature set is $\mathcal{X}$ (lines 2-12). The first pass through lines 2-12 causes every individual feature to be evaluated by itself. The individual feature f_i that induced the best classifier (lines 4-8) is added to the current feature set (lines 9-12). The second pass through lines 2-12 causes every feature set $\{f_i, f_j \mid f_j \in (\mathcal{X} - f_i)\}$ to be evaluated. This process of incrementally adding one feature to the current feature set is repeated until $\mathcal{X}$ itself is evaluated. Following this, the subset of features that induced the best classifier is chosen as the selected subset. The execution of the FS algorithm in Figure 15 on the feature space in Figure 14 is as follows:

1. $currentSubset \leftarrow \emptyset$
2. Sets $\{A\}, \{B\}, \{C\}$ and $\{D\}$ are evaluated, and $\{D\}$ gives the highest score.
3. $currentSubset \leftarrow currentSubset \cup \{D\}$
4. All sets of two features that contain feature D are considered next. Sets $\{A, D\}, \{B, D\}$ and $\{C, D\}$ are evaluated, and $\{A, D\}$ gives the highest score.
5. $currentSubset \leftarrow currentSubset \cup \{A\}$
6. All sets of three features that contain features A and D are considered next. Sets $\{A, B, D\}$ and $\{A, C, D\}$ are evaluated, and $\{A, C, D\}$ gives the highest score.
7. $currentSubset \leftarrow currentSubset \cup \{C\}$
8. Set $\{A, B, C, D\}$ is evaluated.
9. $currentSubset \leftarrow currentSubset \cup \{B\}$
10. Set $\{A, C, D\}$ gave the highest score of all ten feature sets evaluated, and is therefore chosen as the optimal feature set to use.

```
1.  currentSubset ← ∅, bestSubset ← ∅, bestSubsetScore ← 0
2.  While currentSubset ≠ 𝒳
3.      bestScore ← 0, bestFeature ← ∅
4.      For f ∈ (𝒳 − currentSubset)
5.          T ← currentSubset ∪ f
6.          If score(T) > bestScore
7.              bestScore ← score(T)
8.              bestFeature ← f
9.      currentSubset ← currentSubset ∪ bestFeature
10.     If bestScore > bestSubsetScore
11.         bestSubsetScore ← bestScore
12.         bestSubset ← currentSubset
13. Return bestSubset
```

Figure 15: Pseudocode for a FS algorithm.

BE operates in the opposite direction to FS. Rather than moving from the empty set of features to $\mathcal{X}$, BE moves through the feature space from $\mathcal{X}$ to the empty set. Pseudocode for the BE algorithm can be seen in Figure 16. The first step is to initialise the current feature set to $\mathcal{X}$ (line 1). Following this the body of the algorithm is repeated until the current feature set is empty (lines 2-12). The first pass through lines 2-12 considers every possible feature set $\{\mathcal{X} - f_i \mid \forall f_i \in \mathcal{X}\}$. The feature chosen for removal is $f_1 \in \mathcal{X} \mid \forall f_j \in \mathcal{X}, score(\mathcal{X} - f_1) \geq score(\mathcal{X} - f_j)$ (lines 4-8). f_1 is then removed from the current subset, and $\mathcal{X} - f_1$ is recorded as the best feature subset if its score is greater than $\mathcal{X}$ by itself (lines 9-12). The second pass through lines 2-12 causes every feature set $\{(\mathcal{X} - f_1) - f_i \mid \forall f_i \in (\mathcal{X} - f_1)\}$ to be evaluated. This process is repeated until the current feature set is empty. Following this, the subset of features that induced the best classifier is chosen as the selected subset. The execution of the BE algorithm in Figure 16 on the feature space in Figure 14 is as follows:

1. Set $\{A, B, C, D\}$ is evaluated.
2. $currentSubset \leftarrow \{A, B, C, D\}$
3. All sets of three features are evaluated, and $\{A, C, D\}$ gives the highest score.
4. $currentSubset \leftarrow currentSubset - \{B\}$
5. All permutations of two of the features in $\{A, C, D\}$ are evaluated, and $\{A, C\}$ gives the highest score.
6. $currentSubset \leftarrow currentSubset - \{D\}$
7. $\{A\}$ and $\{C\}$ are evaluated next, and $\{A\}$ gives the highest score.
8. $currentSubset \leftarrow currentSubset - \{C\}$
9. Set $\{A, C\}$ gave the highest score of all ten feature sets evaluated, and is therefore chosen as the optimal feature set to use.

```
1.  currentSubset ← 𝒳, bestSubset ← 𝒳, bestSubsetScore ← score(𝒳)
2.  While currentSubset ≠ ∅
3.      bestScore ← 0, worstFeature ← ∅
4.      For f ∈ currentSubset
5.          T ← currentSubset − f
6.          If score(T) > bestScore
7.              bestScore ← score(T)
8.              worstFeature ← f
9.      currentSubset ← currentSubset − worstFeature
10.     If bestScore > bestSubsetScore
11.         bestSubsetScore ← bestScore
12.         bestSubset ← currentSubset
13. Return bestSubset
```

Figure 16: Pseudocode for a BE algorithm.

The smaller number of feature sets that must be evaluated when using FS or BE instead of an exhaustive approach does not come for free. Both methods suffer from a 'nesting effect' due to the fact that the operations they use to add or discard a feature from the current set cannot be reversed (Pudil et al. 1994). This causes the feature sets evaluated at step t to be a superset, in the case of BE, or a subset, in the case of FS, of the best set from step $t + 1$. Essentially both methods follow a trajectory through the state space that cannot be deviated from, or backtracked along. This greedy approach ensures that the methods are sub-optimal, that is they cannot guarantee the quality of the feature set returned. FS and BE also generally remove redundant and weak features, while keeping independent ones (Chen & Jeong 2007). However, using individually weak features, rather than only strong ones, may induce a better classifier (Guyon & Elisseeff 2003), and therefore simply removing redundant and/or weak features may not be the most appropriate method for finding the optimal feature set (Chen & Jeong 2007). Despite their overall similarity, FS and BE do differ in some regards. For example, for the vast majority of datasets FS should be quicker. This is because most learning algorithms can be trained in less time when using fewer features. This benefits FS as, although both methods evaluate $\frac{F(F+1)}{2}$ subsets, the subsets evaluated by FS will on average be smaller than those evaluated by BE. For example, FS will only evaluate two subsets of size $F - 1$, whereas BE will evaluate F. However, with FS the addition of each feature is only evaluated with respect to the features that are already in the feature set. This means that multivariate interactions between features can be missed in FS, whereas BE allows the interactions between all features to be considered.

In order to substantively improve upon the basic FS and BE algorithms, it is necessary to move away from only using operators that add or discard one feature at a time. Metaheuristics, such as genetic algorithms (GAs), achieve this by optimising the search process in addition to the feature set. Rather than impose a trajectory through the state space, GAs attempt to find and explore areas of it that seem most promising. They manage this by maintaining a population of feature sets, with each individual in the population occupying a different state in the state space. Over time the population will change as better feature sets are found. Gradually the individuals in the population will begin to converge to areas in the state space that contain high quality feature sets.

GAs for feature selection will often represent the individual feature sets in the population as bit vectors, with each feature $f_i \in \mathcal{X}$ represented by a corresponding bit b_i. A feature set $S \in \mathcal{X}$ can therefore be represented as a bit vector where $b_i = 1$ if $f_i \in S$, and $b_i = 0$ if $f_i \notin S$. In general, GAs provide two operators, mutation and crossover, that manipulate the bit vectors in order to enable the transition between states. The mutation operator acts on a feature set S by flipping a bit in the bit vector representing it. The flipping of a bit b_i from 1 to 0 corresponds to

feature f_i being removed from S, and is comparable to the operator used in BE. Alternatively, the flipping of a bit b_i from 0 to 1 corresponds to feature f_i being added to S, and is comparable to the operator used in FS. A comparison of the mutation operator with the operators used in FS and BE can be seen in Figure 17. Although the mutation operator is more powerful than the operators used in FS and BE, it is still insufficient for searching the entire state space efficiently. In order to do this it is necessary to enable the search to 'jump' between areas of the state space. Crossover achieves this by enabling two or more bit vectors to be combined in order to produce new ones. This process functions much like biological reproduction, with the individual 'parent' bit vectors producing 'offspring' that are a combination of the bits in the parents. This enables rapid exploration of the state space, as one crossover operation can produce a transition between parent and offspring states that would have required multiple mutation operations. An example of a crossover operation can be seen in Figure 18.

The ability to rapidly move between distant states using crossover, while also maintaining a more local search using mutation, gives GAs a tuneable balance between exploring the state space and exploiting knowledge about localities within it. Despite this, there is no guarantee that a GA will find the optimal solution. Reasons for this include the random nature of the mutation and crossover operators, and the fact that GAs can become stuck in local optima. Once one individual in the population is near a local optimum, the remainder of the population will be progressively drawn towards it through crossover. The attraction towards this one locally optimal state causes the individual bit vectors in the population to become increasingly similar to the optimum and one another. With the parent bit vectors differing by fewer and fewer bits, crossover produces offspring that are progressively more similar to their parents. Eventually the population becomes so homogenous that the search ceases advancing, and a local optimum is reached.

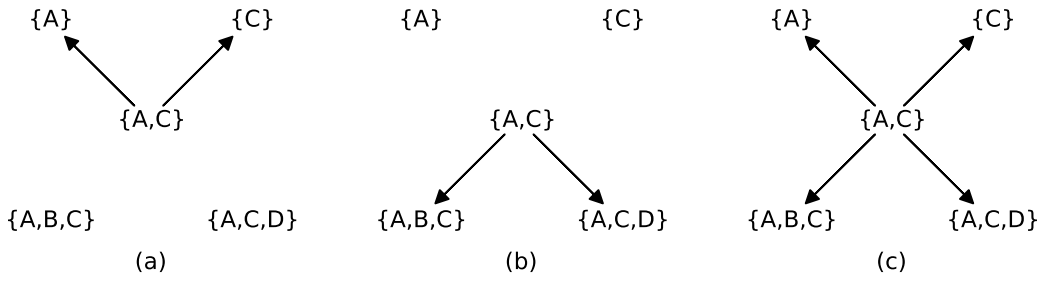

Figure 17: A comparison of feature space traversal by BE, FS and GA mutation. Assuming the current feature subset is $\{A, C\}$, (a) shows how BE can only move to subsets of the current set, (b) shows how FS can only move to supersets of the current set and (c) shows how mutation enables a GA to move in either direction.

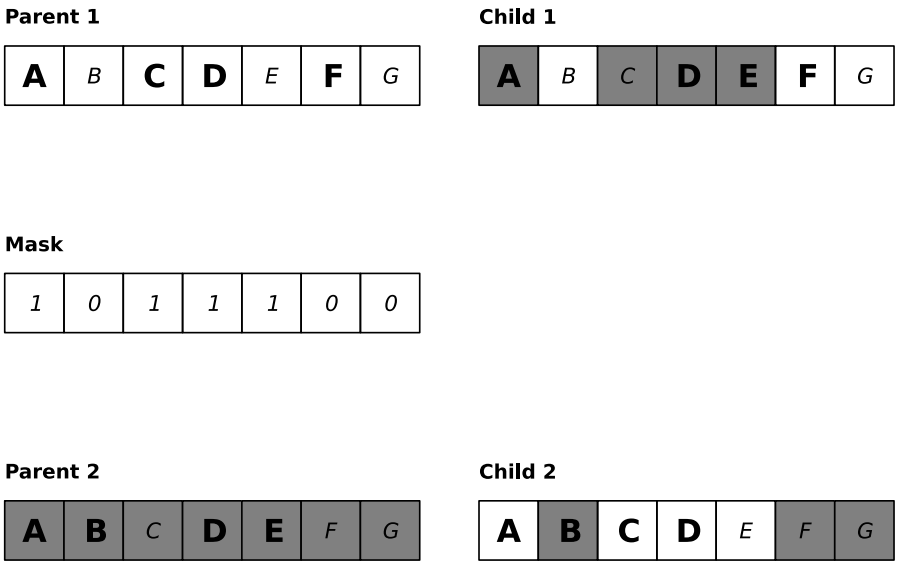

Figure 18: A demonstration of a crossover operation from a GA. Parent 1 and parent 2 are to be crossed over to produce child 1 and child 2. The large bold letters indicate features that are in the feature set. Parent 1 represents the feature set $\{A, C, D, F\}$, parent 2 $\{A, B, D, E\}$, child 1 $\{A, D, E, F\}$ and child 2 $\{A, B, C, D\}$. The mask contains one entry for every feature, and each value in the mask is set to 1 or 0 with equal probability. A 1 in the mask for feature f_i indicates that feature f_i in child 1 will come from parent 2, and feature f_i in child 2 will come from parent 1. A 0 in the mask for feature f_i indicates that feature f_i in child 1 will come from parent 1, and feature f_i in child 2 will come from parent 2.

A typical GA implementation for feature selection will comprise a number of generations; each of which consists of modifying the current population of bit vectors, evaluating the fitness of the newly created bit vectors and updating the population. The fitness of an individual bit vector is a measure of the quality of the bit vector, and in the case of feature selection corresponds to a measure of the quality of the feature subset. Example pseudocode for a GA used for feature selection can be seen in Figure 19. The first step is to initialise the bit vectors in the population (line 2). This can be performed in a number of ways, such as: randomly instantiating each bit of each vector to 0 or 1 with equal probability, ensuring that each bit is a 0 and a 1 the same number of times or initialising the population to inhabit a particular region of the state space. The main body of the GA (lines 4-12) is then repeated until a stopping criterion is met. Potential criterion include the number of generations, the time elapsed, the distribution of features in the population and the number of fitness evaluations performed. Each generation consists of the following steps:

1. Select C pairs of unique individuals (line 5).
2. Perform crossover on each of the C pairs selected in step 1 (line 6). This will generate a number of offspring determined by the crossover operator used.
3. Mutate each of the individuals produced in step 2 (line 7).
4. Select the new population to be a subset of the individuals created in step 4 and the old population (i.e. $P_{i+1} \subseteq P_i \cup O_M$) (line 9).

5. Determine if any individual in P_{i+1} is the fittest individual seen over all the generations (lines 10-12).

Finally, the best individual encountered over all the generations is returned as the best feature set. This approach to performing a GA can be considered to be a steady state approach as it forms population P_{i+1} from the members of P_i and their offspring. A common alternative to the steady state approach is the generational GA. In this approach P_{i+1} is formed only from the offspring of P_i (i.e. $P_{i+1} = O_M$).

Input

$popSize$ – The size of the population.

Pseudocode

1. $gen \leftarrow 0$
2. $P_{Gen} \leftarrow popSize$ randomly generated individuals
3. $bestIndividual \leftarrow \emptyset$
4. While no stopping criterion has been met
5. $Parents \leftarrow \{(p_{i1}, p_{i2}) : i = 1, \dots, C \wedge p_{i1}, p_{i2} \in P_{Gen} \wedge p_{i1} \neq p_{i2}\}$
6. $O_C \leftarrow \{crossover(i,j) : \forall (i,j) \in Parents\}$
7. $O_M \leftarrow \{mutate(i) : \forall i \in O_C\}$
8. $gen \leftarrow gen + 1$
9. $P_{Gen} \leftarrow \{popSize$ individuals from $P_{Gen} \cup O_M\}$
10. $bestInGen \leftarrow$ fittest individual in P_{Gen}
11. If $fitness(bestInGen) > fitness(bestIndiv)$
12. $bestIndiv \leftarrow bestInGen$
13. Return $bestIndiv$

Figure 19: Pseudocode for a GA.

4.2.3 Approaches

Traditional approaches to classification can fall into one of three broad categories. In *unsupervised learning* each observation in $\mathcal{D}$ is given the same class label. This does not mean that all observations belong to the same class, but rather that none of the observations have had their true class ascertained. Therefore, the training of the classifier cannot use knowledge of class membership to help separate the observations into their unique classes. Instead, the system must use only the raw data, and potentially some *a priori* information about the relative importance of the features and observations. This leads unsupervised methods to search for patterns in the data that are too structured to be noise (Ghahramani 2004), and therefore to find clusters of observations rather than specific disjoint partitions.

Semi-supervised learning methods can be employed when only some of the observations in $\mathcal{D}$ are labelled, and have therefore had their class ascertained. In this case, all classes in $\mathcal{Y}$ may

occur as the label of a labelled observation in $\mathcal{D}$, or all observations belonging to some class $c \in \mathcal{Y}$ may be in the set of unlabelled observations. In the latter situation, all information about class c is lost if available unlabelled observations are not used, while in the former it is possible that important properties of a class can be missed if only the labelled observations are used. Irrespective of how the classes are distributed amongst the labelled and unlabelled observations, using unlabelled observations has been shown to be beneficial when training a classifier (Bouchachia 2007; Castelli & Cover 1996).

In contrast to unsupervised and semi-supervised approaches, *supervised learning* approaches require that all observations in $\mathcal{D}$ be associated with a class from $\mathcal{Y}$. They can then use the class information in $\mathcal{D}$ to train a classifier that can predict the class of a new observation, in addition to capturing and explaining the patterns and interactions between features. The training of the classifier will produce a mapping between $\mathcal{X}$ and $\mathcal{Y}$, which can be seen as a general rule to be used to predict the class of an observation given its instantiation of $\mathcal{X}$. Typically a set of observations, $\mathcal{T} \subseteq \mathcal{D}$, is designated to be a *training set*. This training set is used to learn the mapping between $\mathcal{X}$ and $\mathcal{Y}$, and the set of observations $\mathcal{D} - \mathcal{T}$ can then be used to test the ability of the mapping to generalise to observations not seen during training. The quality of a learnt classifier is then based on the ability of the learning algorithm to use $\mathcal{D}$ in order to learn a mapping from $\mathcal{X}$ to $\mathcal{Y}$ that can generalise to $\mathcal{P}$, as well as on how well $\mathcal{D}$ represents the set of all observations that can be drawn from $\mathcal{P}$.

4.2.3.1 Supervised Learning

Approaches to supervised learning can be divided into two broad categories. The first consists of algorithms that attempt to train a single best classifier using $\mathcal{D}$. Examples of algorithms in this category are k-Nearest Neighbours (k-NN), support vector machines (SVMs) and decision trees (DTs). Rather than attempting to learn a single optimal classifier, *ensemble* methods learn many different classifiers, and then combine the outputs of the classifiers to generate a single prediction. The most common examples of ensemble approaches are boosting, bagging and random forests (RFs).

4.2.3.1.1 k-Nearest Neighbours

The k-NN algorithm attempts to predict the class of an observation from the k observations in $\mathcal{T}$ that are nearest to it. By varying the value of k the number of neighbours that are used in the classification can be altered, and the influence of individual observations in $\mathcal{T}$ on the classification can be controlled. This is necessary as different datasets will perform differently

with the same value of k. For example, small values of k will increase the sensitivity of the classifier to outliers and noise present in $\mathcal{T}$ (Wu et al. 2007). Larger values of k can be used to smooth out the outliers, but can also cause poorly represented clusters of observations to be missed. Additionally, larger values of k will increase the likelihood that observations from the wrong class dominate the neighbourhood. This is especially true if the observation to be classified lies close to a boundary between classes. Figure 20 demonstrates some of the affects that changing the value of k can have on classifications.

Two further concerns when using k-NN algorithms are the features used and the lack of a training phase. As features with large scales will dominate frequently used distance metrics, such as the Euclidean distance, feature scaling or standardisation is often necessary. Additionally, k-NN algorithms are sensitive to the presence of irrelevant features due to the equal weighting that they give to all features (Cord & Cunningham 2008). A final concern with k-NN methods is that they defer all computation until an observation needs to be classified. This saves time on training the classifier, as there is no need for training, but causes predictions to take substantially longer. Additionally, the lack of a training phase also means that k-NN classifiers do not learn a general mapping between $\mathcal{X}$ and $\mathcal{Y}$, and are therefore unable to provide information about the importance of individual features or observations in learning an accurate representation of $\mathcal{P}$.

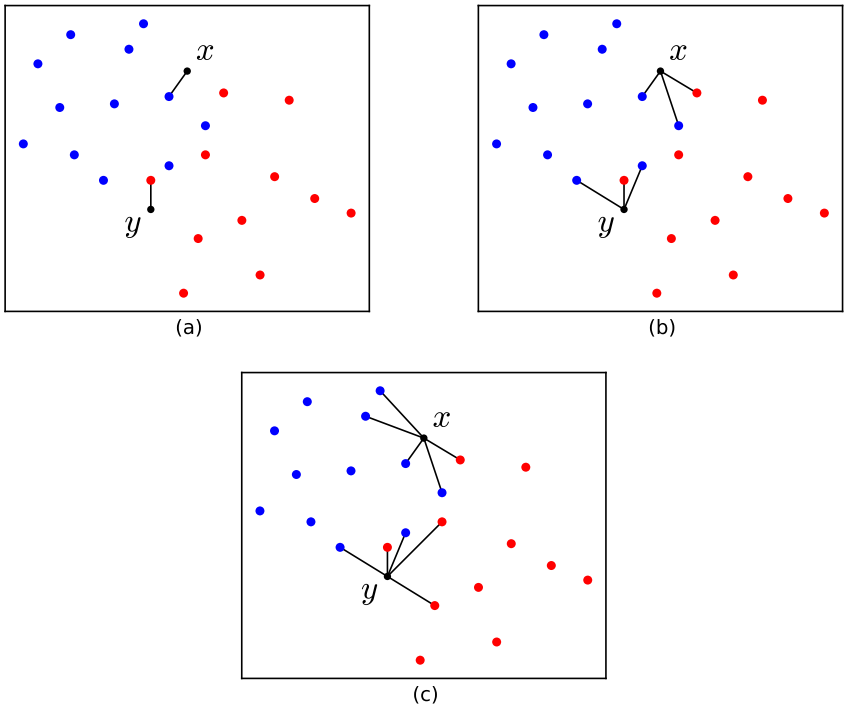

Figure 20: The effects of changing k in a k-NN classififer. A dataset of two classes, class c_1 in blue and c_2 in red, is used to classify observations x and y, in black, under three different values of k. Figure (a) uses $k = 1$, (b) uses $k = 3$ and (c) uses $k = 5$. The k nearest observations to both x and y are indicated by solid black lines. Observation x is assigned to class c_1 using all three values for k. However, observation y is assigned to class c_1 when $k = 3$, but c_2 when $k = 1$ or $k = 5$.

4.2.3.1.2 Support Vector Machines

SVMs represent one alternative to k-NN. In order to classify observations, they attempt to divide the feature space represented by $\mathcal{X}$ into two subspaces. This is done by finding a hyperplane that separates the observations in $\mathcal{T}$ in such a way that each class's observations are in a different subspace. This necessitates that the classification problem be binary, as a single hyperplane can only separate the observations into two sets. Once the hyperplane is found, new observations can be classified based on their position relative to it (Figure 21).

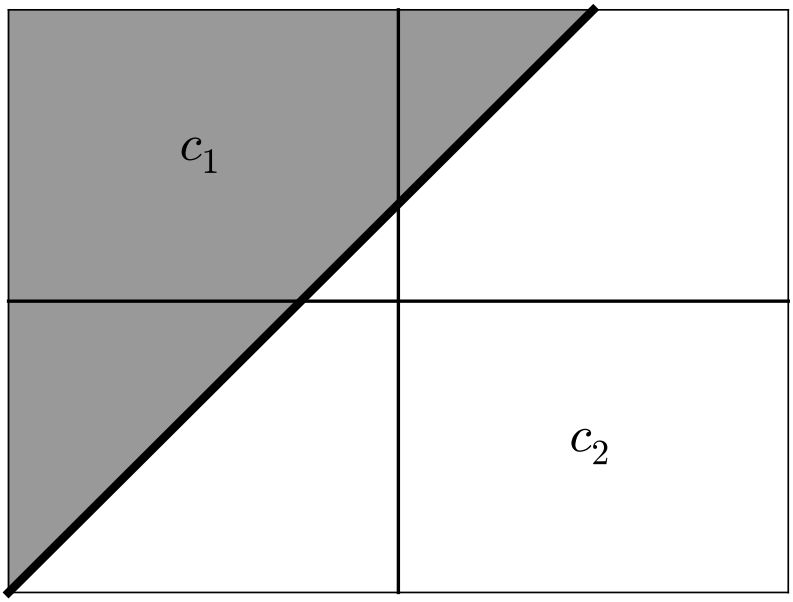

Figure 21: A depiction of a hyperplane bisecting a two dimensional feature space. The hyperplane is represented by the thick black line. Any observations that fall in the grey section of the feature space will be classified as class c_1, while those that fall in the white section will be classified as class c_2.

If a hyperplane that perfectly separates the observations can be found, then there will be an infinite number of them (Ben-Hur et al. 2008). Therefore, in order to choose the best hyperplane from this infinite set you must be able to quantify the quality of each one. This can be done using the concept of a margin. Assuming a binary classification problem with classes c_1 and c_2, the margin of a hyperplane H is the Euclidean distance from H to the closest observation of class c_1 plus the Euclidean distance from H to the closest observation of class c_2. The goal of an SVM then is to choose H such that it perfectly separates $\mathcal{T}$ and also maximises the margin. Choosing the hyperplane in this manner ensures that there will be the largest possible distance between the hyperplane and the training observations on either side, which has been proven to reduce an upper bound on the expected generalisation error (Kotsiantis 2007). The maximum margin hyperplane H can be found by finding a pair of hyperplanes, H_{c_1} and H_{c_2}, such that:

1. H_{c_1} passes through at least one observation from class c_1 in $\mathcal{T}$.
2. H_{c_2} passes through at least one observation from class c_2 in $\mathcal{T}$.
3. H_{c_1} and H_{c_2} are parallel, and there are no observations from $\mathcal{T}$ between them.

Out of all such pairs, find the one that maximises the distance between H_{c_1} and H_{c_2}. The dividing hyperplane H is then set halfway between H_{c_1} and H_{c_2}. The training observations closest to H, and therefore the ones that lie on H_{c_1} or H_{c_2}, are termed support vectors, and are the only training observations that are needed to define H. Therefore, if every observation in $\mathcal{T}$ except for the support vectors was removed from $\mathcal{T}$, or moved in the feature space without crossing H_{c_1} or H_{c_2}, the same dividing hyperplane would be found (Burges 1998). An illustration of the support vectors and the maximisation of the margin can be seen in Figure 22.

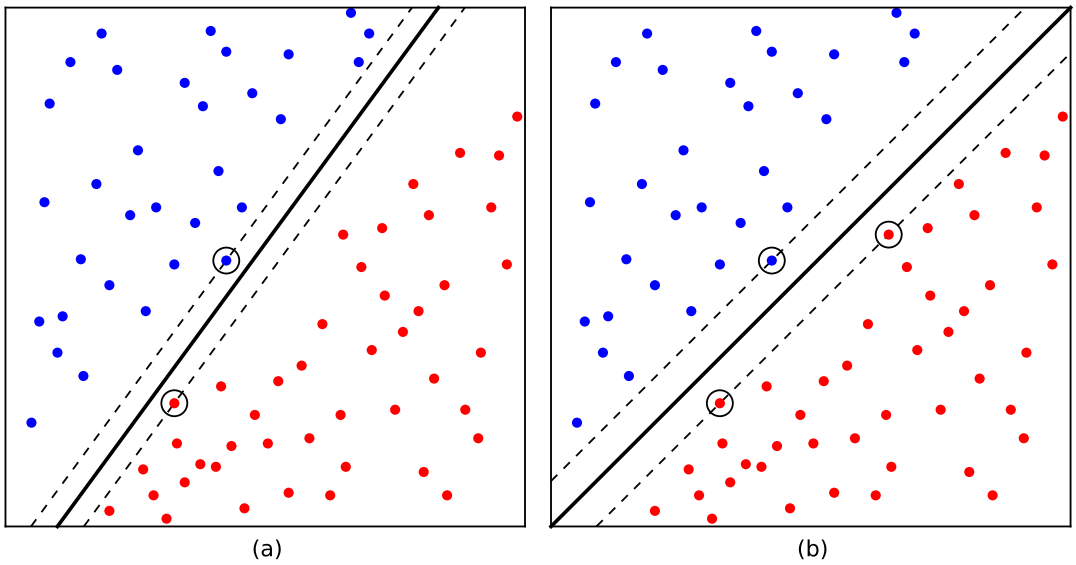

Figure 22: An example of maximising the margin when using SVMs. A two dimensional dataset of two classes, class c_1 in blue and c_2 in red, is separated by two different hyperplanes. Circled observations are the support vectors, and the dashed lines passing through them represent the hyperplanes H_{c_1} and H_{c_2}. In (a) it can be seen that the margin is smaller than the maximum margin that is found in (b).

The approach to SVMs outlined above requires that the training observations be perfectly linearly separable. The problem of datasets that are not perfectly separable can be handled using soft margins, where the dividing hyperplane does not have to perfectly classify all training observations. However, this still requires that the dividing hyperplane be linear, and often it is possible to achieve better classification accuracy using a nonlinear boundary between classes. SVMs can achieve this by mapping the observations in $\mathcal{T}$ into a higher dimensional feature space where it is possible to linearly separate the classes. This works because a linear boundary induced by an SVM in the higher dimensional feature space translates to a non-linear boundary in the

original (lower dimensional) feature space (Cord & Cunningham 2008). An example of a successful transformation inducing a non-linear boundary in the original feature space can be seen in Figure 23.

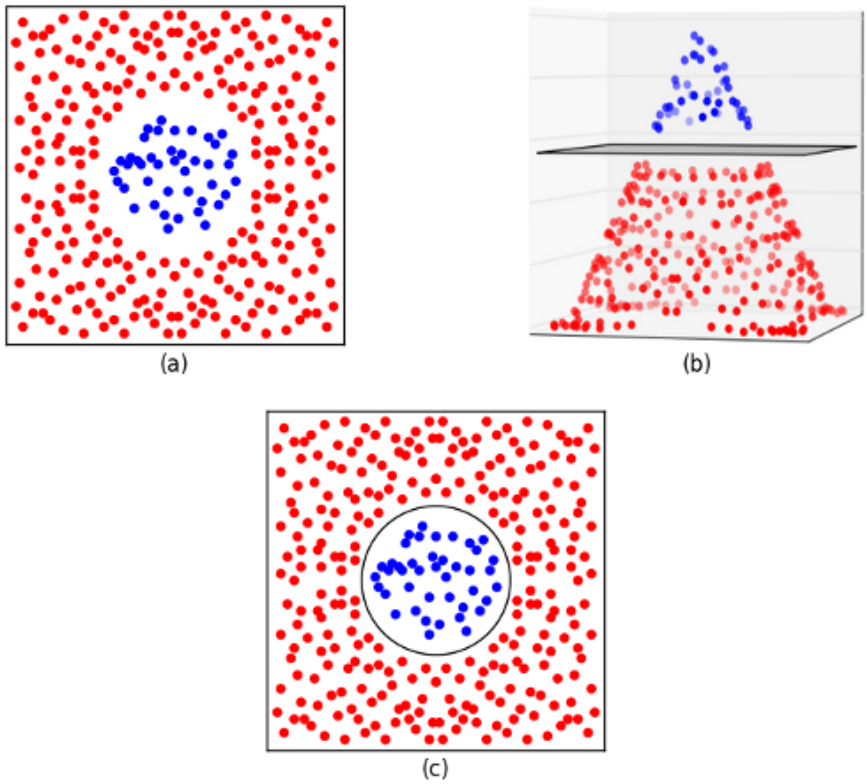

Figure 23: Linear separation of a dataset in higher dimensions. Mapping a dataset into a higher dimensional feature space enables a non-linearly separable dataset to become linearly separable. (a) The original dataset is not linearly separable in two dimensions, but is separable using a non-linear boundary. (b) A projection of the two dimensional data into three dimensions. The third dimension is calculated as the inverse of the distance of an observation from the centre of the two dimensional plot. A hyperplane is then able to linearly separate the three dimensional data. (c) The non-linear boundary that the hyperplane in the three dimensional space induces in the original two dimensional space.

Typically, transforming data into a higher dimensional feature space would raise concerns about increased computational complexity and overfitting. However, SVMs enable the advantages of higher dimensional feature spaces to be exploited without increasing the computational load or likelihood of overfitting. This is done through the utilisation of kernel methods. In order to determine the separating hyperplane in an SVM it is necessary to compute the dot product between all pairs of training observations. If an explicit mapping, $f(\boldsymbol{x})$, from the original to higher dimensional feature space was used, then the expensive computation of $f(\boldsymbol{x}) \cdot f(\boldsymbol{y})$ would be required for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{T}$. In order to avoid this, a kernel function $K(\boldsymbol{x}, \boldsymbol{y})$ can be used. Provided that we can construct a kernel $K(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}) \cdot f(\boldsymbol{y})$ that can be computed efficiently, the mapping $f(\boldsymbol{x})$ never needs to be evaluated or even known. By enabling us to avoid carrying out

the mapping from the input feature space to the high dimensional space where the linear boundary is to be constructed, kernel functions enable SVMs to be efficiently generalised to non-linear classification tasks.

