Evaluating Symbolic AI as a Tool to Understand Cell Signalling

George Aristidis Elder

School of Biological and Chemical Sciences
Queen Mary - University of London

Thesis submitted in partial fulfilment
of the requirements of the Degree of
Doctor of Philosophy

December 2022

This page is intentionally left blank

Statement of Originality

I, George Aristidis Elder, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature: Date: 23/12/2022

digitally signed (George A. Elder)

Details of collaboration and publications:
Collaborators are acknowledged at the end of each chapter.

Abstract

The diverse and highly complex nature of modern phosphoproteomics research produces a high volume of data. Chemical phosphoproteomics especially, is amenable to a variety of analytical approaches. In this thesis we evaluate novel Symbolic AI based algorithms as potential tools in the analysis of cell signalling. Initially we developed a first order deductive, logic-based model. This allowed us to identify previously unreported inhibitor-kinase relationships which could offer novel therapeutic targets for further investigation. Following this we made use of the probabilistic reasoning of ProbLog to augment the aforementioned Prolog based model with an intuitively calculated degree of belief. This allowed us to rank previous associations while also further increasing our confidence in already established predictions. Finally we applied our methodology to a *Saccharomyces cerevisiae* gene perturbation, phosphoproteomics dataset. In this context we were able to confirm the majority of ground truths, i.e. gene deletions as having taken place as intended. For the remaining deletions, again using a purely symbolic based approach we were able to provide predictions on the rewiring of kinase based signalling networks following kinase encoding gene deletions. The explainable, human readable and white-box nature of this approach were highlighted, however its brittleness due to missing, inconsistent or conflicting background knowledge was also examined.

Acknowledgements

I want to express my sincere gratitude to all supervisors and collaborators involved in this project. My greatest thanks goes to Professor Conrad Bessant, who has been a mentor, astute supervisor and was always available, guiding and supporting me through every stage of my PhD endeavour. I wish to also express my deepest gratitude to Professor Pedro R. Cutillas, who has been instrumental, by shaping and supervising the project in the biological direction.

My fellow PhD students have been a pleasure to work with. Sincere appreciation to past lab members Esteban Gea, Naz Nawaz and Onur Ozcan for providing a thriving research environment at the beginning of this journey. Current lab members to which I also extend my gratitude include Nikhil Branson, Antara Labiba, Lewis Palmer and Hajar Saihi. A particular special mention to Magdalena Huebner (and Onur) for forming the core Symbolic AI team and who where there to help make the illogical logical. Additionally, all the fantastic people that I had the pleasure of being around and spending time with in DERI, especially Tiejun Wei and Abbas Khan.

Additionally, I express my sincere gratitude to the members of the Cutillas lab for their valuable and biologically relevant comments and feedback on my research during our lab meeting presentations. Their contributions have been instrumental in shaping this work. Moreover, I am deeply grateful for the generous funding provided by the MRC, without which this research would not have been possible.

Last but not least, I would like to thank my family and friends for all their continuous support and grounding, absolutely necessary in order for me to complete this academic adventure.

Contents

List of Figures

List of Tables

This page is intentionally left blank

Chapter 1

Introduction

1.1 Cell Signalling via phosphorylation

Phosphorylation, a fundamental biochemical process, plays a crucial role in various cellular functions and is intrinsically linked to cell signalling. This process involves the addition of a phosphate group to a protein molecule, catalysed by enzymes known as kinases. The reversible nature of phosphorylation enables these modified proteins to switch between active and inactive states, regulating vital cellular processes such as growth, differentiation, and apoptosis (Eid et al. 2017). Kinases, which form the kinome comprising over 500 known human members, represent approximately 2% of the human genome and exert significant control over cellular pathways (Manning et al. 2002).

The structure and activity of kinases make them attractive targets for understanding the mechanistic basis of diseases, including cancer, inflammatory disorders, and immune-based pathologies (Duong-Ly & Peterson 2013). Allosteric and antagonistic inhibitors can relatively easily target these enzymes, offering potential therapeutic avenues for treating such conditions (described further in ensuing section). To comprehend the complexity of these signalling networks and their dysregulation in disease, models of cell signalling pathways driven mainly by protein and lipid kinases have been developed. These models are invaluable tools in unravelling cell processes and understanding how they regulate various cell phenotypes, particularly in the context of cancer where dysregulation is prominent (Pawson & Scott n.d.).

1.1.1 Phosphorylation Mechanism and Impact

Phosphorylation-mediated responses extend to hormonal, developmental, and metabolic regulation, making this process critical for cellular functioning. The cascade of phosphorylation events involves a series of sequential kinase-mediated modifications, where the activation or inhibition of one kinase leads to the subsequent phosphorylation of another protein and so forth (Pawson & Scott n.d.). This interconnected signalling cascade, as depicted in Figure 1.1, underpins numerous cellular processes, although explaining its mechanistic intricacies remains challenging.

With the advent of modern mass spectrometry techniques, our understanding of protein phosphorylation has advanced significantly. These techniques allow for the rapid detection and quantification of thousands of phosphosites within a matter of days, revolutionising the field of phosphoproteomics (Chan et al. 2016). Complex chemistries for target (phosphosite) enrichment, diverse sample ionisation, and fragmentation methods have contributed to achieving unparalleled accuracy and efficiency, and these capabilities will only improve with time (Riley & Coon 2016).

However, while the ability to collect high-throughput and highly accurate phosphorylation-based datasets has its merits, it presents the challenge of interpreting vast amounts of data points (Rudolph et al. 2016). As such, an in-depth understanding of phosphorylation and its role in cell signalling is indispensable for making sense of these datasets and deciphering their implications in various biological contexts.

1.1.2 Kinase Structure

Kinases are a diverse group of enzymes that share a common catalytic domain known as the kinase domain. This domain is responsible for transferring phosphate groups from ATP (adenosine triphosphate) to specific amino acid residues, predominantly serine, threonine, or tyrosine, on target proteins. The three-dimensional structure of the kinase domain consists of a small N-terminal lobe and a larger C-terminal lobe, with the active site located in the cleft between them. The ATP-binding site resides in the N-terminal lobe, while the substrate-binding site lies in the C-terminal lobe, allowing for the recognition and phosphorylation of specific protein substrates (McClendon et al. 2014).

Kinases can exhibit considerable structural variability outside the conserved kinase domain, which contributes to their diverse functions and substrate specificities. Some kinases have additional regulatory domains that control their activity, localisation, and interactions with other proteins, further adding to the complexity of their regulation.

1.1.3 Regulation of Kinase Activity

The precise and context-dependent responses of kinases are achieved through stringent regulation. This regulation encompasses a range of mechanisms that fall under the umbrella of post translational modifications. Kinases undergo various post-translational modifications that influence their activity. Notably, phosphorylation, ubiquitination (Lu & Hunter 2009), acetylation (Lee et al. 2018), and other modifications can either activate or inhibit kinase function. These modifications can also establish docking sites for other proteins, enabling the formation of crucial multi-protein complexes within signalling cascades.

Phosphorylation of kinases (by other kinases), further exerts control over their functions by disrupting surfaces for protein-ligand interactions without necessitating conformational changes. Conformational changes prompted by phosphorylation significantly rely on the structural environment of the phosphorylated protein. When phosphorylated, the phosphate group affects the kinase's activity by fostering a network of hydrogen bonds among specific neighbouring amino acid residues. This intricate form is dictated by the three-dimensional structure of the phosphorylated kinase and is, therefore, distinctive for each one individually (Cheng et al. 2011).

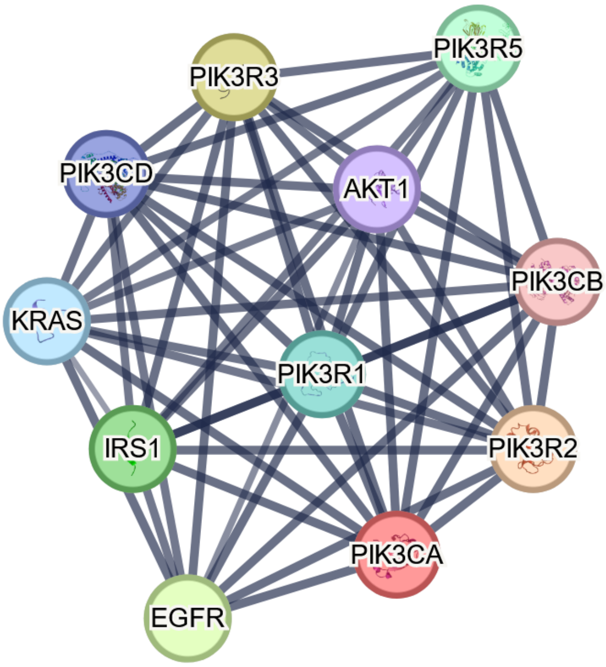


Figure 1.1: The figure depicts a protein kinase-based interaction network, specifically focusing on the PIK3CA STRING-derived network. In this representation, network nodes represent proteins, including kinases, while edges signify protein-protein interactions, encompassing both inferred and physical interactions. To ensure robustness, only interactions with an edge confidence score above 0.7 have been included. Edge confidence quantifies the reliability and strength of protein-protein interactions or associations between two proteins. This illustrative visualisation aims to capture the intricate and interconnected nature of protein-protein and protein-kinase interaction networks, which form complex pathways involving cascades of reactions. Understanding these networks is crucial for unravelling the mechanisms and complexities of cellular processes.(Szklarczyk et al. 2015)

1.1.4 Kinase inhibitors

Kinase signalling is an essential part of a number of cellular activities. It involves the addition of a phosphate group onto a protein (or other kinase) target at a specific location on its structure. This type of signalling as well as dephosphorylation, carried out by phosphatases, is often dysregulated in disease, and specifically for cancer it is considered a hallmark of the disease. Detailed knowledge of the mechanisms underpinning kinase signalling is therefore crucial in our understanding of diseases and how to cure and/or prevent them. Attempts at chemically targeting and resetting or altering these types of signalling, first *in vitro* and then in clinical trials have resulted in the development of 60 small molecule inhibitors (Turdo et al. 2021).

Approaches to study the above signalling usually involve the targeted disruption of kinase activity. This can be achieved through knockdowns (or silencing) of genes which translate into key kinases and/or chemically through the use of selective and non-selective inhibitors. Both of these types of approaches are employed to disambiguate the involvement of kinases in key cell processes, namely growth, cell-cycle progression, apoptosis and metabolism; while in the case of cancerous cells, additionally, proliferation, drug resistance (acquired and innate) and invasiveness. Through this selective silencing or inhibition, insights into kinase importance, their relations, their druggability as well as their relations with other proteins (or kinases) can be gleaned. Kinase druggability refers to the extent to which a kinase target can be effectively modulated or inhibited by small molecule drugs (Yueh et al. 2019).

From a drug discovery, pharmaceutical and monetary perspective, the development of novel kinase inhibitor molecules is of paramount importance for both pharma and medical research industries (Hill et al. 2015). At present, 60 small molecule inhibitors are on the FDA approved list (Turdo et al. 2021) with at least a further 250 in various stages of clinical trials (Klaeger et al. 2017). They are classified into six types, I-VI, depending on the structure of the resulting complex after they have bound on specific regions of their target kinase(s). In their majority (wrt to their approved status) they tend to affect the active site of a kinase, which is located between the N- and C-terminal domains and whose function is to bind ATP (Yueh et al. 2019). Here they exact a conformational change on the affected kinase(s) thus inhibiting its activity to varying degrees.

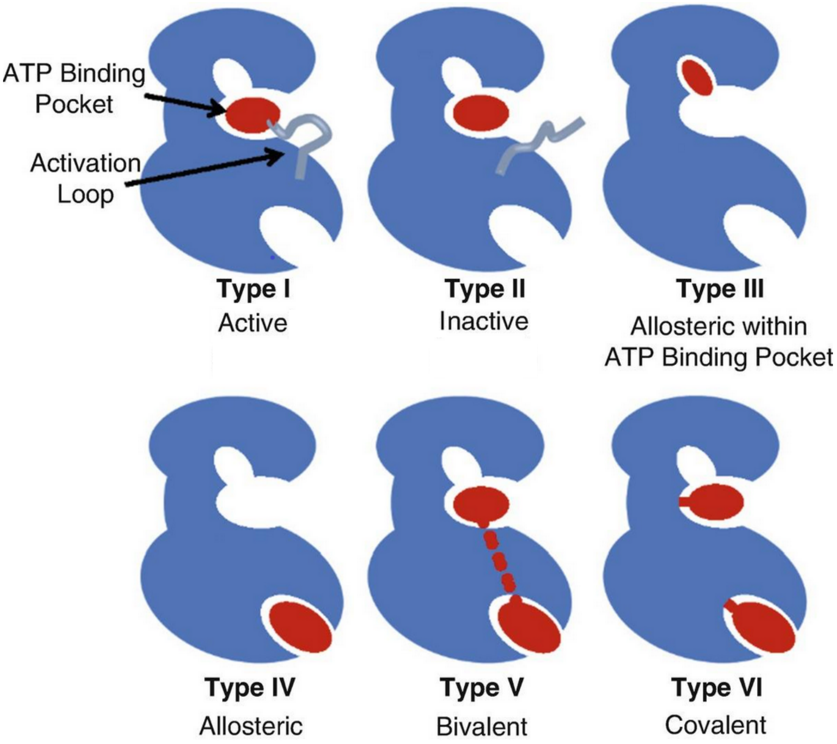


Figure 1.2: Illustration showcasing all six types (I-VI) of kinase inhibitors in simplified representations. The image displays inhibitors bound to their target kinases: Type I inhibitors targeting the active conformation, Type II inhibitors engaging the inactive form, Type III inhibitors binding to the allosteric site within the ATP pocket, Type IV inhibitors acting as allosteric modulators, Type V inhibitors with bivalent interactions, and Type VI inhibitors forming covalent bonds with the kinase. Image amended from (Martinez et al. 2020)

Type I and II target the ATP site of a given kinase in its active and inactive conformation, respectively. They each have their own limitations however. Type I require high concentrations in order to overcome the physiological concentrations of ATP, as this is their main agonist. Additionally, the highly conserved nature of the kinase ATP binding site across different kinases and kinase families, makes type I inhibitors very "promiscuous" i.e. they bind to targets they're not designed for (Roskoski 2016).

Relatively different from the above pharmacodynamics, Type III and IV do not bind to the ATP site of the kinase and are therefore considered to be allosteric in nature. Specifically, type III kinase inhibitors bind to an adjacent area and Type IV bind neither to the ATP nor to other peptide binding sites. Finally, type V bind to two different sites of the kinase structure making them bivalent inhibitors; type VI bond entirely differently to all the previous types, by covalent bonds as opposed to forming hydrogen bonds which is how all previous types bind to their target kinase(s). Due to the difference in strength between these two types of chemical bonds, types I-V are considered reversible whereas type VI inhibitors are not (Roskoski 2016).

1.1.5 Kinase inhibitors as a tool for scientific discovery

Treating cells with kinase inhibitors has become a popular means of perturbing signalling networks in an effort to better understand the signalling circuitry. For example, a cell culture may be treated with an inhibitor of a particular kinase and any changes in occupancy of thousands of phosphosites measured using phosphoproteomics (Hijazi et al. 2020). In theory, signalling cascades could be reconstructed from the data by following reductions in phosphosite occupancy, starting with the direct targets of the inhibited kinase. In practice, drawing reliable conclusions from such datasets is complicated by off target effects of the inhibitor, incomplete knowledge of kinase-substrate relationship, compensation mechanisms and other confounding biological factors.

Large perturbation experiments, in which many different inhibitors are applied (either individually or together), is one approach to disambiguating the cause of observed phosphosite occupancy changes (Hijazi et al. 2020).

However, these experiments produce large datasets that are difficult to interpret down to the level of individual phosphosite occupancy changes. Typically, frequentist statistics is used to hone in on some significant consequences of perturbation and these are then explained by manual interpretation of the data in the context of prior knowledge about kinases, their targets, and their role in signalling networks. It is tempting to consider that artificial intelligence, in the form or machine learning, may automate the interpretation of these datasets but is it poorly suited to the task due to difficulty in incorporating prior knowledge and because the dimensionality of the data make it unsuited for model training. The datasets produced typically contain occupancy data for many thousands of phosphosites per sample. However, despite the large number of phosphosites being analysed, the number of unique perturbation states (distinct experimental conditions or treatments) is usually limited.

1.2 Symbolic AI: Logic Programming and other paradigms

Artificial intelligence (AI) has experienced a boom in recent years both in application and research for its advancement, in academia as well as industry. Its widespread adoption is ever increasing and can be found in the majority of technical areas of modern life. As a fundamental research area of AI, symbolic AI has gone through a comparatively less fruitful era to its sub-symbolic or frequentist counterpart. The latter encompasses techniques that fall under the machine learning tree such as deep learning and neural networks, to name the most in vogue ones. These have benefited not only from advancements in software (n.b. pyTorch (Paszke et al. 2019), scikit-learn (Pedregosa et al. 2011)), but also the availability of large amounts of easily accessible data and advancements in hardware (reviewed in (Berggren et al. 2020)). The field of Symbolic AI on the other hand, experienced its relative heyday much earlier than its sub-symbolic cousin, therefore missed out on the benefits from the aforementioned advancements. Semantic and "under-the-hood" advancements continued to take place in the symbolic field, as well as applications in the field of automated theorem proving. Real world applications however were few and far between, this being especially true in the biosciences and the emerging field of bioinformatics and systems biology (Calegari et al. 2020).