4.2.3.1.3 Decision Trees

Another approach to developing learning systems is to use DTs. These separate the observations in $\mathcal{T}$ using a set of disjoint rules based on the values of the features. For example, one rule, r, for an observation belonging to class c may be $f_1 < 10 \wedge f_2 > 0.5$ for $f_1, f_2 \in \mathcal{X}$. Therefore, any observation, d, that satisfies r will be assigned to class c. As the set of rules is disjoint, it can be structured as a series of choices with disjoint outcomes, and can therefore be represented using a tree. The tree itself is a set of hierarchically organised nodes and edges (Figure 24), where each node has one incoming edge, except for the root node which has none, and two or more outgoing edges. If a node has outgoing edges it is considered to be internal, and terminal (or a leaf) otherwise. Generational relationships for a node, n, can be defined in terms of the tree structure. Assuming that n is not the root node, then its parent, $p = parent(n)$, is the node where n's incoming edge originated. Similarly, assuming that n is not a leaf node, then the children of n, $C = children(n)$, are the nodes $\{C|\forall c \in C, n = parent(c)\}$. Similarly, the ancestors of n, $A = ancestors(n)$, are the nodes $\{A|\forall a \in A, a = parent(n) \vee a \in ancestors(parent(n))\}$, while the descendents of n, $D = descendents(n)$, are the nodes $\{D|\forall d \in D, n \in ancestors(d)\}$.

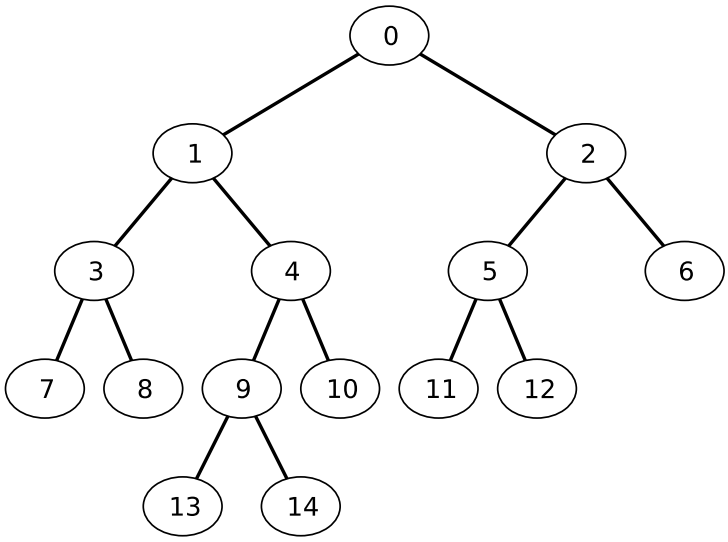

Figure 24: A binary tree. Taking node n to be node 4, then $parent(n)$ is node 1, $children(n)$ is the set of nodes $\{9, 10\}$, $ancestors(n)$ is the set of nodes $\{0, 1\}$ and $descendents(n)$ is the set of nodes $\{9, 10, 13, 14\}$.

A DT uses hyperplanes to recursively partition the feature space into a set of disjoint hyperrectangles, such that observations in $\mathcal{T}$ from the same class are grouped in the same hyperrectangle. The boundaries that make up a hyperrectangle are the conditions that make up the individual classification rules that the DT uses, and therefore there are as many rules as there are hyperrectangles. The reason that a DT induces a set of hyperrectangles, rather than a set of arbitrary multi-dimensional shapes, is because only one feature, f, is considered at each node in the tree. This causes each node to induce a partitioning hyperplane, or set of hyperplanes, orthogonal to the f axis, and parallel to all others. Conceptually, each node, n, represents a different subspace of the entire feature space, such that the subspace $s = subspace(n) = \bigcup_{\forall c \in children(n)} subspace(c)$. Hence, the set of hyperplanes induced by n partition the subspace it represents into c smaller supspaces, where c is the number of children that n has. In this way each child node represents a finer grained subspace of the entire feature space than its parent.

The process of nodes partitioning the subspace that they represent continues recursively until a leaf node is created. The subspace represented by a leaf node is therefore equivalent to one of the hyperrectangles generated by the DT, and also to a classification rule. Each leaf is associated with a single class based on the classes of the observations in $\mathcal{T}$ that fall within the hyperrectangle that it represents. Once the partitioning is complete, an observation can be classified by determining which leaf (hyperrectangle) it falls in. A depiction of a DT can be seen in Figure 25, and the feature space partitioning it induces in Figure 26.

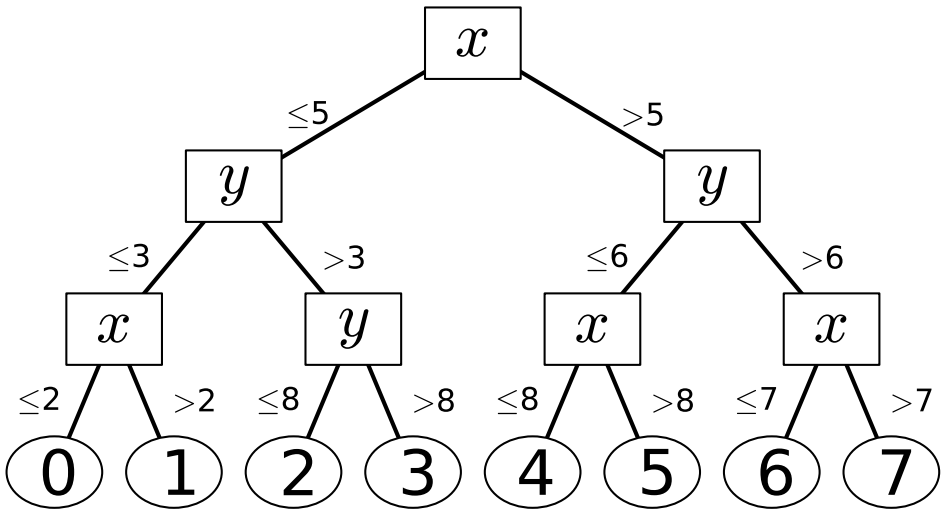

Figure 25: A decision tree. Squares represent internal nodes and ellipses leaf nodes. The letter in an internal node indicates the feature that is used for the node's cutpoint, and the inequalities on the edges leaving it indicate the values of the feature that the node uses to partition the observations in its subspace. For example, the root node uses $x = 5$ as the cutpoint, and therefore any observations where $x \leq 5$ go into the subspace represented by its left child and any observations where $x > 5$ go into the subspace represented by its right child.

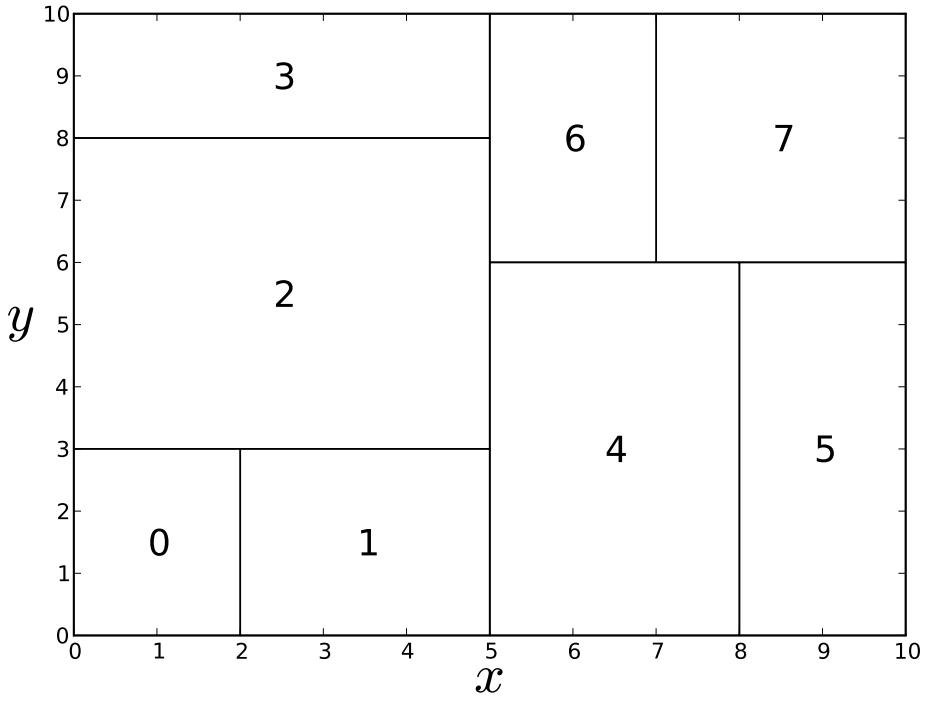

Figure 26: The partitioning of the feature space induced by the DT in Figure 25. Each numbered rectangle represents the portion of the feature space covered by the leaf node in Figure 25 with the same number. For example, leaf node 2 will hold all observations where $x \leq 5$ and $3 < y \leq 8$.

In order to construct a DT, for each node, n, a decision must be made as to the best *cutpoint* for n. A cutpoint is the value, or values, of a feature, f, that are used to partition the subspace represented by n. For example, a binary cutpoint for node n of $f = 10$ would create a partitioning hyperplane perpendicular to the f axis through $f = 10$. Then, the subspace represented by n would be partitioned into two smaller subspaces, one where $f \leq 10$ and one where $f > 10$. The optimal cutpoint for n is the one that enables n and its descendents, and therefore the subspace represented by n, to have the greatest possible classification performance. However, finding the optimal cutpoint in this manner is NP-complete (Kotsiantis 2007), and therefore efficient heuristics for evaluating cutpoints are used instead.

Technically a cutpoint can be multivariate. However, allowing multivariate cutpoints substantially increases the computational complexity of finding the optimal cutpoint. Therefore, the most common approach used when finding a cutpoint is to restrict them to being univariate. In addition to only allowing univariate cutpoints, the optimality of a cutpoint for a node, n, is usually only evaluated with respect to the observations from $\mathcal{T}$ in the subspace represented by n. This means that rather than considering the global optimality of a cutpoint, i.e. with respect to all the partitions that will be induced by n's descendents, most DT heuristics only consider the local optimality of the cutpoint. Therefore, the DT is not grown by making one globally optimal choice from amongst all the possible DTs, but rather by making a series of locally optimal decisions.

However, while this makes the choice of cutpoints tractable, sequentially making locally optimal choices rarely leads to a globally optimal solution (Strobl et al. 2009).

Typically the quality of a cutpoint is measured in terms of the purity of the subspaces that it induces, with a subspace being considered pure when all the observations from $\mathcal{T}$ in it are of the same class. As a subspace is represented by a node in a DT, the quality of a cutpoint for a node, n, can then be measured in terms of the increase in purity between n and n's children. We are then looking for the cutpoint that gives the largest increase in purity, or alternately the largest decrease in impurity. One common measure of impurity used in classification tasks is the Gini index. This calculates the impurity of the set of observations, $\mathcal{N}$, in a node, n, using equation (1), with C being the set of classes that the observations in $\mathcal{N}$ belong to and p_c the proportion of the observations in $\mathcal{N}$ that belong to class c. The impurity of a binary cutpoint is then calculated as the weighted sum of the impurities of the two child nodes, L and R, as seen in equation (2), with $\mathcal{N}_L \subseteq \mathcal{N}$ the set of observations in child node L and $\mathcal{N}_R \subseteq \mathcal{N}$ the set in child node R. An example of the effect that different choices of cutpoint have on the increase in purity can be seen in Figure 27.

$$Gini(n) = 1 - \sum_{c \in C} {p_c}^2 \quad \textbf{(1)}$$

$$Gini_{split}(n) = \frac{\#\mathcal{N}_L}{\#\mathcal{N}} Gini(L) + \frac{\#\mathcal{N}_R}{\#\mathcal{N}} Gini(R) \quad \textbf{(2)}$$

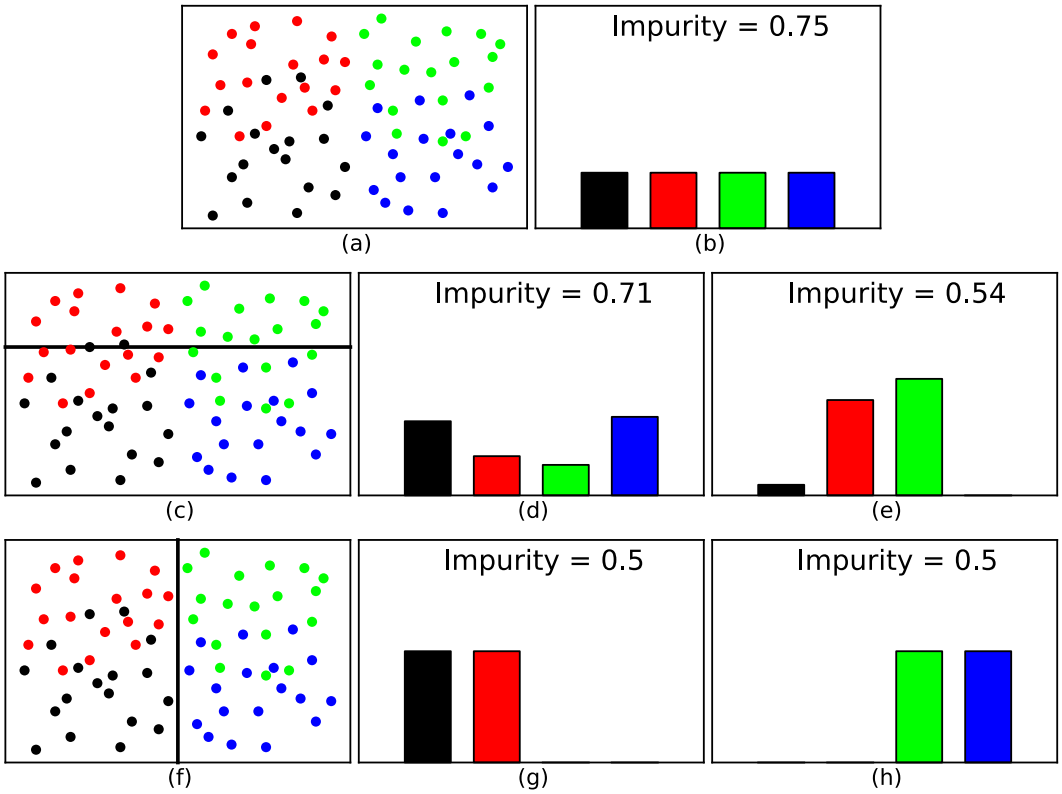

Figure 27: Effect of cutpoint choice on child node purity. Figures (a), (c) and (f) are plots of the four classes, indicated by the four different colours, in a two dimensional dataset. Figure (a) represents the data before it is partitioned by node n, and (b) the relative frequency of each class in the subspace represented by n. The horizontal black line in (c) demonstrates one possible cutpoint for node n, with (d) and (e) showing the relative frequency of each class in the two child nodes that would result from the cutpoint. The vertical black line in (f) represents an alternative cutpoint for n, with (g) and (h) showing the relative frequency of each class in the child nodes induced by the cutpoint. A cutpoint for a node induces purer children, and is therefore preferable, if there are fewer taller bars in the relative frequency plots for its children. Therefore, although the child represented by (e) is relatively pure, the cutpoints shown in (c) and (f) have split impurities of 0.66 and 0.5 respectively, indicating that the cutpoint in (f) is preferable.

Due to the recursive partitioning employed by DTs, they are highly susceptible to overfitting. If a DT training algorithm is run until all leaf nodes are pure, then the DT will classify the observations in $\mathcal{T}$ with perfect accuracy. This is due to the fact that the feature space will have been partitioned such that each hyperrectangle contains only observations of one class. Any observation $d \in \mathcal{T}$ that is classified by the DT, will therefore filter down to a leaf node containing d and possibly other observations of the same class as it. Unless $\mathcal{T}$ is highly representative of the data that can be drawn from $\mathcal{P}$, it is likely that the generalisation accuracy of the DT will be significantly lower than its accuracy on $\mathcal{T}$. However, this propensity for overfitting can be reduced by pruning the DT; the two most common approaches being pre- and post-pruning. In pre-pruning the growth of the DT is stopped before all the leaf nodes are pure (Kotsiantis 2007), while in post-pruning the DT is grown until the leaves are pure before backtracking and undoing cutpoints in order to make some leaves less pure.

Two further concerns when using DTs are the piecewise nature of the boundaries that they induce between classes, and their instability to changes in $\mathcal{T}$. As was mentioned previously, the univariate nature of the cutpoint at each node causes the partitioning hyperplanes to be orthogonal to one axis and parallel to all others. This causes the boundaries between classes induced by a DT to be piecewise, and therefore DTs can have difficulty when a diagonal partition of the entire feature space would be best (Kotsiantis 2007). Rather than a smooth diagonal boundary, the DT will induce a step-wise or 'staircase' boundary between the two classes (Figure 28). In addition to its step-wise nature, the boundary induced by a DT shows a high level of instability when $\mathcal{T}$ changes even slightly. The recursive partitioning that DTs use means that the choice of cutpoint in a node depends on the cutpoints in its ancestral nodes, which are all strongly dependent on the distribution of the observations in $\mathcal{T}$ (Strobl et al. 2009). This means that the entire structure of a DT can be altered if the cutpoint in the root node changes, which can happen from even small changes in $\mathcal{T}$. This instability of single DTs leads to their predictions showing high variability (Strobl et al. 2009).

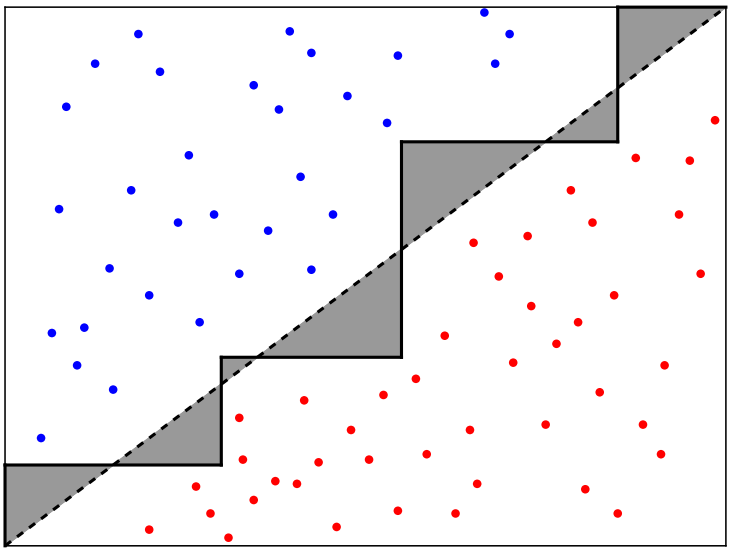

Figure 28: Illustration of the step-wise nature of the boundary induced by a DT. The dataset from Figure 22 is shown separated by the optimal SVM hyperplane, the dashed line, and a DT, the solid lines. The shaded area demonstrates the deviation of the DT boundary from the optimal SVM boundary.

4.2.3.1.4 Ensembles

One alternative to constructing a single optimal classifier is to generate an ensemble consisting of multiple suboptimal classifiers, or base learners. Rather than use the classification produced by a single classifier, an ensemble will gather the classifications from all the base learners in it, and then aggregate them to produce a consensus classification. This approach can prove to be beneficial in situations where choosing between potentially optimal classifiers, or

even finding the optimal classifier, is infeasible (Dietterich 1997). In these situations the ability of ensembles to pool a set of approximations to the optimal solution can be superior to using a single best approximation (Dietterich 1997).

A further advantage of ensembles is their ability to boost the performance of weak learners. A weak learner is a classifier that performs only marginally better than if it was randomly guessing an observation's class, and therefore the probability of it being correct deviates only slightly from 0.5. This is significant, as provided the probability of each individual in a population being correct is not 0.5, and individuals' predictions are not correlated, then $\lim_{n\to\infty} P = 1$, where n is the number of individuals in the population and P is the probability of the population being correct (Condorcet 1785). Therefore, the accuracy of an ensemble can theoretically be improved by adding more and more base learners to it. However, this relies on the classification errors of the individual base learners being uncorrelated, and therefore independent (Dietterich 1997). The accuracy of the ensemble will therefore only continue to increase provided that the diversity of the base learners does not decrease (Cord & Cunningham 2008).

Diversity in the set of base learners can be ensured by manipulating the set of features and/or observations in $\mathcal{T}$. Rather than training all the base learners on $\mathcal{T}$, each one is trained on a separate dataset, $\mathcal{T}_i$, created by selecting only a subset of the features and/or observations in $\mathcal{T}$. However, in order for this approach to provide the necessary diversity, the algorithm used to train the base learners must be sensitive to the composition of the training set, and therefore must be able to reliably induce different classifiers from the different $\mathcal{T}_i$. For this reason ensembles often make use of unstable classifiers, such as DTs, which also benefit from being in ensembles as the variability in their predictions can be 'smoothed out' by the process of aggregating the individual classifications (Strobl et al. 2009). In addition to enabling this 'smoothing out', the diversity allows the induction of a more flexible boundary between classes, which is often able to generalise better than the boundary induced by a single base learner (Wu et al. 2007).

Approaches for selecting the training set, $\mathcal{T}_i$, for each base learner can be categorised as sequential or parallel. Sequential approaches generate a series of training sets where each one depends on the training sets generated before it, while in a parallel approach all training sets are generated independently of one another. Boosting and bagging/RFs are common approaches to ensemble construction that utilise sequential and parallel approaches respectively. In boosting, the diversity of the ensemble is ensured by generating each training set, $\mathcal{T}_i$, from a sample of the observations in $\mathcal{T}$. The sampling process for each $\mathcal{T}_i$ is governed by a distribution of weights over the observations in $\mathcal{T}$, with observations with greater weight being more likely to be selected. By adjusting the weightings it is possible to alter the sampling distribution used to select the training set for the next base learner (Cord & Cunningham 2008). The weightings are updated between

the training of each of the base learners, in such a way that observations that were misclassified by the previous base learner get a larger weight, while ones that were classified correctly get a smaller weight. This serves to reorient the focus of the next learner towards observations that have been misclassified by previous ones. The final boosted ensemble makes use of all the trained base learners, but weights each learner's contribution to the ensemble's final classification according to a performance metric. The process of reorienting the focus of the learners and weighting each one's contribution to the final classification makes boosting a robust method for constructing ensembles. Additionally, it has been shown that they do not overfit, and can continue to improve their generalisation ability even after their error on $\mathcal{T}$ has reached zero (Wu et al. 2007).

Rather than training the base learners to focus on specific subsets of the observations as boosting does, bagging (Breiman 1996) attempts to improve upon a single classifier by training a set of base learners on random samples of the observations in $\mathcal{T}$. Bagging starts by drawing m samples of n observations from $\mathcal{T}$ uniformly and with replacement. A base learner is then trained on each of the m samples, to generate m base learners. As the samples that the base learners are trained on are drawn with replacement, some observations from $\mathcal{T}$ are likely to occur multiple times in a given sample m_i. Although this leads to the training sets containing duplicate observations, the strategy used by bagging is effective because each random sample is a reflection of the same process that generated the data in $\mathcal{T}$ (Strobl et al. 2009). Following the training of the m base learners, their predictions can be aggregated to form the final classification.

While bagging and boosting are both general frameworks, other approaches for creating ensembles are restricted to specific base learners. One extension to bagging that is limited to DTs are RFs (Breiman 2001). These are created in much the same way that a bagged ensemble of DTs would be, except that they alter the method by which the cutpoint for each node is chosen. Instead of considering all the features in $\mathcal{X}$ when determining the optimal cutpoint for a node, RFs consider only a random subset, $mtry_n \subseteq \mathcal{X}$, of features at each node, n. $mtry_n$ is chosen independently of the subsets selected at all other nodes, but will contain the same user specified number of features, $mtry$.

In addition to making training quicker, the selection of a random subset of features at each node allows RFs to inject added diversity into the ensemble when compared to bagging. This can be seen by considering a node n where the locally optimal cutpoint, c, is made using feature f. With bagged DTs, c will always be selected as the cutpoint for n. However, in RFs c can only be selected if $f \in mtry_n$, and therefore if $f \notin mtry_n$ a different cutpoint will be found for n. This makes it easier for weaker features to be included in the ensemble, and can potentially reveal interactions that would have been missed if the cutpoint was always chosen from amongst all the

features in $\mathcal{X}$ (Strobl et al. 2009). One final advantage of randomly choosing a subset of features at each node stems from the fact that the split selection process is only locally optimal. As making a series of locally optimal choices is not guaranteed to find a globally optimal solution, incorporating suboptimal cutpoints, as RFs do, may lead to better global performance.

In addition to the previously mentioned advantages of ensembles, RFs also offer a useful measure than can be used to analyse the forest: the out-of-bag (OOB) error estimate. In order to calculate the OOB error, the concept of an OOB observation must first be defined. Assuming that $\mathcal{T}_i$ is the random sample of observations from $\mathcal{T}$ that was used to train the ith DT, then the OOB observations on i, $OOB_i = \mathcal{T} - \mathcal{T}_i$, is the subset of observations from $\mathcal{T}$ that were not used to train i. The OOB error relies on predicting the class of each observation, $d \in \mathcal{T}$, using only those DTs for which d is OOB, i.e. all trees i where $d \in OOB_i$. By calculating the OOB error in this manner, you obtain an unbiased estimate of the generalisation accuracy of the RF. However, the accuracy of the OOB error estimate depends on the number of DTs in the forest. The required number is dependent on $\mathcal{T}$, but must be sufficiently large that each observation in $\mathcal{T}$ is OOB on enough DTs to get a reliable OOB classification for it. Assuming there are enough DTs in the forest, the OOB error rate should approach the generalisation error of the RF. However, the OOB error estimate has been found to slightly overestimate the actual generalisation error (Bylander 2002), and will therefore approach the generalisation error from above as the number of DTs in the forest increases. Pseudocode for the calculation of the OOB error can be seen in Figure 29.

1. $error_{OOB} \longleftarrow 0$
2. For each observation $t \in \mathcal{T}$:
3. Determine the set of trees, t_{OOB}, for which t is OOB.
4. Classify t using each tree in t_{OOB}.
5. Aggregate the predictions from step 4 to get an overall classification, c_t, for t.
6. Determine the real class, c, of t.
7. If $c \neq c_t$ then $error_{OOB} \longleftarrow error_{OOB} + 1$.
8. $error_{OOB} \longleftarrow {error_{OOB}}/{|\mathcal{T}|}$

Figure 29: Pseudocode for the calculation of the OOB error in a RF.

5 Feature-Based Druggability Prediction - Determining the Druggability of Human Proteins

5.1 Cleaning and Collation of Protein Data

When constructing a proteomics study, ideally all interesting properties of the investigated proteins would have experimentally derived values, preferably with multiple studies providing a body of evidence. However, biases in research topics and experimental limitations make this impossible for all but the simplest of properties, such as those calculated from the sequence. For all other properties, the trade-offs between the benefit of including the property and the potential biases in its data must be considered.

In the case of drug targets, some properties, such as protein protein interactions, variant numbers and expression levels, may potentially appear to be more common in targets than in non-targets due to the fact that targets are likely to have been more thoroughly characterised experimentally, rather than because of any true biological relationship. However, this should not present an impediment to their use, as the influence of these biases is likely to be more systematic than random. Therefore, despite the imperfection of the available data, results from proteomics studies can be used in drawing valid conclusion, provided that the results are interpreted in light of the limitations imposed by the data.

In addition to the general data concerns for a proteomics study like this one, druggability studies present an additional data based difficulty. As discussed in Section 1.3, predicting the druggability of proteins requires both a set of targets and non-targets. The lack of a gold standard set of non-targets is therefore a concern. Although a protein can be confirmed to be a target simply by being targeted by an approved drug, proving that a protein is a non-target requires, as a minimum, published evidence of the protein failing as a proposed target. However, this evidence is not published by drug companies, and even were it to be, there is always the question of how many times, and at what point in the development process, a protein must fail as a putative target before it is considered to be undruggable. It must therefore be noted that the non-targets used here are not validated as non-targets, but are instead simply ones that until now have not been successfully targeted by an approved drug. While this is less than ideal, the lack of any validated non-targets necessitates this approach to druggability studies.

5.1.1 Protein Accession and Name

The UniProt accessions and name of each human protein were extracted from an XML file containing all reviewed human proteins from UniProt release 2012_05, hereafter referred to as

the UniProt XML file. For each protein <entry> element in the file, the accessions were extracted from its <accession> child elements, and the protein's name from its <name> child element. The first <accession> element encountered in the record for a protein was taken to be the protein's representative accession. A mapping between non-representative and representative accessions was produced to enable cross referencing with external databases that may use non-representative accessions.

5.1.2 Simple Sequence Properties

Each protein's sequence was extracted from the <sequence> child element of its <entry> element in the UniProt XML file. Following the extraction of the sequence, its length was determined by counting the number of amino acid residues in it. Information about the presence or absence of a signal peptide was extracted from the <feature> child elements of a protein's <entry> element in the UniProt XML file. Any protein with a <feature> element where the value of the **type** attribute was "signal peptide" was deemed to contain a signal peptide.

The number of PEST motifs, peptide sequences rich in proline, glutamic acid, serine, and threonine, in each protein was calculated using epestfind (http://emboss.bioinformatics.nl/cgi-bin/emboss/epestfind). epestfind returns potential, poor and invalid PEST motifs. Only potential PEST motifs were counted. The number of PEST motifs returned by epestfind was summed to get the total number of PEST motifs for the protein. The program was run with the default parameters.

The number of low complexity regions was calculated using segmasker (Camacho et al. 2009). The number of low complexity intervals returned by segmasker was summed to get the total number of low complexity regions for a protein. The program was run with the default parameters.

The hydrophobicity of a protein was calculated to be the mean of the hydrophobicity values, as determined by the Kyte and Doolittle index (Kyte & Doolittle 1982), of the amino acids in its sequence. This was calculated by summing the hydrophobicity values of all the amino acids in the sequence, and then dividing by the sequence length. Residues B, J and Z were treated as they were for determining the frequency of the individual amino acids (Section 5.1.3).

The isoelectric point of each protein was calculated using the pepstats program (http://emboss.sourceforge.net/apps/cvs/emboss/apps/pepstats.html). The program was run using the -auto parameter.

5.1.3 Amino Acid Composition

Following the extraction of the sequence (Section 5.1.2), the number of occurrences of each of the twenty standard amino acids in the sequence was determined. Ambiguous amino acid codes (B, J and Z) were handled by incrementing the occurrence count for their corresponding amino acids (D/N for B, I/L for Q and E/J for Z) by 0.5. Therefore, each ambiguous code counts as half of each of the amino acids it corresponds to. As an example, a sequence containing 10 Ns and 5 Bs would be given a total occurrence count for N of 12.5. From these occurrence counts, the frequency with which each amino acid occurs in the protein's sequence was determined by dividing the count for the amino acid by the sequence length. Amino acids were also grouped into eight categories: tiny (A, C, G, S and T), small (A, C, D, G, N, P, S, T and V), aliphatic (I, L and V), aromatic (F, H, W and Y), non-polar (A, C, F, G, I, L, M, P, V, W and Y), charged (D, E, H, K and R), positively charged (H, K and R) and negatively charged (D and E). For each protein, the fraction of the amino acids in its sequence that belong to each of the categories was calculated. This was done by summing up the occurrence counts for each of the amino acids in the category, and then dividing by the length of the sequence.

5.1.4 Protein Family

Proteins were classified as being a GPCR, ion channel, kinase, protease or other. Protein family membership was determined using multiple UniProt sources. The first source was the <keyword> child elements of each protein's <entry> element in the UniProt XML file. A protein was determined to be a GPCR if the value of the **id** attribute of a <keyword> element was "KW-0297"; an ion channel if the value was one of "KW-1071", "KW-0851", "KW-0107", "KW-0869", "KW-0407", "KW-0631" or "KW-0894"; a kinase if the value was one of "KW-0418", "KW-0723" or "KW-0829" and a protease if value was one of "KW-0031", "KW-0064", "KW-0121", "KW-0224", "KW-0482", "KW-0645", "KW-0720", "KW-0788" or "KW-0888". A protein was also determined to be a GPCR, kinase or protease if it appeared in the GPCR (http://www.uniprot.org/docs/7tmrlist accessed May 14th 2012), kinase (http://www.uniprot.org/docs/pkinfam accessed May 14th 2012) or protease (http://www.uniprot.org/docs/peptidas accessed May 14th 2012) files respectively.

5.1.5 Posttranslational Modifications

Information about the glycosylation and phosphorylation sites of a protein was extracted from the <feature> child elements of the protein's <entry> element in the UniProt XML file. Information about a glycosylation site was extracted from a <feature> element when the value of its **type** attribute was "glycosylation site". The element's **description** attribute was used to determine whether the glycosylation was *N*-linked or *O*-linked. Information about a

phosphorylation site on the protein was extracted from a <feature> element when the value of its **type** attribute was "modified residue". The element's **description** attribute was used to determine whether a serine, threonine or tyrosine was phosphorylated. For each protein, the number of each of the five types of posttranslational modification site (*O*-glycosylation, *N*-glycosylation, phosphoserine, phosphothreonine and phosphotyrosine) was calculated. The data on phosphorylation sites extracted from UniProt in this manner was also used to calculate the total number of phosphorylation sites, of any type, for each protein.

5.1.6 Secondary Structure

NetSurfP (Petersen et al. 2009) was used to predict the fraction of residues in each protein that participate in exposed α-helices, buried α-helices or β-strands. Although accurate secondary structure information could be obtained from crystal structures, this information is unavailable for the majority of proteins. Predictions were therefore preferable to experimentally determined structural information.

Information about the number of α-helical transmembrane regions of each protein was extracted from the <feature> child elements of the protein's <entry> element in the UniProt XML file. A helical transmembrane region is recorded in a <feature> element when its **type** attribute is "transmembrane region" and the **description** attribute is present and contains 'Helical' (without quotes) as its first characters.

5.1.7 Protein Protein Interactions

The protein protein interaction (PPI) information for a protein was extracted from the <comment> child elements of the protein's <entry> element when the value of the **type** attribute was "interaction". PPIs recorded in UniProt can be binary or unary, and can record interactions between human and non-human proteins. For each protein, the number of unique human proteins that participate in a binary interaction with the protein was calculated.