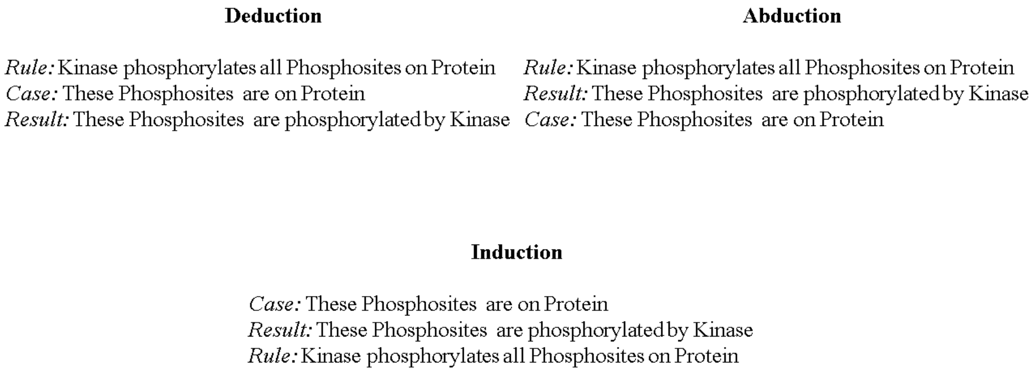
Following is a brief introduction to symbolic AI and its constituent subfields, techniques and applications. Namely, knowledge representation, reasoning and logic programming. Towards the end of the chapter is a description of the issues with explainability, black box modality and interepretability and how these relate and are overcome with approaches that fall under symbolic AI.

1.2.1 Knowledge representation, ways of reasoning and types of inference

Knowledge representation (KR) is an essential aspect of any Artificial intelligence implementation as no form of reasoning can take place without a representation of knowledge to base itself upon. Logic-based KR systems have been proposed in different forms with the aim of capturing terminological knowledge (Baader et al. 1970) via description logics or subjective knowledge via modal logics (Garson 2021).

Reasoning approaches on the other hand are used to describe the different types of inference that are used to rationalise over, draw conclusions from and analyse a collection of knowledge and premises. Different types of reasoning include but are not limited to, *first order logic* (FOL), *higher order logic* (HOL), the aforementioned *modal logic* and *Horn clauses.*

The three types of inferential principles, namely deduction, induction, and abduction, can be ranked based on their level of ”correctness” (Bergman et al. 2010). To understand this ranking, we should consider the level of ”necessity” associated with the inferences drawn from a given scenario. Deduction, represented in Figure 1.3, follows a top-down approach, where the inferences are necessarily true when the premises themselves are true. In other words, a valid conclusion is reached from universally true premises. On the other hand, induction and abduction lead to conclusions that are related to the initial premises but do not strictly depend on the truth of those premises. These two types are less rigidly defined compared to deduction, making the distinction between induction and abduction somewhat fuzzy. Induction involves a bottom-up approach, generalising from a set of facts and incorporating an element of likelihood in its conclusions. Abduction, on the other hand, is more ambiguous. It involves inferences derived from a subset of given true premises, but the validity of these inferences may change (either become less certain or more supported) when all the premises are considered as a whole (Douven 2021).



Deduction

Rule: Kinase phosphorylates all Phosphosites on Protein
Case: These Phosphosites are on Protein
Result: These Phosphosites are phosphorylated by Kinase

Abduction

Rule: Kinase phosphorylates all Phosphosites on Protein
Result: These Phosphosites are phosphorylated by Kinase
Case: These Phosphosites are on Protein

Induction

Case: These Phosphosites are on Protein
Result: These Phosphosites are phosphorylated by Kinase
Rule: Kinase phosphorylates all Phosphosites on Protein

Figure 1.3: The image showcases three examples of the 'beans in a bag' scenario, adapted to fit the context of this study. It illustrates the different types of reasoning employed in the research. Abduction, a form of reasoning, is seen in drawing conclusions based on incomplete evidence. Deduction, another type of reasoning, begins with a general principle and is applied to a specific situation. Meanwhile, induction involves deriving a generalised rule from specific observations.

Our approach employs techniques which belong to the field of logic programming, which in turn are based on deduction. The origins of logic programming in its Prolog (which stands for *PROgrammation en LOGique*) form can be traced to the work of Alain Colmarauer and Robert Kowalski. The latter being responsible for the conceptualisation of the logic programming "philosophy" while the former is credited with implementing the first language (Prolog III) (Colmerauer & Roussel 1992, Kowalski 1974). One of the most popular and relevant implementation of logic programming is the Prolog language is SWI-PROLOG. Being freely available (`https://www.swi-prolog.org/`) with well established semantics and a number of built-ins and libraries it has widely used as a teaching as well as research tool (Wielemaker et al. 2012).

A shortcoming of Prolog is that its facts and rules are assumed to be 100% true, thus rendering it relatively ill-equipped to deal with uncertainty or uncertain statements. This is particularly problematic in biological applications because experimental observations and prior knowledge often comes with some level of uncertainty. A way of augmenting logic programming with a probabilistic modality was first explored in the seminal paper by Sato describing distribution semantics (Sato 1995). This approach was later further established and applied with two probabilistic logic programming systems, PRISM (Sato 2009) and ProbLog (Raedt et al. 2015). The latter is syntactically similar to Prolog but extends the language with Sato's distribution semantics. Overall, it is comprised of two complimentary parts, one part which is Prolog syntax based with definite clauses and the same inference algorithm based on deduction; a second part which defines a probability distribution over the ground and true instances of the former Prolog based part.

1.2.2 Application of logic modelling in the bio-sciences

Examples of logic modelling or the use of a logical formalism to analyse, contextualise and make predictions from biological data exist in a variety of forms. Notably, applications centre around the representation of signalling networks as this form of cell interaction representation lends itself best to symbolic reasoning. Specifically, an early application of said approach resulted in a Boolean model that captured the intricacies of T Cell receptor signalling. This provided testable hypotheses in the form of novel predictions as well as missing links within the known signalling network (Saez-Rodriguez et al. 2007). Gene expression based regulatory networks have also been analysed via logic modelling based techniques to capture their structure in other model organisms such as *E. coli* (Maeyer et al. 2013) or their dysregulation in cancer (Remy et al. 2015).

In cancer research, as mentioned above, the dysregulation of networks, key to cellular processes, can be, and has been, modelled intrinsically or inferred via the use of logic modelling paradigms. These attempts include, but are not limited to, PHONEMeS and KinomeXplorer whose aim is to contextualise phosphosites within a kinase-protein or kinase-kinase interaction network (Terfve et al. 2015, Horn et al. 2014). The former, leverages phosphoproteomics data in a perturbation context, with prior knowledge (in the form of kinase-phosphosite substrate associations) in order to elucidate network wide changes, that these perturbations elicit. It combines a statistics based pre-modelling step, with a quasi-logic programming implementation. KinomeXplorer on the other hand focuses on a motif based approach. Their approach combines previous work of the same group, namely the NetPhorest (Miller et al. 2008) and NetworKIN (Linding et al. 2007) algorithms, which revolve around phosphobinding domains and kinase target recognition motifs, with a Bayesian scoring system. This enabled them to more accurately predict and model the behaviour the kinase networks and their targets in a plethora of experimental scenarios. Another application that is available as an R/Bioconductor package is CellNOptR which includes a collection of methods based on logic formalisms, such as integer linear programming (similar to the updated PHONEMeS implementation), probabilistic logic formalisms and a "Boolean simulator" amongst others. This collection of readily available tools has been used in a number of publications, to analyse specific pathways from diverse dataset origins (microfluidics perturbations (Eduati et al. 2020), single cell mass cytometry (Tognetti et al. 2021) and publicly available phosphoproteomics data (Traynard et al. 2017)).

Other methods that involve extensive testing under various experimental conditions (such as different time points, inhibitor panels, and multiple MS/MS runs) have been developed and made available to the public as computational tools. These studies either require extensive experimental testing or rely on uninterpretable machine learning (ML) "black box" models which offer little to no explanation as to feature selection and importance. In order to circumvent the above issues and pitfalls, we drew inspiration from the study that put forward the concept of the Robot Scientist. The work described therein lies in the fields of logic programming and knowledge representation and tackles the issue of automating the scientific discovery process. This is achieved by combining an inductive logic model to formulate hypotheses, with a robot to carry out the experimental procedure in order to validate or reject them (King et al. 2004). More specifically, they describe the development and application of a robotic system for conducting rational, hypothesis-driven experiments in yeast biology using Inductive Logic Programming. It is mainly the use of logic programming to establish a logic-based model with which one can actively interact in order to interpret the data. Another relatively automated system (Eve as it was baptised) was developed in order to aid in the automation of early-stage drug development, specifically for neglected tropical diseases (King 2017). Interestingly this later implementation has recently been used to confirm, alarmingly, the phenomenon dubbed "reproducibility crisis" Roper et al. (2022).

1.2.3 Explainability of AI

Progressively, we have witnessed our reliance on intelligent systems, which in turn rely on all the diverse forms of AI, increase in all facets of human society. Medical and biological research is no exception to this phenomenon, however, in these fields there is a marked difference in the need to explain the behaviour, predictions and suggestions of such systems. There are a number of fields, but amongst them most obvious are the medical and legislative that require near absolute interpretability in order for any form of machine learning/black box model output to be considered and/or implemented. This is especially true when such outputs are used to inform decisions or actions taken by health professionals that have a direct impact on disease prevention, treatment and outcome. A large section of sub-symbolic AI techniques such as those based on machine learning (Deep Learning, Neural Networks etc.) fall short in this respect. When a model based on deep learning provides a prediction, classification or action suggestion this is based on millions to billions of operations over a vast amount of input data. Understandably, it is impossible for a human to interpret and analyse each of these operations individually, ergo these models are referred to as *black box models* (Crabbé & Schaar 2022).

The need for integrating explainability and interpretability into model building as well as creating "white box" models (as opposed to "black box") is becoming progressively more poignant even since 2018 (Medicine 2018). From 2015 the term Explainable AI (XAI) was floated in a call for proposals by USA's DARPA further highlighting the need for a concerted effort towards achieving the goal for human understandable, auditable and trustworthy AI (Gunning et al. 2021). Approaches to render the predictions of such *black box* models less opaque have been extensively reviewed in (Das et al. 2022), with notable methodologies within the *Symbolic metamodeling* framework and *Post-Hoc Explainability* context. The former (Alaa & Schaar 2019) aims for disambiguation by expressing the totality of the black-box model as a human readable and transparent mathematical equation; whereas the latter augments the predictions of a black-box model with explanations derived from either the individual features upon which they were based or by highlighting prediction-relevant parts of the training set (Crabbé & Schaar 2022).

It is also worth noting that in most mainstream applications of AI, such as speech recognition, computer vision and language translation, the accuracy of the trained model is far more important than understanding how the model works. In scientific research the opposite is often the case: building a model that predicts the outcome of an experiment can be useful but novel discoveries come from understanding the biological mechanisms that led to the outcome.

As can be seen, the above methodologies are employed to essentially provide an explanation layer that sits alongside the chosen sub-symbolic model design. Conversely developing a model within a logical programming framework, be that First Order Logic, Higher Order Logic or augmented with probabilities, by its syntactical nature, renders said model inherently observable and by extension its predictions more easily understandable and auditable (Sokol & Flach 2022). It can be described as inherently observable due to the nature of its active components, namely, the *rules* and *facts*, as is the case for most models developed under a Logic Programming framework. These constituent components can be arranged in a manner that makes the process of *satisfaction* almost self-evident. A model that is set up to be human understandable is also auditable. It is precisely this ethos, alongside the mentality captured by the following quote from (O'Keefe 1990) "Elegance is not optional" that was adhered to when undertaking the logic modelling and symbolic AI tasks that make up the bulk of this thesis.

As evident from the mentioned methodologies, they are employed to provide an explanation layer alongside the chosen sub-symbolic model design. On the other hand, developing a model within a logical programming framework, whether First Order Logic, Higher Order Logic, or enriched with probabilities, inherently makes the model observable. This syntactical nature of the model enhances its understandability and auditability (Sokol & Flach 2022). The active components in such models, namely the *rules* and *facts*, can be organised in a way that makes the process of *satisfaction* almost self-evident, a characteristic common among Logic Programming models. When a model is designed to be human-understandable, it naturally becomes auditable. It is precisely this ethos, alongside the mentality captured by the following quote from (O'Keefe 1990) "Elegance is not optional" that guided the logic modelling and symbolic AI tasks undertaken in this thesis.

1.2.4 Challenges of phosphoproteomic data

Current approaches in phosphoproteomics based analysis are plagued by a number of issues, which include but are not limited to, lack of active prior knowledge use, non-interpretable Machine Learning (ML) models and inherent research bias. Building a model than can logically analyse phosphoproteomic data and offer human readable explanations as to how this is carried out can help us further understand how cell signalling circuitry operates and is disturbed by disease and other perturbations.

With high throughput phosphoproteomics data collection methods comes the burden of sifting through increasingly larger amounts of data points (Rudolph et al. 2016). Sifting through this data to find meaningful associations, test hypotheses or simply in an effort to explain the observations is an arduous task. The challenge lies not only within the data itself but also the approaches employed by computational biologists. First, the data under-represents the system by lack of phosphoproteome coverage or unknown functionality of phosphosites. Secondly, the analyses often focus on a small subsection of phosphosites, usually a handful of the observed, and attempt to make purely statistical i.e. ad-hoc in nature, by way of an arbitrarily chosen cut-off, conclusions and extrapolations (Humphrey et al. 2013). Alternatively, approaches require a range of experimental conditions, including different time points, inhibitor panels, and multiple mass spectrometry runs. These approaches can provide valuable context for phosphorylation data, but also harbour the danger of missing non-canonical associations.

A recent study, and the source of observational data for the main body of this thesis, tried to tackle the above issue with an overview approach, establishing the EBDT (Expectation of Being a Downstream Target) algorithm. This scores potential downstream phosphosite targets of kinases based on the activity response of kinases to perturbagens and the effects of the later on the entire phosphoproteome landscape (Hijazi et al. 2020). Another that tried to offer a similar overview approach, decided to initially mine publicly available data and machine learn a phosphosite functionality score (Ochoa et al. 2020). This was then used to rank phosphosites and key candidates were chosen for further experimentation (knockdown, specific inhibitor targeting etc.). Once again however these studies either require extensive experimental testing or rely on uninterpretable machine learning models which offer little to no explanation as to feature selection and importance.

1.2.5 Issues with Background Knowledge: The Dark Kinome and biases in the study of kinases

Kinases and proteins that interact with each other through phosphorylation cascades usually do so in a temporal manner. This, combined with the fact that kinase inhibitor selectivity, as mentioned above, can be described as type-dependant, makes the study of these interactions challenging. Another issue is that of the approximately 43,000 phosphosites only 20 % have an associated kinase (of the 550 known) that targets them (Dinkel et al. 2011). Combining this with the ever present issue in proteomics of whether a given (phospho-carrying) peptide is even detectable (Li et al. 2010), it is no wonder we have terms such as “dark kinases” (Essegian et al. 2020) and the “dark kinome” (Axtman 2021) to describe our lack of context specific knowledge relating to this highly active subsection of proteins.

The above has lead to an annotation inequality and subsequent research bias, where a group of 5,000 well studied proteins, and by extension an even smaller subgroup of kinases is the “beneficiary” of major research focus. Of note, 9%, approximately 1600 individual proteins are the subject of almost 80% of a representative body of $>$ 2m open access articles from PMC (Sinha et al. 2018). Ultimately this narrowing of research focus and overexpendiditure of resources on a subsection of the proteome and kinome, leads to blind spots in our understanding of normal as well as disease dysregulated protein based cell functionality. As a side note, but also understandably this issue plagues (and has plagued for some time) genomics, where it is described as the street light effect (Dunham 2018).

However weighty these issues may appear, and its kinase specific fallout we have had to deal with in our approach, initiatives such as the one from the US National Institutes of Health (NIH) (Rodgers et al. 2018) have identified a vast number of proteins whose mechanistic interactions are ripe for study and elucidation. An example success story of how the issue of understudied proteins has been overcome is the output of research on LMTK3. When it was first identified in 2011, the lack of relevant characterising literature as well as more direct issues such as the absence of targeting inhibitors and antibodies presented a substantial hurdle to surmount. Over the intervening years it has, through consistent effort, emerged as a key cancer driver, which has been characterised mechanistically, therefore serving as a highlight example on how to tackle the aforementioned issue as well as its wider benefits (Vella et al. 2021).

Another initiative aimed at shedding light in the oft-understudied proteins in order to better describe and mechanistically position them within context, is The Understudied Proteins Initiative. Initially, a survey of interest from researcher participants will highlight and identify candidate proteins, which in turn through a combination of literature mining as well as data generation will then provide hypotheses to be experimentally tested and validated. The scope of this initiative also includes experimental techniques to be used with particular focus on capturing the effects of post-translational modifications such as phosphorylation (Kustatscher et al. 2022).

Pertinent to our approach; described below and in detail in the relevant chapters of this thesis, the lack of phosphorylation related coverage, such as kinase-substrate relationship and the effect of phosphorylation on a given kinases activity, is an issue with which we grappled extensively. Additionally, the varied granularity of the available literature and associated knowledge bases describing these interactions and their effect (Needham et al. 2019) is a matter that is addressed in various databases (Horn et al. 2014).

The above issues, difficulties and blind spots coloured as well as guided our approach and how we applied our tools of choice. This is further described in the ensuing subsection.

1.3 Aims and Objectives

The aim of this PhD project is to evaluate the ability of symbolic AI based algorithms to overcome the limitations of existing tools used for analysis of phosphoproteomics data in the context of prior knowledge.

1.3.1 Research Questions

The underlying research question from the beginning of this endeavour, was can symbolic AI methodologies such as logical reasoning help us make new biological discoveries, specifically in the domain of cell signalling. Based on the fact that new scientific knowledge is discovered by interpreting experimental data within the context of relevant background knowledge, the hypothesis therefore was whether this inference procedure could be automated.

More specifically, we wanted to see whether logical reasoning could discover:

1. Novel inhibitor-kinase relationships (direct and indirect)
2. Kinase/Phosphatase-Protein relationships
3. Effects of phosphorylation on Kinase/Phosphatase activity

In this thesis, we will demonstrate that our initial approach to modelling biological systems used logical reasoning to capture their complexity and provided new insights into cell signalling. We, later enhanced our approach by incorporating probabilistic logic, which allowed us to account for the inherent unpredictability of biological systems. This was achieved by introducing a *"level of belief system"* to our model, which operates alongside logical reasoning. Once the validity of this analysis was ascertained by answering our fundamental research hypothesis, it was then applied to a further dataset from a different model organism, namely yeast *Saccharomyces cerevisiae.* This allowed us to expand our understanding and use of symbolic

AI-based methodology. In this case, we studied the effects of gene knockouts rather than broad-spectrum kinase inhibition, which provided a clearer set of “ground truths” against which to test our methodology. Overall, this helped us to further refine and validate our approach to analysing biological systems.

In Chapter 2, I lay out how this was tackled, with firstly the establishment of the “toy model” as a “proof of concept” whether such an approach could yield correct associations. As we had designed the “toy model” it was trivial to decipher its outputs. Following this, larger and larger parts of the dataset where introduced alongside the expansion (and curation) of the background knowledge base. With successive iterations however the innate brittleness of the approach and by extension the developed models became more and more apparent. The following chapter is an attempt to deal with this as well as other issues.

Chapter 3 contains the expansion of the aforementioned model and knowledge base with a system containing levels of belief. This was achieved through the gradual introduction of “probabilities” in a step-wise fashion as described above. Initially with the development of a “toy model” to establish whether the approach is applicable and then with the piecemeal addition of further experimental facts and background knowledge facts. [1]

Following the application of this approach to our phosphoproteomics data, the next logical step was to see whether such approaches could lend themselves to answering similar questions in phosphoproteomics data derived in a similar manner but from other model organisms. A self evident candidate (due to its well documented proteome, genome and cell functionalities) was *Saccharomyces cerevisiae.* A recent study (Li et al. 2019) that made extensive use of phosphoproteomics data from different strains with gene deletions (specific to kinases) proved an invaluable resource of experimental

[1] Quote marks are needed here in order to avoid the fury of pure statisticians as in our context these calculated weights represent a level of belief in query outputs rather than probabilities in the strictest of terms/definitions.

facts. Chapter 4 explains how these were then populated, in a similar manner to above, with background knowledge, which was also manually curated even though significantly less granular then what is available with cancer cell line associations. The methodology was used to confirm perturbations and provide contextualised cell signalling rewiring due to these perturbations.

All python code, Prolog and ProbLog *rule* containing files can be found here: `https://github.com/Dudelder/Symbolic_AI_Opus`. Background Knowledge *facts* base, experimental *facts* and associated sources can be found here : `https://www.dropbox.com/sh/80tfwiwgvtggglw/AAAASA9WXpb1Qf3YDOZGAMYca?dl=0`.

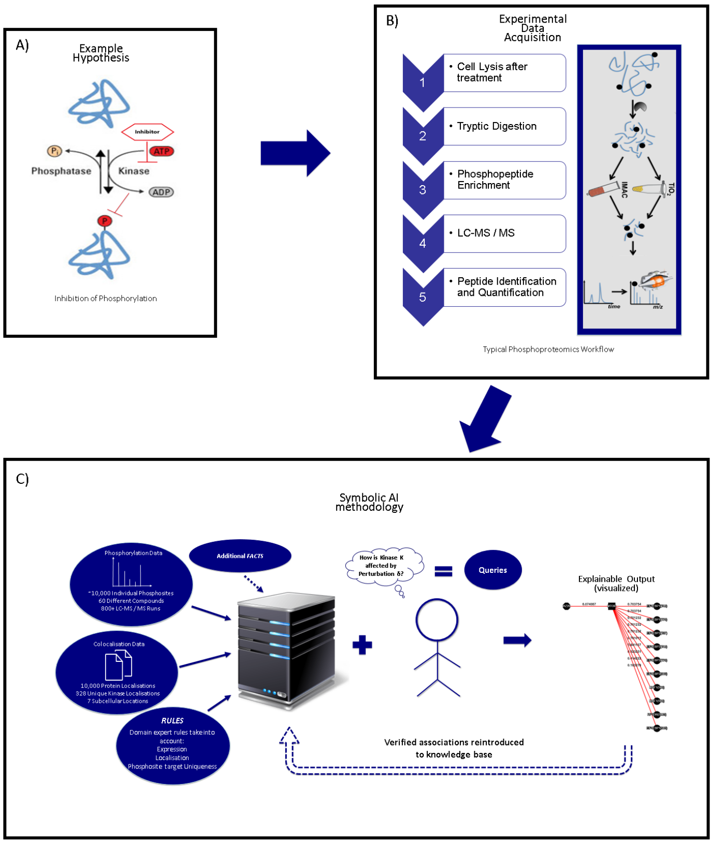

A)
Example Hypothesis
Inhibitor
Pi
ATP
Phosphatase
Kinase
ADP
P
Inhibition of Phosphorylation
B)
Experimental Data Acquisition
1
• Cell Lysis after treatment
2
• Tryptic Digestion
3
• Phosphopeptide Enrichment
4
• LC-MS / MS
5
• Peptide Identification and Quantification
IMAC
TiO2
time
m/z
Typical Phosphoproteomics Workflow
C)
Symbolic AI methodology
Phosphorylation Data
~10,000 Individual Phosphosites
60 Different Compounds
800+ LC-MS / MS Runs
Additional FACTS
Colocalisation Data
10,000 Protein Localisations
328 Unique Kinase Localisations
7 Subcellular Locations
RULES
Domain expert rules take into account:
Expression
Localisation
Phosphosite target Uniqueness
How is Kinase K affected by Perturbation δ?
Queries
Explainable Output (visualized)
Verified associations reintroduced to knowledge base

Figure 1.4: Overview of the study, experimental design and logic enabled analysis of results. A) Formation of hypotheses regarding the effects of inhibitors on the phosphoproteome of a cancer cell line or model organism. B) Phosphoproteomics experiments workflow in order to collect data based on hypotheses. Data can also be collected from relevant publicly available datasets. C) Symbolic AI Enabled Analysis workflow. Via the use of relevant background knowledge in the form of curated databases alongside a domain expert derived *rules* system allowed us to analyse and interpret the data in an explainable and auditable manner.

This page is intentionally left blank

Chapter 2

Building a Logic Program to analyse phosphoproteomics data

2.1 Introduction

2.1.1 Developing a Toy Model

In this chapter, we present a concise and informative toy model designed to illustrate a complex biological scenario involving kinases (k), proteins (p), and their phosphosites (s) in response to an initial perturbation (δ). This model simplifies the system while retaining its essential interactions, offering valuable insights into the intricate network of biochemical events within living organisms.

The first version of our toy model captures the mechanics of signal transduction, distilled to its bear essentials. Starting from the top of the graph visible in Figure 2.1, $\delta_{\mathbf{1}}$ and $\delta_{\mathbf{2}}$ represent two distinct perturbations that affect our model/system. The letter δ was chosen from the Greek word $\delta\iota\alpha\tau\alpha\rho\alpha\chi\acute{\eta}$ which amongst other things translates to perturbation or disturbance. Within the toy model representation of Figure 2.1 and Figure 2.2 purple is associated with perturbagens (heptagonal node), blue is associated with kinases (circular nodes and edges denoting known substrates), green is associated with phosphosites (circular nodes and edges denoting presence of phosphosite on a protein or kinase), experimental *facts* (observations) are denoted by orange edges and finally proteins are represented by teal coloured circular nodes.

Beginning with a perturbation $\delta_{\mathbf{1}}$ for which we have an observation that it has induced a fold change on the abundance of a phosphosite **s** and is known to affect the activity of a kinase **k**. The former represents our experimentally derived observations while the latter makes up part of our background knowledge. The above model was sufficient to capture a three step cascade following a perturbation δ, with the major assumption that phosphosites are uniquely targeted by a kinase.

The network in Figure 2.1 is a graphical representation of our knowledge base of Prolog clauses. Apart from the *facts* that by virtue of the way they have been written out are self explanatory, `perturbs(....)` is the experimental observation *fact*. Within this is the `occupancy(...)` *argument* that describes the log2 fold change between control and treatment for a given perturbation-phosphosite observation pair. Between -1 and 1 this was set as 'unchanged', above 1 as 'up' and below -1 as 'down'. These in turn reflect a halving (-1 and below) or doubling (1 and above) of measured phosphosite intensity between inhibitor treated and control. These were chosen to represent the fold changes present in our actual experimental data which will be described in detail in following sections of this chapter. In Prolog notation this takes the following form:

```
kinase(k1).
kinase(k2).
kinase(k3).
phosphosite(s1).
phosphosite(s2).
phosphosite(s3).
perturbation(delta1).
perturbation(delta2).
protein(p1).
known_target(kinase(k1), phosphosite(s1)).
known_target(kinase(k1), phosphosite(s2)).
known_target(kinase(k2), phosphosite(s2)).
known_target(kinase(k3), phosphosite(s3)).
is_on(phosphosite(s1),kinase(k3)).
is_on(phosphosite(s2),kinase(k3)).
is_on(phosphosite(s3),protein(p1).
perturb(perturbation(delta1),phosphosite(s3),occupancy(
   down)).
occupancy(up).
occupancy(down).
occupancy(unchanged).
cell_line(c1).
```

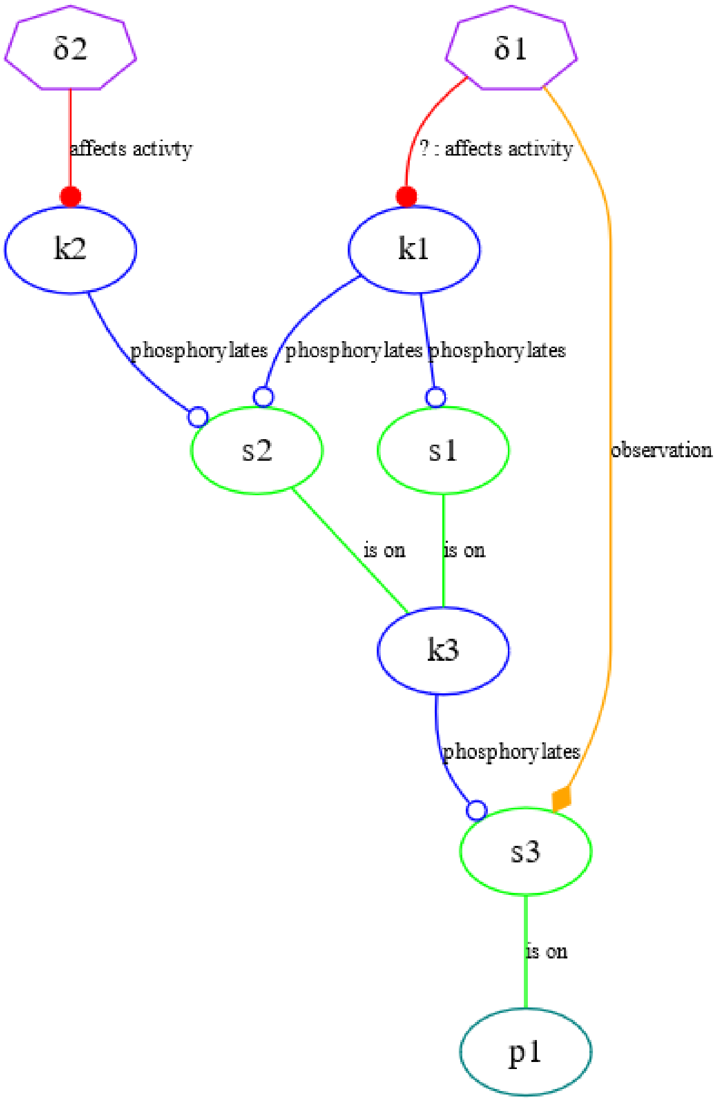


Figure 2.1: A toy model illustrates the minimal number of interactions needed to understand a biological scenario involving kinases (**k**), proteins (**p**), and their phosphosites (**s**) in response to an initial perturbation (δ). This initial version of the toy model represents three step cascade of interactions beginning from a perturbation, that potentially affects the activity of a kinase and our observation of this is via the relative change in abundance of a phosphosite on the protein (that is targeted by said kinase).

Facts of this format, only make up "half" of what a logic program is. The other "half" are *rules*. Within a logic programming setting these take the form of `h :- b`$_1$`, b`$_2$`, . . ., b`$_n$. On the left-hand side of the

`:-` symbol is the *head* of the *rule* and on the right-hand side is the *body* which can be made up of any number of *literals* (b_1, b_2,...,b_n). In the context of Prolog, it was previously mentioned that *facts* and *rules* can be thought of as 'halves.' While this analogy captures a partial truth, it is important to note that *facts* are not entirely separate from *rules.* In Prolog, a *fact* can be considered as a simple *rule* where the *body* (the condition) is always `TRUE`.

This distinction emphasises that *facts* and *rules* share similarities in their structure and representation. While *rules* have a condition (body) that must evaluate to `TRUE` for the rule to be applicable, *facts* can be seen as a specific case of *rules* with an implicit `TRUE` body condition.

This clarification is crucial in understanding the relationship between *facts* and *rules* in Prolog. While they have distinct names and are often treated differently in Prolog programs, they are inherently linked, and *facts* can be viewed as a special kind of *rule.* This insight helps in comprehending the fundamental structure of Prolog programs and how they handle knowledge representation and logical inference.

Therefore *rules* in Prolog can be considered as the framework based upon which the software can make logical connections between the individual *facts*, when queried. As mentioned previously the first query we sought to answer was whether we could infer the effect of a perturbation δ on the activity of a kinase **k** in the context of our toy model. In Prolog notation this question took the form:

```
1 doesDinhibitKinC(perturbation(D), kinase(K),
     cell_line(C)) :-
2    known_target(kinase(K), phosphosite(S)),
3    perturbs(perturbation(D), phosphosite(S),
        occupancy(down)).
```

The first line (1) constitutes the 'head' of the rule and is a predicate with arguments 'perturbation' and 'Kinase' in this case. This is followed by ':-'

which can be read as the logical statement 'if', i.e. the statement preceding :- is `TRUE` if the statement following is `TRUE` / all the 'goals' following can be satisfied at least once. The following two lines (2,3) make up the main *body* of the *rule.* Each of them can be a 'goal', which appears as part of a *rule* or the *head* of a standalone *rule.* The are separated by a comma which represents the logical operator ($\wedge$). Each of these needs to be satisfied in their specified order at least once, for the query to output `TRUE`. As we have set up the model, after consulting (loading in) the appropriate model .pl file, an open ended query to the Prolog interpreter SWI-Prolog takes the form:

```
?- doesDinhibitKinC(A,B,C).
```

By instantiating the *rule* in the query with capital letters A, B and C, as they denote *variables* in the Prolog notation, we are positing an 'open-ended' query. By doing this we are forcing the interpreter to unify our variables with any and all appropriate *facts* that satisfy the individual goals that make up the *rule* called by the query. In this case the query output is:

```
1 ?- doesDinhibitKinC(A,B,C).
2      A = perturbation(delta1),
3      B = kinase(k3),
4      C = cell_line(c1).
```

From the above, (1) is the query and (2-4) are the unifications of the variables A, B and C to the *facts* within our knowledge base. Therefore the inferred deduction is that there exists a perturbation ($\delta_{\mathbf{1}}$) that affects kinase (**k3**), in this case the effect being an inhibition as the occupancy in the *rule* is desired as down and the associated rationale as described above.

The first extension to the toy model was the addition of subcellular location defining *facts* for proteins and kinases to the knowledge base. The assumption behind this was that for a kinase **k** to act upon a phosphosite **s** (carried by a protein **p**), they (**k** and **p**) must both exist in the same subcellular location for a given time frame. Compounds that form part of a signalling cascade cannot interact with each other if they are not colocalised. Therefore the following *facts* were added:

```
plocation(protein(p1),subclocation(loc1),cell_line(c1)).
klocation(kinase(k1),subclocation(loc1),cell_line(c1)).
klocation(kinase(k2),subclocation(loc1),cell_line(c1)).
```

Accordingly the rule described above was amended in order to take into account these new *facts*.

```
1 doesDinhibitKinC2(perturbation(D), kinase(K),
     cell_line(C) :-
2      known_target(kinase(K), phosphosite(S)),
3      (is_on(phosphosite(S),protein(P)) ;
4      is_on(phosphosite(S), kinase(K))),
5      colocalistaion(protein(P), kinase(K)).
```

In (3) above we have a new goal which includes the operator ; that corresponds to ∨ i.e. either/or. This *goal* is satisfied when either of the two subgoals can be unified, i.e. when there exists a phosphosite **s** on either a protein **p** or a kinase **k**. This in effect sets up the next goal which in turn, corresponds to a stand-alone *rule* that takes the below form:

```
colocalisation(protein(P), kinase(K)) :-
    plocation(protein(P), sublocation(L),cell_line(c1)),
    klocation(kinase(K), sublocation(L),cell_line(c1)).
```

Here is where the aforementioned subcellular location *facts* come into play. Specifically, as a logical statement this *rule* takes the form:

Kinase (A) can be found in “Location” within the cell

Protein (P) can be found in “Location” within the cell

Given Location is shared ∗

∗ By assigning the same name “Location” to the variable when posing the query, the interpreter is forced to satisfy only the goals with *facts* that contain the same variable. Therefore the need for an extra *rule* to 'pick' the location is negated.

Both *rules* (colocalisation and 2nd iteration) succeeded as intended with the latter's output being, as expected:

```
?- doesDinhibitKinC2(A,B,C).
    A = perturbation(delta1),
    B = kinase(k3),
    C = cell_line(c1).
```

Such a rule however, does not take into account the potential of interplay between Kinases and their phosphosites (López-Otín & Hunter 2010). In order to overcome this, efforts were made to establish a *rule* that would take this cross-talk into account. The "uniqueness" *rule* was therefore developed in order to only consider phosphosites that are uniquely targeted by a kinase.