5.1.8 External Database References

Data concerning the cross-referencing of UniProt accessions and external database identifiers was extracted from Ensembl (Flicek et al. 2012) using an automated BioMart (Kinsella et al. 2011) XML query. The NCBI Gene IDs, Ensembl Gene IDs, Ensembl Transcript IDs, Ensembl Peptide IDs and UniGene cluster IDs associated with each representative UniProt human protein accession were extracted using the XML query seen in Figure 30.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Query>
<Query client="pythonclient" processor="TSV" limit="-1" header="0" uniqueRows = "1">
    <Dataset name="hsapiens_gene_ensembl" config="ensembl">
        <Filter name="uniprot_swissprot_accession" value="" />
        <Attribute name="ensembl_gene_id" />
        <Attribute name="ensembl_transcript_id" />
        <Attribute name="ensembl_peptide_id" />
        <Attribute name="entrezgene" />
        <Attribute name="unigene" />
        <Attribute name="uniprot_swissprot_accession" />
    </Dataset>
</Query>
```

Figure 30: Ensembl BioMart XML query for extracting external database IDs. The value **attribute of the <Filter> element is a comma separated list of all the representative human UniProt accessions.**

5.1.9 UniGene Expression Clusters

Unigene (Wheeler 2003) was used to extract data relating to the expression profile of the human proteome. Individual transcripts in UniGene are grouped into clusters that are believed to come from the same locus. The expression profile of a cluster is then determined by counting the number of expressed sequence tags (ESTs) in it for each of the body sites and developmental stages recorded in UniGene. The external cross-references extracted from UniProt (Section 5.1.8) were used to map UniProt accessions to UniGene cluster IDs from UniGene build #232. A protein's expression in an individual body site or developmental stage was taken to be the sum of the ESTs in that body site or developmental stage across all UniGene clusters cross-referenced with the protein. In addition to the raw expression values, a derived feature was created that records the number of body sites in which the protein is expressed. This feature was calculated for each protein as the number of body sites in which the expression level was not 0.

5.1.10 Ensembl

Ensembl was used to extract information about the alternative transcripts, paralogs and germline variants of UniProt proteins. The first step in extracting the data was to use the results of the BioMeart XML query from Figure 30 to find the set, E, of all Ensembl Gene IDs that can be cross-referenced with a representative UniProt human protein accession. The number of protein coding transcripts, t_e, generated by each gene $e \in E$ was then determined using another automated BioMart XML query (Figure 31). Once the number of protein coding transcripts associated with each gene was known, the number associated with each representative UniProt human protein accession, p, could be calculated. This was done by first finding the set, $E_p \subseteq E$, of Ensembl Gene IDs that could be cross-referenced with p, and then taking $\max_{e \in E_p} t_e$ to be the number of protein coding transcripts associated with protein p.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Query>
<Query client="pythonclient" processor="TSV" limit="-1" header="0" uniqueRows = "1">
    <Dataset name="hsapiens_gene_ensembl" config="ensembl">
        <Filter name="ensembl_gene_id" value="" />
        <Attribute name="ensembl_gene_id" />
        <Attribute name="ensembl_transcript_id" />
        <Attribute name="transcript_count" />
        <Attribute name="transcript_biotype" />
    </Dataset>
</Query>
```

Figure 31: Ensembl BioMart XML query for extracting transcript information. The value **attribute of the <Filter> element is a comma separated list of the Ensembl Gene IDs in E.**

The number of paralogs associated with each representative UniProt human protein accession was determined using the Ensembl Perl API. For each Ensembl Gene ID, $e \in E$, the Ensembl Gene IDs, e_{hom}, of the genes calculated by Ensembl to be homologous to e were extracted. The set of Ensembl Gene IDs of genes paralogous to e, e_{par}, therefore consists of all genes, $h \in e_{hom}$, where the description field in the homology table entry for the homologous relationship between e and h is 'within_species_paralog'. The number of paralogs of a UniProt accession, p, is then calculated by first determining the set of Ensembl Gene IDs, E_p, that can be cross-referenced with p, and then $\#(\bigcup_{e \in E_p} e_{par})$, the number of unique paralogs of the genes in E_p.

The Ensembl Perl API was also used to extract information about the mutations associated with each representative UniProt human protein accession. In order to do this, every gene, $e \in E$, was associated with four sets of mutations:

- e_3, the set of mutations that occur within the 3' untranslated region of e.
- e_5, the set of mutations that occur within the 5' untranslated region of e.
- e_{non}, the set of nonsynonymous coding mutations that occur within e.
- e_{syn}, the set of synonymous coding mutations that occur within e.

In order to determine these four sets, the set of all mutations associated with e is first extracted. This set is then reduced to contain only those mutations that have at most one consequence for each transcript coded for by e. This cut down set of mutations is termed e_{mut}. e_3 is then the set of mutations in e_{mut} that occur within the 3' untranslated region of e. e_5, e_{non} and e_{syn} are formed from e_{mut} analogously. The number of each of the four mutation consequences of interest associated with a UniProt accession, p, was then calculated by first determining the set of Ensembl Gene IDs, E_p, that can be cross-referenced with p. The number of 3' untranslated region mutations associated with p is then $\sum_{e \in E_p} \#e_3$, the number of 5' untranslated region mutations

$\sum_{e \in E_p} \#e_5$, the number of nonsynonomous coding mutations $\sum_{e \in E_p} \#e_{non}$ and the number of synonymous coding mutations $\sum_{e \in E_p} \#e_{syn}$.

5.1.11 Protein Drug Targets

The protein drug targets were determined using the Therapeutic Target Database (TTD) version 4.3.02 (Zhu et al. 2010) and DrugBank version 3 (Knox et al. 2011). In order to determine the representative UniProt accessions of the approved targets in the DrugBank database, the XML file of the database was first parsed to determine the set, D, of approved small molecule drugs. A drug, d, was only included in D if the **type** attribute of its <drug> element was "small molecule" and the data associated with one of the <group>child elements of its <groups> child element was 'approved'. For each drug $d \in D$, the DrugBank target IDs, T_d, of the proteins that it targets were determined by extracting the **partner** attribute of each <target> child element of d's entry in the XML file. The set of DrugBank target IDs of every target of an approved small molecule is then $T = \bigcup_{d \in D} T_d$. Following this, the file of external database cross-references (http://www.drugbank.ca/system/downloads/current/drug_links.csv.zip accessed July 8th 2013) was parsed to determine the set, U, of UniProt accessions of the targets in T. The accessions in U were then converted to a set of representative accessions, U_{DB}, using the non-representative to representative accession mapping determined in Section 5.1.1.

The TTD database was parsed to extract the UniProt accessions of the targets of approved drugs (those targets with an identifier beginning with 'TTDS') recorded in the database. The accessions were subsequently converted to a set, U_{TTD}, of representative accessions using the non-representative to representative accession mapping created in Section 5.1.1. The accessions in $U_{TDD} - U_{DB}$, i.e. those accessions solely implicated as targets by the TTD, were then further analysed to ensure that the proteins they identify are the target of an approved small molecule drug.

The final number of proteins determined to be the target of an approved small molecule drug was 1324, of which 1249 were found in DrugBank and 313 in the TTD. 238 of the proteins were common to both sources, while 1011 were unique to DrugBank and 75 unique to the TTD.

5.1.12 Cancer Proteins

For the purposes of this work, a cancer protein is one that is implicated in causing cancer or is the target of an antineoplastic drug. Cancer proteins were determined using two sources: the Cancer Gene Census (CGC) (Futreal et al. 2004) and the FDA's database of approved drugs. The

CGC dataset (accessed on June 15th 2012) was parsed in order to determine the NCBI Gene IDs of genes with variants, germline or somatic, that are causally implicated in cancer. Using the cross-references determined in Section 5.1.8, the NCBI Gene IDs of these cancer genes were mapped to representative UniProt human protein accessions.

The FDA's Drugs@FDA database was downloaded (http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM054599.zip accessed April 2013), and processed to determine the set of approved antineoplastic drugs. All drugs approved by the FDA through March 2013 were manually evaluated for evidence of being indicated for antineoplastic use. For each drug, the approved indications for it were determined based on the label data stored by the FDA, or using DrugBank and the TTD if no label data was available. Drugs approved for supportive care (e.g. antiemetics and analgesics), adjunct treatment or non-cancerous cellular proliferation (e.g. actinic keratosis) were excluded from the list, while those approved for precancerous conditions (e.g. myelodysplastic syndrome) were included. Once the final set of approved antineoplastic drugs was created, the DrugBank and TTD IDs of the drugs were determined. The targets of these drugs, as recorded by DrugBank and the TTD, were then determined. These targets then had their representative UniProt accessions determined using the process described in Section 5.1.11.

5.1.13 Final Datasets Generated

The following 104 features were used in the construction of the protein datasets:

- Amino acid composition
    - Twenty amino acid frequencies (Section 5.1.3)
    - Eight amino acid category frequencies (Section 5.1.3)
- Simple sequence properties
    - Sequence length (Section 5.1.2)
    - The number of PEST motifs (Section 5.1.2)
    - The number of low complexity regions (Section 5.1.2)
    - The hydrophobicity of the protein (Section 5.1.2)
    - The isoelectric point (Section 5.1.2)
    - The presence of a signal peptide (Section 5.1.2)
- Posttranslational modifications
    - The number of *O*- and *N*-glyosylated sites (Section 5.1.5)
    - The number of phosphorylated serine, threonine and tyrosine sites (Section 5.1.5)
    - The total number of phosphorylated sites of any type (Section 5.1.5)

- Secondary structures
    - The number of α-helical transmembrane regions (Section 5.1.6)
    - The percentage of residues predicted to participate in an exposed α-helix (Section 5.1.6)
    - The percentage of residues predicted to participate in a buried α-helix (Section 5.1.6)
    - The percentage of residues predicted to participate in a β-strand (Section 5.1.6)
- Germline variants
    - The number of 3' untranslated region, 5' untranslated region, nonsynonymous coding and synonymous coding mutations (Section 5.1.10)
- Inter-protein relationships
    - The number of binary PPIs (Section 5.1.7)
    - The number of alternative transcripts (Section 5.1.10)
    - The number of paralogs (Section 5.1.10)
- Expression levels
    - Seven developmental stage expression levels (Section 5.1.9)
    - Forty-five body site expression levels (Section 5.1.9)
    - Derived feature recording the number of body sites the protein is expressed in (Section 5.1.9)

Six categories were created from the annotated human proteins. Within each category the proteins can be considered to be either positive or negative, positive proteins being those proteins that are approved drug targets and negative proteins those that are not. However, not all positive proteins will have been identified as such yet. Therefore, the set of negative proteins will contain both proteins that will never be the target of an approved drug and those that are not currently but will be in the future. The categories were therefore divided into positive and unlabelled proteins, rather than positive and negative, where the unlabelled proteins contain both negative and nominally mislabelled positive proteins. Each protein in the human proteome was evaluated against a set of criteria to determine which of the categories it belongs in, and then evaluated against a separate criterion for each category to determine whether it is a positive protein in that specific category. The six categories, along with their criteria, can be seen in Table 8.

Category Name	Criterion for Inclusion in Category	Criterion for Inclusion in Positive Class
AllTargets	All proteins are included.	The protein must be a target protein.
Cancer	The protein must be a cancer protein.	The protein must be the target of an antineoplastic drug.
GPCR	The protein must be a GPCR.	The protein must be a target protein.
IonChannel	The protein must be an ion channel.	The protein must be a target protein.
Kinase	The protein must be a kinase.	The protein must be a target protein.
Protease	The protein must be a protease.	The protein must be a target protein.

Table 8: Dataset inclusion criteria. The criteria that a protein must meet to be included in each of the dataset categories, along with the criterion that must be met for each dataset in order to be considered a positive protein in it.

5.2 Investigation of the Effects of Redundancy Removal

Most algorithms for removing redundancy from a protein dataset will define the distance between two proteins in the dataset to be a function of their sequences. However, when attempting to induce a classifier using the dataset, the proteins are embedded in a space defined by the dataset's features. The distance between two proteins in this space is therefore determined by the feature vectors that define them and the classification algorithm used, and may be independent of the sequence similarity distance. Therefore, the distance between two proteins during the redundancy removal may be substantially different to the distance between them during the induction of the classifier. If differences in the distance measures cause proteins that are distant in the feature space to be considered too similar by the redundancy removal algorithm, then the removal of one of the too similar proteins may cause information about the distribution of the proteins in the feature space to be lost, to the detriment of the induced classifier's capabilities.

In order to evaluate the effect that the difference between the two distance measures has on the induction of a classifier, non-redundant datasets were generated using multiple sequence identity thresholds, and then used to induce RFs. As a lower sequence identity threshold causes there to be a greater difference between the original dataset and the non-redundant one generated from it, using a range of thresholds enables classifier capability to be evaluated when the redundancy removal has different levels of influence on the dataset used for training (the non-redundant dataset). In order to compare the capabilities of the induced classifiers, they were used to classify the proteins in the entire dataset from which their non-redundant training dataset was generated. This enables a RF induced using a non-redundant dataset to be evaluated in terms of its capability of generalising to the entire dataset, and therefore allows the loss of information about the distribution of the proteins in the feature space, caused by the redundancy removal, to be assessed.

5.2.1 Approach Taken

In order to determine the optimal sequence identity threshold for generating the non-redundant dataset of each category, nine non-redundant datasets were created from each of the $CancerTarg$, $GPCR$, $IonChannel$, $Kinase$ and $Protease$ categories. The $AllTargets$ category was not tested as the number of proteins in the category makes the process of experimentally determining the optimal threshold prohibitively time consuming. Rather, the final threshold used was determined based on a consensus of the optimal thresholds for the other five categories. The pairwise sequence identity between all pairs of sequences in the human proteome (extracted from UniProt as in Section 5.1.2) was determined using PSI-BLAST version 2.2.25, from the NCBI BLAST+ package, with the default scoring matrix. Each protein was BLASTed sequentially against every other protein in order to determine all N^2 possible sequence identities. The arguments used for the BLASTing were:

- -evalue 1
- -inclusion_ethresh 0.0001
- -num_iterations 3
- -gap_trigger 18
- -num_descriptions 10000
- -num_alignments 10000
- -dbsize 0
- -outfmt "7 qseqid sseqid pident length evalue"

For each of the five categories, non-redundant datasets were created using the Leaf algorithm (Section 2.3.2.2) with pairwise sequence identity thresholds of 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%. A non-redundant dataset generated using a threshold of 100% contains all proteins in the category, i.e. no proteins are considered to be redundant. For each combination of pairwise sequence identity and category, redundancy was removed from the set of positive proteins and the set of unlabelled proteins in the category separately. The non-redundant positive and unlabelled proteins were then combined to make the final non-redundant dataset.

Each of the nine non-redundant datasets generated for a category underwent the same evaluation process. For a dataset, d, this process was:

1. Optimise the positive class weighting, with $mtry = 10$, using the approach in Section 5.3.1.
2. Perform feature selection using the CHC-GA algorithm in Section 5.3.2 with $mtry = 10$, $numberTrees = 1000$ and the positive class weighting found in step 1. The CHC-GA was

allowed to converge once, and the feature subset/random seed pair that induced the RF with the greatest G mean was taken to be the optimal combination.

3. Train a RF using the optimal positive class weighting, feature subset and random seed, along with $mtry = 10$ and $numberTrees = 1000$.
4. Using the RF generated in step 3, predict the class of each protein in d using the forest's OOB predictions, and the class of the redundant proteins in the category using the entire forest.
5. Use the predictions generated in step 4 to calculate the G mean of the forest.

For each category, the threshold that produced the dataset that led to the largest G mean in step 5 was deemed to be the optimal threshold to use. The optimal threshold is therefore the one that enables the induction of a RF that forms the best predictions of all proteins in the category.

5.2.2 Results

When classifying the proteins in a non-redundant $Cancer$ dataset, the threshold used to generate the dataset made little difference, as evidenced by the fact that the induced RFs all have G means within 0.02 of each other (Table 9). The G means of the classifications of the proteins in the entire $Cancer$ dataset show slightly more variation, but as the lowest G mean is no more than 0.03 lower than that achieved by the RF associated with the 100% threshold, the redundancy removal process is unlikely to have led to a substantial loss of the information in the entire $Cancer$ dataset. Despite this, the dataset generated using a 20% threshold induced a RF that classified the non-redundant proteins with a greater G mean than it did the entire set of proteins. It is therefore likely that the use of this particular threshold caused the distribution of the proteins in the non-redundant dataset to be different to that of the proteins in the entire dataset. The decision boundary induced using the non-redundant dataset would then fit the proteins in the entire dataset worse than it does those in the non-redundant dataset, thereby leading the RF to overfit the non-redundant dataset to the point where it classifies the entire dataset with a lower G mean.

Threshold	Non-redundant Observations (Pos/Unl)	Non-redundant Dataset G Mean	Entire Dataset				
			TP	FP	TN	FN	G Mean
20%	403 (178/225)	0.84	293	50	394	94	0.82
30%	519 (236/283)	0.83	309	53	391	78	0.84
40%	625 (285/340)	0.83	326	66	378	61	0.85
50%	695 (316/379)	0.84	328	67	377	59	0.85
60%	742 (343/399)	0.84	313	52	392	74	0.85
70%	785 (367/418)	0.84	312	53	391	75	0.84
80%	806 (379/427)	0.84	318	59	385	69	0.84
90%	818 (385/433)	0.85	318	55	389	69	0.85
100%	831 (387/444)	0.85	332	71	373	55	0.85

Table 9: Comparison of RFs induced using non-redundant subsets of the $Cancer$ dataset. For each threshold, a non-redundant dataset was generated using Leaf (Section 2.3.2.2) and used to induce a RF. The RF was then used to classify the proteins in both the non-redundant dataset it was trained on and the entire $Cancer$ dataset. The TPs/FNs are the number of positive proteins in the entire dataset predicted correctly/incorrectly, and the TNs/FPs are the number of unlabelled proteins predicted correctly/incorrectly.

The threshold used to generate the non-redundant $GPCR$ dataset had a substantial effect on the classifications of the proteins in both the non-redundant training set and the entire $GPCR$ dataset, with smaller thresholds generally leading to the induction of RFs with lower G means (Table 10). Despite this trend, the largest G mean on the non-redundant datasets is associated with a threshold of 20%. This is likely due to the large fraction of proteins removed at this threshold causing the remaining proteins to be sparsely distributed throughout the feature space, thereby making the decision boundary easier to optimise to fit the distribution of the non-redundant proteins. In addition to this, the 20% and 30% thresholds lead to the induction of RFs that classify the non-redundant proteins with a greater G mean than they do the entire set of proteins. As with the other datasets, this is likely due to the subset of proteins kept by the redundancy removal having a different distribution when compared to the proteins in the entire dataset, thereby leading the RF to overfit the non-redundant dataset to the point where it classifies the proteins in the entire dataset with a lower G mean. Unlike 20% and 30%, the thresholds between 40% and 90% led to the induction of RFs that performed better on the entire $GPCR$ dataset than on the non-redundant dataset they were trained on. This is likely due to the redundancy removal predominantly thinning out clusters of proteins in the feature space that belong to a single class, thereby disproportionately removing those proteins that are easier to classify and so reducing the G mean. If this is the case, the distribution of the proteins in the non-redundant dataset will be similar to that of the proteins in the entire dataset. The non-redundant dataset will therefore still contain enough information to correctly classify the vast majority of the removed proteins, and as a result the G mean will be greater when classifying the entire dataset. In addition, this shows that at these thresholds the redundancy removal and classification

distance measures are well correlated, causing the redundancy removal to remove proteins that are both similar in terms of sequence identity and their location within the feature space.

Threshold	Non-redundant Observations (Pos/Unl)	Non-redundant Dataset G Mean	Entire Dataset				
			TP	FP	TN	FN	G Mean
20%	57 (14/43)	0.87	80	211	501	35	0.70
30%	150 (39/111)	0.76	94	372	340	21	0.62
40%	276 (66/210)	0.76	93	104	608	22	0.83
50%	409 (86/323)	0.79	90	75	637	25	0.84
60%	556 (102/454)	0.82	93	81	631	22	0.85
70%	665 (111/554)	0.83	93	85	627	22	0.84
80%	735 (113/622)	0.85	98	95	617	17	0.86
90%	779 (114/665)	0.85	99	107	605	16	0.86
100%	827 (115/712)	0.86	100	109	603	15	0.86

Table 10: Comparison of RFs induced using non-redundant subsets of the $GPCR$ dataset. For each threshold, a non-redundant dataset was generated using Leaf (Section 2.3.2.2) and used to induce a RF. The RF was then used to classify the proteins in both the non-redundant dataset it was trained on and the entire $GPCR$ dataset. The TPs/FNs are the number of positive proteins in the entire dataset predicted correctly/incorrectly, and the TNs/FPs are the number of unlabelled proteins predicted correctly/incorrectly.

The results for the $IonChannel$ (Table 11), $Kinase$ (Table 12) and $Protease$ (Table 13) datasets show the same general trends as the $Cancer$ and $GPCR$ ones, such as there being less variation in the G means of the non-redundant dataset classifications and larger thresholds leading to the induction of RFs that better classify the entire dataset. All three datasets also exhibit the same trend that once the threshold falls below a certain value, 50% in the case of the $IonChannel$ dataset and 40% in the case of the $Kinase$ and $Protease$ datasets, the G mean of the classifications of the entire dataset becomes sizably less than the G mean of the classifications of the non-redundant dataset. Similar to the $Cancer$ and $GPCR$ datasets, this is likely due to differences in the distribution of the proteins in the non-redundant dataset and those in the entire dataset.

Threshold	Non-redundant Observations (Pos/Unl)	Non-redundant Dataset G Mean	Entire Dataset				
			TP	FP	TN	FN	G Mean
20%	68 (25/43)	0.82	114	79	86	41	0.62
30%	106 (41/65)	0.81	119	56	109	36	0.71
40%	146 (58/88)	0.80	128	59	106	27	0.73
50%	187 (76/111)	0.82	117	37	128	38	0.77
60%	227 (95/132)	0.81	125	30	135	30	0.81
70%	270 (124/146)	0.82	119	14	151	36	0.84
80%	306 (145/161)	0.85	121	13	152	34	0.85
90%	319 (155/164)	0.85	122	15	150	33	0.85
100%	320 (155/165)	0.85	122	15	150	33	0.85

Table 11: Comparison of RFs induced using non-redundant subsets of the $IonChannel$ dataset. For each threshold, a non-redundant dataset was generated using Leaf (Section 2.3.2.2) and used to induce a RF. The RF was then used to classify the proteins in both the non-redundant dataset it was trained on and the entire $IonChannel$ dataset. The TPs/FNs are the number of positive proteins in the entire dataset predicted correctly/incorrectly, and the TNs/FPs are the number of unlabelled proteins predicted correctly/incorrectly.

Threshold	Non-redundant Observations (Pos/Unl)	Non-redundant Dataset G Mean	Entire Dataset				
			TP	FP	TN	FN	G Mean
20%	102 (18/84)	0.79	51	196	371	43	0.60
30%	198 (26/172)	0.85	49	165	402	45	0.61
40%	332 (49/283)	0.78	75	184	383	19	0.73
50%	432 (67/365)	0.79	72	120	447	22	0.78
60%	497 (77/420)	0.81	77	132	435	17	0.79
70%	569 (83/486)	0.79	72	118	449	22	0.78
80%	625 (88/537)	0.80	72	112	455	22	0.78
90%	650 (94/556)	0.79	69	90	477	25	0.79
100%	661 (94/567)	0.80	72	98	469	22	0.80

Table 12: Comparison of RFs induced using non-redundant subsets of the $Kinase$ dataset. For each threshold, a non-redundant dataset was generated using Leaf (Section 2.3.2.2) and used to induce a RF. The RF was then used to classify the proteins in both the non-redundant dataset it was trained on and the entire $Kinase$ dataset. The TPs/FNs are the number of positive proteins in the entire dataset predicted correctly/incorrectly, and the TNs/FPs are the number of unlabelled proteins predicted correctly/incorrectly.

Threshold	Non-redundant Observations (Pos/Unl)	Non-redundant Dataset G Mean	Entire Dataset				
			TP	FP	TN	FN	G Mean
20%	117 (15/102)	0.90	37	92	380	22	0.71
30%	197 (25/172)	0.84	46	107	365	13	0.78
40%	312 (38/274)	0.86	46	91	381	13	0.79
50%	402 (49/353)	0.83	46	49	423	13	0.84
60%	464 (55/409)	0.85	47	53	419	12	0.84
70%	486 (57/429)	0.85	47	41	431	12	0.85
80%	496 (59/437)	0.86	49	46	426	10	0.87
90%	504 (59/445)	0.86	49	46	426	10	0.87
100%	531 (59/472)	0.87	50	49	423	9	0.87

Table 13: Comparison of RFs induced using non-redundant subsets of the $Protease$ dataset. For each threshold, a non-redundant dataset was generated using Leaf (Section 2.3.2.2) and used to induce a RF. The RF was then used to classify the proteins in both the non-redundant dataset it was trained on and the entire $Protease$ dataset. The TPs/FNs are the number of positive proteins in the entire dataset predicted correctly/incorrectly, and the TNs/FPs are the number of unlabelled proteins predicted correctly/incorrectly.

Despite the discussed commonalities in the results for the five datasets, there is a clear difference between the affect that the redundancy removal has on the $Cancer$ dataset and the affect that it has on the datasets based on protein family membership. This can be seen most easily through a comparison of the proportion of proteins in the entire dataset that remain following redundancy removal (Table 14). For all thresholds except 90%, the $Cancer$ dataset has the greatest proportion of proteins remaining, likely due to the more heterogeneous nature of the proteins in the dataset leading to fewer intra-class similarities between proteins (Section 5.4.7.3). This lower proportion of proteins removed is also likely responsible for the differences in the threshold at which the induced RFs classify the proteins in the non-redundant dataset with a greater G mean than those in the entire dataset, and for the performance of the RFs induced from non-redundant $Cancer$ datasets degrading at a much slower rate than those induced from non-redundant protein family based datasets.

Threshold	Fraction of Proteins Remaining				
	$Cancer$	$GPCR$	$IonChannel$	$Kinase$	$Protease$
20%	0.48	0.07	0.21	0.15	0.22
30%	0.62	0.18	0.33	0.30	0.37
40%	0.75	0.33	0.46	0.50	0.59
50%	0.84	0.49	0.58	0.65	0.76
60%	0.89	0.67	0.71	0.75	0.87
70%	0.94	0.80	0.84	0.86	0.92
80%	0.97	0.89	0.96	0.95	0.93
90%	0.98	0.94	1.00	0.98	0.95

Table 14: Fraction of the number of proteins in the entire dataset in each non-redundant dataset.

5.3 Development of Machine Learning Approach Used

5.3.1 Random Forest Parameter Optimisation

RFs rely on two primary parameters to control their growth: $mtry$, the size of the random subset of features evaluated at each node and $numberTrees$, the number of trees in the forest. In addition to these two parameters, each observation in the dataset can be given a weighting, in the form of a penalty/reward for incorrectly/correctly classifying it. In addition to simply representing the relative importance of an observation, weightings can be used to mitigate the detrimental effects of class imbalances in a dataset. Assuming a binary classification problem, an imbalance in class distribution will naturally cause the algorithm to prefer strong performance in classifying the majority class. However, by giving the observations of the minority class a larger weighting than those of the majority class it is possible to ensure that correctly/incorrectly classifying the minority class observations is rewarded/penalised more heavily, and therefore to force the classifier to treat both classes more equally.

In order to mitigate the class imbalance in the datasets used here, the weighting given to observations of the unlabelled class was held at 1 while that of the observations in the positive class was varied. A grid search was used to simultaneously optimise the value of the $mtry$ parameter and the positive class weighting. For each combination of $mtry$ and positive class weighting, 100 RFs were grown with $numberTrees = 1000$. The OOB predictions from each of the 100 forests were then collated in order to determine the total number of positive proteins predicted correctly (TPs) positive proteins predicted incorrectly (FNs), unlabelled proteins predicted correctly (TNs) and unlabelled proteins predicted incorrectly (FPs). The sensitivity (3) and specificity (4) of the predictions were then calculated, and used to determine the G mean (5) for the parameter combination. Once the search was complete, the optimal parameter combination for the dataset was taken to be the one that produced the forests with the greatest G mean. In order to ensure that the variation in the performance of the classifiers was solely dependent on changing $mtry$ and the positive class weighting, the same set of 100 random seeds were used to grow the RFs for each parameter combination.

Sensitivity	$TPs/(TPs + FNs)$	**(3)**
Specificity	$TNs/(TNs + FPs)$	**(4)**
G mean	$\sqrt{Sensitivity * Specificity}$	**(5)**

The G mean was used to evaluate the performance of the RFs due to the class imbalance and the equal importance assigned to correctly predicting observations of both classes. Commonly, either the accuracy (6) or error rate (7) is used to determine a classifier's quality. However, when training a binary classifier on an imbalanced dataset a high accuracy (or low error rate) can often be achieved simply by predicting all observations as belonging to the majority class. As this imbalance increases, the perceived quality of this majority predicting classifier will only increase. Therefore, despite classifiers of this form potentially having the largest possible accuracy, their classifications have no real meaning due to their complete disregard for the observations in the minority class. A better approach to measuring a classifier's performance on an imbalanced dataset is to use a quality measure that penalises poor performance on any class. Three examples of such measures are the F measure (8), Matthews correlation coefficient (9) and the aforementioned G mean. However, as we are placing equal importance on maximising both the specificity and sensitivity, the G mean is preferable as it is a direct average of the two.

Accuracy	$(TPs + TNs)/(TPs + FNs + FPs + TNs)$	**(6)**
Error rate	$(FPs + FNs)/(TPs + FNs + FPs + TNs)$	**(7)**
F measure	$2 * TPs/(2 * TPs + FNs + FPs)$	**(8)**
MCC	$\frac{(TPs * TNs) - (FPs * FNs)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}$	**(9)**

The RF implementation created for this work can be found at https://github.com/SimonCB765/RandomForest.

5.3.2 Feature Selection

Feature selection was performed using a modified CHC genetic algorithm (CHC-GA) (Eshelman 1991). A CHC-GA combines a conservative selection strategy, preserving the best individuals after every generation, with a crossover operator that is both disruptive and produces offspring that are maximally different to their parents. Given a set of features, F, and the number of individuals to generate in each generation, P, the algorithm first sets the $threshold$ parameter to $\#F/4$. The $threshold$ parameter controls how dissimilar two parents must be before they can be combined to produce offspring via crossover. This functions as a form of incest prevention, since parents that are too similar are prevented from mating.

Following the initialisation of the $threshold$ parameter, the initial population P_0 is created. Each individual in P_0 is initialised to be a set of $\#F/2$ features. The distribution of the features throughout P_0 is controlled in such a way that for every pair of features, $i, j \in F$, the

number of individuals in P_0 that contain i is at most one more than the number that contains j. This ensures that, provided the size of the population is large enough, no feature's influence is curtailed due to the vagaries of the initialisation process. Following the initialisation of P_0, the fitness of each individual, $p_i \in P_0$, is calculated by growing a RF from the features in $F - p_i$. The G mean is then calculated from the OOB predictions of the RF, with larger values for the G mean corresponding to fitter individuals. The RF is grown using the optimal parameters determined in Section 5.3.1.

After initialising P_0 and calculating the fitness of each individual in it, the generational portion of the GA can begin. The first step in each generation, g, is to select the pairs of parents, M_g, that will undergo crossover in order to produce a set of offspring, O_g. This is done by randomly selecting $P/2$ pairs of individuals, with replacement, from P_g. In order to undergo crossover, the Hamming distance between the parents in a given pair, $(p_i, p_j) \in M_g$, must be greater than the $threshold$. As individuals in P_g are sets of features, the Hamming distance between any two individuals is equal to the number of features in their symmetric difference, i.e. $\#(p_i \triangle p_j)$. Provided that the parent pairing is permitted to undergo crossover, half uniform crossover is used to produce two offspring by swapping a random subset of half the features that differ between the parents, i.e. half the features in $p_i \triangle p_j$. Assuming that the set of features to be swapped is s, the two offspring, o_i and o_j, are created such that $o_i = p_i \triangle s$ and $o_j = p_j \triangle s$. If no offspring are created through crossover, then the $threshold$ is decremented by one. This ensures that future pairs of parents do not have to be as dissimilar as the pairs in the current generation. Once the $threshold$ reaches 0 the run stops as the population is deemed to have converged.

After O_g has been produced, the fitness of each individual in it is calculated in the same manner used for the individuals in P_0. The population for the next generation, P_{g+1}, is then determined from O_g and P_g by setting P_{g+1} to be the fittest individuals from $P_g \cup O_g$, such that $\#(P_g) = \#(P_{g+1})$. It is possible, especially when a run nears convergence, that the members of O_g will all be less fit than the members of P_g, causing $P_g = P_{g+1}$. If this occurs repeatedly the run can get stuck in a loop, whereby the $threshold$ does not decrease because the individuals in P_g are not too similar, but the offspring are not fit enough to make it into P_{g+1}. In order to prevent this, a check is inserted that causes the $threshold$ to be decremented by one if a set number of generations pass without the population changing.

The CHC-GA implementation created for this work can be found at https://github.com/SimonCB765/RandomForest.

5.4 Identification of Targets and Their Properties

5.4.1 Approach Taken

For each category in Section 5.1.13, the values for the positive class weighting and $mtry$ parameters were optimised using the approach in Section 5.3.1. Once the optimal parameter values had been found, feature selection was performed using the CHC-GA algorithm from Section 5.3.2. In order to allow the GA to converge to potentially different feature sets, multiple repetitions of the CHC-GA were performed. These repetitions were repeated with different values for the $numberTrees$ parameter, in order to determine the forest size that gave the best performance. The values of $numberTrees$ tested were 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500 and 5000. The optimal feature set, random seed and $numberTrees$ value were taken to be those that induced the fittest individual across all CHC-GA repetitions.