Initially, the *rule* defining what constitutes a shared target was developed. As can be seen below, the interpreter collates the kinases (**k**), up to two at a time, that share a particular phosphosite (**s**) (1,2 below) while the 3 sub-goal is there to ensure two different 'known_target' *facts* are chosen.

```
1 sharedtarget(phosphosite(S)) :-
2     known_target(kinase(K1), phosphosite(S)),
3     known_target(kinase(K2), phosphosite(S)),
4     kinase(K1) \= kinase(K2).
```

Testing the *rule* by querying the interpreter we get the below output. As expected given our model (Figure 2.1) we get that the shared phosphosite is **s2** (2, below).

```
1 ?- sharedtarget(phosphosite(S)).
2     S = phosphosite(s2).
3     true.
```

The above *rule* then makes up the main body of the *rule* below. Here the new symbol \+ is introduced. This represents negation (¬) in the Prolog syntax. A point of distinction here however as the inclusion of this symbol affects the overall semantics of the Logic Program we developed. Up to this point our Logic Program can be considered a definite program as it lacked negation, under the well-founded semantics. In a negation-free Logic Program the Least Herbrand Model and well-founded (semantics based) model are identical. In essence these models are the collection of all ground atoms, i.e. all satisfied *rules* containing all *facts* with which they are satisfied. Further, our models exist within the semantics of the Closed World Assumption (CWA), wherein everything that is not directly implied within our model to be `TRUE` is assumed to be `FALSE`. Practically, this means that for a given Logic Program and its ground instance, the left side of a *rule* (anything before the : −) is true *if and only if* or *iff* there exists one *rule* body (anything that is after the : −) that is `TRUE`. Further details can be found here (Dung 1992) and (Gelder et al. 1991).

From above, the negation of the “sharedtarget” *rule* needs to be satisfied, i.e. there needs to exist a phosphosite *s* that is “uniquely” (as per our definition) targeted by a kinase *k* for the *goal* and therefore the *rule* to succeed.

```
uniquetarget(phosphosite(S)) :-
    phosphosite(S),
    \+ sharedtarget(phosphosite(S)).
```

Testing the “uniquetarget” *rule* by querying the interpreter we get the below output. As expected given our model (Figure 2.1) we get that the shared phosphosite is **s3** (2, below).

```
1 ?- uniquetarget(phosphosite(S)).
2     S = phosphosite(s3).
3     true.
```

Following on from the definition of the “Uniqueness” *rule*, it was added, in turn, to the body of the 3rd iteration of our *rule* (4, below) which aims to identify which perturbation (δ), leads to the inhibition of a kinase **k**.

```
doesDinhibitKinC3(perturbation(D), kinase(K), cell_line(C)) :-
    known_target(kinase(K), phosphosite(S)),
    perturbs(perturbation(D), phosphosite(S), occupancy(down)),
    uniquetarget(phosphosite(S)),
    (is_on(phosphosite(S), protein(P)) ;
    is_on(phosphosite(S), kinase(K))),
    colocalisation(protein(P),kinase(K)).
```

Accordingly, when testing the above iteration of the “doesDinhibitKinC” *rule* we query the interpreter and receive the output in the manner below. As can be seen and again according to our toy model we receive the expected output of the δ**1** perturbation inhibiting kinase **k3**.

```
?- doesDinhibitKinC3(A,B,C).
    A = perturbation(delta1),
    B = kinase(k3),
    C = cell\_line(c1).
```

Having reached this stage of complexity and logical rigour in the “doesDinhibitKinC” *rule*, a potential fallacy became apparent. The interpreter could potentially yield a `TRUE` response based on only one *fact* regarding negative phosphosite fold change making it through the aforementioned logical “gates”, despite there being potentially a number of *facts* stating positive phosphosite fold change. The *rules* and *facts* added to the Logic Program were set up to check whether a majority of phosphosites (**s**) targeted by a kinase (**k**) had occupancy **down**, as well as satisfying all the previous iterations’ requirements. The rationale behind this was that for a given kinase **k** and its known target phosphosites **s**, their respective occupancy should be greater in number for those described as **down** over **up**. A graph representation of the amended toy model can be seen in Figure 2.2. This includes new phosphosites **s4** and **s5**, new observation facts (edges connecting perturbation nodes to phosphosite nodes).

As a first step, this required the development of a *rule* that can be considered the analogue of the “doesDinhibitKinC” *rule* but for excitation. The predicate was named accordingly “doesDactivateKinC/2”. The key difference with the latest, third iteration, was the substitution of the occupancy(down) sub-goal with occupancy(up). This new *rule* was then combined with the aforementioned iteration as well as the addition of the built-in ’findall/3’ predicate. This collates all the Target1 and Target2 phosphosites that satisfy the activate/inhibit (*rule*) sub-goals into two Lists which are then compared with the > operator. Alongside these *rules*, *facts* relating to phosphosites (**s4**, **s5** and a protein **p2** as well as their related colocalisation and observation *facts*) were added to the knowledge base.

```
majoritycheck(perturbation(D), kinase(K)) :-
    ((findall(Target1, (doesDinhibitKinC(perturbation(D)
       , kinase(K), cell_line(C),  phosphosite(S1))),
       TargetList1),
    findall(Target2, (doesDactivateKinC(perturbation(D),
        kinase(K), cell_line(C),  phosphosite(S2))),
       TargetList2)),
    length(TargetList1,Length1), length(TargetList2,
       Length2))
    compare(>, Length1, Length2).
```

Additionally the ’is_on/2’ predicated was amended accordingly to include the new phosphosites on proteins being considered. Therefore, the fourth (and final) iteration of the rule took the following form in Prolog notation:

```
doesDinhibitKinC4(perturbation(D), kinase(K), cell_line(
   C)) :-
    known_target(kinase(K), phosphosite(S)),
    uniquetarget(phosphosite(S)),
    (is_on(phosphosite(S), protein(P)) ;
    is_on(phosphosite(S), kinase(K))),
    colocalisation(protein(P),kinase(K)),
    majoritycheck(perturbation(D), kinase(K)).
```

It includes five subgoals that need to be satisfied in order for it to yield a TRUE response. Of these, three ("Colocalisation", "Uniqueness" and "Majority") are *rules* themselves.

These iterations of the "inhibited" *rule* have evolved organically, in an effort to capture the intricate nature of cascadal, kinase based cell signalling, as well as following a number of targeted Prolog sessions with collaborators. In the following section of Results, analysis of the query outputs has been based on all possible combinations of the three *rules* named above "Colocalisation", "Uniqueness" v1 and v2 and "Majority" (n.b. a number of amendments and improvements were made to the toy model developed rules in order for them to be applicable to the entirety of the dataset. These are described in further detail below).

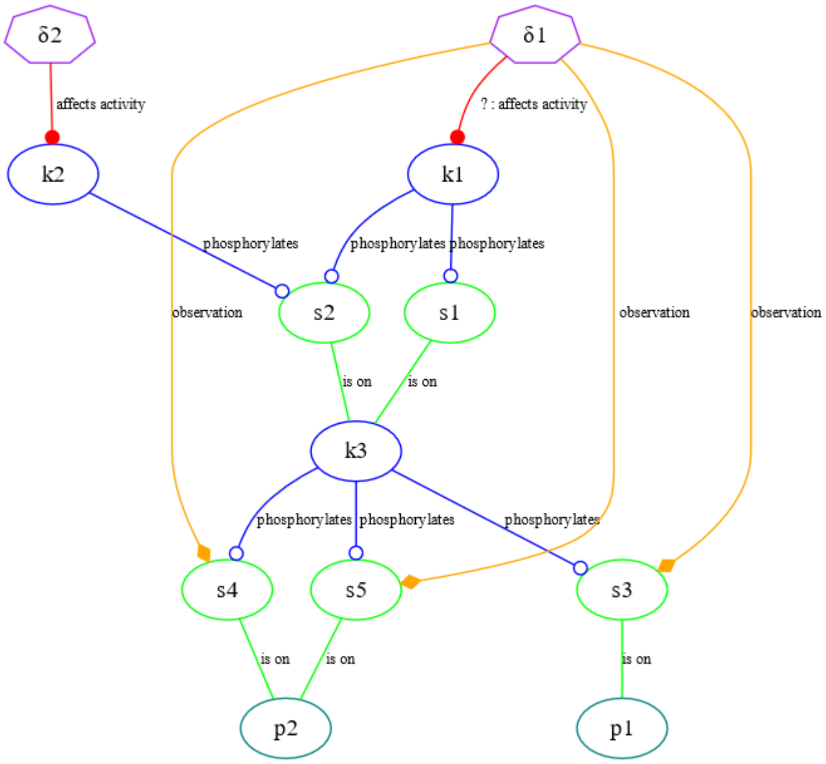


Figure 2.2: Amended version of the first toy model. Includes two new phosphosites **s4** and **s5** found on another protein **p2**. Their inclusion serves the purpose of providing *facts* for the the 'majoritycheck/2' predicate as part of the fourth iteration of the *rule*. The rationale behind this being that majority of known targets, which are also unique, for a given kinase need to have a decreased occupancy. With this predicate and the associated *facts* the aim was to capture an overall effect on the network specific to a kinase following a perturbation.

2.2 Methods

2.2.1 From phosphoproteomics data to *facts*

The core data set that formed the basis upon which we developed and trialled the methodology, described in this thesis was primarily taken from the following study (Hijazi et al. 2020). In brief, the data contains the effect of 63 kinase inhibitors on the extensive phosphoproteome of a panel of 3 cell lines, namely MCF-7, HL60 and NTERA2. For the purposes of our method development we focused solely on MCF7. An in depth detailed description of the cell culture methods, LC MS/MS protocols and kinase selectivity assays can be found in the methods section of (Hijazi et al. 2020). In this section an overview of these as well as the data processing steps needed to arrive to our *Facts* will be presented.

2.2.1.1 Sample run through LC MS/MS Instrument

LC MS/MS was carried out on samples from the treated cell lines. Each treatment was performed twice forming a biological duplicate and in turn each of these was analysed in duplicate thus making a total of four replicates. Following tryptic digestion as per published MS methodologies, TiO_2 enrichment for phosphopeptides was performed (Montoya et al. 2011). These are then introduced to the LC column and after elution injected into the mass spectrometer. Here they are ionised, accelerated and then go through the first stage of mass spectrometry MS1. As this is a Data Dependent Acquisition method (DDA), the mass spectrometer at the MS1 stage looks at all peaks, takes the ones that are highest in intensity and then picks these to further fragment (Davies et al. 2021). This secondary fragmentation constitutes MS2. These spectra represent fragmented peptides which in theory are phosphorylated as they have undergone the aforementioned TiO_2 enrichment process. TiO_2 enrichment is a way of pulling the phosphorylated peptides out, with the hope that the sample with mostly phosphorylated peptides enters the Liquid Chromatography column. Even with the above DDA method a large number of spectra is produced therefore there is a trade-off between how many spectra one wants to collect vs how clean of unnecessary noise these spectra are. One MS1 spectrum that fulfils the intensity requirement results in approximately 10 MS2 spectra being collected based on the highest abundance peaks, and this is then repeated until the end of the run. These are then collated across the run and form the .raw format file that is the final output of the instruments.

2.2.1.2 Process pipeline from .raw to log2 fold change stage

Initial processing of the .raw file takes place in Mascott (MacCoss et al. 2002). These contain lists of peptide sequences that arise from a model organism genome, in our case human. They match our acquired .raw spectra with these and then map them to proteins. In essence our spectra is compared to a database of known peptides and the most likely matches are determined. The software outputs .dat files which are manually collated and populated with meta data such as the run, the batch, the cell line and treatment they correspond to. These make up the ctam db available here (`https://ctamdb.sbcs.qmul.ac.uk/`) which also includes the above .raw files as well as information on individual peptides identified and associated scoring metrics.

The above process is required before the phosphorylation site quantification takes place. This is done by in house developed software based on the PESCAL methodology (Casado & Cutillas 2011), titled PESCAL++. In brief this pipeline takes as input the .raw files and the .dat peptide identification files from the Mascot software. It then chooses an appropriate retention time, finds the peak in the MS1 spectrum corresponding to the peptide and quantifies the intensity associated by calculating the area under the peak. It also carries this out and matches the spectrum across runs. This is done for all identified peptides. Therefore all identified peptides are matched to an Area Under the Peak (AUP) value. The approach can be described as identification lead as the samples containing the peptides are themselves unlabelled.

2.2.1.3 Transcribing data into *facts*

Once the initial instance of a queryable Prolog program was established, (toy model as described above) and was yielding responses, that made sense based on the limited amount of *facts*, the next step was to expand both the set of *facts* and the accompanying *rules*.

As the information stored in the database presented a large amount of intricately organised data, it was first essential to establish certain cut-offs in order to avoid redundant or dubious data being turned into facts. For each table that was accessed, due to its specific set up and format, different cut-off points were set. Taking as example the "Observation" table, from which the data relative to changes measured in individual phosphosites was taken, a threshold p-value $\prec 0.05$ was set. Apart from the data described in the "Toy Model" section a Python script was written which automatically transcribed data and background knowledge into *facts*.

Specifically for the experimental/observation data and the background knowledge included in the Chemphopro database, the process of this Python Program is threefold. Initially, it connects to the SQL based database, which allows for queries to be passed unto it. Following this, it creates a dataframe based on the query output, upon which it applies the aforementioned cut-offs. Finally, taking the formatted dataframe, it outputs in the correct file format as well as internal Prolog notation, *facts* based on the data extracted from the specified Chemphopro database table. In the case where background knowledge was not included in the SQL based database, an amended python script was used that scraped the data from a .csv format file or similar. A schema depicting the overall configuration of the ChemPhoPro database can be seen here Figure 2.3, additionally a website to access this data can be found here `http://chemphopro.org/`.

A significant portion of the data transcription process described above involved Regular Expressions as this was deemed the most effective way to parse and format the selected data. Each instance of transcribing data into Prolog facts has had its own Python software written. This was due to formatting issues specific to each table of data from the Chemphopro as well as the SubCellBarCode paper databases. Additional features of other Python scripts include a methylation observations removal script and naming convention checking script.

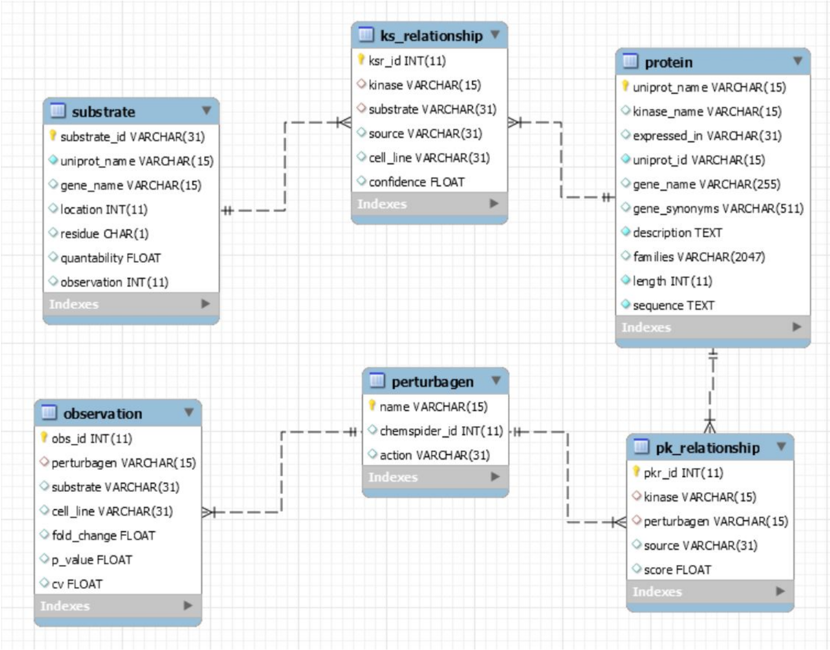


Figure 2.3: Database comprised of six tables, namely “Protein”, “Perturbagen”, “Observation”, “Substrate”, “KS_relationship” and “PK_relationship”. Per table these contain, Protein: Mainly naming sequence and length information, Perturbagen: Compound naming and synonyms, Observation: Main data table containing substrate responses to compound treatment, Substrate: Phosphosite specific information, KS_relationship: Kinase and substrate relation data from UNIPROT and PDT, PK_relationship: Known Perturbagen and Kinase combinations with source of information

2.2.2 Background Knowledge Databases

2.2.2.1 Known target sources

Our knowledge base consists of three types of *facts*. Experimentally derived, logically derived and *facts* curated from trusted background knowledge sources. This distinction is not passed to the interpreter but serves the purpose of defining the origin of the *facts* contained in our knowledge base. A key subset of curated background knowledge *facts* are those describing known phosphosite targets of kinases. These fall within all three types mentioned above and are instrumental in our approach. They provide the logical bridge between the response of individual phosphosites to drug treatments and the activity of kinases acting upon said phosphosites. The main background knowledge sources are described below.

As mentioned above in Chapter 1 the main source of experimental data (”Observations”) for this body of work came from the EBDT study (Hijazi et al. 2020). Part of their study involved the development of the Expectation of Being a Downstream Target (EBDT) algorithm. This scores potential downstream phosphosite targets of kinases based on the activity response of kinases to perturbagens and the effects of the later on the entire phosphoproteome landscape. The main source however, of known kinase targets contained in the chemphopro db comes directly from UNIPROT (Bateman et al. 2017). This curated database includes known kinase and phosphatase targets which were transcribed to *facts* in the same way as described above. UNIPROT being mostly manually curated from literature is expected to offer a reliable source background knowledge.

An issue that became apparent and constituted a considerable stumbling block throughout the main study and its constituent projects is the overlap, or significant lack thereof, between experimental *facts* and background knowledge *facts*.

In an effort to demonstrate this Figure 2.4 has 3 Venn diagrams which represent how many phosphosites present in our experimentally derived *facts* have a corresponding *fact* detailing which kinase(s) target them.

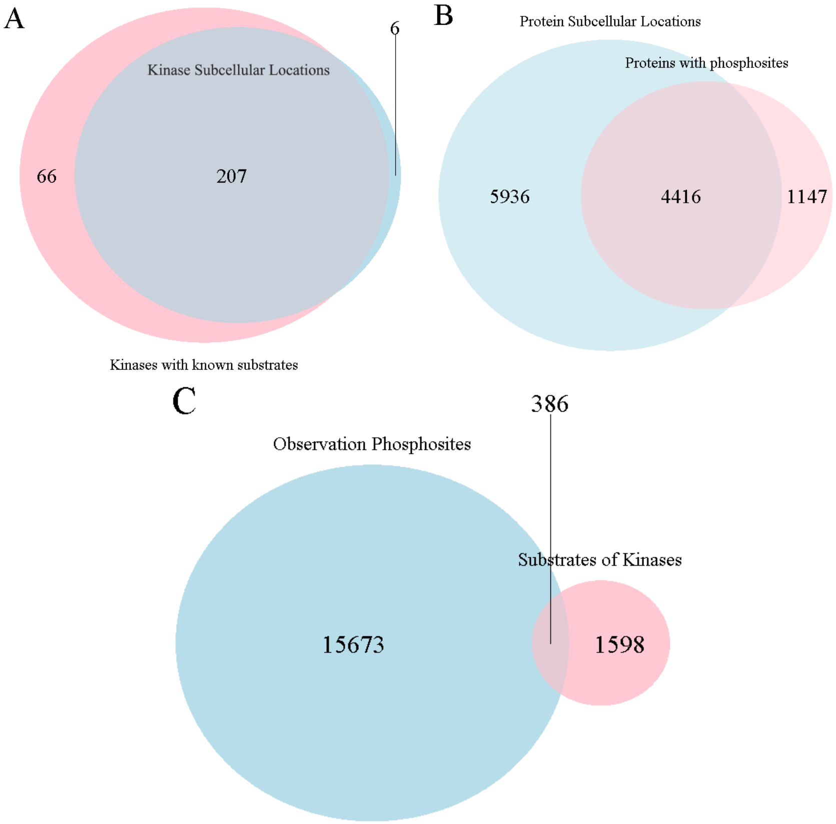


Figure 2.4: Venn Diagrams depicting the overlap between sets of *facts* that make up our background knowledge base. A: Overlap between kinase subcellular locations and kinases with known phosphosite substrates. B: Overlap between protein subcellular locations and proteins with phosphosites (relevant to the 'is_on/2' predicate). C: Overlap (or lack thereof) between observational data phosphosites and phosphosites for which we have a kinase association. Specific sources include EBDT (Hijazi et al. 2020) (phosphosite/protein lists), UNIPROT (Bateman et al. 2017) (kinase phosphosite associations), SubCellBarCode (Orre et al. 2019) (subcellular locations for both kinases and proteins)

Facts List	Source	Size (Number of Facts)
Kinase - Substrate Associatons	UNIPROT and EBDT	3,995
Inhibitor - Kinase Association	Klaeger, EBDT and vendors	722
Phosphosite presence on Protein	UNIPROT	18,795
Protein Subcellular locations	SubCellBarCode	10,353
		Total = 33,856

Table 2.1: Table listing the title, sources and associated number of *Facts* that make up the Background Knowledge section of the logic model. Specific sources referenced Klaeger (Brand et al. 2015), EBDT (Hijazi et al. 2020), UNIPROT (Bateman et al. 2017), SubCellBarCode (Orre et al. 2019)

2.2.2.2 Subcellular protein and kinase location source: SubCellBarCode

For reasons that will become clearer in the following section, (Rule combinations and their effects) another database needed to be mined for relevant *Facts*. The database in question (referred to as SubCellBarCode from now on) was taken from (Orre et al. 2019) which reported the subcellular location of proteins mapping to approximately 12000 individual genes. The four locations reported in the study were Mitochondria, Secretory, Cytosol and Nucleus. The rationale behind this, explained further below, was that in order for a kinase to act upon a protein (or another kinase) they both need to be located in the same subcellular location. Based on the Python scripts developed for the Chemphopro database, a new set was developed and amended to work with this databases' format. The resulting *Facts* described the subcellular location of both kinases and proteins of interest and became of intrinsic importance as the complexity and logical rigour of the rules advanced.

2.3 Results

2.3.1 Rule combinations and their effects

Data collection from Prolog query outputs is a multi-step process. Initially, it involves posing an open-ended query to the interpreter which in turn tries to unify the variables with all possible values that satisfy the *rules* and *facts* in the knowledge base. In essence, it outputs all the possible instantiations to the variables defined. An open-ended query calling forth the 3rd iteration of the inhibited rule (explained above), would take the form:

```
doesXinhibitAinC3(Perturbagen, Kinase, Cell_Line).
```

This, returns sequentially, all combinations of `(Perturbagen, Kinase, Cell Line)` that are logical implications, defined as part of the *rule* invoked. Prolog also includes predicates 'findall/3' and 'bagof/3', with which all solutions of a query can be collated in lists. The latter of the two predicates parses the list and removes duplicate entries. These were used extensively in the first part of the data acquisition process.

During the application of the toy model developed *rules* to the main dataset and extended knowledge base an addition to the 'Uniqueness' *rule* was made. The second version of the *rule*, referred to and explained as U2 includes a colocalisation check as part of its 'sharedtarget' sub-goal. The rationale behind this was that even though two Kinases might have the same known substrate, this should not be taken into account if they were not also colocalised. Specifically, on the first version of the "Uniqueness" *Rule*, a sub-goal was added in the "shared_kinase" *Rule*. This was, in essence, a modified version of the aforementioned "Protein and Kinase Colocalisation" *Rule*.

In order to examine the effect of the different "sub-goals" and standalone *rules* on the query outputs of Prolog the following combinations were considered. These can be found in Table 2.2, below. Zero refers to the basal iteration of the Inhibited *Rule* as described above. This was used across all combinations tested as it forms the basis and poses the highest logical "hurdle" to be overcome.

Rule Combination	Description
ZERO	See text
1ST	1st Iteration or Zero + Colocalisation
2ND	2nd Iteration or 1st Iter. + Uniqueness 1
3RD	3rd Iteration or 1st Iter. + Uniqueness 2
COL_M	Colocalisation + Majority
U1	Uniqueness1
U1_M	Uniqueness 1 + Majority
U2	Uniqueness 2
U2_M	Uniqueness 2 + Majority
4TH	4th Iteration or 3rd Iter. + Majority
4TH_U1	Uniqueness 1 + Majority

Table 2.2: Contains a brief description of what each acronym used in the section below refers to.

	ZERO	COL	U1	U2	M	COL_M
ZERO	■	1ST	U1	U2	■	COL_M
COL	1ST	■	2ND	3RD	COL_M	
U1	U1	2ND	■	■	U1_M	4TH_U1
U2	U2	3RD	■	■	U2_M	4TH
M	■	COL_M	U1_M	U2_M	■	■
COL_M	COL_M	■	4TH_U1	4TH	■	■

Table 2.3: Contains all the combinations that were considered in the analysis. Note: Black boxes signify redundant or overlapping combinations which were not considered.

Once the relevant queries, based on the combinations described above, were put to the Prolog interpreter, its output was collected in the aforementioned "lists" and parsed into Python. Here, a series of tailored Python scripts was employed across all combinations to further analyse and quantify the "lists". The main data analysed revolves around the number of compounds predicted to inhibit a given kinase, as well as the number of targets predicted to be inhibited by a given compound.

For the purposes of analysis, a new metric, dubbed modified Positive Predictive Value (mPPV) is introduced. Based on the formula for Positive Predictive Value, a well-established metric, the False Positive value was replaced by the Predicted Positive ($\mathrm{P_rP}$) value. The above is defined as:

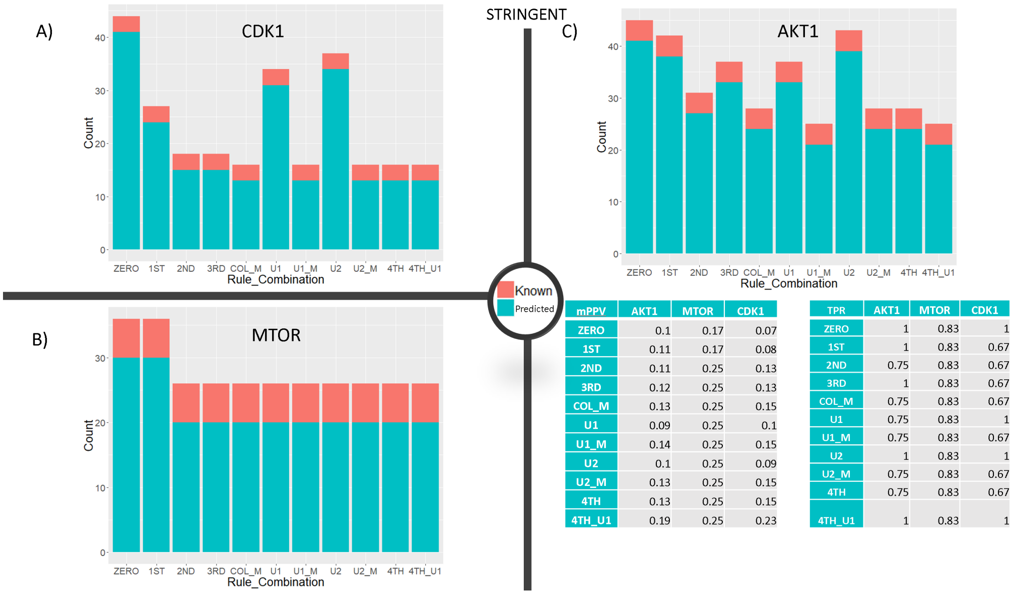
$$mPPV = \frac{TP}{P_rP}, \; P_rP = P_r - TP$$

where

$$P_r = All\ Predictions$$

In addition to the above, the metric for True Positive Rate was calculated for each Kinase and Perturbagen based on all possible query combinations. Analysis was carried out on two sets of Observation *facts*, differentiated by cut-offs on log2fold-change. One set had a cut-off of < 0 for when Phosphosite can be considered as perturbed “down” and the other had a cut-off of < -1 (i.e. Greater than 50% reduction). These are referred to as “Lenient” and “Stringent” in the remainder of the chapter, respectively.

All subsequent graphs were produced via R/Rstudio using ggplot and plotly packages available via CRAN.



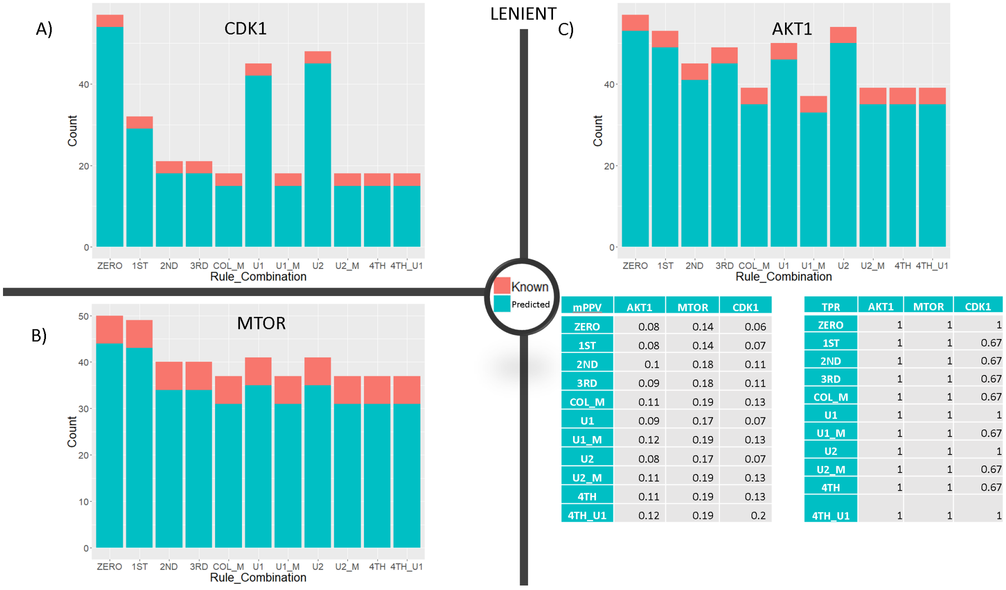
mPPV	AKT1	MTOR	CDK1
ZERO	0.1	0.17	0.07
1ST	0.11	0.17	0.08
2ND	0.11	0.25	0.13
3RD	0.12	0.25	0.13
COL_M	0.13	0.25	0.15
U1	0.09	0.25	0.1
U1_M	0.14	0.25	0.15
U2	0.1	0.25	0.09
U2_M	0.13	0.25	0.15
4TH	0.13	0.25	0.15
4TH_U1	0.19	0.25	0.23

TPR	AKT1	MTOR	CDK1
ZERO	1	0.83	1
1ST	1	0.83	0.67
2ND	0.75	0.83	0.67
3RD	1	0.83	0.67
COL_M	0.75	0.83	0.67
U1	0.75	0.83	1
U1_M	0.75	0.83	0.67
U2	1	0.83	1
U2_M	0.75	0.83	0.67
4TH	0.75	0.83	0.67
4TH_U1	1	0.83	1

Figure 2.5: Bar charts illustrating the comparison between Predicted and Known counts of Perturbagens for different combinations of subgoals (acronyms described in Table 2.2). The dataset is based on the "Stringent" set of Observation *facts*. Panels A, B, and C present results for CDK1, MTOR, and AKT1 Kinases respectively. Accompanying each panel, tables provide comprehensive mPPV (modified Positive Predictive Value) and TPR (True Positive Rate) metric values for each evaluated combination of rules.

The two Figures 2.5 and 2.6, contain bar charts with the Predicted and Known counts for a panel of 3 Kinases, namely CDK1, AKT1, MTOR. These were selected based on their key role in the CDK1-PDK1-PI3K/Akt pathway, well studied and described in homeostasis as well as dysregulation in diseases such as cancer (Koundouros & Poulogiannis 2018, Koundouros et al. 2020). The tables included within both figures list the values for the metrics mPPV and TPR.

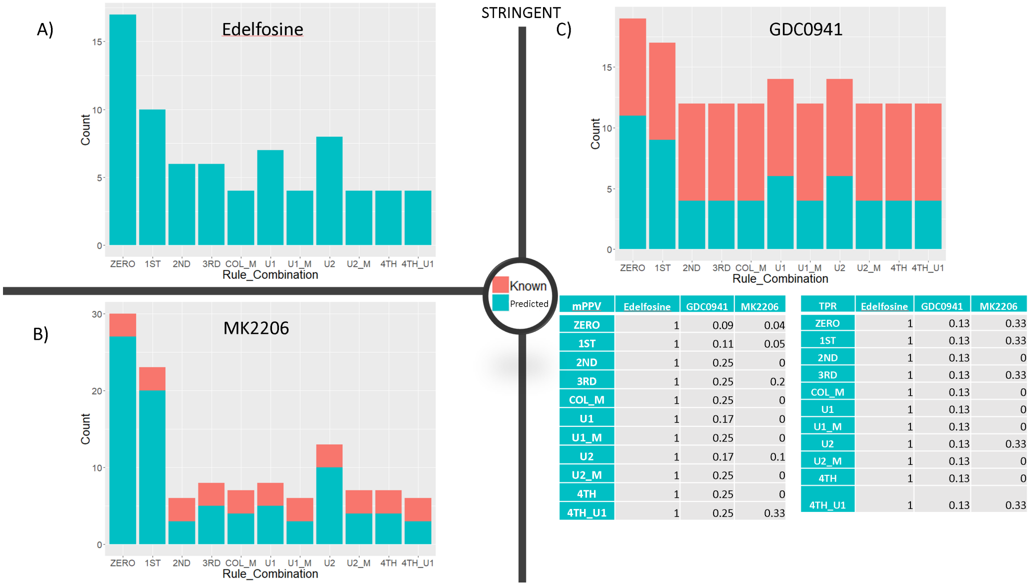
Figure 2.7 can be considered to represent the reverse perspective of Figures 2.5, 2.6. The bar charts and tables of metrics are based on the number of predicted Kinase Targets for a panel of Inhibitors. GDC 0941 is a PI3K inhibitor shown to significantly decrease AKT phosphorylation both *in vitro* as well as *in vivo* (Usman et al. 2018). MK2206 is a well-known pan-AKT



mPPV	AKT1	MTOR	CDK1
ZERO	0.08	0.14	0.06
1ST	0.08	0.14	0.07
2ND	0.1	0.18	0.11
3RD	0.09	0.18	0.11
COL_M	0.11	0.19	0.13
U1	0.09	0.17	0.07
U1_M	0.12	0.19	0.13
U2	0.08	0.17	0.07
U2_M	0.11	0.19	0.13
4TH	0.11	0.19	0.13
4TH_U1	0.12	0.19	0.2

TPR	AKT1	MTOR	CDK1
ZERO	1	1	1
1ST	1	1	0.67
2ND	1	1	0.67
3RD	1	1	0.67
COL_M	1	1	0.67
U1	1	1	1
U1_M	1	1	0.67
U2	1	1	1
U2_M	1	1	0.67
4TH	1	1	0.67
4TH_U1	1	1	1

Figure 2.6: Bar charts illustrating the comparison between Predicted and Known counts of Perturbagens for different combinations of subgoals (acronyms described in Table 2.2). The dataset is based on the "Lenient" set of Observation *facts*. Panels A, B, and C present results for CDK1, MTOR, and AKT1 Kinases respectively. Accompanying each panel, tables provide comprehensive mPPV (modified Positive Predictive Value) and TPR (True Positive Rate) metric values for each evaluated combination of rules.



mPPV	Edelfosine	GDC0941	MK2206
ZERO	1	0.09	0.04
1ST	1	0.11	0.05
2ND	1	0.25	0
3RD	1	0.25	0.2
COL_M	1	0.25	0
U1	1	0.17	0
U1_M	1	0.25	0
U2	1	0.17	0.1
U2_M	1	0.25	0
4TH	1	0.25	0
4TH_U1	1	0.25	0.33

TPR	Edelfosine	GDC0941	MK2206
ZERO	1	0.13	0.33
1ST	1	0.13	0.33
2ND	1	0.13	0
3RD	1	0.13	0.33
COL_M	1	0.13	0
U1	1	0.13	0
U1_M	1	0.13	0
U2	1	0.13	0.33
U2_M	1	0.13	0
4TH	1	0.13	0
4TH_U1	1	0.13	0.33

Figure 2.7: Bar charts showcase the comparison between Predicted and Known counts of Perturbagens influencing a range of Kinases. The dataset is derived from the 'Stringent' set of Observation facts. Panels A, B, and C present the results for Edelphosine, MK2206, and GDC0941 Perturbagens, respectively. For each perturbagen, tables provide mPPV (modified Positive Predictive Value) and TPR (True Positive Rate) metric values for every evaluated combination of rules studied.

inhibitor capable of efficiently targeting all three isoforms (Hirai et al. 2010). Also present are graphs for Edelfosine, one of the perturbagens with no specific known targets that has been ascribed an amount of potential kinase targets, varying in number with combinations of subgoals.