Once the optimal feature subset/random seed pair had been determined, the final predictions for the proteins in the category were generated from a RF trained on the dataset using the optimal positive class weighting, $mtry$, $numberTrees$, feature subset and random seed. The final classification for an individual protein consists of two parts: the RF's weighted vote for the unlabelled class and its weighted vote for the positive class. From these two values the positive similarity of a protein can be calculated as the fraction of the RF's total vote for the positive class. This similarity can be thought of as the confidence of the RF in its prediction, and can therefore be used as a measure of a protein's drug target likeness. The final classification of a protein can then be determined from its similarity by defining a cutoff, such that proteins are classified as positive only if they have a positive similarity above the cutoff. A cutoff of 0.5 was used here, as a similarity greater than this indicates that the majority of the RF's votes were for the positive class.

In addition to forming predictions for the class of each protein in a category, the importance of the features in the category's dataset was determined using a test of statistical significance and calculating a measure of the size of the effect of the difference between the positive and unlabelled observations. The statistical significance of each feature was determined using a two-tailed Mann-Whitney U test, with significance determined at the 0.05 level and multiple comparisons corrected for using the Bonferroni method. The effect size was calculated by estimating the probability of superiority (PS), also known as the common language effect size (McGraw & Wong 1992), using equation (10), where U is the U statistic of the positive observations from the Mann-Whitney U test, m the number of positive observations in the dataset and n the number of unlabelled observations. Here the PS is the fraction of all possible pairs of a positive and unlabelled protein in which the positive observation has a greater value for the feature than the unlabelled observation. The expression levels extracted from UniGene were

not tested for significance, but the derived feature recording the number of body sites a protein is expressed in was. Similarly, the proportions of tiny and small amino acids were not tested for significance.

$$PS = U/m * n \quad \textbf{(10)}$$

5.4.2 Results – Human Proteome

5.4.2.1 Target Properties

The results from the analysis of the features in the $AllTargets$ dataset (using the method described in Section 5.4) can be seen in Table 15. Compared to the unlabelled proteins in the dataset, the positive ones are proportionally more non-polar ($PS = 0.65$). Of the individual amino acid proportions, the only non-polar amino acids that occur in a greater proportion in unlabelled proteins are cysteine and proline, and the only polar amino acids that occur in a greater proportion in positive proteins are asparagine, aspartic acid and threonine. However, as the effect sizes for all of these is either small or very small, the differences in the proportions of the individual amino acids can be seen to be largely in line with the difference in proportion of non-polar amino acids. Although the effects of the differences in non-polar and individual amino acids are not large, together they indicate that the positive proteins are consistently more non-polar than the unlabelled ones. This is further demonstrated by the fact that the positive proteins are moderately more hydrophobic ($PS = 0.67$), as would be expected due to their greater proportion of non-polar amino acids and smaller proportion of polar ones. As positive proteins are more likely to contain a transmembrane helix (43% of positive proteins compared to 24% of unlabelled ones), tend to have a greater number of transmembrane α-helices ($PS = 0.60$) and have a greater percentage of their residues in buried α-helices ($PS = 0.66$), the amino acid composition results are likely due to membrane proteins making up a greater fraction of the set of positive proteins. This is perhaps unsurprising due to the large fraction of membrane proteins (e.g. GPCRs and transport proteins) that are believed to be targeted by approved drugs, and the vital roles in transport and signal transduction that many membrane proteins play. Besides the amino acid proportions, the only other feature with a sizeable effect was the number of *N*-linked glycosylation sites. As *N*-linked glycosylation has been associated with increased protein stability and protection against degradation and denaturation, in addition to ensuring the correct folding of proteins (Dalziel et al. 2014; Helenius & Aebi 2004), the greater number of *N*-linked glycosylation sites likely indicates that positive proteins have a greater half-life *in vivo*. Additionally, due to glycosylation almost solely occurring on extracellular amino acids, proteins with glycosylation sites are predominantly either transmembrane or secreted. Therefore, the greater number of *N*-linked glycosylation sites on positive proteins provides further evidence that they tend to be membrane bound.

Protein interaction and pathway data from KEGG (Kanehisa et al. 2014), Reactome (Croft et al. 2014) and STRING (Jensen et al. 2009) were also analysed for the proteins in the dataset. However, the low coverage of these databases made accurate analyses of the proteins in the dataset infeasible. Analysis of the enrichment of Gene Ontology (Ashburner et al. 2000) terms between the unlabelled and positive proteins was also investigated using the DAVID (Huang et al.

2009) functional annotation tool. However, when setting the background to the entire proteome and looking for enriched terms in the positive proteins, no terms were found to be significantly enriched. Identical results were also found when checking for enrichment in the unlabelled proteins.

Feature	P-value	PS	Positive Median	Unlabelled Median
Alanine *	3.47×10^{-04}	0.53	0.07	0.07
Arginine *	1.28×10^{-13}	0.44	0.05	0.06
Asparagine *	1.33×10^{-15}	0.57	0.04	0.03
Aspartic Acid *	5.90×10^{-08}	0.54	0.05	0.05
Cysteine	1.53×10^{-01}	0.49	0.02	0.02
Glutamic Acid *	3.71×10^{-19}	0.43	0.06	0.07
Glutamine *	2.57×10^{-65}	0.36	0.04	0.04
Glycine *	2.19×10^{-10}	0.55	0.07	0.06
Histidine *	1.35×10^{-05}	0.46	0.02	0.02
Isoleucine *	1.10×10^{-72}	0.65	0.05	0.04
Leucine *	3.33×10^{-05}	0.53	0.10	0.10
Lysine	1.80×10^{-01}	0.49	0.05	0.05
Methionine *	1.31×10^{-33}	0.60	0.02	0.02
Phenylalanine *	5.31×10^{-78}	0.65	0.04	0.04
Proline *	9.94×10^{-12}	0.44	0.05	0.06
Serine *	1.37×10^{-60}	0.37	0.07	0.08
Threonine	2.87×10^{-03}	0.52	0.05	0.05
Tryptophan *	4.64×10^{-24}	0.58	0.01	0.01
Tyrosine *	1.61×10^{-52}	0.63	0.03	0.03
Valine *	7.98×10^{-64}	0.64	0.07	0.06
Aliphatic *	6.09×10^{-70}	0.65	0.22	0.20
Aromatic *	6.68×10^{-56}	0.63	0.12	0.10
Charged *	1.61×10^{-23}	0.42	0.24	0.26
Negatively Charged *	2.05×10^{-06}	0.46	0.11	0.11
Non-polar *	1.24×10^{-72}	0.65	0.56	0.53

Feature	P-value	PS	Positive Median	Unlabelled Median
Positively Charged *	7.98×10^{-23}	0.42	0.13	0.14
Sequence Length *	2.13×10^{-14}	0.56	474	410
PEST Motifs *	2.66×10^{-13}	0.45	0	0
Low Complexity Regions *	1.83×10^{-08}	0.45	2	2
Hydrophobicity *	3.28×10^{-93}	0.67	-0.19	-0.38
Isoelectric Point	1.31×10^{-01}	0.49	7.31	7.47
Signal Peptide *	8.10×10^{-11}	0.53	0	0
***O*-glycosylation Sites ***	3.62×10^{-04}	0.51	0	0
***N*-glycosylation Sites ***	1.35×10^{-64}	0.60	0	0
Phosphoserine Sites	7.02×10^{-01}	0.50	0	0
Phosphothreonine Sites	3.02×10^{-02}	0.51	0	0
Phosphotyrosine Sites *	1.66×10^{-25}	0.54	0	0
Total Phosphorylation Sites *	1.98×10^{-04}	0.53	0	0
Transmembrane α-helices *	3.16×10^{-62}	0.60	0	0
Exposed α-helices *	1.92×10^{-05}	0.54	0.13	0.12
Buried α-helices *	2.47×10^{-89}	0.66	0.22	0.14
β Strands *	2.40×10^{-12}	0.56	0.12	0.09
3' Untranslated	7.32×10^{-01}	0.50	1	1
5' Untranslated	3.41×10^{-01}	0.51	0	0
Nonsynonymous Coding *	6.66×10^{-16}	0.57	15	11
Synonymous Coding *	2.50×10^{-10}	0.54	0	0
Binary PPIs *	5.02×10^{-14}	0.56	1	0
Alternative Transcripts *	2.44×10^{-18}	0.57	3	2
Paralogs *	5.73×10^{-07}	0.53	0	0
Body Sites Expressed In *	5.31×10^{-12}	0.56	27	26

Table 15: Results of the feature analysis for the $AllTargets$ dataset. The p-values and the PS were calculated as in Section 5.4. Shaded features are ones for which the $PS \geq 0.5$. The amino acid, exposed α-helix, buried α-helix and β strand features are all proportions (e.g. the Alanine feature for a protein is the number of alanine residues in the sequence divided by the sequence length), while all other features are absolute numbers. Features with significant differences are indicated with an *.

5.4.2.2 Target Predictions

The best combination of parameters and feature set for classifying the proteins in the $AllTargets$ dataset was $numberTrees = 1000$, $mtry = 5$, a weight of 110 given to each observation in the in positive class, a random seed of 3079726279227244970 and the following forty features out of the original 105 from Section 5.1.13:

- Amino acid compositions
    - The proportion of cysteine, glycine, leucine, serine, tyrosine and aromatic residues.
- Simple sequence properties
    - The hydrophobicity, isoelectric point, number of low complexity regions, presence of a signal peptide and sequence length.
- Posttranslational modifications
    - The number of *N*-linked glycosylation, *O*-linked glycosylation and phosphotyrosine sites.
- Secondary structure
    - The fraction of residues predicted to participate in β-strands, buried α-helices and exposed α-helices.
    - The number of transmembrane α-helices.
- Germline variants
    - The number of 3' untranslated region mutations
- Inter-protein relationships
    - The number of binary PPIs.

- Developmental stage expression:
    - The neonate expression level.
- Body site expression
    - The bladder, brain, connective tissue, ear, embryonic tissue, esophaus, eye, heart, larynx, liver, lung, salivary gland, skin, spleen, testis, tonsil, umbilical cord and uterus expression levels.
    - The number of body sites that the protein is expressed in.

The positive similarity of the proteins in the $AllTargets$ dataset can be seen in Figure 32. Using a cutoff of 0.5, the RF's predicted classifications were as follows:

Positive Observations				Unlabelled Observations				G Mean
Total	TPs	FNs	Sensitivity	Total	TNs	FPs	Specificity	
1324	1018	306	0.77	18919	15021	3898	0.79	0.78

A G mean of 0.78 indicates that the RF had difficulty classifying proteins in the $AllTargets$ dataset. While this may be due to the inadequacy of RFs for the task, the performance of the RFs on other datasets indicates that it is more likely due to the $AllTargets$ dataset itself. As the $AllTargets$ dataset contains a more heterogeneous set of proteins, due to there being no additional membership criteria, the distinction between positive and unlabelled proteins may be more difficult to make, as unlabelled proteins in one family may overlap with positive proteins in others. Additionally, proteins from smaller families will likely form poorly defined clusters. These proteins will therefore be more difficult to classify correctly, and will also increase the difficulty involved in the classification of the proteins in the larger families. In order to test this theory, two new datasets were created from the $AllTargets$ dataset. The first dataset, $LargeFamilies$, consisted of all proteins in the $GPCR$, $IonChannel$, $Kinase$ and $Protease$ datasets, and the second dataset, $SmallFamilies$, consisted of all proteins in $AllTargets - LargeFamilies$. RFs were optimised for these two datasets using the procedures described in Section 5.3. The optimised RFs were then used to classify the proteins in the dataset that they were trained on (Section 5.4). The G mean of the RF optimised for the $LargeFamilies$ dataset was 0.81, and the G mean of the RF optimised for the $SmallFamilies$ dataset was 0.76. As expected, the proteins in the smaller families were more difficult to classify, and the proteins in the larger families were classified with a G mean greater than that of the $AllTargets$ dataset. These results indicate that it is likely to be the combination of the protein families that makes accurate classifications more difficult, and that including smaller families is detrimental to the classification of proteins in general.

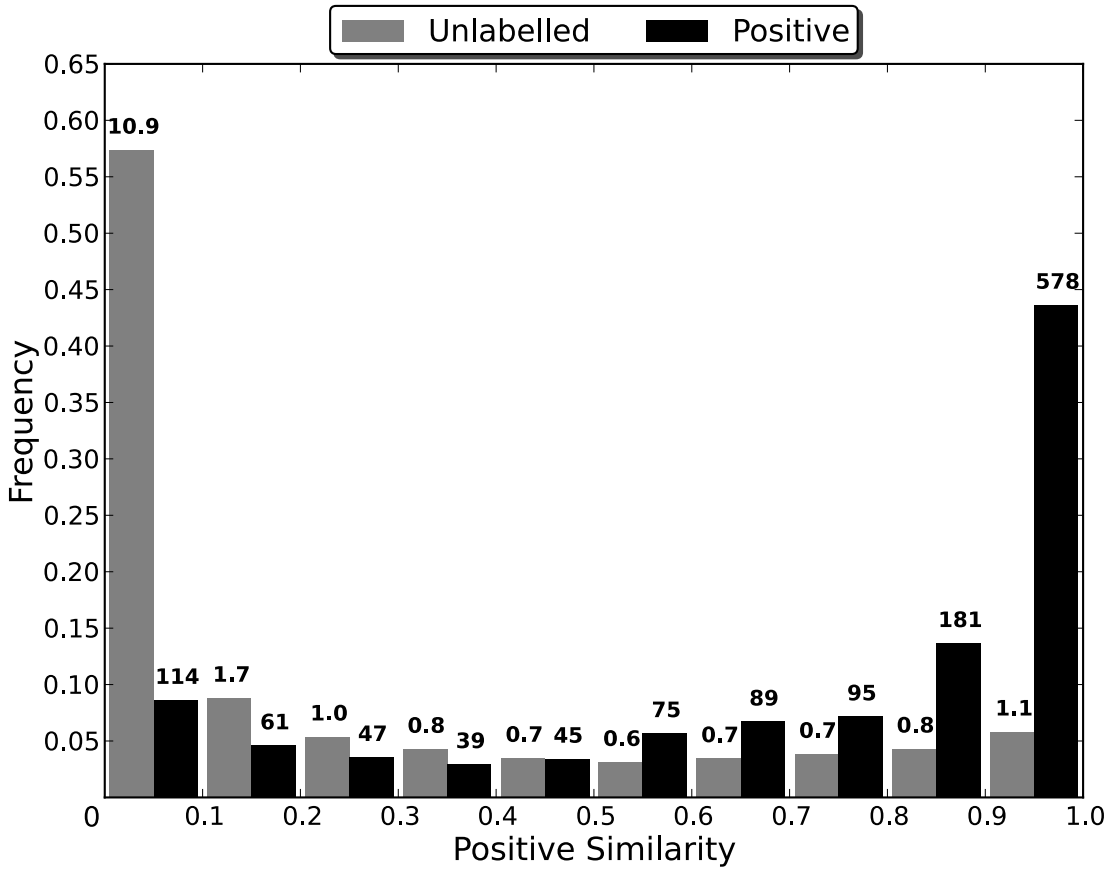

Figure 32: Weighted predictions of the proteins in the $AllTargets$ dataset. The positive similarity of a given protein is equal to the fraction of the forest's votes that are for the positive class. The values over the bars indicate the number of proteins in the bin (in raw numbers for the positive (black) bars and in thousands for the unlabelled (grey) bars). The $AllTargets$ dataset contained 18919 unlabelled proteins and 1324 positive ones.

5.4.3 Results - Cancer Proteins

5.4.3.1 Target Properties

The results from the analysis of the features in the $Cancer$ dataset (using the method described in Section 5.4) can be seen in Table 16. Compared to the unlabelled proteins in the dataset, the positive ones have a much greater proportion of non-polar amino acids ($PS = 0.74$). Additionally, the only polar amino acids that occur in a greater proportion in positive proteins are asparagine and threonine (both of which have inconsequential differences in their proportions), while proline is the only non-polar amino acid that occurs in a greater proportion in unlabelled proteins. The positive proteins are also substantially more hydrophobic ($PS = 0.82$), as would be expected due to their greater proportion of non-polar amino acids and smaller proportion of polar ones. As positive proteins are more likely to contain a transmembrane helix than unlabelled ones (55% compared to 11%), tend to have a much greater number of transmembrane helices ($PS = 0.73$) and have a much greater percentage of their residues in buried α-helices ($PS = 0.75$), the amino acid composition results are likely due to membrane proteins making up a greater fraction of the set of positive proteins.

As entry to the secretory pathway in humans is controlled by the presence of a signal peptide at the N-terminus of a protein, positive proteins are slightly more likely to be secreted than unlabelled ones due to their increased likelihood of containing a signal peptide ($PS = 0.61$). Additionally, the positive proteins in the $Cancer$ dataset are likely to have a longer *in vivo* half-life, due to their greater number of *N*-linked glycosylation sites ($PS = 0.71$), which have been associated with a longer half-life *in vivo*, and smaller number of PEST motifs ($PS = 0.40$), which are associated with proteins with a shorter intracellular half-life (Rogers et al. 1986).

The results also indicate that specific and reliable activity of a cancer protein is likely important in its being targeted by antineoplastic drugs. One example of this is the smaller number of 5' untranslated ($PS = 0.34$), 3' untranslated ($PS = 0.32$) and nonsynonymous coding ($PS = 0.38$) variants that are found in the positive proteins. As the untranslated regions of a gene are important for the regulation of mRNA translation and protein expression (Chatterjee & Pal 2009; Li & Lu 2013; Mignone et al. 2002) and nonsynonymous coding variants can lead to alterations in the expression and structure/function of a protein, the activity of a protein with fewer of these variants is likely to be more consistent between individuals.

Further examples of the preference for proteins with reliable activity come from the smaller number of phosphorylation sites ($PS = 0.40$), binary PPIs ($PS = 0.42$) and low complexity regions ($PS = 0.38$) that are found in positive proteins. As protein phosphorylation is frequently altered in cancerous cells, by having fewer phosphorylation sites it is possible that the positive proteins will be less affected by aberrant phosphorylation, thereby ensuring that their activity and

its regulation is minimally affected by the cancerous microenvironment. Participating in fewer binary PPIs can also be seen in this light, as a limited set of interactions may make a protein's activity less susceptible to alterations in the activity or regulation of other proteins. Similarly, it has been shown that low complexity region containing proteins have more binding partners (Coletta et al. 2010), that hub proteins in PPI networks contain significantly more low complexity regions (Dosztányi et al. 2006; Ekman et al. 2006) and that many known disordered regions in proteins are implicated in signalling and regulation (Romero et al. 2004). It therefore seems likely that low complexity regions enable a protein to interact with other proteins more readily, whether in a signalling or regulatory capacity. Having fewer of them may then indicate that a protein is involved in fewer interactions with other proteins, which would in turn imply that the protein's activity and expression is less amenable to modification by the cancerous microenvironment.

The smaller number of germline mutation variants of all types in positive proteins is possibly a reflection of the predisposition to cancer caused by some germline mutations, or may indicate that having fewer viable germline variants means that a protein is less amenable to somatic mutations that leave the protein functional. This would be advantageous for an antineoplastic target, as the cancer microenvironment makes it more likely that genetic mutations will arise in the gene coding for a given protein. If these mutations leave the protein functional, then drugs targeting the protein could have unexpected effects. By targeting proteins that are less susceptible to mutations that leave them viable, the activity of an antineoplastic drug would be more reliable, as the expression and function of the protein itself is more reliable.

The expression of the positive proteins in fewer body sites ($PS = 0.34$) means that the effects of a drug's modulatory activity can be limited to a more specific range of tissues. Not only can this help to limit undesirable side effects, but also to restrict the activity of the drug to a narrow range of tissues where the cancerous cells originate from. This may be particularly important for antineoplastic drugs, as they can often be more harmful to normal cells than non-antineoplastic medications.

Feature	P-value	PS	Positive Median	Unlabelled Median
Alanine	1.46×10^{-01}	0.53	0.07	0.07
Arginine *	8.88×10^{-05}	0.42	0.05	0.05
Asparagine	5.26×10^{-02}	0.54	0.04	0.04
Aspartic Acid	1.22×10^{-02}	0.45	0.05	0.05
Cysteine *	4.34×10^{-09}	0.62	0.02	0.02
Glutamic Acid *	3.73×10^{-12}	0.36	0.06	0.07
Glutamine *	2.43×10^{-25}	0.29	0.04	0.05
Glycine	8.61×10^{-01}	0.50	0.06	0.06
Histidine	1.80×10^{-03}	0.44	0.02	0.02
Isoleucine *	4.15×10^{-27}	0.72	0.05	0.04
Leucine *	4.44×10^{-16}	0.66	0.10	0.09
Lysine	1.29×10^{-03}	0.44	0.05	0.06
Methionine *	1.25×10^{-05}	0.59	0.02	0.02
Phenylalanine *	1.80×10^{-42}	0.77	0.04	0.03
Proline *	2.04×10^{-13}	0.35	0.05	0.07
Serine *	7.94×10^{-10}	0.38	0.07	0.08
Threonine	8.66×10^{-03}	0.55	0.05	0.05
Tryptophan *	6.38×10^{-27}	0.72	0.02	0.01
Tyrosine *	1.11×10^{-15}	0.66	0.03	0.02
Valine *	2.70×10^{-30}	0.73	0.07	0.05
Aliphatic *	5.83×10^{-43}	0.78	0.22	0.19
Aromatic *	1.85×10^{-35}	0.75	0.12	0.09
Charged *	5.95×10^{-16}	0.34	0.24	0.26
Negatively Charged *	3.09×10^{-10}	0.37	0.11	0.12
Non-polar *	1.26×10^{-33}	0.74	0.55	0.51

Feature	P-value	PS	Positive Median	Unlabelled Median
Positively Charged *	2.63×10^{-15}	0.34	0.13	0.14
Sequence Length	4.15×10^{-01}	0.48	505	557
PEST Motifs *	1.06×10^{-08}	0.40	0	1
Low Complexity Regions *	1.17×10^{-09}	0.38	2	4
Hydrophobicity *	1.41×10^{-57}	0.82	-0.19	-0.57
Isoelectric Point	1.56×10^{-01}	0.53	7.04	6.81
Signal Peptide *	1.11×10^{-15}	0.61	0	0
***O*-glycosylation Sites**	7.76×10^{-01}	0.50	0	0
***N*-glycosylation Sites ***	2.81×10^{-38}	0.71	1	0
Phosphoserine Sites *	8.17×10^{-12}	0.37	0	1
Phosphothreonine Sites *	2.13×10^{-06}	0.42	0	0
Phosphotyrosine Sites	3.30×10^{-03}	0.54	0	0
Total Phosphorylation Sites *	3.31×10^{-07}	0.40	1	2
Transmembrane α-helices *	5.21×10^{-45}	0.73	1	0
Exposed α-helices	2.28×10^{-01}	0.52	0.12	0.11
Buried α-helices *	1.65×10^{-35}	0.75	0.22	0.10
β Strands *	2.67×10^{-06}	0.59	0.10	0.06
3' Untranslated *	2.55×10^{-19}	0.32	0	3
5' Untranslated *	1.36×10^{-16}	0.34	0	2
Nonsynonymous Coding *	3.62×10^{-09}	0.38	14	29
Synonymous Coding *	2.04×10^{-04}	0.44	0	0
Binary PPIs *	6.83×10^{-05}	0.42	1	2
Alternative Transcripts	1.08×10^{-02}	0.45	3	4
Paralogs	5.50×10^{-03}	0.45	0	0
Body Sites Expressed In *	3.93×10^{-16}	0.34	24	32

Table 16: Results of the feature analysis for the $Cancer$ dataset. The p-values and the PS were calculated as in Section 5.4. Shaded features are ones for which the $PS \geq 0.5$. The amino acid, exposed α-helix, buried α-helix and β strand features are all proportions (e.g. the Alanine feature for a protein is the number of alanine residues in the sequence divided by the sequence length), while all other features are absolute numbers. Features with significant differences are indicated with an *.

5.4.3.2 Target Predictions

The best combination of parameters and feature set for classifying the proteins in the $Cancer$ dataset was $numberTrees = 1000$, $mtry = 5$, a weight of 1.3 given to each observation in the in positive class, a random seed of - 4923865346116695007 and the following thirty-six features out of the original 104 from Section 5.1.13:

- Amino acid compositions
    - The proportion of alanine, arginine, asparagine, cysteine, glycine, histidine, methionine, proline, serine, tryptophan and aromatic residues.
- Simple sequence properties
    - The presence of a signal peptide.
- Posttranslational modifications
    - The number of *N*-linked glycosylation, phosphothreonine and phosphotyrosine sites.
- Secondary structure
    - The fraction of residues predicted to participate in β-strands and buried α-helices.
    - The number of transmembrane α-helices
- Germline variants:
    - The number of 3' untranslated mutations.
- Inter-protein relationships
    - The number of paralogs.
- Developmental stage expression
    - The blastocyst, neonate and juvenile expression levels.
- Body site expression
    - The adrenal gland, brain, cervix, ear, embryonic tissue, heart, lymph node, prostate, skin, stomach, testis and uterus expression levels

The positive similarity of the proteins in the $Cancer$ dataset can be seen in Figure 33. Using a cutoff of 0.5, the RF's predicted classifications were as follows:

Positive Observations				Unlabelled Observations				G Mean
Total	TPs	FNs	Sensitivity	Total	TNs	FPs	Specificity	
387	334	53	0.86	444	380	64	0.86	0.86

As 55% of positive proteins are membrane bound, compared to 11% of unlabelled ones, it would be expected that a substantial fraction of the misclassified unlabelled proteins are also membrane bound. This was found to be the case, with 55% of unlabelled proteins with a positive similarity > 0.5 being membrane bound, and 65% of the unlabelled proteins most likely to be suitable targets, those with positive similarity ≥ 0.75, being membrane bound. This tendency to

be membrane bound is also reflected in the function of the forty-six unlabelled proteins with positive similarity ≥ 0.75, as a large fraction of them are membrane bound receptors involved in signal transduction. Many of the proteins are also putative proto-oncogenes or tumour suppressors, and are often involved in a process or processes that can contribute to the distinguishing characteristics of cancer. For example, the unlabelled protein with the greatest positive similarity was protein patched homolog 1 (PTCH1) (UniProt accession Q13635). In addition to being a known tumour suppressor, PTCH1 is a receptor for hedgehog ligands, which are involved in proliferation and differentiation during embryogenesis (Agren et al. 2004). Even when a direct connection between the protein and cancer is speculative or unknown, a connection between them can often be hypothesised. For example, although sodium-dependent phosphate transport protein 2B (UniProt accession O95436) has no clear oncogenic or tumour suppression function, it is regulated by epidermal growth factor (Xu et al. 2001), the expression of which is often altered in cancerous cells as part of their achieving unregulated growth. While connections between the proteins and cancer provide some validation for the usefulness of the misclassified unlabelled proteins as antineoplastic targets, at least one, programmed cell death 1 ligand 1 (UniProt accession Q9NZQ7), is known to be the target of a compound currently undergoing phase II clinical trials as an antineoplastic drug (MPDL3280A). All unlabelled proteins in the $Cancer$ dataset predicted to be positive can be found in Appendix A Section I. These proteins are those that have been causatively linked to cancer, without being used as an antineoplastic target, and appear most suitable for consideration as future antineoplastic drug targets.

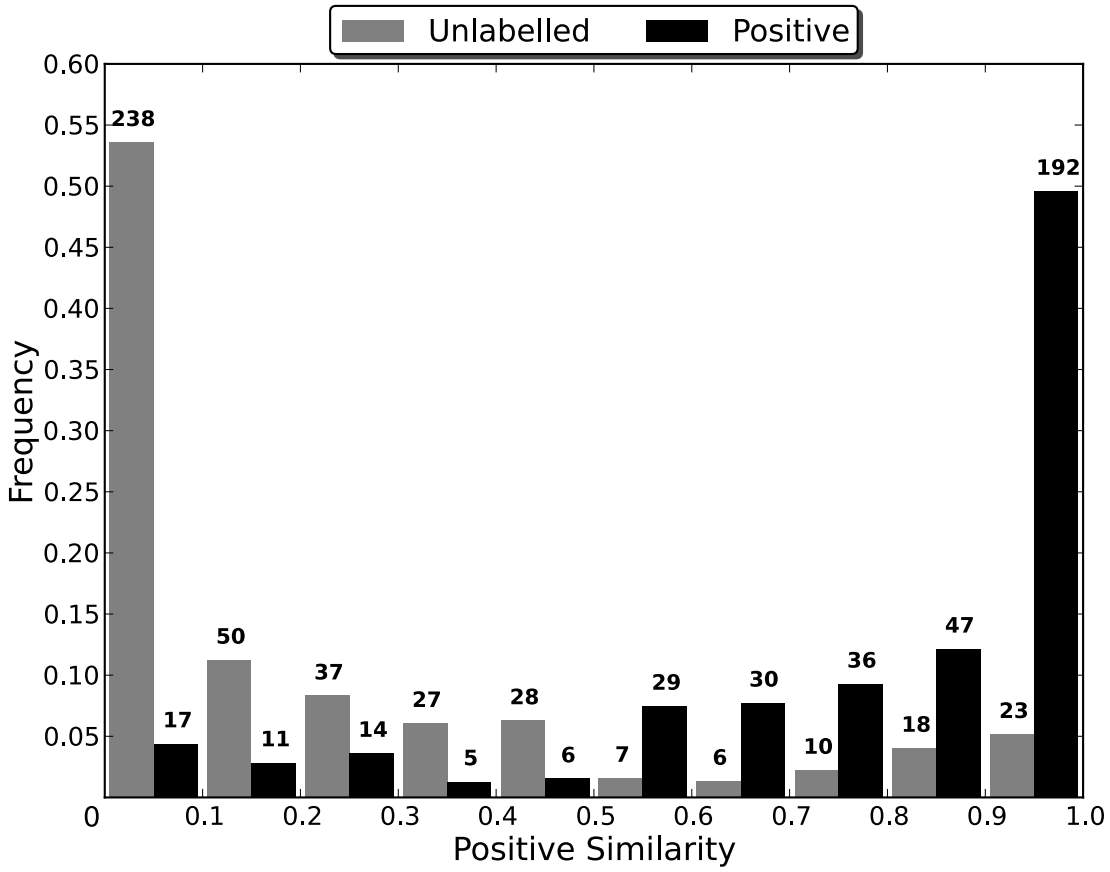

Figure 33: Weighted predictions of the proteins in the $Cancer$ dataset. The positive similarity of a given protein is equal to the fraction of the forest's votes that are for the positive class. The values over the bars indicate the number of proteins in the bin. The $Cancer$ dataset contained 444 unlabelled proteins and 387 positive ones.

5.4.4 Results – GPCRs

5.4.4.1 Target Properties

The results from the analysis of the features in the $GPCR$ dataset (using the method described in Section 5.4) can be seen in Table 17. Considering the size and composition of the $GPCR$ dataset, when compared to the other datasets investigated, the number of features with meaningful effect sizes is surprisingly large. This was indicative of either substantial differences between the positive and unlabelled proteins, or of a large subpopulation of GPCRs (most likely unlabelled ones) that are considerably different to the other proteins in the dataset. A likely contender for this subpopulation is odorant/olfactory GPCRs. While odorant GPCRs are restricted to cells specialised for the detection of external stimuli, e.g. odours and tastes, non-odorant GPCRs are differentially expressed throughout the body, respond to a variety of endogenous ligands and regulate various vital physiological processes (Regard et al. 2008). Therefore, non-odorant GPCRs should be more likely to be targeted by drugs. Analysis of the $GPCR$ dataset supports this belief, as of the 421 odorant GPCRs in the dataset, none were classified as likely to be a potential drug target (Section **Error! Reference source not found.**) or were the target of an pproved drug.

In order to evaluate the impact of the odorant GPCRs on the feature analysis, a second dataset, $GPCR\_NO$, was constructed from the $GPCR$ dataset by removing all odorant GPCRs from it. The results of the analysis of this second dataset (using the method described in Section 5.4) can be seen in Table 18. For all features, except the fraction of residues in exposed α-helices, the effect size was smaller in the $GPCR\_NO$ dataset than in the $GPCR$ dataset. Additionally, only five features were deemed to have significant differences, compared to thirty-seven features in the $GPCR$ dataset.

When compared to the unlabelled proteins, the positive proteins in the $GPCR\_NO$ dataset have a slightly smaller proportion of non-polar amino acids ($PS = 0.43$) and lower hydrophobicity ($PS = 0.38$). The positive proteins in the $GPCR\_NO$ dataset were also slightly more likely to have a longer sequence length than the unlabelled ones ($PS = 0.59$). As GPCRs contain seven transmembrane regions and the positive proteins have a slightly smaller fraction of residues in buried α-helices ($PS = 0.44$), the difference in the sequence length likely comes from positive proteins having more extra and intracellular residues. Unlike the amino acids in the transmembrane regions, non-transmembrane residues are likely to be more hydrophilic as they are exposed to the extra and intracellular environments rather than being embedded in a membrane. This likely accounts for the smaller proportion of non-polar and aromatic amino acids in positive proteins and for their lower hydrophobicity. Similarly, the greater proportion of

charged and negatively charged amino acids in the positive proteins is likely due to the increased sequence length.

Feature	P-value	PS	Positive Median	Unlabelled Median
Alanine *	4.07×10^{-08}	0.66	0.08	0.06
Arginine *	9.88×10^{-22}	0.77	0.05	0.04
Asparagine	1.27×10^{-03}	0.59	0.04	0.03
Aspartic Acid *	5.45×10^{-08}	0.66	0.03	0.03
Cysteine	3.63×10^{-03}	0.42	0.03	0.03
Glutamic Acid *	6.09×10^{-14}	0.71	0.03	0.03
Glutamine	6.78×10^{-03}	0.58	0.03	0.03
Glycine *	2.50×10^{-04}	0.61	0.05	0.05
Histidine *	1.52×10^{-16}	0.27	0.02	0.03
Isoleucine *	7.68×10^{-06}	0.37	0.07	0.08
Leucine *	1.86×10^{-15}	0.28	0.12	0.14
Lysine *	8.30×10^{-04}	0.60	0.04	0.03
Methionine *	1.17×10^{-13}	0.29	0.02	0.03
Phenylalanine *	1.93×10^{-17}	0.26	0.05	0.07
Proline *	4.39×10^{-11}	0.69	0.05	0.04
Serine	5.60×10^{-02}	0.44	0.08	0.08
Threonine *	7.45×10^{-04}	0.40	0.06	0.06
Tryptophan *	3.41×10^{-21}	0.76	0.02	0.01
Tyrosine *	1.64×10^{-08}	0.34	0.03	0.04
Valine	2.47×10^{-01}	0.47	0.08	0.08
Aliphatic *	8.59×10^{-24}	0.22	0.26	0.30
Aromatic *	3.23×10^{-17}	0.26	0.12	0.15
Charged *	7.68×10^{-22}	0.77	0.18	0.15
Negatively Charged *	9.12×10^{-18}	0.74	0.07	0.05
Non-polar *	2.45×10^{-12}	0.30	0.61	0.65

Feature	P-value	PS	Positive Median	Unlabelled Median
Positively Charged *	1.75×10^{-13}	0.71	0.11	0.10
Sequence Length *	1.97×10^{-30}	0.81	408	320
PEST Motifs *	2.31×10^{-07}	0.59	0	0
Low Complexity Regions *	2.78×10^{-07}	0.64	2	1
Hydrophobicity *	1.06×10^{-33}	0.17	0.31	0.68
Isoelectric Point *	5.97×10^{-06}	0.63	9.02	8.52
Signal Peptide	1.38×10^{-03}	0.55	0	0
***O*-glycosylation Sites**	1.92×10^{-02}	0.51	0	0
***N*-glycosylation Sites ***	6.47×10^{-12}	0.68	2	1
Phosphoserine Sites *	1.43×10^{-06}	0.57	0	0
Phosphothreonine Sites *	6.08×10^{-06}	0.54	0	0
Phosphotyrosine Sites	5.49×10^{-03}	0.52	0	0
Total Phosphorylation Sites *	7.87×10^{-08}	0.59	0	0
Transmembrane α-helices	9.85×10^{-01}	0.50	7	7
Exposed α-helices	3.35×10^{-01}	0.53	0.09	0.09
Buried α-helices *	6.29×10^{-19}	0.25	0.47	0.58
β Strands *	1.86×10^{-12}	0.30	0.03	0.04
3' Untranslated	1.81×10^{-02}	0.53	0	0
5' Untranslated	5.56×10^{-02}	0.53	0	0
Nonsynonymous Coding *	3.92×10^{-16}	0.70	2	0
Synonymous Coding	7.95×10^{-02}	0.52	0	0
Binary PPIs	6.93×10^{-03}	0.54	0	0
Alternative Transcripts *	6.68×10^{-18}	0.72	1	0
Paralogs	3.72×10^{-02}	0.52	0	0
Body Sites Expressed In *	5.85×10^{-27}	0.79	12	5

Table 17: Results of the feature analysis for the $GPCR$ dataset. The p-values and the PS were calculated as in Section 5.4. Shaded features are ones for which the $PS \geq 0.5$. The amino acid, exposed α-helix, buried α-helix and β strand features are all proportions (e.g. the Alanine feature for a protein is the number of alanine residues in the sequence divided by the sequence length), while all other features are absolute numbers. Features with significant differences are indicated with an *.

Feature	P-value	PS	Positive Median	Unlabelled Median
Alanine	4.75×10^{-02}	0.56	0.08	0.07
Arginine	5.63×10^{-03}	0.59	0.05	0.05
Asparagine	1.77×10^{-01}	0.54	0.04	0.04
Aspartic Acid *	7.06×10^{-04}	0.61	0.03	0.03
Cysteine	4.56×10^{-01}	0.48	0.03	0.03
Glutamic Acid	5.66×10^{-02}	0.56	0.03	0.03
Glutamine	4.25×10^{-01}	0.47	0.03	0.03
Glycine	4.89×10^{-01}	0.52	0.05	0.05
Histidine *	9.56×10^{-09}	0.32	0.02	0.02
Isoleucine	4.73×10^{-01}	0.52	0.07	0.06
Leucine *	1.44×10^{-04}	0.38	0.12	0.13
Lysine	7.18×10^{-02}	0.56	0.04	0.04
Methionine	4.69×10^{-01}	0.52	0.02	0.02
Phenylalanine	2.43×10^{-03}	0.40	0.05	0.06
Proline	1.77×10^{-03}	0.60	0.05	0.04
Serine	1.82×10^{-01}	0.46	0.08	0.08
Threonine	6.27×10^{-01}	0.48	0.06	0.06
Tryptophan	7.86×10^{-01}	0.49	0.02	0.02
Tyrosine	4.28×10^{-01}	0.47	0.03	0.03
Valine	5.84×10^{-01}	0.48	0.08	0.08
Aliphatic	3.39×10^{-03}	0.41	0.26	0.27
Aromatic *	6.54×10^{-06}	0.36	0.12	0.14
Charged	2.63×10^{-03}	0.60	0.18	0.17
Negatively Charged	1.64×10^{-03}	0.60	0.07	0.06
Non-polar	3.51×10^{-02}	0.43	0.61	0.63

Feature	P-value	PS	Positive Median	Unlabelled Median
Positively Charged	1.92×10^{-01}	0.54	0.11	0.11
Sequence Length	2.87×10^{-03}	0.59	408	373
PEST Motifs	2.40×10^{-01}	0.53	0	0
Low Complexity Regions	4.71×10^{-01}	0.48	2	2
Hydrophobicity *	1.84×10^{-04}	0.38	0.31	0.43
Isoelectric Point	1.81×10^{-01}	0.54	9.02	8.68
Signal Peptide	5.96×10^{-01}	0.48	0	0
***O*-glycosylation Sites**	7.97×10^{-02}	0.51	0	0
***N*-glycosylation Sites**	2.04×10^{-01}	0.54	2	2
Phosphoserine Sites	7.94×10^{-02}	0.53	0	0
Phosphothreonine Sites	5.07×10^{-03}	0.53	0	0
Phosphotyrosine Sites	2.78×10^{-01}	0.51	0	0
Total Phosphorylation Sites	3.51×10^{-02}	0.55	0	0
Transmembrane α-helices	9.58×10^{-01}	0.50	7	7
Exposed α-helices	5.36×10^{-02}	0.44	0.09	0.10
Buried α-helices	6.55×10^{-02}	0.44	0.47	0.51
β Strands	4.84×10^{-02}	0.44	0.03	0.03
3' Untranslated	2.41×10^{-01}	0.48	0	0
5' Untranslated	1.85×10^{-01}	0.47	0	0
Nonsynonymous Coding	2.98×10^{-01}	0.53	2	1
Synonymous Coding	4.03×10^{-01}	0.49	0	0
Binary PPIs	3.70×10^{-01}	0.48	0	0
Alternative Transcripts	6.69×10^{-03}	0.58	1	1
Paralogs	7.19×10^{-01}	0.50	0	0
Body Sites Expressed In	1.73×10^{-01}	0.54	12	11

Table 18: Results of the feature analysis for the $GPCR\_NO$ dataset. The p-values and the PS were calculated as in Section 5.4. Shaded features are ones for which the $PS \geq 0.5$. The amino acid, exposed α-helix, buried α-helix and β strand features are all proportions (e.g. the Alanine feature for a protein is the number of alanine residues in the sequence divided by the sequence length), while all other features are absolute numbers. Features with significant differences are indicated with an *.

5.4.4.2 Target Predictions

The best combination of parameters and feature set for classifying the proteins in the $GPCR$ dataset was $numberTrees = 4000$, $mtry = 5$, a weight of 12 given to each observation in the in positive class, a random seed of -4568194888819162440 and the following forty-two features out of the original 104 from Section 5.1.13:

- Amino acid compositions
    - The proportion of asparagine, glycine, histidine, isoleucine, serine, tryptophan, valine, aliphatic and non-polar residues.
- Simple sequence properties
    - The number of PEST motifs and the presence of a signal peptide.
- Posttranslational modifications
    - The number of *N*-linked glycosylation, *O*-linked glycosylation and phosphothreonine sites.
- Secondary structure
    - The fraction of residues predicted to participate in β-strands, buried α-helices and exposed α-helices.
- Germline variants
    - The number of 5' untranslated mutations.
- Inter-protein relationships
    - The number of alternative transcripts and paralogs.
- Developmental stage expression
    - The embryoid body and neonate expression levels.
- Body site expression
    - The bone, bone marrow, brain, mammary gland, mouth, salivary gland, testis, thymus, thyroid, trachea, umbilical cord, uterus and vascular expression levels.
    - The number of body sites that the protein is expressed in.

The positive similarity of the proteins in the $GPCR$ dataset can be seen in Figure 34. Using a cutoff of 0.5, the RF's predicted classifications were as follows:

Positive Observations				Unlabelled Observations				G Mean
Total	TPs	FNs	Sensitivity	Total	TNs	FPs	Specificity	
115	104	11	0.90	712	613	99	0.86	0.88

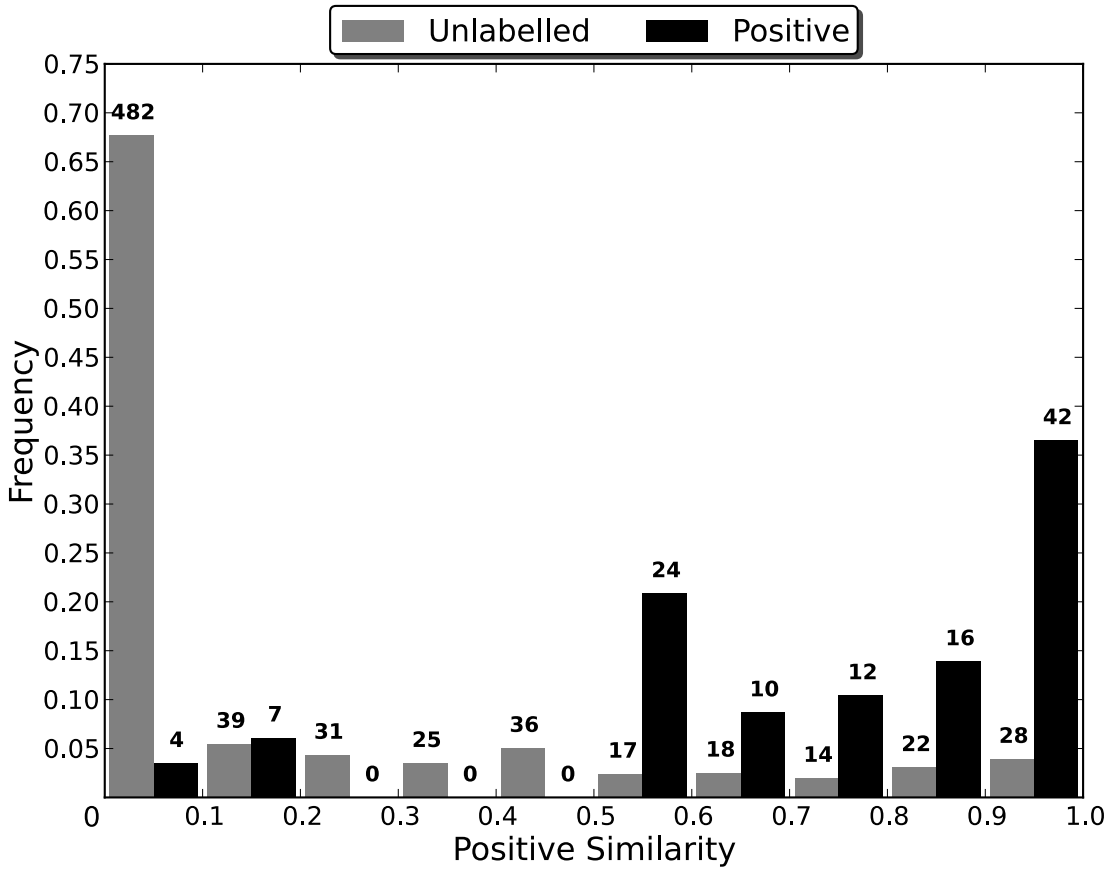

Figure 34: Weighted predictions of the proteins in the $GPCR$ dataset. The positive similarity of a given protein is equal to the fraction of the forest's votes that are for the positive class. The values over the bars indicate the number of proteins in the bin. The $GPCR$ dataset contained 712 unlabelled proteins and 115 positive ones.

As with the results of the analysis of the features in the $GPCR$ dataset (Section 5.4.3.2), the distribution of the positive similarities of the proteins in the dataset is likely to be heavily skewed by the presence of the odorant/olfactory GPCRs. Of the 421 odorant/olfactory GPCRs in the dataset, 419 were given a positive similarity below 0.1, with all 421 having a positive similarity below 0.5. In order to assess the impact on the classifications of including the odorant/olfactory GPCRs in the dataset, a new dataset, $GPCR\_NO$, was constructed from all proteins in the $GPCR$ dataset that are not odorant/olfactory receptors. The best combination of parameters and feature set for classifying the proteins in the $GPCR\_NO$ dataset was $numberTrees = 4000$, $mtry = 5$, a weight of 3.6 given to each observation in the in positive class, a random seed of -251746180866936552 and the following forty-seven features out of the original 104 from Section 5.1.13:

- Amino acid compositions
    - The proportion of glutamine, histidine, proline, serine, negatively charged and polar residues.
- Simple sequence properties
    - The sequence length and presence of a signal peptide.
- Posttranslational modifications
    - The number of *O*-linked glycosylation, phosphothreonine and phosphotyrosine sites.
- Secondary structure
    - The fraction of residues predicted to participate in β-strands and buried α-helices.
    - The number of transmembrane α-helices.
- Germline variants
    - The number of 3' untranslated mutations.
- Inter-protein relationships
    - The number of paralogs and binary PPIs.
- Developmental stage expression
    - The embryoid body, blastocyst, fetus and infant expression levels.
- Body site expression
    - The adrenal gland, bladder, blood, bone, brain, connective tissue, ear, embryonic tissue, eye, intestine, kidney, liver, lung, nerve, ovary, placenta, prostate, spleen, stomach, testis, thymus, thyroid, tonsil, umbilical cord and uterus expression levels.
    - The number of body sites that the protein is expressed in.

The lower G mean of the RF trained on the $GPCR\_NO$ dataset indicates that the dissimilarities between the positive and unlabelled proteins are not as great as the results generated using the $GPCR$ dataset would purport to show. The positive and unlabelled GPCRs are in fact quite similar, once the odorants are removed, as can be seen from the large overlap and relatively low frequencies in their positive similarities (Figure 35) and the small effect size of their differences (Table 18).

Using a cutoff of 0.5, substantial differences can be seen in the classifications of the non-odorant GPCRs by the RFs trained on the $GPCR$ and $GPCR\_NO$ datasets (Table 19). Although no odorant receptors were misclassified by the RF trained on the $GPCR$ dataset, removing the odorants from the dataset led to forty-nine fewer unlabelled proteins being misclassified as positive. Although this may appear counterintuitive, it is in fact unsurprising. This is because the presence of the large subpopulation of unlabelled odorants allows the weight given to the

positive observations to be increased, thereby improving the sensitivity of the RF at the cost of increasing the number of misclassified unlabelled proteins. As only non-odorant unlabelled proteins will be misclassified, due to the dissimilarity between the odorants and positive proteins, the resultant decrease in specificity will be small and can be more than compensated for by the increase in sensitivity. Therefore, removing the odorants will not only negate the artificial boost to the specificity that they provide, but also necessitate a decrease in the weight given to the positive proteins, as can be seen by the optimal weight for the $GPCR$ dataset being 12, compared to 3.6 for the $GPCR\_NO$ dataset. As the G mean of the RF trained on the $GPCR\_NO$ dataset is substantially more sensitive to misclassified unlabelled proteins, due to the smaller number of unlabelled proteins in the dataset, the number of misclassified unlabelled proteins must be brought down in order to achieve a respectable G mean. However, the increase in specificity that this provides will be accompanied by a sizeable decrease in sensitivity, and therefore a lower G mean.

Of the twenty-three unlabelled proteins with the greatest likelihood of being suitable drug targets, those with positive similarity ≥ 0.75, 15 are class A GPCRs, 7 are class B and 1 is class C. Irrespective of class, the GPCRs are predominantly expressed in the brain and the central nervous system. In terms of the ligands of the misclassified unlabelled proteins, seven of the twenty-three are orphan receptors with no known ligand, while the remainder are predominantly receptors for neurotransmitters and neuropeptides (in line with their tendency to be expressed in the brain). All unlabelled proteins in the $GPCR\_NO$ dataset predicted to be positive can be found in Appendix A Section II.

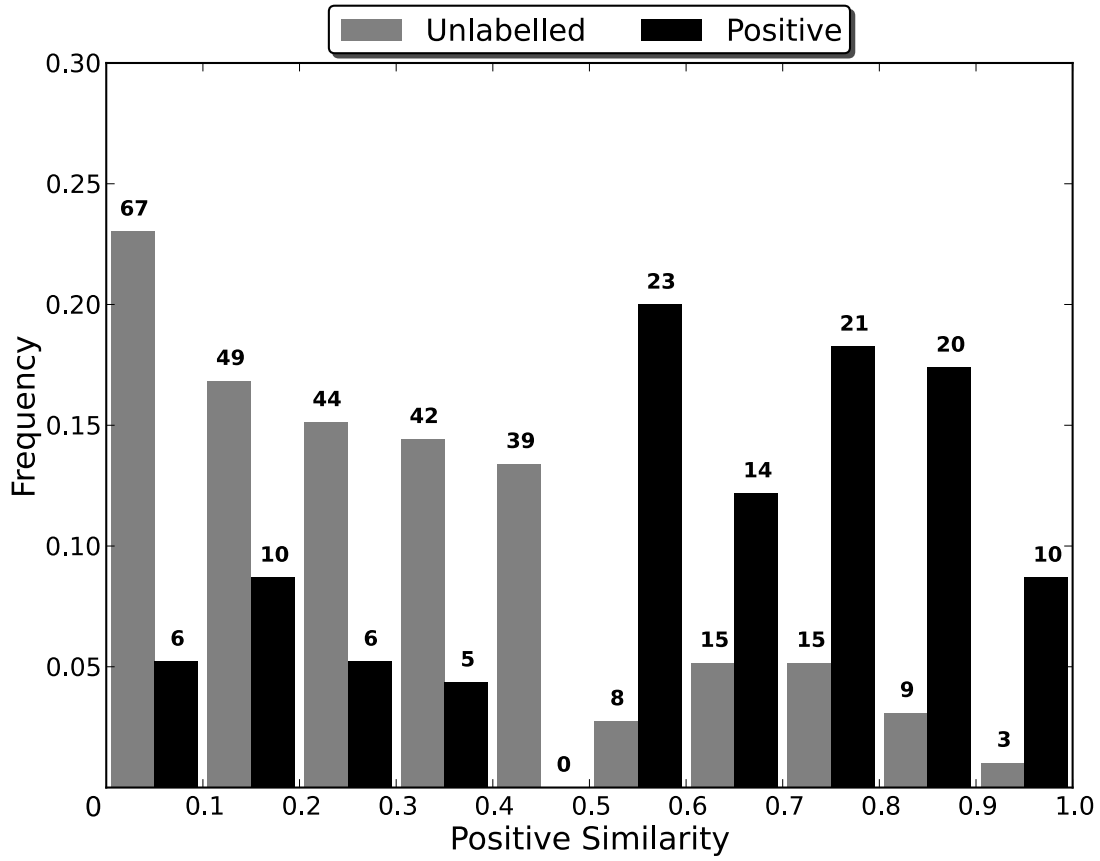

Figure 35: Weighted predictions of the proteins in the $GPCR\_NO$ dataset. The positive similarity of a given protein is equal to the fraction of the forest's votes that are for the positive class. The values over the bars indicate the number of proteins in the bin. The $GPCR\_NO$ dataset contained 291 unlabelled proteins and 115 positive ones.

Dataset Trained On	Positive Observations				Unlabelled Observations				G Mean
	Total	TPs	FNs	Sensitivity	Total	TNs	FPs	Specificity	
$GPCR$	115	104	11	0.90	291	241	99	0.71	0.74
$GPCR\_NO$	115	88	27	0.77	291	241	50	0.83	0.80

Table 19: A comparison of the predictions of the non-odorant GPCRs. Predictions were made by the optimised RF trained on the $GPCR$ dataset and the one trained on the $GPCR\_NO$ dataset.

5.4.5 Results - Ion Channels

5.4.5.1 Target Properties

The results from the analysis of the features in the $IonChannel$ dataset (using the method described in Section 5.4) can be seen in Table 20. The differences between the positive and unlabelled proteins in terms of their amino acid proportions is minimal, with only a small difference in the proportion of non-polar amino acids ($PS = 0.43$). The tendency of the positive proteins to have a slightly smaller proportion of non-polar amino acids can likely be explained by the greater sequence length of the positive proteins ($PS = 0.61$). As the difference in the number of transmembrane helices in positive and unlabelled proteins is minimal ($PS = 0.48$) and the positive proteins have a smaller fraction of residues in buried α-helices ($PS = 0.41$), the longer sequence length of the positive proteins can likely be explained by them having more residues in the intra and/or extracellular space. Unlike the amino acids in the transmembrane regions, these residues are likely to be more hydrophilic as they are exposed to the extra and intracellular environments rather than being embedded in a membrane. The positive proteins would therefore have a slightly smaller proportion of non-polar amino acids.

The increased number of extra and intracellular amino acids could also account for the tendency of the positive proteins to have an increased number of *N*-linked glycosylation ($PS = 0.68$), phosphoserine ($PS = 0.58$) and total phosphorylation sites ($PS = 0.57$). In order to test this, the PS of the three features was tested after accounting for the length of the protein (by dividing the feature value for a protein by the number of residues in its sequence). Following this the positive proteins still had greater values for *N*-linked glycosylation ($PS = 0.68$), phosphoserine ($PS = 0.56$) and total phosphorylation ($PS = 0.55$) sites. However, the effect for the phosphoserine and total phosphorylation sites is now too small to be meaningful, indicating that without the difference in sequence length there would likely be no consequential effect for the phosphoserine or total phosphorylation sites. In contrast to the phosphorylation sites, the PS of the *N*-linked glycosylation sites is the same after controlling for the differences in sequence length, meaning that positive ion channels are likely to have greater *in vivo* half-lives and be more stable. Due to their being more likely to contain a signal peptide, positive ion channels are also more likely to be secreted.

Feature	P-value	PS	Positive Median	Unlabelled Median
Alanine	5.21×10^{-02}	0.44	0.06	0.07
Arginine	5.91×10^{-01}	0.52	0.06	0.05
Asparagine *	9.78×10^{-04}	0.61	0.04	0.03
Aspartic Acid	1.57×10^{-03}	0.60	0.05	0.04
Cysteine	1.36×10^{-01}	0.45	0.02	0.02
Glutamic Acid	2.63×10^{-01}	0.46	0.06	0.06
Glutamine	1.91×10^{-03}	0.40	0.03	0.04
Glycine	2.50×10^{-01}	0.46	0.06	0.06
Histidine	7.60×10^{-01}	0.49	0.02	0.02
Isoleucine	1.54×10^{-02}	0.58	0.06	0.06
Leucine *	2.13×10^{-06}	0.35	0.10	0.11
Lysine	2.39×10^{-01}	0.54	0.05	0.05
Methionine	2.36×10^{-02}	0.57	0.03	0.02
Phenylalanine	2.79×10^{-01}	0.46	0.05	0.05
Proline	3.47×10^{-01}	0.53	0.05	0.05
Serine	2.59×10^{-01}	0.54	0.08	0.07
Threonine *	2.16×10^{-04}	0.62	0.05	0.05
Tryptophan	9.73×10^{-01}	0.50	0.02	0.02
Tyrosine	7.06×10^{-01}	0.49	0.03	0.03
Valine	6.60×10^{-03}	0.59	0.07	0.06
Aliphatic	2.54×10^{-01}	0.46	0.23	0.23
Aromatic	2.67×10^{-01}	0.46	0.12	0.13
Charged	8.33×10^{-01}	0.51	0.23	0.23
Negatively Charged	5.43×10^{-01}	0.52	0.11	0.10
Non-polar	1.59×10^{-02}	0.42	0.55	0.57

Feature	P-value	PS	Positive Median	Unlabelled Median
Positively Charged	8.46×10^{-01}	0.51	0.13	0.13
Sequence Length *	4.85×10^{-04}	0.61	613	509
PEST Motifs	6.64×10^{-01}	0.51	0	0
Low Complexity Regions	9.72×10^{-02}	0.55	3	3
Hydrophobicity	1.90×10^{-01}	0.46	-0.11	-0.08
Isoelectric Point	8.49×10^{-01}	0.51	7.38	7.56
Signal Peptide *	1.68×10^{-10}	0.66	0	0
***O*-glycosylation Sites**	NA	NA	0	0
***N*-glycosylation Sites ***	1.02×10^{-08}	0.68	2	1
Phosphoserine Sites	2.58×10^{-03}	0.58	0	0
Phosphothreonine Sites	2.22×10^{-01}	0.52	0	0
Phosphotyrosine Sites	1.59×10^{-01}	0.53	0	0
Total Phosphorylation Sites	9.02×10^{-03}	0.57	0	0
Transmembrane α-helices	5.26×10^{-01}	0.48	4	5
Exposed α-helices	2.85×10^{-02}	0.43	0.14	0.16
Buried α-helices	3.53×10^{-03}	0.41	0.27	0.32
β Strands	1.26×10^{-03}	0.60	0.11	0.06
3' Untranslated	6.21×10^{-01}	0.49	0	0
5' Untranslated	2.74×10^{-01}	0.47	0	0
Nonsynonymous Coding	8.18×10^{-02}	0.56	4	3
Synonymous Coding	1.45×10^{-03}	0.45	0	0
Binary PPIs	2.89×10^{-01}	0.47	0	0
Alternative Transcripts	4.44×10^{-02}	0.56	3	2
Paralogs	7.27×10^{-02}	0.54	0	0
Body Sites Expressed In	4.14×10^{-01}	0.53	15	15

Table 20: Results of the feature analysis for the $IonChannel$ dataset. The p-values and the PS were calculated as in Section 5.4. Shaded features are ones for which the $PS \geq 0.5$. The amino acid, exposed α-helix, buried α-helix and β strand features are all proportions (e.g. the Alanine feature for a protein is the number of alanine residues in the sequence divided by the sequence length), while all other features are absolute numbers. Features with significant differences are indicated with an *. The NAs for the *O*-glycosylation sites are due to no ion channels containing an *O*-glycosylation site.

5.4.5.2 Target Predictions

The best combination of parameters and feature set for classifying the proteins in the $IonChannel$ dataset was$numberTrees = 1000$, $mtry = 10$, a weight of 1.2 given to each observation in the in positive class, a random seed of 2641231349290994133 and the following forty features out of the original 104 from Section 5.1.13:

- Amino acid compositions
    - The proportion of alanine, cysteine, glycine, lysine, phenylalanine, tryptophan and tiny residues.
- Simple sequence properties
    - The number of PEST motifs, presence of a signal peptide and sequence length.
- Posttranslational modifications
    - The number of *O*-linked glycosylation and phosphoserine sites.
- Secondary structure
    - The fraction of residues predicted to participate in β-strands.
- Germline variants
    - The number of 5' untranslated, nonsynonymous coding and synonymous coding mutations
- Inter-protein relationship
    - The number of alternative transcripts.
- Developmental stage expression
    - The blastocyst and juvenile expression levels.
- Body site expression
    - The ascites, bone, esophagus, intestine, kidney, liver, lung, lymph, lymph node, mouth, muscle, nerve, ovary, pancreas, pharynx, prostate, thymus, thyroid, umbilical cord, uterus and vascular expression levels.

The positive similarity of the proteins in the $IonChannel$ dataset can be seen in Figure 36. The distribution of the proteins in the $IonChannel$ dataset likely indicates that there is a strong similarity between the positive and unlabelled proteins, as there are no particularly large peaks in any of the more extreme bins (0.0-0.1 and 0.9-1.0). Using a cutoff of 0.5, the RF's predicted classifications were as follows:

Positive Observations				Unlabelled Observations				G Mean
Total	TPs	FNs	Sensitivity	Total	TNs	FPs	Specificity	
155	133	22	0.86	165	144	21	0.87	0.87

Of the ten unlabelled proteins with the greatest likelihood of being suitable drug targets, those with positive similarity ≥ 0.75, six are known to be voltage-gated, three ligand-gated and one of unknown gating. Of the voltage-gated channels, two are selective for calcium, three for potassium and one for sodium, with the potassium channels both being inward rectifying ones. All three ligand gated channels were selective for cations, with the ligands being zinc for one channel and serotonin for the other two. All unlabelled proteins in the $IonChannel$ dataset predicted to be positive can be found in Appendix A Section III.

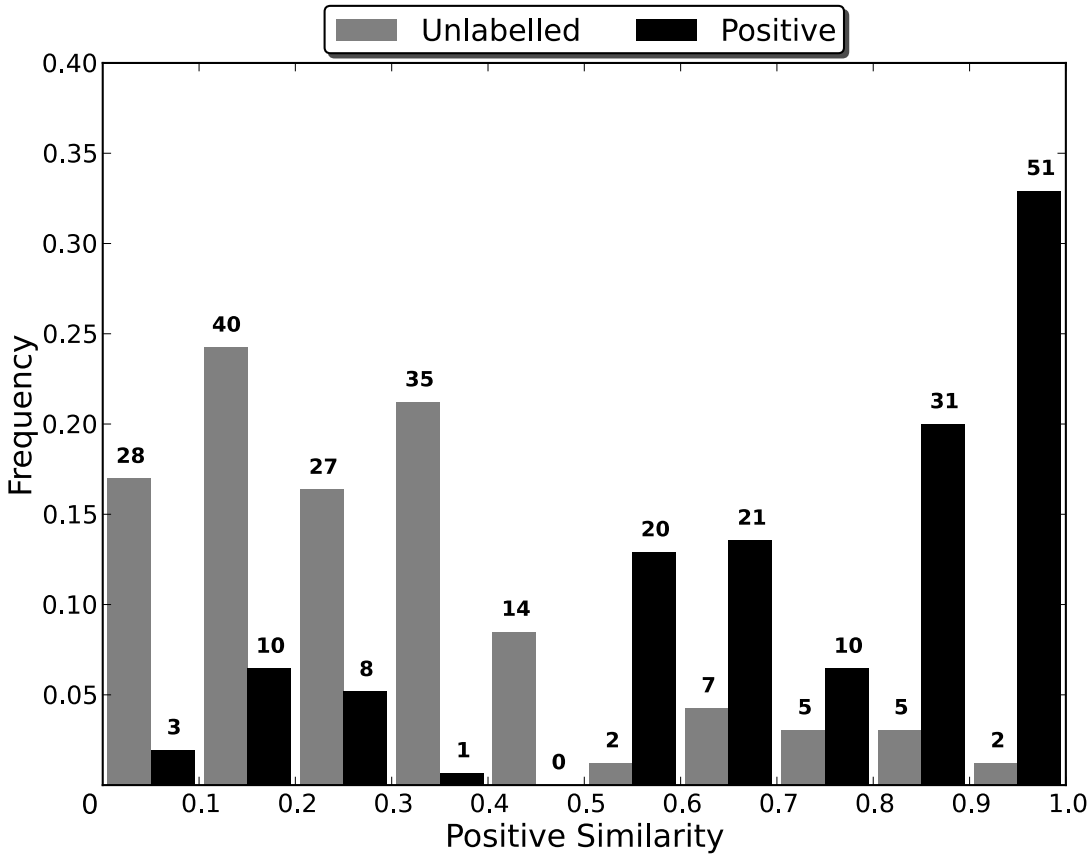

Figure 36: Weighted predictions of the proteins in the $IonChannel$ dataset. The positive similarity of a given protein is equal to the fraction of the forest's votes that are for the positive class. The values over the bars indicate the number of proteins in the bin. The $IonChannel$ dataset contained 165 unlabelled proteins and 155 positive ones.

5.4.6 Results – Kinases

5.4.6.1 Target Properties

The results from the analysis of the features in the $Kinase$ dataset (using the method described in Section 5.4) can be seen in Table 21. Although the results indicate that there are significant differences between the positive and unlabelled proteins, the substantial differences in their compositions, specifically the much larger proportion of tyrosine kinases in the positive proteins (Table 23), are potentially influencing the results. The significant differences seen for the $Kinase$ dataset could then simply be a reflection of the differences between serine/threonine and tyrosine kinases. Although the presence of kinases of an unknown type complicates this generalisation somewhat, in general their feature values closely follow those of the serine/threonine kinases, not the tyrosine ones. They are therefore likely to be at best neutral with regards to the differences between the serine/threonine and tyrosine kinases.