2.4 Discussion

From the kinase perspective of predicted counts of inhibitors, Figures 2.5, 2.6, two main conclusions can be drawn. Firstly, regarding the increasing "logical complexity" of the "inhibited" *Rule*. The hypothesis was that the number of predicted inhibitors will decrease with the increasing logical "complexity" of the *rule*. The trend visible in the bar charts of both figures supports this. Additionally, via the inclusion of most *rule* combinations, the effect of taking into account colocalisation *facts* can be noted. Notably, in Figure 2.5 across the selected kinases, both versions of the "uniqueness" *rule*, in combination solely with the basal Zero iteration seem to have little effect on the total count of predicted inhibitors.

When developing the 3rd iteration of the "Inhibited Rule", which includes colocalisation and v2 of the uniqueness rules, it was hypothesised that the count of total predicted perturbagens per kinase would increase. This iteration of the rule was more logically rigorous as it involved the second version of the uniqueness rule, however this version was more lenient with respect to the number of phosphosites it would let through.

The effect of a relatively more stringent log2 fold change cut-off on the count of predicted inhibitor kinase relations is also evident from Figures 2.5, 2.6. Additional support to it comes from the tables in both figures for the mPPV metric. Values across all rule combinations based on Lenient observations facts are lower, a trend that suggest prediction confidence increases with cut-off stringency.

From the Perturbagen perspective of predicted counts of targets, similar conclusions can be drawn. However, of note is the consistent appearance of Edelfosine as a high number of targets perturbagen. This is noted across all rule combinations, as seen in Figure 2.7. This is of particular interest, given the fact that no specific mechanism of action has been reported for Edelfosine. Targets in glycerophospholipid metabolism as well as the Fas receptor (CD95) have been reported (Consuelo & Faustino 2002). Interestingly, Edelfosine features in the top 10, consistently across *rule* combinations, for highest number of predicted targets. In the rule iterations it fell below (Zero, U2, and 4TH_U1), it's still within the highest 20.

This page is intentionally left blank

Chapter 3

Application of probabilistic reasoning to cell signalling research

3.1 Introduction

In the previous chapter, we explored the application of logic programming, which is a clausal form of first order logic, to build a model for addressing our research questions. As has been shown, the results and predictions were promising however they failed to make validation worthy predictions and included a relatively high number of false positives and negatives.

The need therefore arose to enhance, in a manner, our model with a quantifiable level of belief able to capture the uncertain nature of kinase-kinase and kinase-protein interactions. This uncertainty can be attributed to imperfections in the data acquisition phosphoproteomics, contradictory or incomplete background information and the reality that a brittle yet abstract constrained rule can hold water but with notable leaks. These imperfections are not unique to our chosen research area of cell signalling research. Indeed, imperfect data and uncertain prior knowledge are common across all biological research because of many immeasurable or uncontrollable factors, leading to emergent stochasticity in our observations.

Another way to view this is via the framing of the qualification problem. As introduced by (Mccarthy 1986) it states that there will always be additional requirements or conditions to be added and/or satisfied when attempting to model the successful outcome of an action. Within this framework, issues with lack of knowledge affect both our ability to form rules as well as applying the rules proficiently. Therefore, including aspects of probability theory as part of our approach will allow us to better capture the uncertainty inherent in the protein based interactions we are trying to model. Our approach does not fall under fuzzy logic as probabilities used in the following (probabilistic) logic program represent degrees of belief rather than degrees of truth, as is the case in fuzzy logic (Cintula et al. 2021). Furthermore, it will become clear later in this chapter that our approach does not constitute a formal statistical treatment, as the measures of uncertainty in the input data are not all probabilities. The output of our analysis cannot therefore be interpreted as the probability of specific facts being true, but it is sufficient to rank facts according to how likely they are to be true. This is sufficient for our purpose as we seek to rank hypotheses for subsequent laboratory validation.

Probabilistic logic programming is an analytical and modelling approach that combines logic programming languages (in our case Prolog) with probabilistic representations and lies within the broad field of AI. Its main aim is to overcome the limitation of a rigid First Order Logic model by augmenting Logic Programming with probabilities. In the context of probabilistic logic programming, learning of weights is achieved through gradient-based approaches and associating these weights to *facts* or *clauses* (as defined in a logic program) (Sato & Kameya 2011). This is directly applicable to our project, as it provides a way to more closely and accurately model perturbagen kinase interactions and predict novel ones.

The aim of this chapter is to revisit the scientific question tackled in the previous chapter, but this time using a reasoning framework that take uncertainty into account.

3.2 Proof of concept and toy database application

3.2.1 Building a proof of concept or toy model

As before, our first step was to build a simplified representation or toy interaction model. In the case of building a toy model, within the context of probabilistic logic programming we have chosen to augment our already set model by making use of the relevant semantics. Annotating our *facts* with values, rather than directly relating to probabilities, are considered degrees of belief in the query outputs.

Employing the same methodological approach as in Chapter 2, our probabilistic toy model takes the form that can be seen in Figure 3.1. As a structure it is identical to the one in the previous chapter. The major addition to the model and by extension the methodology is the association of probabilities with individual *facts*. Extending from above w.r.t. probabilistic and fuzzy logic, these associations stem from a frequentist approach i.e. they are directly related to measurements or are derived from measurements with methodologies that preserve the notion of probabilities.

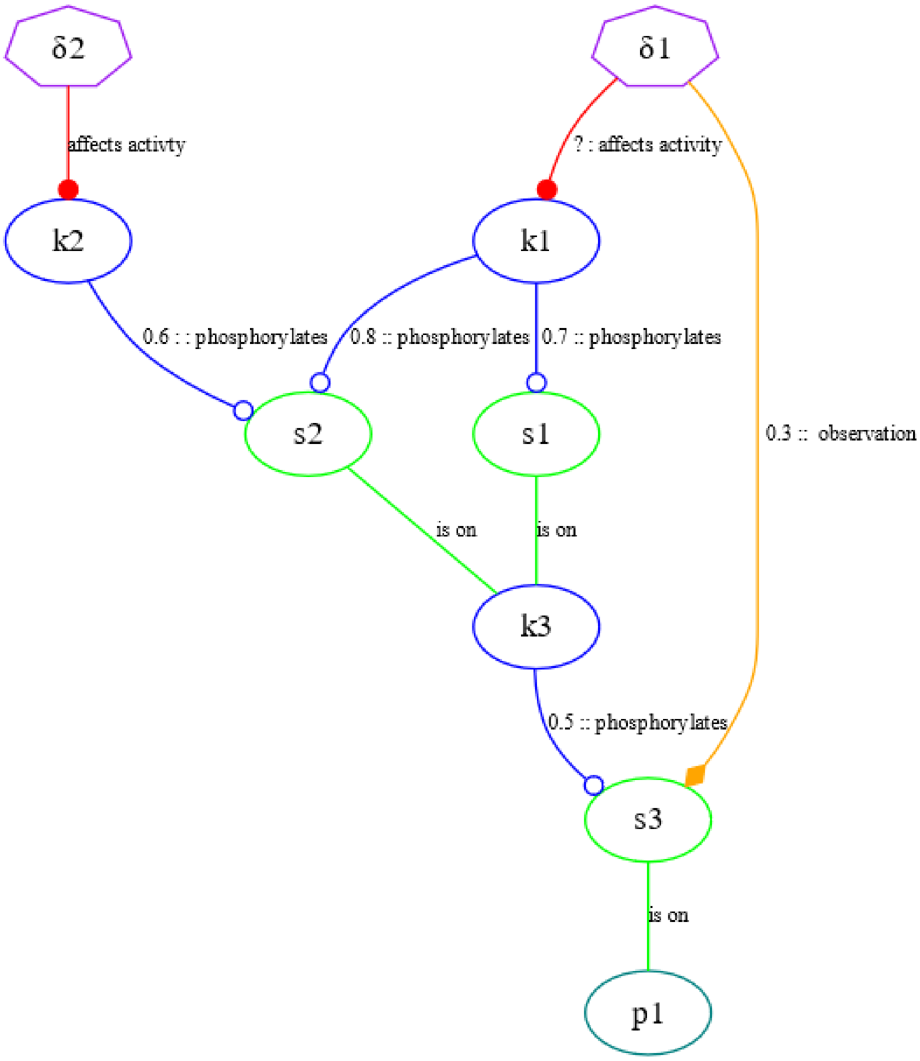


Figure 3.1: This graphical representation of the probabilistic toy model is identical in structure with Figure 2.1. Specific edges however are annotated with their corresponding Degree of Belief. In this way the transferability of a ground logic program between the Prolog and ProbLog implementation can be seen. Specifically, and as can be seen in the list of *facts* within the text, the annotations on the edges capture the epistemic uncertainty in the kinase phosphosite associations as well as the mix of alleatoric and epistemic uncertainty that characterises the process of Mass Spectrometry based data acquisition and related post processing.

As mentioned above our toy model was augmented into a probabilistic toy model. This was implemented in a straightforward manner as the difference between Prolog and ProbLog notation is the addition of the operator `::` to individual facts. Therefore the *facts* of our toy model from the previous chapter have taken the following form:

```
kinase(k1).
kinase(k2).
kinase(k3).
phosphosite(s1).
phosphosite(s2).
phosphosite(s3).
perturbation(delta1).
perturbation(delta2).
protein(p1).
0.7 :: knowntarget(kinase(k1), phosphosite(s1)).
0.8 :: knowntarget(kinase(k1), phosphosite(s2)).
0.6 :: knowntarget(kinase(k2), phosphosite(s2)).
0.5 :: knowntarget(kinase(k3), phosphosite(s3)).
ison(phosphosite(s1), kinase(k3)).
ison(phosphosite(s2), kinase(k3)).
ison(phosphosite(s3), protein(p1).
0.3 :: perturbs(perturbation(delta1), phosphosite(s3),
    occupancy(down)).
occupancy(up).
occupancy(down).
occupancy(unchanged).
cell_line(c1).
```

Taking, `0.7 :: knowntarget(kinase(k1), phosphosite(s2)).` as an example, this indicates that the *fact* phosphosite **s2** is a known target of kinase **k1** with probability of 0.7 of being `TRUE` and 1-0.7=0.3 of being `FALSE`. Within the context of our probabilistic toy model this can thought of as representing the level or degree of belief we have in the accuracy of this information. In general such associations are mined from literature and are contained in databases which associate with them a *score*. This usually reflects the amount, quality and consistency of evidence backing such an association. As can be noted, not all *facts* have an associated probability with them. In these cases the probability is 1.0 and therefore does not need to be explicitly specified for the *facts* themselves. The *rules* maintain their notation as this is also shared between Prolog and ProbLog. ProbLog also supports probabilities in the head of clauses or rules, however this notation was not used in our probabilistic toy model or probabilistic Logic Program. Associating a probability with a *rule* defines the likelihood that the rule is `TRUE`. In our implementation the *rules* are assumed to be `TRUE` as their aim is to capture the degree of belief associated with their ground instances and therefore are not preset in this implementation.

As with the toy model in the Prolog implementation, the probabilistic Toy Model changed to reflect the iterations of the rules. Specifically the *facts* base was altered to also reflect the aforementioned ProbLog weights/probabilities inclusive syntax. Taking for example the sub cellular location *facts* they took the following form:

```
0.8 :: plocation(protein(p1), subclocation(loc1),
   cell_line(c1)).
0.5 :: klocation(kinase(k1), subclocation(loc1),
   cell_line(c1)).
0.5 :: klocation(kinase(k2), subclocation(loc1),
   cell_line(c1)).
```

For the subcellular location of protein **p1**, **loc1**, the degree of belief we have associated with it is 0.8. In essence this means that the probability of “finding” the protein at that location is 0.8. What is also implied is that the probability of the location where one finds said protein (**p1**) not being **loc1** is 0.2. Accordingly when trying to implement the 4th iteration of our main *rule* (n.b. the 4th iteration includes a *predicate* with goals that succeed if the majority of a kinase’s known targets are of occupancy *down* vs those with occupancy *up*) in ProbLog notation, further *facts* were added and the probabilistic toy model took the form shown in: Figure 3.2.

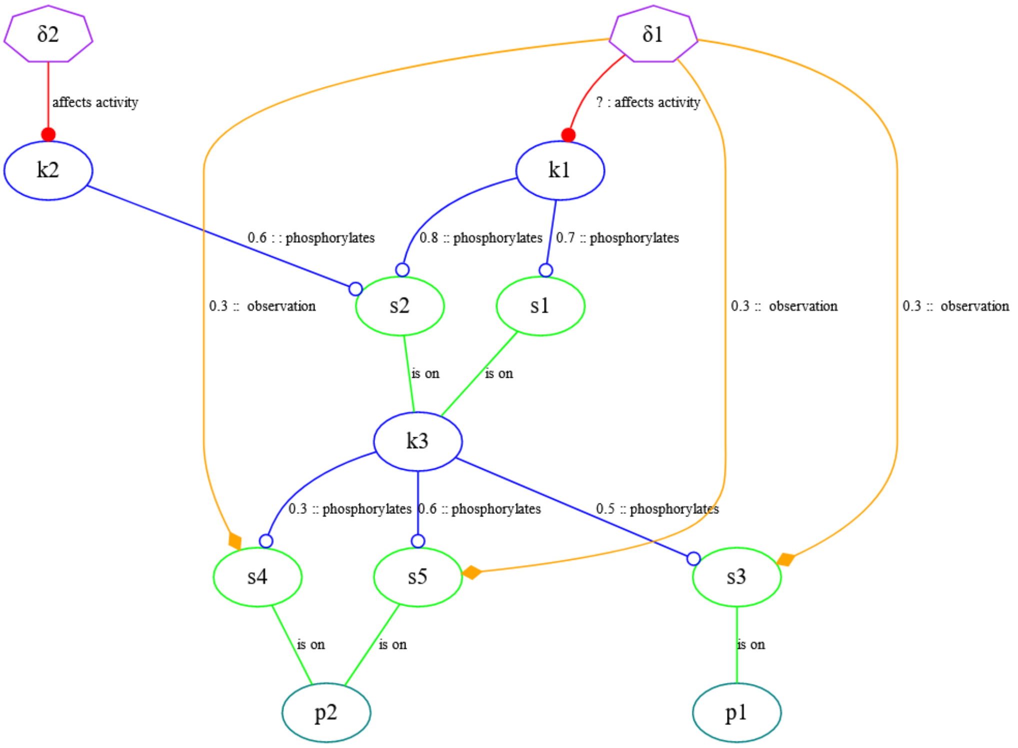


Figure 3.2: Network representation of update Toy Model version. Includes two more phosphosites **s4** and **s5** and relevant degree of belief annotations. Additionally, observational *facts* were added relating to how perturbation δ_2 affects these two phosphosites. As with the toy model in the Prolog implementation, these new *facts* are as well as the accompanying degrees of belief are intended to look at the overall effect on a kinase specific signalling network following a perturbation

Our main aim remained the investigation of whether a given perturbation affects a kinase by confirming known associations as well as establishing novel ones. By using the ProbLog pipeline (specifically ProbLog2 (Dries et al. 2015)) we are able to reason over the ground logic program while extrapolating with degrees of belief. A visualisation of the pipeline used for the probabilistic Toy Model and for the subsequent extension to our overall data can be seen in Figure 3.3. One of the main differences compared to the toy model was how the probabilistic Toy Model was deployed and interacted with. While in the toy model version of the previous chapter the queries were mainly posed via the interactive SWI-Prolog environment, for the ProbLog pipeline individual queries are specified within the probabilistic toy model (later also in the full probabilistic Logic Program). This allows for a tighter control of the output format as well as making use of the different modalities of ProbLog, namely "explain" and "ground" (explained further below).