The influence of the differences between kinase types was evaluated by creating two new datasets. The $Kinase\_TK$ dataset was constructed from the $Kinase$ dataset by removing all positive proteins that were not tyrosine kinases, while the $Kinase\_NTK$ dataset was constructed from the $Kinase$ dataset by removing all positive proteins that were tyrosine kinases. Both of these datasets had their features analysed in terms of significance and effect size using the methods described in Section 5.4. A comparison between the features that are significant in each of the three datasets can be seen in Table 22. For each feature, the deviation of the effect size from 0.5 in the $Kinase$ dataset can be seen to be between the deviations for the $Kinase\_TK$ and $Kinase\_NTK$ datasets. However, if the positive serine/threonine and tyrosine kinases shared similar properties, then the features with the greatest deviations in effect size in the $Kinase$ dataset would be expected to have the greatest deviations in the $Kinase\_TK$ and $Kinase\_NTK$ datasets. The pattern of deviations therefore indicates that there are distinct differences between the positive serine/threonine kinases and the positive tyrosine ones. Additionally, the positive tyrosine kinases can be seen to be dominating the effects seen in the $Kinase$ dataset, as the $Kinase\_TK$ dataset shows very large deviations for those features that are significant in the $Kinase$ dataset while the $Kinase\_NTK$ dataset has very small deviations for them. Additionally, of the nine significant features in the $Kinase$ dataset, all nine were found to be significant in the $Kinase\_TK$ dataset, while none were significant in the $Kinase\_NTK$ dataset. These results indicate that the differences between the positive and unlabelled proteins in the $Kinase$ dataset are highly likely to be an artefact of the makeup of the dataset, rather than a true reflection of the properties that make a kinase a suitable drug target.

Despite the entire set of kinases being unsuitable for analysis, it is possible that informative differences can be found by comparing positive tyrosine kinases with unlabelled ones.

However, in addition to the differences between the serine/threonine and tyrosine kinases, the $Kinase$ dataset also has a biased set of positive tyrosine kinases. Although it may be hypothesised that this bias would be due to a preference for receptor tyrosine kinases, due to drug targets being predominantly membrane bound, the fraction of receptor tyrosine kinases in the positive proteins and in the set of all tyrosine kinases is very similar. Rather, the source of the bias comes from the specific disease that the drugs targeting the tyrosine kinases are intended to treat: cancer. Of the forty positive tyrosine kinases, thirty-four are the target of an antineoplastic drug, while a further three are causally implicated in cancer (and therefore in the $Cancer$ dataset). However, only four of the fifty unlabelled tyrosine kinases are causally implicated in cancer. Any comparison of positive and unlabelled tyrosine kinases is therefore more of a comparison between those tyrosine kinases that have been implicated in cancer and those that have not.

While the positive serine/threonine kinases have no biases as evident as those of the tyrosine kinases, there are very few of them. Additionally, these kinases may be unrepresentative in that they may have been selected as early targets for specific reasons, i.e. properties that they possess, that will not extrapolate to future targets. Therefore, until the set of positive kinases is more representative, or the set of positive serine/threonine kinases increases in size, it will be difficult to get an accurate picture of the properties, besides their ability to transfer phosphoryl groups, that make a general kinase a suitable drug target.

Feature	P-value	PS	Positive Median	Unlabelled Median
Alanine	1.19×10^{-02}	0.42	0.06	0.07
Arginine	6.76×10^{-02}	0.44	0.06	0.06
Asparagine	3.38×10^{-03}	0.59	0.04	0.03
Aspartic Acid	2.39×10^{-02}	0.57	0.05	0.05
Cysteine	3.63×10^{-03}	0.59	0.02	0.02
Glutamic Acid	4.14×10^{-01}	0.47	0.07	0.07
Glutamine	1.32×10^{-02}	0.42	0.04	0.04
Glycine	1.67×10^{-01}	0.54	0.07	0.06
Histidine	4.38×10^{-01}	0.48	0.03	0.03
Isoleucine	8.21×10^{-02}	0.56	0.05	0.05
Leucine	8.32×10^{-01}	0.49	0.10	0.10
Lysine	2.25×10^{-01}	0.46	0.06	0.06
Methionine	1.25×10^{-01}	0.55	0.02	0.02
Phenylalanine	2.42×10^{-01}	0.54	0.04	0.04
Proline	2.82×10^{-01}	0.47	0.06	0.06
Serine	3.86×10^{-02}	0.43	0.07	0.07
Threonine	6.87×10^{-02}	0.56	0.05	0.05
Tryptophan *	7.12×10^{-05}	0.63	0.01	0.01
Tyrosine *	3.69×10^{-04}	0.61	0.03	0.03
Valine	8.16×10^{-02}	0.56	0.06	0.06
Aliphatic	1.44×10^{-01}	0.55	0.21	0.21
Aromatic	1.83×10^{-03}	0.60	0.11	0.10
Charged	6.80×10^{-02}	0.44	0.26	0.27
Negatively Charged	6.31×10^{-01}	0.52	0.12	0.12
Non-polar	7.77×10^{-02}	0.56	0.53	0.53

Feature	P-value	PS	Positive Median	Unlabelled Median
Positively Charged	3.66×10^{-03}	0.41	0.14	0.15
Sequence Length	1.21×10^{-01}	0.55	682	587
PEST Motifs	4.16×10^{-02}	0.44	0	0
Low Complexity Regions	3.05×10^{-01}	0.47	2	2
Hydrophobicity	4.61×10^{-02}	0.56	-0.35	-0.38
Isoelectric Point	4.59×10^{-03}	0.41	6.87	7.12
Signal Peptide *	2.74×10^{-10}	0.63	0	0
***O*-glycosylation Sites**	2.64×10^{-01}	0.50	0	0
***N*-glycosylation Sites ***	3.84×10^{-12}	0.64	0	0
Phosphoserine Sites	1.78×10^{-01}	0.54	2	1
Phosphothreonine Sites	2.94×10^{-02}	0.56	1	0
Phosphotyrosine Sites *	3.49×10^{-18}	0.74	2	0
Total Phosphorylation Sites *	4.05×10^{-08}	0.67	8	3
Transmembrane α-helices *	4.34×10^{-08}	0.62	0	0
Exposed α-helices *	9.94×10^{-06}	0.36	0.10	0.13
Buried α-helices	9.53×10^{-02}	0.45	0.13	0.15
β Strands *	5.12×10^{-10}	0.70	0.17	0.13
3' Untranslated	2.16×10^{-01}	0.54	2	1
5' Untranslated	2.41×10^{-01}	0.54	2	1
Nonsynonymous Coding	5.29×10^{-03}	0.59	24	17
Synonymous Coding	4.58×10^{-01}	0.52	0	0
Binary PPIs	1.20×10^{-02}	0.58	2	1
Alternative Transcripts	2.12×10^{-01}	0.54	4	3
Paralogs	9.48×10^{-01}	0.50	0	0
Body Sites Expressed In	8.89×10^{-03}	0.58	33	31

Table 21: Results of the feature analysis for the $Kinase$ dataset. The p-values and the PS were calculated as in Section 5.4. Shaded features are ones for which the $PS \geq 0.5$. The amino acid, exposed α-helix, buried α-helix and β strand features are all proportions (e.g. the Alanine feature for a protein is the number of alanine residues in the sequence divided by the sequence length), while all other features are absolute numbers. Features with significant differences are indicated with an *.

Feature	$Kinase$	$Kinase\_NTK$	$Kinase\_TK$
Phosphotyrosine Sites	*** 0.24**	0.07	*** 0.48**
β Strands	*** 0.20**	0.08	*** 0.35**
Total Phosphorylation Sites	*** 0.17**	0.08	*** 0.30**
Exposed α-helices	***-0.14**	0.03	*** -0.37**
***N*-Glycosylation Sites**	*** 0.14**	0.03	*** 0.29**
Signal Peptide	*** 0.13**	0.02	*** 0.27**
Tryptophan	*** 0.13**	0.03	*** 0.25**
Transmembrane α-helices	*** 0.12**	0.01	*** 0.26**
Tyrosine	*** 0.11**	0.03	*** 0.22**
Aromatic	0.10	0.08	0.12
Arginine	0.09	0.04	*** 0.16**
Cysteine	0.09	0.05	0.15
Positively Charged	-0.09	0.00	*** -0.22**
Isoelectric Point	-0.09	-0.04	*** -0.16**
Nonsynonymous Coding	0.09	0.11	0.06
Body Sites Expressed In	0.08	0.11	0.05
Alanine	-0.08	-0.09	-0.07
Glutamine	-0.08	-0.03	-0.15
Binary PPIs	0.08	0.12	0.02
Aspartic Acid	0.07	*** 0.14**	-0.01

Table 22: Comparison of the feature effect sizes across the three datasets of kinases. Effect size deviation from $PS = 0.5$ (no effect) for the twenty features with the largest effect size in the $Kinase$ dataset. A negative value indicates that the positive proteins have smaller values than the unlabelled ones, while a positive value indicates that they have greater ones. Features with significant differences in a dataset are indicated with an *.

5.4.6.2 Target Predictions

The best combination of parameters and feature set for classifying the proteins in the $Kinase$ dataset was $numberTrees = 1000$, $mtry = 5$, a weight of 23 given to each observation in the in positive class, a random seed of -6712145332927501964 and the following thirty-two features out of the original 104 from Section 5.1.13:

- Amino acid compositions
    - The proportion of asparagine, aspartic acid, cysteine, glycine, lysine, methionine, serine, tryptophan, aliphatic, charged, negatively charged and non-polar residues.
- Simple sequence properties
    - The number of PEST motifs.
- Posttranslational modifications
    - The number of *N*-linked glycosylation, *O*-linked glycosylation, phosphothreonine and phosphotyrosine sites.
- Inter-protein relationships
    - The number of binary PPIs.
- Developmental stage expression
    - The embryoid body, fetus, neonate and juvenile expression levels.
- Body site expression
    - The bone, cervix, ear, liver, lymph node, parathyroid, salivary gland, thymus, umbilical cord and uterus expression levels.

The positive similarity of the proteins in the $Kinase$ dataset can be seen in Figure 37. The distribution of the proteins in the $Kinase$ dataset likely indicates that there is a strong similarity between the positive and unlabelled proteins, as there are no particularly large peaks in any of the more extreme bins (0.0-0.1 and 0.9-1.0). Using a cutoff of 0.5, the RF's predicted classifications were as follows:

Positive Observations				Unlabelled Observations				G Mean
Total	TPs	FNs	Sensitivity	Total	TNs	FPs	Specificity	
94	76	18	0.81	567	460	107	0.81	0.81

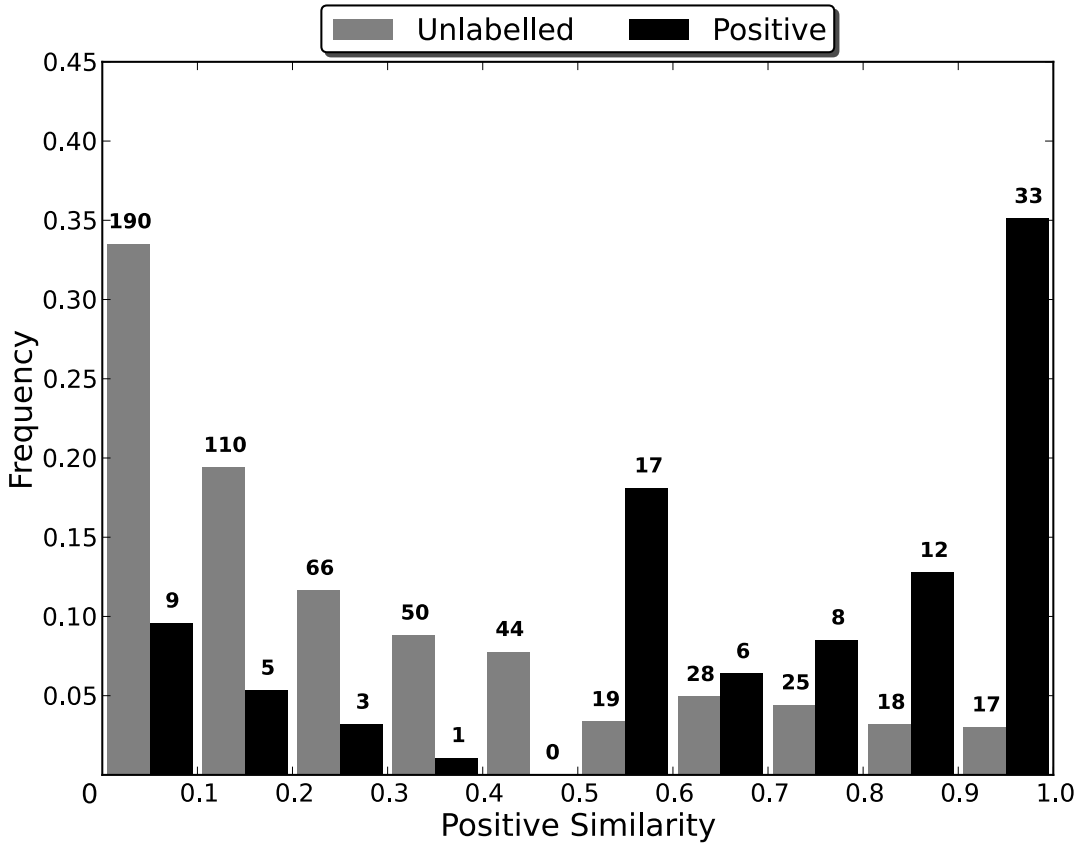

Figure 37: Weighted predictions of the proteins in the $Kinase$ dataset. The positive similarity of a given protein is equal to the fraction of the forest's votes that are for the positive class. The values over the bars indicate the number of proteins in the bin. The $Kinase$ dataset contained 567 unlabelled proteins and 94 positive ones.

Two clear trends can be discerned by looking at the types of the kinases in the $Kinase$ dataset (Table 23). Firstly, atypical kinases make poor targets. Of the twenty-seven kinases known to be atypical, only one is the target of an approved drug. Similarly, only one unlabelled atypical kinase is misclassified as positive, although with a positive similarity < 0.75. There are therefore no atypical kinases amongst the forty-nine unlabelled kinases that are most likely to be suitable targets, those with positive similarity ≥ 0.75. The second clear trend is the preferential targeting of tyrosine kinases. Despite only 14% of all kinases of known type being tyrosine kinases, they comprise 43% of the positive kinases of known type. Additionally, if the misclassified unlabelled proteins are included, then 73 of the 90 kinases that are known to be tyrosine kinases are targets or are believed to be suitable future targets. Further evidence of the disproportionate importance of tyrosine kinases as drug targets can be seen in the fact that 31% of the misclassified unlabelled proteins, and 53% of the unlabelled proteins with positive similarity ≥ 0.75, are tyrosine kinases.

In addition to the type of the kinase, the misclassified unlabelled proteins share with the positive proteins a tendency to be membrane bound. Although 17% of unlabelled and 34% of positive proteins are membrane proteins, 24% of the unlabelled proteins with positive similarity

> 0.5 are. However, of the unlabelled proteins with positive similarity ≥ 0.75, twenty (41%) are membrane bound. This further highlights the influence of tyrosine kinases on the prediction of kinase drug targets, as nineteen of the twenty misclassified membrane bound unlabelled proteins with positive similarity ≥ 0.75 are receptor tyrosine kinases. When considering receptor and non-receptor tyrosine kinases separately, the receptor tyrosine kinases make up the largest fraction of the misclassified unlabelled proteins with positive similarity ≥ 0.75. The importance of being membrane bound to the likelihood of a kinase being a suitable drug target is therefore likely to be more of a reflection of the importance of being a receptor tyrosine kinase. All unlabelled proteins in the $Kinase$ dataset predicted to be positive can be found in Appendix A Section IV.

	Serine/Threonine	Tyrosine	Atypical	Unknown
Entire Dataset	390 (59%)	90 (14%)	27 (4%)	154 (23%)
Unlabelled Proteins	355 (63%)	50 (9%)	26 (5%)	136 (24%)
Unlabelled Proteins With Positive Similarity > 0.5	56 (52%)	33 (31%)	1 (1%)	17 (16%)
Unlabelled Proteins With Positive Similarity ≥ 0.75	16 (33%)	26 (53%)	0 (0%)	7 (14%)
Positive Proteins	35 (37%)	40 (43%)	1 (1%)	18 (19%)

Table 23: Division of positive and unlabelled kinases by type. Table showing the distribution of proteins in the entire $Kinase$ dataset, all unlabelled proteins, misclassified unlabelled proteins and all positive proteins by kinase type. Unknown kinases are ones where it is not known whether they are Serine/Threonine, Tyrosine or atypical kinases.

5.4.7 Results – Proteases

5.4.7.1 Target Properties

The results from the analysis of the features in the $Protease$ dataset (using the method described in Section 5.4) can be seen in Table 24. Although these results indicate that significant differences between the positive and unlabelled proteins can be found, care must be taken due to the composition of the set of positive proteins. As 61% of positive proteins are metallo proteases it is possible that differences in the dataset are reflecting differences between a specific subset of the metallo proteases (the positive proteins) and proteases in general (the unlabelled ones). This is of particular concern due to the high level of similarity between the positive metallo proteases and low level of similarity between the positive metallo and non-metallo proteases (Section 5.4.7.3). In order to evaluate the effect of this subpopulation of positive metallo proteases, two further datasets were constructed. The $Protease\_MP$ dataset was constructed from the $Protease$ dataset by removing all positive proteins that were not metallo proteases, while the $Protease\_NMP$ dataset was constructed from the $Protease$ dataset by removing all positive proteins that were metallo proteases. Both of these datasets had their features analysed in terms of significance and effect size using the methods described in Section 5.4. A comparison between the features that are significant in each of the three datasets can be seen in Table 25. For each feature, the deviation of the effect size from 0.5 in the $Protease$ dataset can be seen to be between the deviations for the $Protease\_MP$ and $Protease\_NMP$ datasets, except for the glutamine proportion and synonymous coding variants. However, if the positive metallo proteases and non-metallo proteases shared similar properties, then the features with the greatest effect size deviations in the $Protease$ dataset would be expected to have the greatest deviations in the $Protease\_MP$ and $Protease\_NMP$ datasets. The pattern of deviations therefore indicates that there are distinct differences between the positive metallo proteases and the positive non-metallo ones. This can be seen most clearly in the proportion of cysteine in the proteins, with both the $Protease\_MP$ and $Protease\_NMP$ datasets having sizable effects for it, but with the positive proteins having a greater proportion of cysteine in the $Protease\_NMP$ dataset and a smaller proportion in the $Protease\_MP$ one. Additionally, of the ten significant features in the $Protease$ dataset, eight were found to be significant in the $Protease\_MP$ dataset, while only one was significant in the $Protease\_NMP$ dataset. These results indicate that the differences in the $Protease$ dataset are likely reflecting the differences between the positive metallo proteases and the unlabelled proteases, rather than capturing properties of protease drug targets in general.

As the metallo proteases (both positive and unlabelled) appear to cluster together, one method for overcoming the problems with the $Protease$ dataset would be to compare positive metallo proteases to unlabelled ones. The same could then be done for non-metallo proteases, or alternatively for further subsets of the $Protease$ dataset (e.g. serine proteases). However, this

approach would be problematic due to the small size of the set of positive proteins, as there are only thirty-six positive metallo proteases, and the possible bias towards certain metallo proteases having been selected as early targets due to their specific properties or the simplicity of targeting them. Therefore, more positive proteases are needed in order to accurately determine the properties of protease drug targets.

Feature	P-value	PS	Positive Median	Unlabelled Median
Alanine	8.99×10^{-02}	0.57	0.07	0.06
Arginine	2.25×10^{-02}	0.59	0.06	0.05
Asparagine	7.89×10^{-01}	0.49	0.04	0.04
Aspartic Acid *	5.34×10^{-05}	0.66	0.06	0.05
Cysteine *	3.71×10^{-05}	0.34	0.01	0.03
Glutamic Acid	1.70×10^{-01}	0.45	0.05	0.06
Glutamine *	3.56×10^{-04}	0.36	0.04	0.04
Glycine	1.23×10^{-01}	0.56	0.08	0.08
Histidine	3.58×10^{-01}	0.46	0.03	0.03
Isoleucine	2.03×10^{-01}	0.45	0.04	0.05
Leucine	8.30×10^{-03}	0.40	0.09	0.09
Lysine	2.98×10^{-01}	0.46	0.05	0.05
Methionine	9.92×10^{-01}	0.50	0.02	0.02
Phenylalanine *	1.47×10^{-04}	0.65	0.04	0.04
Proline	3.20×10^{-01}	0.54	0.06	0.06
Serine *	1.43×10^{-05}	0.33	0.06	0.07
Threonine	1.17×10^{-01}	0.56	0.05	0.05
Tryptophan	1.32×10^{-01}	0.56	0.02	0.02
Tyrosine *	2.83×10^{-06}	0.68	0.04	0.03
Valine	2.80×10^{-03}	0.38	0.06	0.06
Aliphatic *	2.41×10^{-05}	0.33	0.19	0.21
Aromatic *	3.36×10^{-06}	0.68	0.13	0.12
Charged	5.48×10^{-01}	0.52	0.25	0.25
Negatively Charged	1.69×10^{-01}	0.55	0.11	0.11
Non-polar	1.33×10^{-01}	0.56	0.56	0.55

Feature	P-value	PS	Positive Median	Unlabelled Median
Positively Charged	6.29×10^{-01}	0.52	0.14	0.13
Sequence Length	6.61×10^{-01}	0.48	478	497
PEST Motifs	8.80×10^{-02}	0.44	0	0
Low Complexity Regions	4.34×10^{-01}	0.47	1	2
Hydrophobicity	2.23×10^{-02}	0.41	-0.39	-0.30
Isoelectric Point	6.88×10^{-01}	0.48	6.97	7.06
Signal Peptide *	8.10×10^{-04}	0.62	1	0
***O*-glycosylation Sites ***	6.08×10^{-08}	0.57	0	0
***N*-glycosylation Sites**	1.81×10^{-02}	0.59	1	0
Phosphoserine Sites	2.21×10^{-01}	0.47	0	0
Phosphothreonine Sites	3.58×10^{-01}	0.48	0	0
Phosphotyrosine Sites	7.18×10^{-01}	0.49	0	0
Total Phosphorylation Sites	5.74×10^{-01}	0.48	0	0
Transmembrane α-helices	7.23×10^{-01}	0.51	0	0
Exposed α-helices	4.04×10^{-01}	0.47	0.07	0.09
Buried α-helices	9.19×10^{-01}	0.50	0.11	0.11
β Strands	9.23×10^{-02}	0.57	0.21	0.17
3' Untranslated	5.15×10^{-01}	0.48	0	0
5' Untranslated	1.78×10^{-01}	0.45	0	0
Nonsynonymous Coding	3.23×10^{-01}	0.54	13	12
Synonymous Coding	2.53×10^{-02}	0.57	0	0
Binary PPIs	8.77×10^{-01}	0.51	0	0
Alternative Transcripts	3.39×10^{-01}	0.46	2	2
Paralogs	1.22×10^{-01}	0.46	0	0
Body Sites Expressed In	5.70×10^{-01}	0.52	25	25

Table 24: Results of the feature analysis for the $Protease$ dataset. The p-values and the PS were calculated as in Section 5.4. Shaded features are ones for which the $PS \geq 0.5$. The amino acid, exposed α-helix, buried α-helix and β strand features are all proportions (e.g. the Alanine feature for a protein is the number of alanine residues in the sequence divided by the sequence length), while all other features are absolute numbers. Features with significant differences are indicated with an *.

Feature	$Protease$	$Protease\_NMP$	$Protease\_MP$
Tyrosine	*** 0.18**	0.12	*** 0.22**
Aromatic	*** 0.18**	-0.02	*** 0.31**
Serine	*** -0.17**	-0.02	*** -0.27**
Aliphatic	*** -0.17**	-0.10	*** -0.21**
Cysteine	*** -0.16**	0.11	*** -0.34**
Aspartic Acid	*** 0.16**	0.04	*** 0.23**
Phenylalanine	*** 0.15**	-0.06	*** 0.29**
Glutamine	*** -0.14**	-0.14	-0.14
Valine	-0.12	-0.04	*** -0.17**
Signal Peptide	*** 0.12**	0.07	*** 0.15**
Leucine	-0.10	-0.13	-0.09
Hydrophobicity	-0.09	0.00	-0.15
Arginine	0.09	0.14	0.06
***N*-Glycosylation Sites**	0.09	0.06	0.10
***O*-Glycosylation Sites**	*** 0.07**	*** 0.17**	0.01
Synonymous Coding	0.07	0.09	0.07
Alanine	0.07	0.02	0.10
β Strands	0.07	0.16	0.01
Threonine	0.06	0.14	0.01
Glycine	0.06	0.14	0.01

Table 25: Comparison of the feature effect sizes across the three datasets of proteases. Effect size deviation from $PS = 0.5$ (no effect) for the twenty features with the largest effect size in the $Protease$ dataset. A negative value indicates that the positive proteins have smaller values than the unlabelled ones, while a positive value indicates that they have greater ones. Features with significant differences in a dataset are indicated with an *.

5.4.7.2 Target Predictions

The best combination of parameters and feature set for classifying the proteins in the $Protease$ dataset was$numberTrees = 1000$, $mtry = 10$, a weight of 20 given to each observation in the in positive class, a random seed of 8716758538734970127 and the following thirty-five features out of the original 104 from Section 5.1.13:

- Amino acid compositions
    - The proportion of arginine, cysteine, isoleucine, glycine, proline, tryptophan and aromatic residues.
- Simple sequence properties
    - The number of PEST motifs.
- Posttranslational modifications
    - The number of *O*-linked glycosylation, *N*-linked glycosylation, phosphothreonine and phosphotyrosine sites.
- Secondary structure
    - The fraction of residues predicted to participate in buried α-helices.
- Germline variants
    - The number of 3' and 5' untranslated mutations.
- Inter-protein relationship
    - The number of paralogs.
- Developmental stage expression
    - The embryoid body, neonate and infant expression levels.
- Body site expression
    - The adipose tissue, blood, bone, cervix, connective tissue, ear, embryonic tissue, mouth, nerve, placenta, prostate, salivary gland, thymus, trachea, umbilical cord and uterus expression levels.

The positive similarity of the proteins in the $Protease$ dataset can be seen in Figure 33. The distribution of the proteins in the $Protease$ dataset likely indicates that there is a strong similarity between the positive and unlabelled proteins, as there are no particularly large peaks in any of the more extreme bins (0.0-0.1 and 0.9-1.0). Using a cutoff of 0.5, the RF's predicted classifications were as follows:

Positive Observations				Unlabelled Observations				G Mean
Total	TPs	FNs	Sensitivity	Total	TNs	FPs	Specificity	
59	52	7	0.88	472	419	53	0.89	0.88

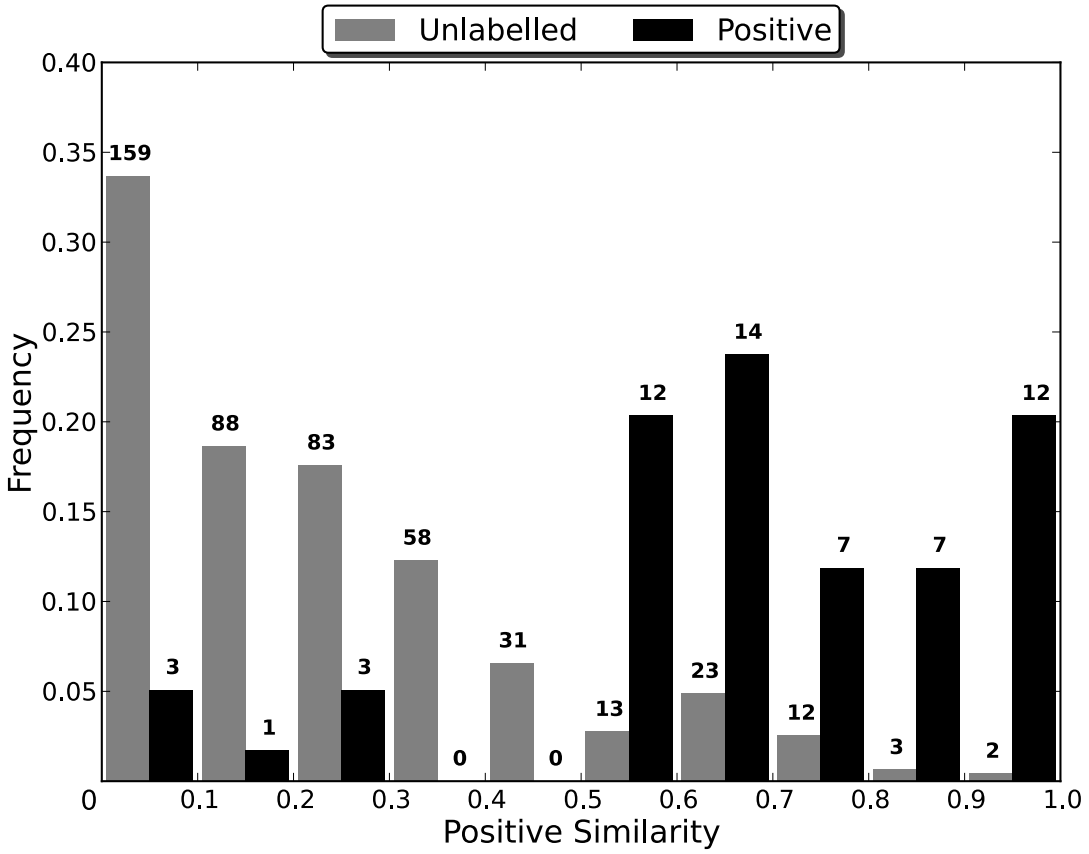

Figure 38: Weighted predictions of the proteins in the $Protease$ dataset. The positive similarity of a given protein is equal to the fraction of the forest's votes that are for the positive class. The values over the bars indicate the number of proteins in the bin. The $Protease$ dataset contained 472 unlabelled proteins and 59 positive ones.

As can be seen from Table 26, the distribution of the types of all misclassified proteases closely follows that of the entire $Protease$ dataset, rather than the set of positive proteins. However, when only those unlabelled proteases that are most likely to make suitable drug targets are considered, those with a positive similarity ≥ 0.75, the distribution of the types of the misclassified unlabelled proteins is much closer to that of the positive proteins. For example, although four unlabelled aspartic proteases are misclassified, none of them have a positive similarity ≥ 0.75. Similarly, although 34% of all misclassified unlabelled proteases are metallo proteases, 50% of misclassified unlabelled proteases with positive similarity ≥ 0.75 are. The potential drug targets with positive similarity ≥ 0.75 are also more similar to the positive proteins in terms of their propensity to be membrane bound. While 23% of all misclassified unlabelled proteins are membrane bound, 36% of the unlabelled proteins with positive similarity ≥ 0.75 are, in close agreement with the 35% of positive proteins that are membrane bound. The small number of misclassified unlabelled proteins with the more confident predictions is likely due to the small number of positive proteins. As the set of positive proteins increases, it is likely that what constitutes similarity to a positive protein will begin to broaden, and more unlabelled proteins will be deemed to be potential drug targets. Similarly, as non-metallo proteases begin to

compose more of the set of positive proteins the fraction of the most confident potential drug target predictions that are metallo proteases will likely decrease. All unlabelled proteins in the $Protease$ dataset predicted to be positive can be found in Appendix A Section V.

	Aspartic	Cysteine	Metallo	Serine	Threonine
Entire Dataset	31 (6%)	135 (25%)	167 (31%)	161 (30%)	19 (4%)
Unlabelled Proteins	29 (6%)	133 (28%)	131 (28%)	148 (31%)	13 (3%)
Unlabelled Proteins With Positive Similarity > 0.5	4 (8%)	9 (17%)	18 (34%)	14 (26%)	5 (9%)
Unlabelled Proteins With Positive Similarity ≥ 0.75	0 (0%)	2 (14%)	7 (50%)	3 (21%)	2 (14%)
Positive Proteins	2 (3%)	2 (3%)	36 (61%)	13 (22%)	6 (10%)

Table 26: Division of positive and unlabelled proteases by type. Table showing the distribution of all unlabelled proteins, misclassified unlabelled proteins and all positive proteins by protease type. The 18 unlabelled proteases of an unknown type are not shown.

5.4.7.3 Dataset Homogeneity

In order to further clarify the reasons for the differences in the G means of the RFs, the homogeneity of each dataset was estimated. For each dataset, the pairwise sequence identity between all pairs of proteins in the dataset was calculated using BLAST (following the method in Section 5.2). The proteins in a pair were considered to be similar if their pairwise sequence identity was at least 20%. This threshold was chosen as it is generally the lowest threshold at which sequence alignments can still be considered reasonable estimates of homology. As the maximum number of similar pairs, excluding identity pairs, for a set of N proteins is $\binom{N}{2} = \frac{N!}{2!(N-2)!}$, the percentage of all possible pairs of two positive proteins, two unlabelled proteins and one positive and one unlabelled protein that are similar could be calculated for each dataset. This percentage indicates the level of similarity between proteins in a given set, with a higher percentage indicating that the proteins in the set are more similar and interconnected. The results of the similarity comparisons can be seen in Table 27.

Dataset	All Pairs	Pairs of Two Positive Proteins	Pairs of Two Unlabelled Proteins	Pairs of One Unlabelled and One Positive Protein
$AllTargets$	0.47%	0.92%	0.49%	0.28%
$Cancer$	1.23%	2.59%	1.29%	0.60%
$GPCR$	33.41%	35.47%	39.17%	15.38%
$GPCR\_NO$	20.20%	35.47%	16.81%	21.46%
$IonChannel$	5.27%	9.69%	4.40%	3.66%
$Kinase$	31.45%	42.44%	30.67%	32.89%
$Protease$	6.54%	18.81%	6.41%	6.26%

Table 27: Comparison of pairs of proteins with pairwise sequence identity of at least 20%. For each dataset, the values indicate the percentage of all possible pairs, pairs consisting of two positive proteins, pairs consisting of two unlabelled proteins and pairs consisting of one positive and one unlabelled protein for which the pairwise sequence identity was at least 20%.