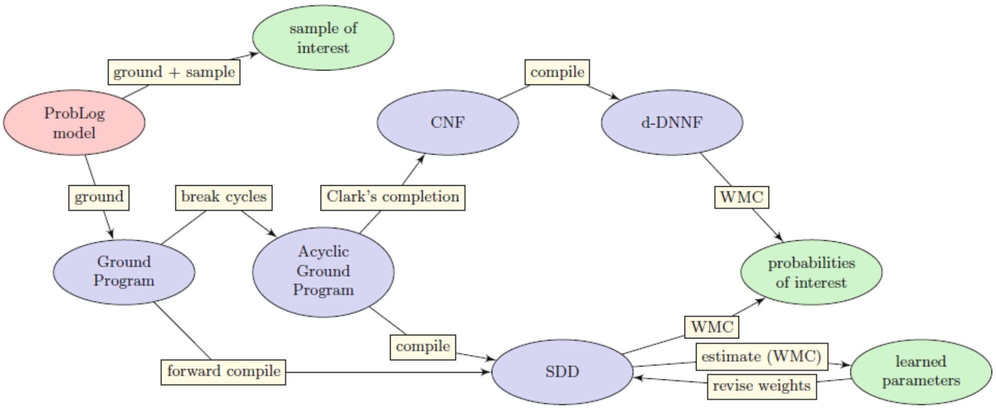


Figure 3.3: ProbLog pipeline for converting a weighted logic program to a formula in propositional logic. Step 1 represents the grounding of the ProbLog model using a Prolog-based grounder to create a ground program that may contain cycles. Step 2 represents the conversion of the ground program to a formula in propositional logic, which involves handling cycles. Different options are available for this conversion, as shown in Step 3. Bottom route represents the use of forward compilation to create sentential decision diagrams (SDDs). Top-most route represents the removal of cycles from the ground program, followed by either the transformation of the resulting acyclic ground program into conjunctive normal form and compilation into a d-DNNF (Top-most route continued) or compilation directly into a SDD (Bottom route). In Step 4 both of these normal forms, support efficient weighted model counting to obtain the final probabilities of interest. Graphic representation taken from (Dries et al. 2015)

Starting from the above list of *facts* and using the rule below (n.b. syntax is identical to Prolog implementation), the probabilistic toy model also includes the query (7, below):

```
1 doesDinhibitKinC2(perturbation(D), kinase(K),
     cell_line(C) :-
2      known_target(kinase(K), phosphosite(S)),
3      (is_on(phosphosite(S),protein(P)) ;
4      is_on(phosphosite(S), kinase(K))),
5      colocalistaion(protein(P), kinase(K)).
6
7 query(doesDinhibitKinC2(perturbation(D), kinase(K),
     cell_line(c1))).
```

Once the probabilistic Toy Model containing script is loaded on the Python based ProbLog environment, a Prolog-like grounder is used to ground the weighted model, i.e. pick all associated *facts* that can satisfy the variables defined in the above query (and their subsequent instantiations in the *sub-goals*). For this specific query a visualisation of this grounded program can be seen in Figure 3.4 (obtained from the "ground" mode of ProbLog2). The following parts of the pipeline take the ground model, and through a transformation step turn it into its conjunctive normal form i.e. a collection of ANDs ($\wedge$) . This is then compiled into a d-DNNF (deterministic, decomposable negation normal form, which can be visualised also by removing the nodes containing probabilities in Figure 3.4) by the DSHARP compiler (Muise et al. 2012). This is a formalism that allows for efficient and accurate weighted model counting (WMC), which is the final step (detailed description of d-DNNF related theory, semantics and how they enable weighted model counting to take place in non polynomial time can be found in (Darwiche 2002) and in-depth (Darwiche 2009*b*) & (Darwiche 2009*a*)). The WMC takes place with this compiled form in order to obtain the final probabilities of the query.

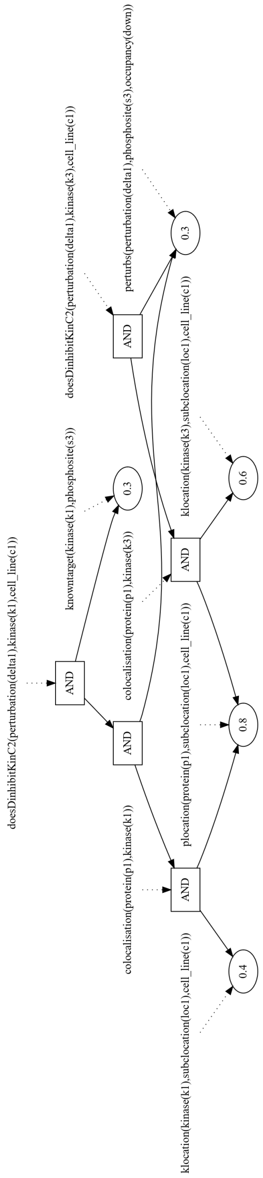

doesDinhibitKinC2(perturbation(delta1),kinase(k1),cell_line(c1))
AND
knowntarget(kinase(k1),phosphosite(s3))
0.3
AND
colocalisation(protein(p1),kinase(k3))
colocalisation(protein(p1),kinase(k1))
doesDinhibitKinC2(perturbation(delta1),kinase(k3),cell_line(c1))
AND
perturbs(perturbation(delta1),phosphosite(s3),occupancy(down))
0.3
AND
AND
klocation(kinase(k3),subcelocation(loc1),cell_line(c1))
0.6
plocation(protein(p1),subcelocation(loc1),cell_line(c1))
0.8
klocation(kinase(k1),subcelocation(loc1),cell_line(c1))
0.4

Figure 3.4: This figure contains the complete graphical representation of a ground logic program before cycle breaking. In this case the query posed was `doesD-inhibitKinC2(perturbation(D), kinase(K), cell_line(c1).` Note that the topmost nodes are the open ended variables D and K substituted with the appropriate *facts* from within the probabilistic Toy Model list of *facts*. Furthermore it contains the grounded individual predicate colocalisation/2 as well as other called *facts* such as knowntarget/2, perturbs/3 and ison/2. The degrees of belief are represented as end nodes in this visualisation

The ProbLog pipeline when run as a command line interface offers a number of operations, amongst these are `explain` and `ground` . The latter was used to generate Figure 3.4 as it outputs the ground probabilistic Logic Program in a GraphViz representation format. The former, `explain`, offers insight into how the probabilities of a given query are computed for a probabilistic Logic Program. Part of its output is a list of proofs with their associated probability. From these the success probability is calculated by simply taking the sum of individual probabilities associated with each proof. In the case of the above example query the output looks is:

```
Proofs
------
doesDinhibitKinC2(perturbation(delta1), kinase(k3),
           cell_line(c1)) :-
    knowntarget(kinase(k3), phosphosite(s3)),
    perturbs(perturbation(delta1), phosphosite(s3),
               occupancy(down)),
    ison(phosphosite(s3),protein(p1)),
    plocation(protein(p1), subclocation(loc1), cell_line
       (c1)),
    klocation(kinase(k3), subclocation(loc1), cell_line(
       c1)).
    P=0.072

Probabilities
-------------
doesDinhibitKinC2(perturbation(delta1), kinase(k3),
        cell_line(c1)): 0.072
```

From the above we can visually confirm that our probabilistic Toy Model is taking into account the appropriate *facts* as part of satisfying the called *rules*. It is also computing the probabilities accordingly as described above (Figure 3.3 and in text).

3.3 Methods

3.3.1 Probabilistic annotated experimental and background knowledge *facts*

An iterative approach was followed when transitioning from the probabilistic Toy Model to the fully fledged probabilistic Logic Program similarly to the Prolog implementation of Chapter 2. A number of different steps were taken in order to augment the *facts* with probabilities.

Initially the Python scripts that produced the lists of *facts* were amended in order to output the appropriate format for ProbLog. By setting the probability or degree of belief before the `::` for each one of the *facts* (where applicable) they were redefined as probabilistic *facts*.

Before this could take place the issue of lack of extensive overlap between our observational *facts* and background knowledge *facts* needed to be addressed. Starting with our observational/experimental data transcription into *facts* a new approach was followed with respect to how we arrived at log2-fold change and p-values. The latter being used for the degree of belief in our probabilistic *fact* format.

The source of the experimental data remains the same (Hijazi et al. 2020). To reiterate, this study contains phosphopeptide abundance data for three cell lines treated with 61 kinase inhibitors in biological and technical duplicates. As with the Prolog implementation the main focus remains the MCF-7 cell line. Using an in-house pipeline (detailed description can be found in the Appendix A.1) a total of 140,424 Perturbagen - Phosphosite observations (covering 14,488 unique phosphosites) were derived. The new calculated p-values were used as the probabilistic annotation for these *facts*.

Looking at Table 3.1 outlining the number of *facts* that make up our Knowledge Base, the amount of relevant *facts* changed for the implementation of the probabilistic Logic Program. This is true for all of them apart from the subcellular locations of proteins and kinases. For these the amount of *facts* did not change, simply the format. The degree of belief metric used here was the likelihood associated with *finding* a specific protein or kinase in a given subcellular location, namely cytosol, nucleus, mitochondria, secretory system or in an undefined location. We set the minimum level of confidence or belief for named locations at 0.5, and for undefined or unclassified locations at 0.1. The lower limit for undefined locations was chosen because it was the lowest score associated with proteins and kinases in the study (Orre et al. 2019) when it was not specifically clear which subcellular location had been assigned.

One of the databases assessed and part of which was included into our knowledge base was OmniPath (Türei et al. 2020). This database represents a collection of over 100 individual resources organised into sections of particular interest. These include signalling networks, protein complexes, protein annotations relating to phenotype, function, subcellular localisation as well as a section on other intercellular signalling. Similarly to above we selected and transcribed into *facts* annotations relating to kinase-protein interactions and whether a specific phosphosite can be found on a kinase or protein.

Using OmniPath as a source offered greater coverage of kinase-substrate relationships as our *facts* are extracted from a closely curated collation of databases that includes UniProt, which was the sole source of these associations for the ProLog implementation. However, no metric was used as a degree of belief annotation (for the majority of *facts*) for two reasons. Firstly, it was not immediately available as part of the Omnipath db and secondly, as will be described further in the Results section, it would lead to prohibitively long computational times for even single perturbagen-kinase queries at a time.

Facts List	Source	Size (Number of Facts)
Kinase - Substrate Associatons	OmniPath	24 688
Inhibitor - Kinase Association	Klaeger, EBDT and vendors	722
Phosphosite presence on Protein	UNIPROT	18,795
Protein Subcellular locations	SubCellBarCode	10,353
		Total = 54,558

Table 3.1: Table listing the title, sources and associated number of *Facts* that make up the Background Knowledge section of the logic model. Specific sources referenced Klaeger (Brand et al. 2015), Omnipath (Türei et al. 2020), SubCellBarCode (Orre et al. 2019)

3.3.2 Results

As with the Prolog implementation, so with the implementation described in this chapter, the collection of results is a multistage process. However, ProbLog2 (as well as ProbLog) as a pipeline can be called and used via the command line and is written as a Python package. This enables access to the different modes of ProbLog (such as Explain and Ground, mentioned in the probabilistic Toy Model section) and the deployment of the probabilistic Logic Program in an Integrated Development Environment (IDE).

When using ProbLog via an IDE, or in the command line, the queries are specified as part of the model as is the output file and the format (choices include “prolog” or “text” with the difference being whether each successful query output is in a new line or as one). Appropriate *rule* and query structuring in combination with the “prolog” format choice for the output of the ProbLog pipeline can create .pl format files that contain the outputs of said queries. These, in turn and following appropriate verification can themselves be used as *fact* containing files to be included in the Knowledge base and make up part of the probabilistic Logic Program. Following this the results file is parsed with a Python script that includes ReGex search patterns and outputs a .csv format file. These .csv files are then read into Rstudio in order to produce the following graphs using ggplot and plotly packages available via CRAN.

3.3.2.1 Rule iterations 1-4 and their overall accuracy

Initially, the accuracy of the probabilistic Logic Program across the four iterations of the doesDinhibitKinC *rule*. In Figures 3.5, 3.7, and 3.10 heatmaps are presented that show the degree of belief across all perturbagen-kinase combinations that we are able to provide a query result for. That is to say, with successive iterations of the rule, the logical constraints to be satisfied increase in number and complexity. Therefore, we can provide more stringent predictions for a decreasing amount of kinase-perturbagen combinations with increasing complexity.

With the first *rule* iteration (knowntarget/2 + perturbation/5 predicates), as can be seen in Figure 3.5 our queries cover the entirety of the potential perturbagen-kinase combinations. Entirely void tiles represent queries which could not be answered at all, either due to lack of background knowledge or no overlap between background knowledge and observation *facts*. Dots in the tiles are known perturbagen-kinase associations. Sources, as with the Logic Program implementation, include (Brand et al. 2015), (Hijazi et al. 2020) amongst others. Coverage of known associations might initially appear weak, however given the lack of significant overlap between our observational *facts* and background knowledge base, we are achieving a high degree of belief for the majority of the known associations (darker colour, higher Degree of Belief) for which this is computable.

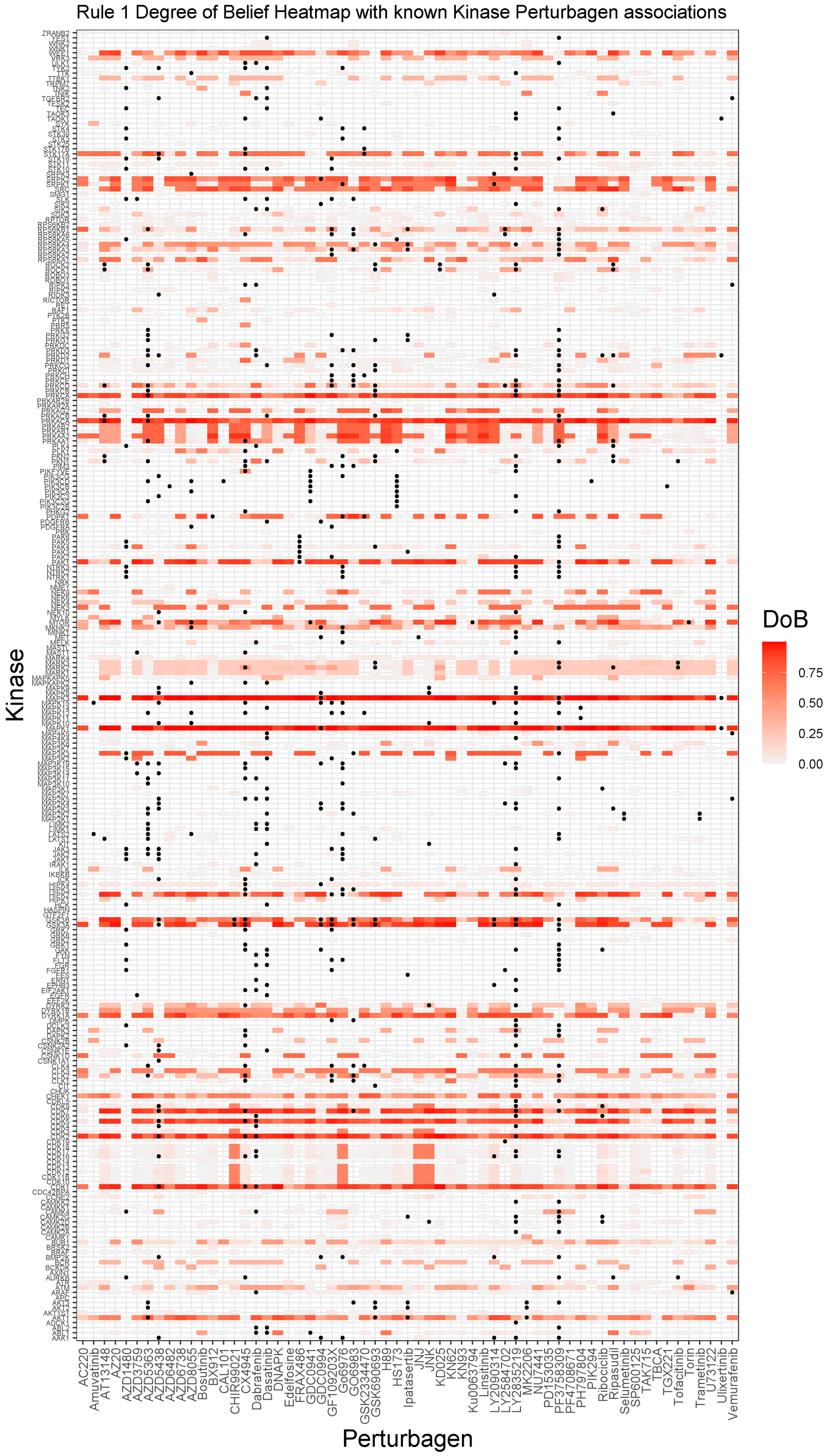


Figure 3.5: Heatmap showing the Degree of Belief from iteration 1 of the doesDinhibitKinC/2 *rule*. Kinases on the y-axis with perturbagens on the x-axis. Darker shades of red correspond toa higher degree of belief associated with the Kinase-Perturbagen query posed to the *rule* iteration.Values filtered for DoB $\geq$ 4.81E-09 (lowest Degree of Belief associated with a known perturbagen-kinase pair) which reduces overall coverage (TP) vs known associations (black dots) but also reduces erroneous predictions (FP).

It is apparent that a for a number of kinases (eg. MAPK1, CDK1, PRKACA etc.) we are able to offer predictions and confirmations regarding their response to the majority of the 61 inhibitors in our panel. As the only major logical hurdle to be overcome in the context of the first iteration of the *rule*, is the 'knowntarget/2' association. Kinases for which we have extensive knowledge are therefore over-represented in this manner, in the first two rule iterations. Looking at the correlation (0.53 and 0.579) in Figure 3.6 for R1 and R2 (*rule* iterations 1 and 2) respectively, it indicates a moderate positive association between the number of known kinase-substrate associations and kinase-perturbagen predictions (n.b. known ones included within this calculation).