From the results it can be seen that datasets with a large percentage of similar pairs, the $GPCR\_NO$ and $Kinase$ datasets, induce RFs with low G means. The high percentage of similar inter-class pairs is likely to be particularly problematic for a RF's classifications, as this indicates that the positive and unlabelled proteins are highly similar and likely more difficult to separate and classify well. This problem can best be seen in the differences between the results for the $GPCR$ and $GPCR\_NO$ datasets. While the inter-class similarity is high in the $GPCR$ dataset, it is six percentage points greater in the $GPCR\_NO$ dataset. Additionally, although the 291 non-odorants make up 41% of the unlabelled proteins, they are involved in 57% of the similar inter-class pairs, while the 421 odorants are involved in 43%.The odorants can therefore be seen to be much less similar to the positive proteins than the unlabelled non-odorants, and likely form a highly

interconnected cluster separate from a second cluster of non-odorants. The removal of the odorants from the dataset will therefore remove a large source of proteins that were relatively simple to classify correctly, along with highlighting the true similarity between the proteins that have potential to serve as targets (non-odorants). This will result in a substantially lower G mean for the RF trained on the $GPCR\_NO$ dataset when compared to the one trained on the $GPCR$ dataset.

In contrast to the $GPCR\_NO$ and $Kinase$ datasets, the $AllTargets$ dataset has very low percentages for all pair types, but still induced a RF with a low G mean. This poor performance indicates that protein datasets can induce poorly performing RFs as a result of being too heterogeneous as well as by being too homogeneous. However, in the case of heterogeneous datasets, it is likely the low level of intra-class similarity that is problematic, as with a low level of intra-class similarity the clustering that the RF relies on to provide accurate classifications is absent. Additionally, highly heterogeneous datasets will cause the individual trees in the forest to show greater variance in their classifications of a given feature subspace, resulting in less confident aggregate predictions.

Although there were no difficulties in obtaining a large G mean for RFs induced from the $Protease$ dataset, the large proportion of metallo proteases in the positive proteins, and the high level of similarity between them, could prove problematic. By looking at the similarity between specific subpopulations of the $Protease$ dataset (Table 28) it can be seen that the similarities between the proteins in the dataset are largely intra-type, i.e. between two metallo or two non-metallo proteases. Out of the 9231 pairs of proteins in the dataset that are similar, only 185 (2.0%) of the pairs include one metallo and one non-metallo protease. The lack of similarity is particularly striking for the pairs of positive proteins, where only 2 (0.6%) pairs of similar proteins consist of a metallo and non-metallo protease. These results demonstrate that the $Protease$ dataset is divided into a minimum of two clusters, one of metallo and one of non-metallo proteases, with the cluster of non-metallo proteases potentially containing further subclusters (such as of serine proteases). While this clustering will prove problematic for the analysis of the important features in the $Protease$ dataset, as there is no one cluster of 'drug target-like proteins', there is no reason to expect that it would prove to be problematic for the RF classifications, as RFs can easily work with datasets that contain distinct clusters in separate subspaces.

		Positive		Unlabelled	
		Metallo	Non-metallo	Metallo	Non-metallo
Positive	Metallo	257	2	457	12
	Non-metallo		74	42	1233
Unlabelled	Metallo			583	129
	Non-metallo				6442

Table 28: Similarities between pairs of proteins in the $Protease$ dataset. Each cell corresponds to one of the possible protein pair combinations. For example, there are 257 pairs consisting of two positive metallo proteases and 129 consisting of an unlabelled metallo protease and an unlabelled non-metallo protease.

6 Discussion

6.1 Improving Redundancy Removal

6.1.1 Protein Datasets

The increased improvement over PISCES by all algorithms, excluding GLP, on larger datasets and smaller thresholds is due to the increased potential for redundancy in these datasets. As the number of proteins in the dataset increases, the number of potential pairs of proteins that have a sequence identity greater than the given threshold increases. There is therefore likely to be more redundancy in the dataset, and consequently more room for improvement over PISCES. Similarly, as the threshold decreases, more pairs of proteins will have a pairwise sequence identity greater than the threshold, and therefore the protein similarity graph of the dataset will become denser. A denser graph means that there is more redundancy in the dataset, and therefore more opportunities for improving on the size of the non-redundant dataset generated by PISCES.

The poor performance of GLP is likely due to its reliance on random search combined with the time limit imposed on it. GLP builds up an independent set in a manner similar to FIS, except that the choices of initial vertex, vertices to add and vertices to swap are made with no attempt to determine the 'best' vertex. As the algorithm progresses through a local search, it effectively performs a random walk through a localised subsection of the space of possible maximal independent sets. After determining that this subspace has been searched thoroughly enough, or that the search has left the subspace it is restricted to, the algorithm is restarted by selecting a new random initial vertex. Even without limiting the run time of the algorithm, the reliance on a purely randomised approach means that it cannot be guaranteed that an MIS, or even a large maximal one, will ever be found. Rather, the longer that the algorithm is allowed to run for, the more likely it is to randomly begin its search in a subspace containing a maximum, or large maximal, independent set. Therefore, as the permissible run time becomes shorter, the algorithm is able to search a smaller and smaller fraction of the search space, and without the direction enforced on the search by evaluating the quality of the vertices, there is no way to ensure that the search is focussed on subspaces with larger independent sets. The size of the non-redundant dataset found by GLP therefore depends heavily on the severity of the constraints imposed on the search by the time limit, the topography of the search space (e.g. how easy it is to find a large maximal independent set in it) and that one of the restarts by chance begins the search in a subspace with many large maximal independent sets. Unfortunately, as the dataset gets larger the required time limit becomes increasingly large, due to the increased size of the search space, and therefore GLP is inappropriate for larger graphs when a time limit is necessary.

6.1.2 BHOSLIB

The difference in the performances of the algorithms on the BHOSLIB and protein similarity graphs is due to the increased density and high connectivity of the BHOSLIB ones. This causes the vertex quality measurement methods used by the algorithms to be unable to identify a unique vertex that should be removed. The algorithms therefore resort to selecting arbitrary vertices, and in the case of Leaf, falling back onto the NeighbourCull removal method. Although arbitrary removals will become less frequent as the algorithms progress, the initial need for arbitrary choices will hamper the ability of the algorithms to find large independent sets. Despite the difficulty of evaluating vertex quality being the same for all algorithms, FIS and GLP are able to overcome this constraint through the use of local search and randomisation. For FIS, the permutations that it employs after finding an initial maximal independent set allow it to perform a local search, thereby further exploring an area of the search space without the need for evaluating the quality of the vertices. This enables it to perform a more thorough search, and also to mitigate the adverse effects of the initial arbitrary choices. In the case of GLP, its random nature means that the density and connectivity of the graph has no effect on its ability to find a large maximal independent set. Additionally, as the BHOSLIB graphs are dense, their complement, which GLP searches through, is sparser than the complement of the protein similarity graphs. This means that GLP can search the space of maximal independent sets more quickly, and therefore more thoroughly.

6.1.3 Conclusions

GLP is clearly an inappropriate replacements for PISCES, as it performs worse than the other graph based algorithms, and often worse than PISCES itself. Even with the time limit extended to 500 times that of Leaf's, the non-redundant datasets returned by GLP are smaller than those found by Leaf, and occasionally than those of PISCES. Of the remaining algorithms, Leaf consistently generated the largest non-redundant datasets from model organism proteomes and random subsets of the human proteome. However, it is not a general solution for finding an MIS, as evidenced by it being the third best algorithm on the BHOSLIB benchmarks. Although Cliquer could be used to find an MIS for datasets of up to 1000 proteins, other algorithms often found one as well, and in a shorter time. For larger protein datasets, where Cliquer is too slow to use, Leaf proved to be the best method, as it finds the largest non-redundant datasets in a suitably short time. Leaf is therefore the most suitable algorithm for generating non-redundant protein datasets.

6.2 Sequence Identity Comparison

By basing the definition of redundancy on sequence similarity, the proteins in the dataset are being placed in a similarity space, where the distance between any two proteins is related to their pairwise sequence identity. In this space, groups of similar proteins will form clusters, while dissimilar ones will be scattered farther apart. The protein similarity graph captures this information, and simplifies it by only connecting proteins that reach a certain level of similarity. As redundancy removal is achieved by ensuring that no two proteins in the non-redundant dataset share an edge, it conceptually functions by thinning out user-defined clusters of proteins in the similarity space through the replacement of a cluster by a representative subset of its proteins.

There are two situations where this thinning out of clusters is necessary: when you want to generate a representative dataset or when you believe that your dataset is biased. The goal when generating a representative dataset is to cover the same subset of the similarity space covered by the original dataset, maximise coverage, while using as few proteins as possible, minimise 'redundancy'. Bias in a dataset, in the context of redundancy removal, is taken to mean that the distribution of the proteins in the dataset throughout the similarity space is not the same as the true distribution of the entire population of proteins. Certain similarity subspaces will therefore contain more proteins than they would under the true distribution, and consequently have a disproportionate influence on conclusions drawn from the dataset. Thinning out clusters of proteins in subsections of the similarity space overpopulated due to biases can therefore be used to rebalance the dataset back towards the true distribution.

For our particular application, the generation of a representative set is much less important, due to computational resources not being stretched, than bias removal. However, bias is only a concern when the dataset is a sample that has been drawn from a population of proteins and is being used to draw conclusions about the population. In the case of the $AllTargets$ dataset, the population of proteins under consideration is the entire human proteome, while in the case of the $GPCR$, $IonChannel$, $Kinase$ and $Protease$ datasets, the populations are the set of all human GPCRs, ion channels, kinases and proteases respectively. As the dataset being used in all five cases is the same as the population of interest, there is no potential source of selection bias or problems due to generalising to proteins outside the dataset. Unlike the datasets based on protein families, the proteins in the $Cancer$ dataset are influenced by past discoveries and historical research preferences, none of which are without bias, and can therefore be seen to be a biased sample of the entire human proteome. However, as no generalisations are being made to proteins outside this biased sample, the conclusions drawn about the proteins in the $Cancer$ dataset will not themselves be biased. Therefore, as bias is not a concern for any of the datasets used here, there is no theoretical reason for them to undergo redundancy removal.

Although the similarity based removal of proteins from the datasets is not necessary, the removal of observations from a dataset can potentially improve the quality of an algorithm trained on it. However, in the case of the protein datasets examined here, the best RFs were always induced using the entire dataset, indicating that the removal of proteins with similar sequences does not improve a classifier's performance. Despite this, there is no indication that measuring similarity between proteins based on their sequences and defining the decision boundary based on the dataset's features was particularly detrimental. Rather, it was likely the act of removing proteins in general that led to the decrease in performance. Therefore, due to the lack of theoretical need or practical benefit, redundancy removal was avoided when determining the properties of and classifying drug target proteins.

6.3 Identification of Targets and Their Properties

6.3.1 Target Prediction and Properties

One noticeable trend across the datasets is the lack of effect for features that could be considered to represent interactions between proteins. The primary measures of this were the number of binary PPIs and the number of phosphorylation sites, as phosphorylation sites are indicative of a protein's involvement in regulatory networks. Despite the biological importance of interactions between proteins, the size of the effect of both the difference in binary PPIs and phosphorylation sites was small for all datasets, with the difference in total phosphorylation sites in the $Cancer$ dataset having the largest effect ($PS = 0.40$). Although it is unclear whether interactions between proteins would be expected to be more or less likely to occur in targets, the lack of importance is perhaps surprising given the importance of the regulation of proteins and the interactions between them.

Another set of features that were minimally important across the datasets is the germline variants. Of the four variant types investigated, consequential effects were seen for synonymous coding variants in the $Cancer$ dataset (with a very small effect), 3' and 5' untranslated variants in the $Cancer$ dataset (both with moderately large effects) and nonsynonymous coding variants in the $AllTargets$ and $Cancer$ datasets (with very small and moderate effects respectively). The unique pattern of effects for the $Cancer$ dataset is predominantly a result of the greater number of variants in the unlabelled proteins, and is most likely due to the characteristics of cancer, rather than the fact that the $Cancer$ dataset is composed of proteins implicated in a disease instead of based on protein family membership. Although the number of germline variants is relatively unimportant for current drug targets, were personalised medicine to become commonplace, it is possible that proteins with a larger number of known mutations that alter their expression or

activity would become more likely to be drug targets, as a greater number of variants would mean that there is more potential for targeting them.

Although the number of PEST motifs and the number of *N*-linked glycosylation sites are both believed to be important in degradation control, their effects do not correlate strongly. As the number of *N*-linked glycosylation sites is greater in the positive proteins for all tested datasets, it would be expected that the positive proteins have fewer PEST motifs, due to them having a longer *in vivo* half-life. However, this is only true for two of the four datasets where the features could be analysed accurately. Even in the datasets where there are fewer PEST motifs in the positive proteins, the effect is always small in both absolute terms and relative to that of the *N*-linked glycosylation sites. These results indicate that there must be some substantial difference in the degradation protection provided by having fewer PEST motifs and more *N*-linked glycosylation sites, or that additional functions of the two are important in helping to determine the differences in their effect sizes.

The clearest difference between positive and unlabelled proteins is in the proportion of non-polar/polar amino acids and in the likelihood of being membrane bound. If the $GPCR\_NO$ and $IonChannel$ datasets are discounted, as they consist solely of transmembrane proteins, the number of transmembrane helices can be seen to have a moderate to large effect for both the $AllTargets$ and $Cancer$ datasets. However, having a greater number of transmembrane helices does not by itself indicate that the positive proteins are more likely to be membrane bound. Rather, the fact that the percentage of positive proteins with a transmembrane helix is substantially greater than the percentage of unlabelled ones in the $AllTargets$ (43% compared to 24%) and $Cancer$ (55% compared to 11%) datasets indicates that positive proteins are much more likely to be membrane bound. When coupled with the size of the effect for the sequence length in the $GPCR\_NO$ and $IonChannel$ datasets, the tendency of proteins to be membrane bound explains the differences in the fraction of non-polar/polar amino acid residues. Being membrane bound is therefore important for, and highly indicative of, a protein being a drug target, and raises the question of whether the results are capturing properties that are truly indicative of being a drug target, or simply of being a membrane bound protein. However, it is believed to be unlikely that the results are simply highlighting properties of membrane bound proteins due to the differences in the effects of the important features for each dataset, and the results of previous experiments (Bakheet & Doig 2009) having shown that approaches similar to those taken here do not solely capture the properties of membrane bound proteins. The performances of the RFs would also seem to indicate that there are real differences between the datasets that cannot be explained by the predisposition of targets to being membrane bound, as

the difference in transmembrane helices does not correlate strongly with the G mean of the optimised classifier.

6.3.2 Dataset Homogeneity

The homogeneity of the datasets is of particular importance when attempting to predict potential targets or determine the properties important for the successful targeting of a protein. Of the three datasets that induced poorly performing RFs ($AllTargets$, $GPCR\_NO$ and $Kinase$), the $GPCR\_NO$ and $Kinase$ datasets were very homogenous, as seen by the high percentage of pairs of proteins that were similar. Conversely, the $AllTargets$ dataset was shown to induce poorly performing RFs due to the heterogeneity of the dataset. The performance of the RFs induced using the $LargeFamilies$ and $SmallFamilies$ datasets indicate that this is likely due to a combination of the presence of overlapping subpopulations and the difficulty of classifying proteins from smaller families. The differences in the subpopulations likely cause there to be more overlap between clusters of unlabelled and positive proteins, while the proteins from smaller families likely negatively impact the clustering of proteins from larger families. The G mean of the RF trained on the $AllTargets$ dataset is also likely to be optimistic for the same reasons that the RF trained on the $GPCR$ dataset was. However, in the case of the $AllTargets$ dataset there are possibly more subpopulations than just the odorant GPCRs that are overly simple to classify, potentially increasing the favourable bias in the results.

Despite the low G mean of the RFs induced using the $AllTargets$, $GPCR\_NO$ and $Kinase$ datasets, any features that are determined to have an effect on the likelihood of a protein being a suitable drug target are not invalidated by the homogeneity of the datasets. Rather, features are simply less likely to be found to be important when the positive and unlabelled proteins are excessively similar or dissimilar. Conversely, with the $Protease$ dataset the homogeneity of the positive proteins proves to be problematic for the determination of the important features but not for the capabilities of the RF induced from it. This is because RFs can easily handle datasets with distinct subpopulations, due to their partitioning of the feature space. The cluster of metallo proteases will therefore not influence the RFs performance on the cluster of non-metallo proteases, as the clusters occupy different feature subspaces. The classifications are therefore not unfavourably affected by the large proportion of positive metal proteases. Rather it simply makes the RF better at determining potential metallo protease drug targets than it is at determining potential non-metallo protease targets.

6.3.3 Random Forests Mitigate the Potential for Overfitting

For most problems, using the same dataset to optimise the parameters, select the optimal feature set and train and evaluate the final classifier would lead to severe overfitting. However, the atypical nature of the problem addressed here lends itself to this approach without risking overfitting. This is because overfitting occurs when a classifier fits limited training observations too closely, and therefore describes the properties of the training set rather than the underlying relationships between features. However, in our case the data available for training is the entire population rather than a sample of it, and there are therefore no observations that can be generalised to. The ideal classifier would therefore be optimised for performance on the training set, as this will optimise the classifier for performance on the entire population.

Despite the lack of need for any generalisation capabilities, we would like to extract potential future drug targets from the unlabelled proteins. This requires the ability to make informed predictions, rather than simply describing the differences between the positive and unlabelled proteins. However, training and evaluating the classifier on the same dataset would in general lead to severely biased predictions. This bias would make unlabelled proteins overly likely to be classified as unlabelled, and vice versa for positive proteins, thereby causing potential drug targets to be missed. We would therefore like to use our entire dataset for both optimising the parameters and training the final classifier, while still being able to use the final classifier to make unbiased predictions about the observations in our dataset.

In order to make an unbiased prediction about an observation, i, the classifier used to make the prediction must not have been trained on a dataset that included i. This dilemma leads to the concept of internal generalisation, whereby we want to be able to generalise from our dataset, $\mathcal{D}$, to an observation $i \in \mathcal{D}$ using some subset of observations, $\mathcal{T} \subset \mathcal{D}$, where $i \notin \mathcal{T}$. For the majority of classification algorithms the best way to do this would be to train $|\mathcal{D}|$ classifiers. Each classifier, c_i, is trained using the set of observations $\mathcal{T}_i = \mathcal{D} - i$, and is used to predict the class of observation i. This is similar to the leave-one-out cross validation approach used to test classifier performance, but is instead being used to form the final prediction of an observation. However, this approach requires training too many classifiers to be feasible even for small datasets. Rather than using cross validation to train a set of classifiers, a single RF, R, can be trained using $\mathcal{D}$ as the training set. Once R has been trained, each observation $i \in \mathcal{D}$ is predicted using only those trees in R for which it is OOB, thereby giving an unbiased prediction of the class of i. The parameters and feature set used to train R can therefore be optimised using $\mathcal{D}$, while still allowing unbiased predictions of the observations in $\mathcal{D}$ to be made. In this manner RFs can enable a population dataset to be used as both the training set and the set of observations that are to be predicted, without worrying about the final predictions being biased.

6.3.4 Conclusions

Subdividing the entire human proteome in order to create a relatively homogenous subset of proteins is necessary for forming an accurate picture of the features that are important for determining a protein's drug target likeness. While the heterogeneous $AllTargets$ dataset does provide some information about drug targets in general, the effect sizes of the individual features are small and the classifications inaccurate. In contrast, datasets formed from more homogenous subsets generally had features with larger effect sizes, and all induced RFs with greater G means. However, protein datasets can quickly become too homogeneous, negatively impacting the classification capability of a RF trained on them. Care is therefore needed when deciding on a subset of the human proteome to use, as certain subdivisions will produce datasets with vastly different levels of homogeneity. The ideal dataset would have distinct subpopulations (like the $Protease$ dataset) or a small but sufficient level of homogeneity (like the $Cancer$ and $IonChannel$ datasets). This homogeneity does not have to be based on family membership or even disease class, but could come from structural, functional or other properties of proteins instead. Despite the restrictions placed on the datasets by the homogeneity requirements, potential targets can be predicted and properties important for the targeting of proteins determined.

The properties that were most important in differentiating targets from non-targets were found to be the proteins' hydrophobicities, *in vivo* half-lives, propensity for being membrane bound and the fraction of non-polar amino acids in their sequences. Taken together, the importance of these properties indicates that drug targets are predominantly membrane bound proteins, and therefore non-polar, with long *in vivo* half-lives. However, in the case of the datasets that consist solely of membrane bound proteins, the $GPCR\_NO$ and $IonChannel$ datasets, the targets are predominantly more polar, rather than non-polar, due to the greater proportion of their sequence that resides in the extra and intracellular spaces. Whilst the primary importance of these general properties holds for all datasets, the $Cancer$ dataset contained additional properties of secondary importance. These secondary features were predominantly associated with the specific and reliable activity/expression of the proteins (e.g. phosphorylation sites and germline variants), and likely indicate that from amongst all proteins involved in cancer, those with the most specific and reliable activity are preferentially chosen to be antineoplastic targets. As the $Cancer$ dataset also showed the most pronounced effect for the importance of the general properties, the range and strength of the importance of the properties in the $Cancer$ dataset when compared to the other datasets, along with the high quality of the RF induced using it, likely indicates that subdivisions based on a disease possess the most promise for informative future analysis.

7 Bibliography

Adams, C.P. & Brantner, V.V., 2010. Spending on new drug development. *Health economics*, 19(2), pp.130–141.

Adams, J.A., 2001. Kinetic and Catalytic Mechanisms of Protein Kinases. *Chemical Reviews*, 101(8), pp.2271–2290.

Agren, M., Kogerman, P., Kleman, M.I., Wessling, M. & Toftgård, R., 2004. Expression of the PTCH1 tumor suppressor gene is regulated by alternative promoters and a single functional Gli-binding site. *Gene*, 330, pp.101–14.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J., 1990. Basic local alignment search tool. *Journal of molecular biology*, 215(3), pp.403–10.

Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D.J., 1997. Gapped BLAST and PSI-BLAST : a new generation of protein database search programs. , 25(17), pp.3389–3402.

Andersen, O.S., 2008. Perspectives on how to drug an ion channel. *The Journal of general physiology*, 131(5), pp.395–7.

Armstrong, C.M. & Hille, B., 1998. Voltage-gated ion channels and electrical excitability. *Neuron*, 20(3), pp.371–80.

Arora, A. & Scholar, E.M., 2005. Role of tyrosine kinase inhibitors in cancer therapy. *The Journal of pharmacology and experimental therapeutics*, 315(3), pp.971–9.

Artandi, S.E. & DePinho, R.A., 2010. Telomeres and telomerase in cancer. *Carcinogenesis*, 31(1), pp.9–18.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. & Sherlock, G., 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1), pp.25–9.

Ashcroft, F.M., 2006. From molecule to malady. *Nature*, 440(7083), pp.440–7.

Baeriswyl, V. & Christofori, G., 2009. The angiogenic switch in carcinogenesis. *Seminars in cancer biology*, 19(5), pp.329–37.

Bagal, S.K., Brown, A.D., Cox, P.J., Omoto, K., Owen, R.M., Pryde, D.C., Sidders, B., Skerratt, S.E., Stevens, E.B., Storer, R.I. & Swain, N.A., 2013. Ion channels as therapeutic targets: a drug discovery perspective. *Journal of medicinal chemistry*, 56(3), pp.593–624.

Bakheet, T.M. & Doig, A.J., 2009. Properties and identification of human protein drug targets. *Bioinformatics*, 25(4), pp.451–7.

Balaji, S., Swaminathan, V. & Kannan, K., 2010. A Simple Algorithm to Optimize Maximum Independent Set. *Advanced Modeling and Optimization*, 12(1), pp.107–118.

Barton, G., 1996. Protein sequence alignment and database scanning. In M. J. E. Sternberg, ed. *Protein structure prediction - A practical approach*. Oxford University Press.

Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L.L., Studholme, D.J., Yeats, C. & Eddy, S.R., 2004. The Pfam protein families database. *Nucleic acids research*, 32(Database issue), pp.D138–41.

Ben-Hur, A., Ong, C.S., Sonnenburg, S., Schölkopf, B. & Rätsch, G., 2008. Support vector machines and kernels for computational biology. F. Lewitter, ed. *PLoS computational biology*, 4(10), p.e1000173.

Bergers, G. & Benjamin, L.E., 2003. Tumorigenesis and the angiogenic switch. *Nature reviews. Cancer*, 3(6), pp.401–10.

Bertino, J.R., Blume-Jensen, P. & Hunter, T., 2002. Signal Transduction Mechanisms Initiated by Receptor Tyrosine Kinases. In *Encyclopedia of Cancer*. pp. 213–234.

Bird, P.I., Trapani, J.A. & Villadangos, J.A., 2009. Endolysosomal proteases and their inhibitors in immunity. *Nature reviews. Immunology*, 9(12), pp.871–82.

Blum, A.L. & Langley, P., 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2), pp.245–271.

Blume-Jensen, P. & Hunter, T., 2001. Oncogenic kinase signalling. *Nature*, 411(6835), pp.355–65.

Booth, B. & Zemmel, R., 2004. Prospects for productivity. *Nature reviews. Drug discovery*, 3(5), pp.451–6.

Börjesson, S.I., Parkkari, T., Hammarström, S. & Elinder, F., 2010. Electrostatic tuning of cellular excitability. *Biophysical journal*, 98(3), pp.396–403.

Bouchachia, A., 2007. Learning with partly labeled data. *Neural Computing and Applications*, 16(3), pp.267–293.

Breiman, L., 1996. Bagging Predictors. *Machine Learning*, 24(2), pp.123–140.

Breiman, L., 2001. Random forests. *Machine learning*, 45(1), pp.5–32.

Bull, S.C., Muldoon, M.R. & Doig, A.J., 2013. Maximising the size of non-redundant protein datasets using graph theory. A. Tramontano, ed. *PloS one*, 8(2), p.e55484.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2), pp.121–167.

Butcher, S.P., 2003. Target discovery and validation in the post-genomic era. *Neurochemical research*, 28(2), pp.367–71.

Bylander, T., 2002. Estimating Generalization Error on Two-Class Datasets Using Out-of-Bag Estimates. *Machine Learning*, 48(1-3), pp.287–297.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K. & Madden, T.L., 2009. BLAST+: architecture and applications. *BMC bioinformatics*, 10(1), p.421.

Carmeliet, P. & Jain, R.K., 2000. Angiogenesis in cancer and other diseases. *Nature*, 407(6801), pp.249–57.

Castelli, V. & Cover, T.M., 1996. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6), pp.2102–2117.

Cavanaugh, A., Huang, Y. & Breitwieser, G.E., 2012. Behind the curtain: cellular mechanisms for allosteric modulation of calcium-sensing receptors. *British journal of pharmacology*, 165(6), pp.1670–7.

Chaffer, C.L. & Weinberg, R.A., 2011. A perspective on cancer cell metastasis. *Science*, 331(6024), pp.1559–64.

Chandra, N., 2009. Computational systems approach for drug target discovery. *Expert Opinion on Drug Discovery*, 4(12), pp.1221–1236.

Chang, C. & Werb, Z., 2001. The many faces of metalloproteases: cell growth, invasion, angiogenesis and metastasis. *Trends in cell biology*, 11(11), pp.S37–43.

Chatterjee, S. & Pal, J.K., 2009. Role of 5'- and 3'-untranslated regions of mRNAs in human diseases. *Biology of the cell*, 101(5), pp.251–62.

Chen, X. & Jeong, J.C., 2007. Enhanced recursive feature elimination. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*. IEEE, pp. 429–435.

Chen, X.P. & Du, G.H., 2007. Target validation: A door to drug discovery. *Drug discoveries & therapeutics*, 1(1), pp.23–9.

Cheng, A., Coleman, R., Smyth, K., Cao, Q., Soulard, P., Caffrey, D., Salzberg, A. & Huang, E., 2007. Structure-based maximal affinity model predicts small-molecule druggability. *Nature biotechnology*, 25(1), pp.71–5.

Christopoulos, A., 2002. Allosteric binding sites on cell-surface receptors: novel targets for drug discovery. *Nature reviews. Drug discovery*, 1(3), pp.198–210.

Clare, J., Tate, S., Nobbs, M. & Romanos, M., 2000. Voltage-gated sodium channels as therapeutic targets. *Drug discovery today*, 5(11), pp.506–520.

Cohen, P., 2002. Protein kinases--the major drug targets of the twenty-first century? *Nature reviews. Drug discovery*, 1(4), pp.309–15.

Coletta, A., Pinney, J.W., Solís, D.Y.W., Marsh, J., Pettifer, S.R. & Attwood, T.K., 2010. Low-complexity regions within protein sequences have position-dependent roles. *BMC systems biology*, 4(1), p.43.

Collier, R., 2009a. Drug development cost estimates hard to swallow. *Canadian Medical Association journal*, 180(3), pp.279–80.

Collier, R., 2009b. Rapidly rising clinical trial costs worry researchers. *Canadian Medical Association journal*, 180(3), pp.277–8.

Condorcet, J.A., 1785. Essay on the Application of Analysis to the Probability of Majority Decisions.

Congreve, M. & Marshall, F., 2010. The impact of GPCR structures on pharmacology and structure-based drug design. *British journal of pharmacology*, 159(5), pp.986–96.

Conn, P.J., Christopoulos, A. & Lindsley, C.W., 2009. Allosteric modulators of GPCRs: a novel approach for the treatment of CNS disorders. *Nature reviews. Drug discovery*, 8(1), pp.41–54.

Cord, M. & Cunningham, P. eds., 2008. *Machine Learning Techniques for Multimedia*, Berlin, Heidelberg: Springer Berlin Heidelberg.

Croft, D., Mundo, A.F., Haw, R., Milacic, M., Weiser, J., Wu, G., Caudy, M., Garapati, P., Gillespie, M., Kamdar, M.R., Jassal, B., Jupe, S., Matthews, L., May, B., Palatnik, S., Rothfels, K., Shamovsky, V., Song, H., Williams, M., Birney, E., Hermjakob, H., Stein, L. & D'Eustachio, P., 2014. The Reactome pathway knowledgebase. *Nucleic acids research*, 42(Database issue), pp.D472–7.

Dalziel, M., Crispin, M., Scanlan, C.N., Zitzmann, N. & Dwek, R.A., 2014. Emerging principles for the therapeutic exploitation of glycosylation. *Science*, 343(6166), p.1235681.

Davis, M.M., Butchart, A.T., Wheeler, J.R.C., Coleman, M.S., Singer, D.C. & Freed, G.L., 2011. Failure-to-success ratios, transition probabilities and phase lengths for prophylactic vaccines versus other pharmaceuticals in the development pipeline. *Vaccine*, 29(51), pp.9414–6.

Dickson, M. & Gagnon, J.P., 2004. Key factors in the rising cost of new drug discovery and development. *Nature reviews. Drug discovery*, 3(5), pp.417–29.

Dietterich, T.G., 1997. Machine-Learning Research: Four Current Directions. *AI Magazine*, 18(4), pp.97–136.

Dilly, S., Lamy, C., Marrion, N. V, Liégeois, J.-F. & Seutin, V., 2011. Ion-channel modulators: more diversity than previously thought. *Chembiochem : a European journal of chemical biology*, 12(12), pp.1808–12.

DiMasi, J. a, Feldman, L., Seckler, a & Wilson, A., 2010. Trends in risks associated with new drug development: success rates for investigational drugs. *Clinical pharmacology and therapeutics*, 87(3), pp.272–7.

DiMasi, J. a, Hansen, R.W. & Grabowski, H.G., 2003. The price of innovation: new estimates of drug development costs. *Journal of health economics*, 22(2), pp.151–85.

Dimitri, N., 2011. An assessment of R&D productivity in the pharmaceutical industry. *Trends in pharmacological sciences*, 32(12), pp.683–5.

Dorsam, R.T. & Gutkind, J.S., 2007. G-protein-coupled receptors and cancer. *Nature reviews. Cancer*, 7(2), pp.79–94.

Dosztányi, Z., Chen, J., Dunker, A.K., Simon, I. & Tompa, P., 2006. Disorder and sequence repeats in hub proteins and their implications for network evolution. *Journal of proteome research*, 5(11), pp.2985–95.

Drag, M. & Salvesen, G.S., 2010. Emerging principles in protease-based drug discovery. *Nature reviews. Drug discovery*, 9(9), pp.690–701.

Drews, J., 1996. Genomic sciences and the medicine of tomorrow. *Nature biotechnology*, 14(11), pp.1516–8.

Ekman, D., Light, S., Björklund, A.K. & Elofsson, A., 2006. What properties characterize the hub proteins of the protein-protein interaction network of Saccharomyces cerevisiae? *Genome biology*, 7(6), p.R45.

Engh, R.A. & Bossemeyer, D., 2002. Structural aspects of protein kinase control—role of conformational flexibility. *Pharmacology & Therapeutics*, 93(2), pp.99–111.

Eshelman, 1991. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In G. J. E. Rawlings, ed. *Foundations of Genetic Algorithms*. Morgan Kaufmann, pp. 265–283.

Eugene Tuv, A.B., 2009. Feature selection with ensembles, artificial variables, and redundancy elimination. *The Journal of Machine Learning Research*, 10, pp.1341–1366.

Evan, G.I. & Vousden, K.H., 2001. Proliferation, cell cycle and apoptosis in cancer. *Nature*, 411(6835), pp.342–8.

Faivre, S., Demetri, G., Sargent, W. & Raymond, E., 2007. Molecular basis for sunitinib efficacy and future clinical development. *Nature reviews. Drug discovery*, 6(9), pp.734–45.

Farady, C.J. & Craik, C.S., 2010. Mechanisms of macromolecular protease inhibitors. *Chembiochem*, 11(17), pp.2341–6.

Fauman, E.B., Hopkins, A.L. & Groom, C.R., 2003. Structural bioinformatics in drug discovery. *Methods of biochemical analysis*, 44, pp.477–97.

Fauman, E.B., Rai, B.K. & Huang, E.S., 2011. Structure-based druggability assessment--identifying suitable targets for small molecule therapeutics. *Current opinion in chemical biology*, 15(4), pp.463–8.

Fear, G., Komarnytsky, S. & Raskin, I., 2007. Protease inhibitors and their peptidomimetic derivatives as potential drugs. *Pharmacology & therapeutics*, 113(2), pp.354–68.

Ferguson, S.S., 2001. Evolving concepts in G protein-coupled receptor endocytosis: the role in receptor desensitization and signaling. *Pharmacological reviews*, 53(1), pp.1–24.

Ferrara, N. & Kerbel, R.S., 2005. Angiogenesis as a therapeutic target. *Nature*, 438(7070), pp.967–74.

Fidler, I.J., 2003. The pathogenesis of cancer metastasis: the “seed and soil” hypothesis revisited. *Nature reviews. Cancer*, 3(6), pp.453–8.

Filmore, D., 2004. It’s a GPCR world. *Modern Drug Discovery*, 7(11), pp.24 – 28.

Flicek, P., Amode, M.R., Barrell, D., Beal, K., Brent, S., Carvalho-Silva, D., Clapham, P., Coates, G., Fairley, S., Fitzgerald, S., Gil, L., Gordon, L., Hendrix, M., Hourlier, T., Johnson, N., Kähäri, A.K., Keefe, D., Keenan, S., Kinsella, R., Komorowska, M., Koscielny, G., Kulesha, E., Larsson, P., Longden, I., McLaren, W., Muffato, M., Overduin, B., Pignatelli, M., Pritchard, B., Riat, H.S., Ritchie, G.R.S., Ruffier, M., Schuster, M., Sobral, D., Tang, Y.A., Taylor, K., Trevanion, S., Vandrovcova, J., White, S., Wilson, M., Wilder, S.P., Aken, B.L., Birney, E., Cunningham, F., Dunham, I., Durbin, R., Fernández-Suarez, X.M., Harrow, J., Herrero, J., Hubbard, T.J.P., Parker, A., Proctor, G., Spudich, G., Vogel, J., Yates, A., Zadissa, A. & Searle, S.M.J., 2012. Ensembl 2012. *Nucleic acids research*, 40(Database issue), pp.D84–90.

Fredriksson, R., Lagerström, M.C., Lundin, L.-G. & Schiöth, H.B., 2003. The G-protein-coupled receptors in the human genome form five main families. Phylogenetic analysis, paralogon groups, and fingerprints. *Molecular pharmacology*, 63(6), pp.1256–72.

Futreal, P.A., Coin, L., Marshall, M., Down, T., Hubbard, T., Wooster, R., Rahman, N. & Stratton, M.R., 2004. A census of human cancer genes. *Nature reviews. Cancer*, 4(3), pp.177–83.

Gabashvili, I.S., Sokolowski, B.H.A., Morton, C.C. & Giersch, A.B.S., 2007. Ion channel gene expression in the inner ear. *Journal of the Association for Research in Otolaryngology*, 8(3), pp.305–28.

Gadsby, D.C., 2009. Ion channels versus ion pumps: the principal difference, in principle. *Nature reviews. Molecular cell biology*, 10(5), pp.344–52.

Garuti, L., Roberti, M. & Bottegoni, G., 2010. Non-ATP competitive protein kinase inhibitors. *Current medicinal chemistry*, 17(25), pp.2804–21.

Gashaw, I., Ellinghaus, P., Sommer, A. & Asadullah, K., 2011. What makes a good drug target? *Drug discovery today*, 16(23-24), pp.1037–43.

Gavrin, L.K. & Saiah, E., 2013. Approaches to discover non-ATP site kinase inhibitors. *MedChemComm*, 4(1), p.41.

Gerber, D.E., 2008. Targeted therapies: a new generation of cancer treatments. *American family physician*, 77(3), pp.311–9.

Ghahramani, Z., 2004. Unsupervised Learning. In O. Bousquet, U. Von Luxburg, & G. Ratsch, eds. *Advanced Lectures in Machine Learning. Lecture Notes in Computer Science 3176*. Berlin: Springer-Verlag, pp. 72–112.

Gordon, J., Pilkington, G.J., Parker, K. & Murray, S.A., 2008. Approaches to mitochondrially mediated cancer therapy. *Seminars in Cancer Biology*, 18(3), pp.226–235.

Gouaux, E. & Mackinnon, R., 2005. Principles of selective ion transport in channels and pumps. *Science*, 310(5753), pp.1461–5.

Grosso, A., Locatelli, M. & Pullan, W., 2007. Simple ingredients leading to very efficient heuristics for the maximum clique problem. *Journal of Heuristics*, 14(6), pp.587–612.

Guyon, I. & Elisseeff, A., 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3, pp.1157–1182.

Hahn, W.C. & Meyerson, M., 2001. Telomerase activation, cellular immortalization and cancer. *Annals of medicine*, 33(2), pp.123–9.

Hanahan, D. & Weinberg, R.A., 2011. Hallmarks of cancer: the next generation. *Cell*, 144(5), pp.646–74.

Hanahan, D. & Weinberg, R.A., 2000. The hallmarks of cancer. *Cell*, 100(1), pp.57–70.

Hanks, S.K. & Hunter, T., 1995. Protein kinases 6. The eukaryotic protein kinase superfamily: kinase (catalytic) domain structure and classification. *FASEB journal*, 9(8), pp.576–96.

Harley, C.B., 2008. Telomerase and cancer therapeutics. *Nature reviews. Cancer*, 8(3), pp.167–79.

Helenius, A. & Aebi, M., 2004. Roles of N-linked glycans in the endoplasmic reticulum. *Annual review of biochemistry*, 73, pp.1019–49.

Hille, B., 2001. *Ion Channels of Excitable Membranes, Third Edition*, Sinauer Associates, Inc.

Hille, B., 1978. Ionic channels in excitable membranes. Current problems and biophysical approaches. *Biophysical Journal*, 22(2), pp.283–294.

Hopkins, A. & Groom, C., 2002. The druggable genome. *Nature reviews. Drug discovery*, 1(9), pp.727–30.

Hua, J., Xiong, Z., Lowey, J., Suh, E. & Dougherty, E.R., 2005. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8), pp.1509–15.

Huang, C., Zhang, R., Chen, Z., Jiang, Y., Shang, Z., Sun, P., Zhang, X. & Li, X., 2010. Predict potential drug targets from the ion channel proteins based on SVM. *Journal of theoretical biology*, 262(4), pp.750–6.

Huang, D.W., Sherman, B.T. & Lempicki, R.A., 2009. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature protocols*, 4(1), pp.44–57.

Hughes, J.P., Rees, S., Kalindjian, S.B. & Philpott, K.L., 2011. Principles of early drug discovery. *British journal of pharmacology*, 162(6), pp.1239–49.

Imming, P., Sinning, C. & Meyer, A., 2006. Drugs, their target and the nature and number of drug targets. *Nature Reviews Drug Discovery*, 5(10), pp.821–835.

Janecek, A., Gansterer, W., Demel, M. & Ecker, G., 2008. On the Relationship Between Feature Selection and Classification Accuracy. In *Journal of Machine Learning Research: Workshop and Conference Proceedings 4*. pp. 90 – 105.

Jensen, L.J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., Bork, P. & von Mering, C., 2009. STRING 8--a global view on proteins and their functional interactions in 630 organisms. *Nucleic acids research*, 37(Database issue), pp.D412–6.

Jiang, Y., Lee, A., Chen, J., Cadene, M., Chait, B.T. & MacKinnon, R., 2002. Crystal structure and mechanism of a calcium-gated potassium channel. *Nature*, 417(6888), pp.515–22.

Kaitin, K.I. & DiMasi, J. a, 2011. Pharmaceutical innovation in the 21st century: new drug approvals in the first decade, 2000-2009. *Clinical pharmacology and therapeutics*, 89(2), pp.183–8.

Kaitin, K.I. & DiMasi, J.A., 2000. Measuring the Pace of New Drug Development in the User Fee ERA. *Drug Information Journal*, 34(3), pp.673–680.

Kanehisa, M., Goto, S., Sato, Y., Kawashima, M., Furumichi, M. & Tanabe, M., 2014. Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic acids research*, 42(Database issue), pp.D199–205.

Keramidas, A., Moorhouse, A.J., Schofield, P.R. & Barry, P.H., 2004. Ligand-gated ion channels: mechanisms underlying ion selectivity. *Progress in biophysics and molecular biology*, 86(2), pp.161–204.

Keyhani, S., Diener-West, M. & Powe, N., 2006. Are development times for pharmaceuticals increasing or decreasing? *Health affairs*, 25(2), pp.461–8.

Khan, A.R. & James, M.N., 1998. Molecular mechanisms for the conversion of zymogens to active proteolytic enzymes. *Protein science*, 7(4), pp.815–36.

Kinsella, R.J., Kähäri, A., Haider, S., Zamora, J., Proctor, G., Spudich, G., Almeida-King, J., Staines, D., Derwent, P., Kerhornou, A., Kersey, P. & Flicek, P., 2011. Ensembl BioMarts: a hub for data retrieval across taxonomic space. *Database*, 2011, p.bar030.

Klees, J.E. & Joines, R., 1997. Occupational health issues in the pharmaceutical research and development process. *Occupational medicine*, 12(1), pp.5–27.

Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., Pon, A., Banco, K., Mak, C., Neveu, V., Djoumbou, Y., Eisner, R., Guo, A.C. & Wishart, D.S., 2011. DrugBank 3.0: a comprehensive resource for "omics" research on drugs. *Nucleic acids research*, 39(Database issue), pp.D1035–41.

Kohavi, R. & John, G.H., 1997. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1-2), pp.273–324.

Kola, I., 2008. The state of innovation in drug development. *Clinical pharmacology and therapeutics*, 83(2), pp.227–30.

Kola, I. & Landis, J., 2004. Can the pharmaceutical industry reduce attrition rates? *Nature reviews. Drug discovery*, 3(8), pp.711–5.

Kornev, A.P., Haste, N.M., Taylor, S.S. & Eyck, L.F. Ten, 2006. Surface comparison of active and inactive protein kinases identifies a conserved activation mechanism. *Proceedings of the National Academy of Sciences of the United States of America*, 103(47), pp.17783–8.

Kotsiantis, S.B., 2007. Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31, pp.249–268.

Kristiansen, K., 2004. Molecular mechanisms of ligand binding, signaling, and regulation within the superfamily of G-protein-coupled receptors: molecular modeling and mutagenesis approaches to receptor structure and function. *Pharmacology & Therapeutics*, 103(1), pp.21–80.

Krupnick, J.G. & Benovic, J.L., 1998. The role of receptor kinases and arrestins in G protein-coupled receptor regulation. *Annual review of pharmacology and toxicology*, 38, pp.289–319.

Kyte, J. & Doolittle, R.F., 1982. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1), pp.105–132.

Lane, J.R., Abdul-Ridha, A. & Canals, M., 2013. Regulation of G protein-coupled receptors by allosteric ligands. *ACS chemical neuroscience*, 4(4), pp.527–34.

Lang, P., Yeow, K., Nichols, A. & Scheer, A., 2006. Cellular imaging in drug discovery. *Nature reviews. Drug discovery*, 5(4), pp.343–56.

Leber, M.F. & Efferth, T., 2009. Molecular principles of cancer invasion and metastasis. *International journal of oncology*, 34(4), pp.881–95.

Lefkowitz, R.J., 1998. G protein-coupled receptors. III. New roles for receptor kinases and beta-arrestins in receptor signaling and desensitization. *The Journal of biological chemistry*, 273(30), pp.18677–80.

Leung, D., Abbenante, G. & Fairlie, D.P., 2000. Protease inhibitors: current status and future prospects. *Journal of medicinal chemistry*, 43(3), pp.305–41.

Levinson, N.M., Kuchment, O., Shen, K., Young, M.A., Koldobskiy, M., Karplus, M., Cole, P.A. & Kuriyan, J., 2006. A Src-like inactive conformation in the abl tyrosine kinase domain. *PLoS biology*, 4(5), p.e144.

Lew, J., 2003. MAP kinases and CDKs: kinetic basis for catalytic activation. *Biochemistry*, 42(4), pp.849–56.

Li, J. & Lu, X., 2013. The emerging roles of 3' untranslated regions in cancer. *Cancer letters*, 337(1), pp.22–5.

Li, Q. & Lai, L., 2007. Prediction of potential drug targets based on simple sequence properties. *BMC bioinformatics*, 8, p.353.

Lindsay, M.A., 2003. Target discovery. *Nature Reviews Drug Discovery*, 2(10), pp.831–838.

Lipsky, M.S. & Sharp, L.K., 2001. From idea to market: the drug approval process. *The Journal of the American Board of Family Practice*, 14(5), pp.362–7.

Liu, P., Zeng, Z., Qian, Z., Feng, K. & Cai, Y., 2009. A Graph Theoretic Algorithm for Removing Redundant Protein Sequences. In *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*. IEEE, pp. 1–3.

Liu, Y. & Gray, N.S., 2006. Rational design of inhibitors that bind to inactive kinase conformations. *Nature chemical biology*, 2(7), pp.358–64.

Lowe, S.W., 2000. Apoptosis in cancer. *Carcinogenesis*, 21(3), pp.485–495.

Lowe, S.W., Cepero, E. & Evan, G., 2004. Intrinsic tumour suppression. *Nature*, 432(7015), pp.307–15.

Luo, J., Solimini, N.L. & Elledge, S.J., 2009. Principles of cancer therapy: oncogene and non-oncogene addiction. *Cell*, 136(5), pp.823–37.

Luttrell, L.M. & Lefkowitz, R.J., 2002. The role of beta-arrestins in the termination and transduction of G-protein-coupled receptor signals. *Journal of cell science*, 115(3), pp.455–65.

Magalhaes, A.C., Dunn, H. & Ferguson, S.S.G., 2012. Regulation of GPCR activity, trafficking and localization by GPCR-interacting proteins. *British journal of pharmacology*, 165(6), pp.1717–36.

Mandal, S., Moudgil, M. & Mandal, S.K., 2009. Rational drug design. *European journal of pharmacology*, 625(1-3), pp.90–100.

Manning, G., Whyte, D.B., Martinez, R., Hunter, T. & Sudarsanam, S., 2002. The protein kinase complement of the human genome. *Science*, 298(5600), pp.1912–34.

May, L.T., Leach, K., Sexton, P.M. & Christopoulos, A., 2007. Allosteric modulation of G protein-coupled receptors. *Annual review of pharmacology and toxicology*, 47, pp.1–51.

McGraw, K.O. & Wong, S.P., 1992. A common language effect size statistic. *Psychological Bulletin*, 111(2), pp.361–365.

McNulty, M.M., Edgerton, G.B., Shah, R.D., Hanck, D.A., Fozzard, H.A. & Lipkind, G.M., 2007. Charge at the lidocaine binding site residue Phe-1759 affects permeation in human cardiac voltage-gated sodium channels. *The Journal of physiology*, 581(2), pp.741–55.

Mignone, F., Gissi, C., Liuni, S. & Pesole, G., 2002. Untranslated regions of mRNAs. *Genome biology*, 3(3), p.REVIEWS0004.

Millar, R.P. & Newton, C.L., 2010. The year in G protein-coupled receptor research. *Molecular endocrinology*, 24(1), pp.261–74.

Morgan, P., Van Der Graaf, P.H., Arrowsmith, J., Feltner, D.E., Drummond, K.S., Wegner, C.D. & Street, S.D., 2012. Can the flow of medicines be improved? Fundamental pharmacokinetic and pharmacological principles toward improving Phase II survival. *Drug discovery today*, 17(9-10), pp.419–24.

Munos, B., 2009. Lessons from 60 years of pharmaceutical innovation. *Nature reviews. Drug discovery*, 8(12), pp.959–68.

Nayal, M. & Honig, B., 2006. On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. *Proteins*, 63(4), pp.892–906.

Nazarenko, I., Hede, S.-M., He, X., Hedrén, A., Thompson, J., Lindström, M.S. & Nistér, M., 2012. PDGF and PDGF receptors in glioma. *Upsala journal of medical sciences*, 117(2), pp.99–112.

Needleman, S.B. & Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3), pp.443–53.

Negus, S.S., 2006. Some implications of receptor theory for in vivo assessment of agonists, antagonists and inverse agonists. *Biochemical pharmacology*, 71(12), pp.1663–70.

Nilsson, R., Peña, J.M., Björkegren, J. & Tegnér, J., 2007. Consistent Feature Selection for Pattern Recognition in Polynomial Time. *The Journal of Machine Learning Research*, 8, pp.589–612.

Niskanen, S. & Östergård, P.R.J., 2003. Cliquer User's Guide, Version 1.0. *Communications Laboratory, Helsinki University of Technology*.

Noble, M.E.M., Endicott, J.A. & Johnson, L.N., 2004. Protein kinase inhibitors: insights into drug design from structure. *Science*, 303(5665), pp.1800–5.

Nolen, B., Taylor, S. & Ghosh, G., 2004. Regulation of protein kinases; controlling activity through activation segment conformation. *Molecular cell*, 15(5), pp.661–75.

Nussenbaum, F. & Herman, I.M., 2010. Tumor angiogenesis: insights and innovations. *Journal of oncology*, 2010, p.24.

Östergård, P.R.J., 2002. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1-3), pp.197–207.

Overington, J.P., Al-Lazikani, B. & Hopkins, A.L., 2006. How many drug targets are there? *Nature reviews. Drug discovery*, 5(12), pp.993–6.

Paul, S.M., Mytelka, D.S., Dunwiddie, C.T., Persinger, C.C., Munos, B.H., Lindborg, S.R. & Schacht, A.L., 2010. How to improve R&D

productivity: the pharmaceutical industry's grand challenge. *Nature reviews. Drug discovery*, 9(3), pp.203–14.

Peck, C.C., Barr, W.H., Benet, L.Z., Collins, J., Desjardins, R.E., Furst, D.E., Harter, J.G., Levy, G., Ludden, T., Rodman, J.H., Sanathanan, L., Schentag, J.J., Shah, V.P., Sheiner, L.B., Skelly, J.P., Stanski, D.R., Temple, R.J., Viswanathan, C.T., Weissinger, J. & Yacobi, A., 1992. Opportunities for integration of pharmacokinetics, pharmacodynamics, and toxicokinetics in rational drug development. *Journal of Pharmaceutical Sciences*, 81(6), pp.605–610.

Peck, C.C., Rubin, D.B. & Sheiner, L.B., 2003. Hypothesis: a single clinical trial plus causal evidence of effectiveness is sufficient for drug approval. *Clinical pharmacology and therapeutics*, 73(6), pp.481–90.

Perozo, E., 2002. New structural perspectives on K(+) channel gating. *Structure*, 10(8), pp.1027–9.

Petersen, B., Petersen, T.N., Andersen, P., Nielsen, M. & Lundegaard, C., 2009. A generic method for assignment of reliability scores applied to solvent accessibility predictions. *BMC structural biology*, 9(1), p.51.

Pierce, K.L., Premont, R.T. & Lefkowitz, R.J., 2002. Seven-transmembrane receptors. *Nature reviews. Molecular cell biology*, 3(9), pp.639–50.

Pinna, L.A. & Ruzzene, M., 1996. How do protein kinases recognize their substrates? *Molecular Cell Research*, 1314(3), pp.191–225.

Pitcher, J.A., Freedman, N.J. & Lefkowitz, R.J., 1998. G protein-coupled receptor kinases. *Annual review of biochemistry*, 67, pp.653–92.

Pudil, P., Novovičová, J. & Kittler, J., 1994. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11), pp.1119–1125.

Puente, X.S. & López-Otín, C., 2004. A genomic analysis of rat proteases and protease inhibitors. *Genome research*, 14(4), pp.609–22.

Puente, X.S., Sánchez, L.M., Overall, C.M. & López-Otín, C., 2003. Human and mouse proteases: a comparative genomic approach. *Nature reviews. Genetics*, 4(7), pp.544–58.

Rask-Andersen, M., Almén, M.S. & Schiöth, H.B., 2011. Trends in the exploitation of novel drug targets. *Nature reviews. Drug discovery*, 10(8), pp.579–90.

Rawlins, M.D., 2004. Cutting the cost of drug development? *Nature reviews. Drug discovery*, 3(4), pp.360–4.

Regard, J.B., Sato, I.T. & Coughlin, S.R., 2008. Anatomical profiling of G protein-coupled receptor expression. *Cell*, 135(3), pp.561–71.

Reményi, A., Good, M.C. & Lim, W.A., 2006. Docking interactions in protein kinase and phosphatase networks. *Current Opinion in Structural Biology*, 16(6), pp.676–85.

Rogers, S., Wells, R. & Rechsteiner, M., 1986. Amino acid sequences common to rapidly degraded proteins: the PEST hypothesis. *Science (New York, N.Y.)*, 234(4774), pp.364–8.

Romero, P., Obradovic, Z. & Dunker, A.K., 2004. Natively disordered proteins: functions and predictions. *Applied bioinformatics*, 3(2-3), pp.105–13.

Russ, A.P. & Lampel, S., 2005. The druggable genome: an update. *Drug discovery today*, 10(23-24), pp.1607–10.

Schechter, I. & Berger, A., 1967. On the size of the active site in proteases. I. Papain. *Biochemical and biophysical research communications*, 27(2), pp.157–62.

Schmid, E.F. & Smith, D.A., 2005. Is declining innovation in the pharmaceutical industry a myth? , 10(15), pp.1031–1039.

Schramm, V.L., 2011. Enzymatic transition states, transition-state analogs, dynamics, thermodynamics, and lifetimes. *Annual review of biochemistry*, 80, pp.703–32.

Shawver, L.K., Slamon, D. & Ullrich, A., 2002. Smart drugs: tyrosine kinase inhibitors in cancer therapy. *Cancer cell*, 1(2), pp.117–23.

Shay, J.W., Pereira-Smith, O.M. & Wright, W.E., 1991. A role for both RB and p53 in the regulation of human cellular senescence. *Experimental cell research*, 196(1), pp.33–9.

Shen, A., 2010. Allosteric regulation of protease activity by small molecules. *Molecular bioSystems*, 6(8), pp.1431–43.

Shugar, D., Besant, P.G. & Attwood, P. V., 2005. Mammalian histidine kinases. *Proteins and Proteomics*, 1754(1), pp.281–290.

Smyth, T.P., 2004. Substrate variants versus transition state analogues as noncovalent reversible enzyme inhibitors. *Bioorganic & medicinal chemistry*, 12(15), pp.4081–8.

Steinmetz, K.L. & Spack, E.G., 2009. The basics of preclinical drug development for neurodegenerative disease indications. *BMC Neurology*, 9(Suppl 1), p.S2.

Strobl, C., Malley, J. & Tutz, G., 2009. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4), pp.323–48.

Sun, Y., Olson, R., Horning, M., Armstrong, N., Mayer, M. & Gouaux, E., 2002. Mechanism of glutamate receptor desensitization. *Nature*, 417(6886), pp.245–53.

Suvarna, V., 2010. Phase IV of Drug Development. *Perspectives in clinical research*, 1(2), pp.57–60.

Szymkowski, D.E. & Avenue, W.L., 2003. Target validation joins the pharma fold. *TARGETS*, 2(1), pp.8–9.

Thain, D., Tannenbaum, T. & Livny, M., 2005. Distributed computing in practice: the Condor experience. *Concurrency and Computation: Practice and Experience*, 17(2-4), pp.323–356.

The UniProt Consortium, 2010. The Universal Protein Resource (UniProt) in 2010. *Nucleic acids research*, 38(Database issue), pp.D142–8.

Turk, B., 2006. Targeting proteases: successes, failures and future prospects. *Nature reviews. Drug discovery*, 5(9), pp.785–99.

Ubersax, J.A. & Ferrell, J.E., 2007. Mechanisms of specificity in protein phosphorylation. *Nature reviews. Molecular cell biology*, 8(7), pp.530–41.

Wang, G. & Dunbrack, R.L., 2003. PISCES: a protein sequence culling server. *Bioinformatics*, 19(12), pp.1589–1591.

Wheeler, D.L., 2003. Database resources of the National Center for Biotechnology. *Nucleic Acids Research*, 31(1), pp.28–33.

Wishart, D.S., Knox, C., Guo, A.C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B. & Hassanali, M., 2008. DrugBank: a knowledgebase for

drugs, drug actions and drug targets. *Nucleic acids research*, 36(Database issue), pp.D901–6.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J. & Steinberg, D., 2007. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1), pp.1–37.

Xu, H., Collins, J.F., Bai, L., Kiela, P.R. & Ghishan, F.K., 2001. Regulation of the human sodium-phosphate cotransporter NaP(i)-IIb gene promoter by epidermal growth factor. *American journal of physiology. Cell physiology*, 280(3), pp.C628–36.

Xu, H., Xu, H., Lin, M., Wang, W., Li, Z., Huang, J., Chen, Y. & Chen, X., 2007. Learning the drug target-likeness of a protein. *Proteomics*, 7(23), pp.4255–63.

Yang, Y., Adelstein, S.J. & Kassis, A.I., 2009. Target discovery from data mining approaches. *Drug discovery today*, 14(3-4), pp.147–54.

Yu, L. & Liu, H., 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In T. Fawcett & N. Mishra, eds. *Proceedings of the Twentieth International Conference on Machine Learning*. Washington DC: Amer Assn for Artificial, p. 856.

Zheng, C.J., Han, L.Y., Yap, C.W., Ji, Z.L., Cao, Z.W. & Chen, Y.Z., 2006. Therapeutic targets: progress of their exploration and investigation of their characteristics. *Pharmacological reviews*, 58(2), p.259.

Zhu, F., Han, B., Kumar, P., Liu, X., Ma, X., Wei, X., Huang, L., Guo, Y., Han, L., Zheng, C. & Chen, Y., 2010. Update of TTD: Therapeutic Target Database. *Nucleic acids research*, 38(Database issue), pp.D787–91.

Appendix A: Predicted Target Proteins

I Cancer Proteins

Accession	Positive Similarity	Accession	Positive Similarity
Q13635	1.00	P60568	0.87
Q9HC73	0.99	P31994	0.85
O95436	0.99	O15393	0.83
P08922	0.99	Q93063	0.83
Q9UKU0	0.99	P20151	0.82
Q16288	0.99	O60674	0.82
Q86VZ1	0.99	P05091	0.82
P25106	0.98	Q99643	0.82
Q96JT2	0.98	P24864	0.80
Q9UM73	0.97	Q16549	0.79
Q9BQ51	0.96	Q8WVQ1	0.79
P53675	0.95	Q06136	0.78
Q6UXM1	0.95	P10415	0.78
Q9NZQ7	0.95	P02786	0.75
P40259	0.94	Q9UKJ5	0.74
P42702	0.94	P43405	0.74
P48735	0.93	Q9Y5W5	0.74
O95573	0.93	P18074	0.74
P51654	0.92	Q8NG68	0.73
P42336	0.91	Q16394	0.69
Q9HBE5	0.91	Q9NPI8	0.67
P36894	0.90	P55287	0.66
P11912	0.90	O14521	0.66
Q9Y693	0.90	Q9ULV8	0.64
Q008S8	0.89	P12830	0.63
O75874	0.89	Q9HB96	0.59
Q9H2T7	0.88	Q01860	0.59
Q8NFG4	0.88	Q92989	0.58
Q96PJ5	0.88	Q99983	0.58
P49589	0.88	O15360	0.57
Q00597	0.87	Q02223	0.52
P14222	0.87	P35125	0.51

II GPCRs

Accession	Positive Similarity	Accession	Positive Similarity
O15303	0.93	P21730	0.72
O43613	0.92	Q9UHM6	0.70
O60883	0.91	Q9BPV8	0.70
Q99705	0.89	Q15391	0.69
Q9NYM4	0.89	Q9Y5X5	0.69
Q8IZP9	0.88	Q9UBY5	0.68
O43614	0.87	P51684	0.68
P49190	0.86	Q99680	0.68
O95838	0.85	P32241	0.65
Q86Y34	0.85	P34998	0.65
P41146	0.82	Q16538	0.64
P30989	0.81	Q15761	0.64
Q15760	0.80	O15354	0.64
Q9BZJ8	0.80	Q9NPB9	0.63
Q96P65	0.79	P51686	0.63
Q9NSD7	0.79	Q9H461	0.61
Q8IZ08	0.79	Q99677	0.60
Q8IZF7	0.77	Q8IZF4	0.60
P32247	0.77	Q9BZJ7	0.60
P48546	0.76	Q9UP38	0.58
P25090	0.76	Q96K78	0.58
P29371	0.75	Q7RTX1	0.57
P41586	0.75	Q9H1Y3	0.57
P41587	0.74	P21453	0.53
Q9GZQ4	0.72	O95977	0.53

III Ion Channels

Accession	Positive Similarity
Q7Z3S7	0.96
Q15878	0.96
Q03721	0.84
Q7Z442	0.84
Q401N2	0.83
Q99712	0.81
Q01118	0.81
A5X5Y0	0.79
O95264	0.76
Q9NPI9	0.76
Q9NTG1	0.74
Q9UHC3	0.74
O00168	0.70
Q9UQD0	0.69
Q7Z443	0.69
Q8TD43	0.65
P63252	0.63
O60928	0.63
Q8IZF0	0.61
Q9Y210	0.52
Q70Z44	0.50

IV Kinases

Accession	Positive Similarity	Accession	Positive Similarity	Accession	Positive Similarity
Q16288	0.99	P36897	0.79	Q8N4C8	0.66
P09769	0.98	P54756	0.79	P78368	0.65
P41743	0.97	P51812	0.79	Q05513	0.65
Q06187	0.97	Q16539	0.79	P05129	0.65
Q05397	0.96	P08237	0.78	Q6XUX3	0.65
Q15375	0.95	P42356	0.78	Q8TAS1	0.64
P41240	0.95	Q9UF33	0.78	Q00535	0.64
P54764	0.94	P19367	0.76	P51451	0.62
Q9Y4K4	0.94	O00329	0.76	Q01974	0.62
P21860	0.93	Q15139	0.76	Q13418	0.62
P54760	0.92	P08922	0.76	Q13163	0.62
Q15303	0.92	P21709	0.76	P34925	0.62
P29320	0.91	P35590	0.75	O43353	0.62
P42680	0.91	P17858	0.74	Q9NVE7	0.62
P54753	0.91	P42681	0.74	Q01973	0.61
P29323	0.91	Q13705	0.73	P15531	0.60
Q8TD19	0.90	P22392	0.73	Q9NY57	0.60
P43405	0.89	O43252	0.73	Q9HC98	0.59
Q06418	0.88	Q9BZL6	0.73	Q9UEW8	0.59
Q15746	0.88	Q16644	0.73	P46734	0.59
P50750	0.88	P68400	0.73	Q15418	0.59
Q12866	0.87	P43403	0.72	Q16877	0.59
Q13308	0.87	Q00534	0.71	P53350	0.58
P30530	0.87	Q9UHD2	0.70	Q9HBH9	0.58
P29597	0.86	P06493	0.70	P45984	0.58
P45983	0.86	Q9H479	0.69	Q9BUB5	0.57
O75716	0.84	P27037	0.69	O14976	0.54
P14616	0.84	P51813	0.69	O14936	0.54
Q14680	0.83	Q96GD4	0.68	P34947	0.53
P24723	0.83	Q8IVH8	0.67	Q8TDX7	0.53
Q01813	0.83	P11802	0.67	Q8WZ42	0.53
O95340	0.82	Q13546	0.67	P52429	0.53
Q13557	0.82	P31751	0.67	Q38SD2	0.53
P00558	0.81	Q04759	0.67	Q16875	0.52
P53671	0.80	Q9UBF8	0.67	P78527	0.51
P36894	0.80	P50613	0.66		

V Proteases

Accession	Positive Similarity	Accession	Positive Similarity
Q9UKU6	0.93	P48052	0.65
P15169	0.90	Q9Y6M0	0.65
Q12884	0.84	Q9NVE5	0.65
P25787	0.81	Q8N6M6	0.65
Q96T52	0.81	P52948	0.64
Q9UQQ1	0.80	P43234	0.64
Q9UI42	0.79	P25788	0.63
Q9HBA9	0.77	Q9UHL4	0.63
Q9NY33	0.77	O14672	0.62
P17655	0.76	P49720	0.61
P42892	0.75	Q8IUX7	0.61
Q53GS9	0.75	Q9Y6W3	0.61
P06681	0.75	Q96FV2	0.60
P28072	0.75	P29122	0.57
P12955	0.74	Q8NBH2	0.57
O95084	0.73	P55786	0.57
P61009	0.73	Q14623	0.57
Q8TB40	0.69	P0DJD8	0.57
Q86TI2	0.69	Q9Y3Q0	0.56
Q9H3G5	0.68	Q9UJW2	0.56
P05156	0.68	Q9Y5B9	0.55
P00736	0.68	O96009	0.55
Q9Y5Z0	0.67	Q7Z2K6	0.54
O43895	0.67	P55210	0.53
Q2TV78	0.67	Q53RT3	0.50
Q99436	0.66	P07384	0.50
Q9UQ90	0.65		