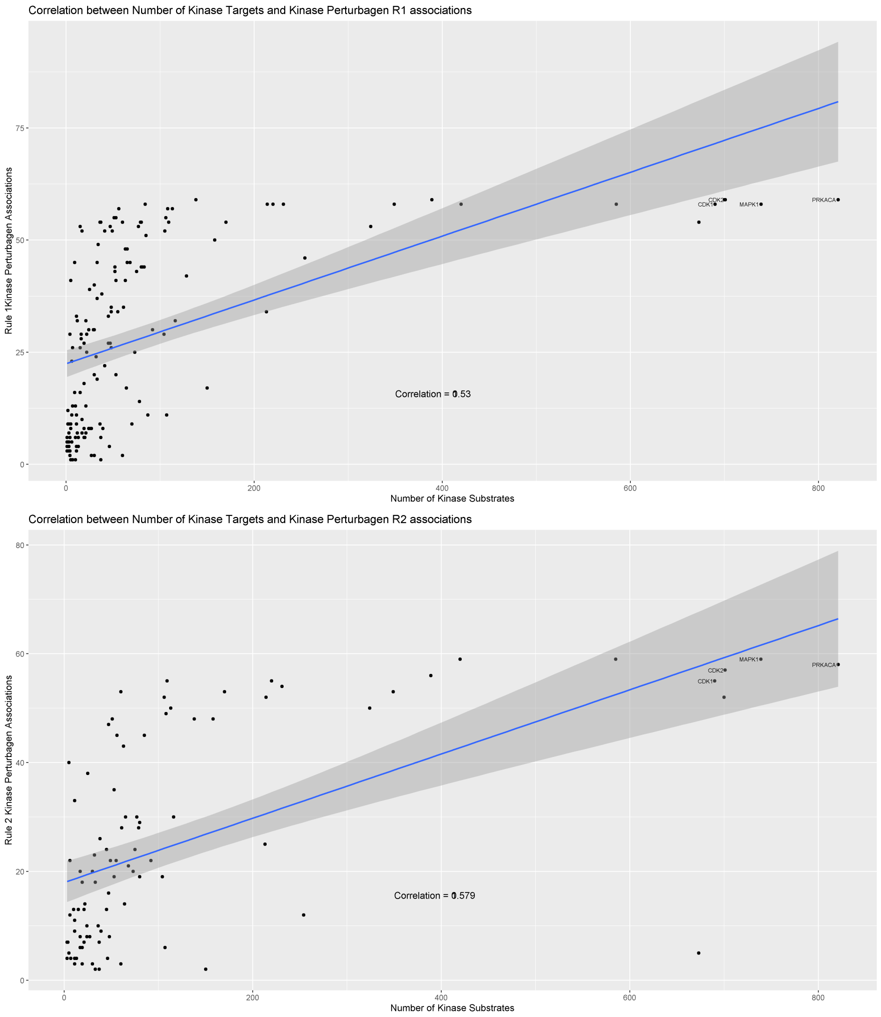


Figure 3.6: Figure includes two scatter plots with number of known kinase phosphosite targets on the x-axis and number of kinase-perturbagen associations, predicted by rule iterations 1 and 2. The correlation, 0.53 and 0.569 (for Iteration 1 and 2, respectively) indicating a relatively moderate positive correlation. Highlighted are CDK1, CDK2, MAPK1 and PRKACA which have the highest number of known targets as well as highest number of predictions made.

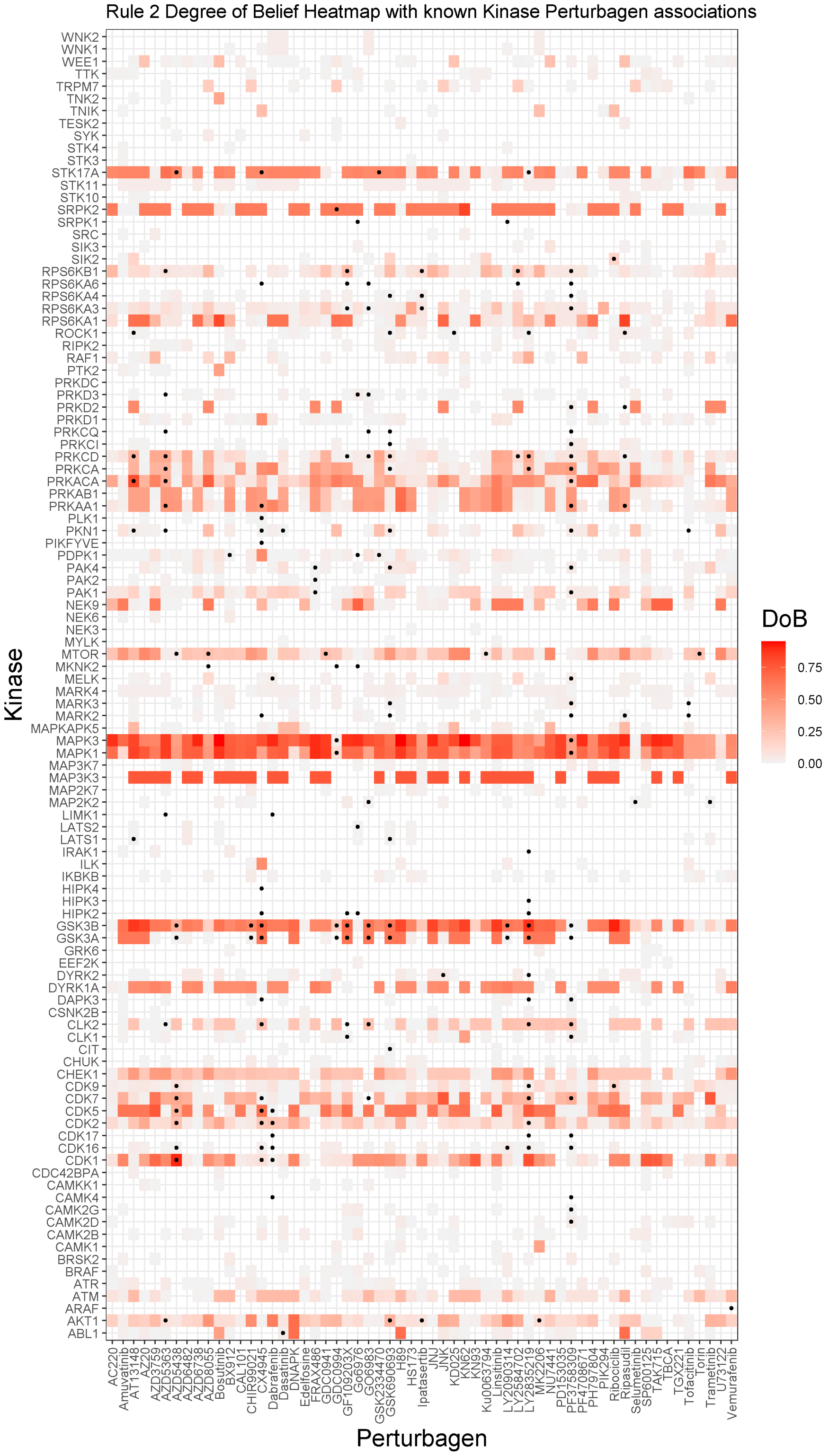


Figure 3.7: Heatmap showing the degree of belief for each combination of kinases (on the y-axis) and perturbagens (on the x-axis) after the second iteration of the doesDinhibitKinC2/2 rule. The degree of belief is a measure of the strength of the association between the kinase and perturbagen, with darker shades of red corresponding to higher degrees of belief. Values for the degree of belief are filtered to show values greater than or equal to 3.99E-09, which reduces the overall coverage of the heatmap (i.e. the number of kinase-perturbagen pairs that are included) but also reduces the number of erroneous predictions. Known associations between kinases and perturbagens are represented by black dots on the heatmap.

However, when looking at the same correlation but for known kinase-perturbagen associations (Figure 3.8) the value of 0.098 indicates a negligible relationship between the two. Here again however, it must be noted that we are able to calculate this using data for 165 out of 583 Kinases. That is the overlap between our known Perturbagen-Kinase and Kinase-Target associations w.r.t. the Kinases. In Figure 3.7 the trend continues, with more of the known perturbagen kinase associations (for which we can offer predictions) being capture by the Degree of Belief based output.

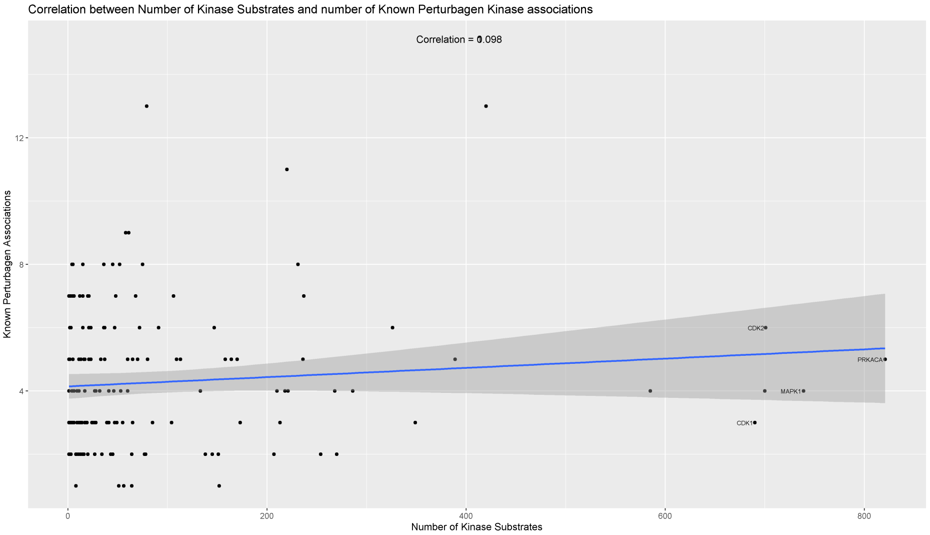


Figure 3.8: Scatter plot with correlation between number of known inhibitors and number of known substrates for a given kinase within the knowledge base. 0.098 Correlation reveals very weak positive association between the two sets of *facts* in our background knowledge base.

For the third iteration of the *rule* however there is a marked reduction in the total number of Kinase-Perturbagen combinations we can offer predictions as shown in Figure 3.10. However, for the combinations for which we can offer a scored, rankable association (both in the case of known and novel associations) our confidence in their accuracy is increased. Similarly to before, the correlation between number of known kinase targets and number of perturbagen kinase predictions remains moderately positive (Figure 3.9) with nearly identical distribution and correlation for 4th iteration *rule* query outputs. Finally, differences in coverage and predictions between the 3rd and 4th iterations of the *rule* are negligible; with the latter offering approximately the same number of known perturbagen kinase associations without increased Degree of Belief but with marked increase in computational time and resources

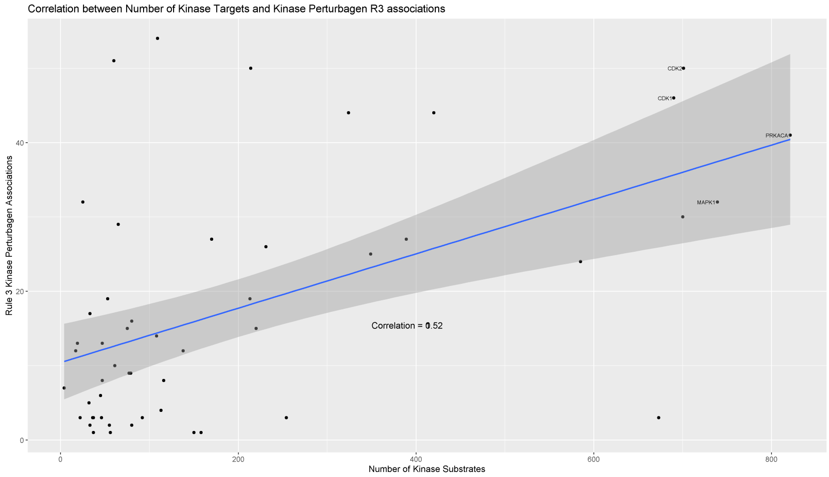


Figure 3.9: For the third iteration the trend continues with a similar correlation of 0.52 which is however slightly less than previous iterations. Similarly the kinases for which we are able to make the most predictions are the ones with the highest number of known phosphosite targets.

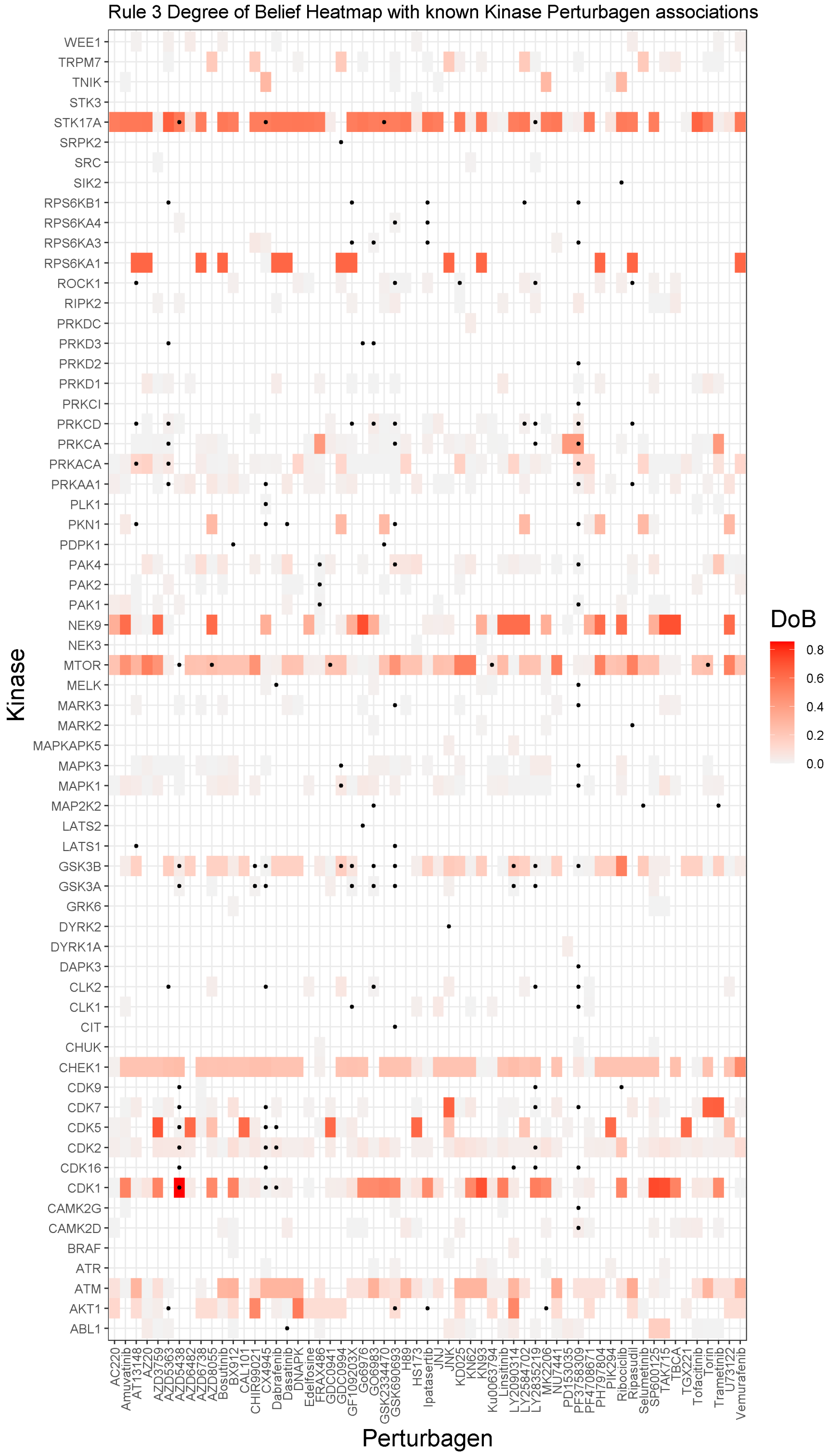


Figure 3.10: Heatmap showing the degree of belief for each combination of kinases (on the y-axis) and perturbagens (on the x-axis) after the second iteration of the doesDinhibitKinC3/2 rule. Darker shades of red indicating a higher degree of belief. Values for the degree of belief are filtered to only include values greater than or equal to 2.22E-08 (lowest degree of belief attributed to known perturbagen-kinase association), which reduces the overall coverage of the heatmap but also reduces the number of erroneous predictions. Known associations between kinases and perturbagens are marked with black dots on the heatmap

3.3.2.2 Confirming known and predicting novel associations

From the above, it was decided that particular focus would be given on the results of the 3rd iteration of the *rule*. Through its implementation we were able to extract both known, and potentially novel kinase-perturbagen associations. Exploring the results we will focus on a few associations that stand out for positive as well as negative reasons.

Picking a well studied kinase such as CDK1, the distributions of the Degree of Belief across 3 of the 4 iterations of the *rule* are bimodal Figure 3.11. With CDK1 having 690 known targets, and looking at the aforementioned moderate correlation between this number and Degree of Belief we expect the grounded instances of the 3rd iteration *rule* to be associated with a relatively high Degree of Belief. The highest ranked drug (by Degree of Belief) predicted to affect CDK1 is AZD5438 (Degree of Belief= 0.851) which within our knowledge base is registered as a vendor reported known association. In the remaining predictions we find both other perturbagens known to target CDK1, namely CX4945 and Dabrafenib. These are scored 0.001 and 0.022 while being ranked 39th and 44th out of 54.

Diving deeper into how ProbLog arrives at the aforementioned Degree of Belief we make use of the "Explain" mode. This offers a breakdown of individual proofs that are formulated as well as their associated Degree of Belief(in this case directly probabilities) and how these are in turn aggregated by summing to provide the final Degree of Belief metric. In general, within the context of a ProbLog implementation, a proof is a sequence of steps or inferences that are used to establish the truth or validity of a logical statement. Proofs are used to determine the probability of a given logical statement, based on the known facts and rules specified in the ProbLog program. Specifically for the AZD5438-CDK1 pair and the third iteration of the *rule*, we are focusing on which is the most influential proof (i.e. the combination *facts* and *rules* that most influence the outcome).

```
1.0 :: knowntarget(kinase(’CDK1’), phosphosite(’PKN2(
   S535)’)),
0.9868 :: klocation(kinase(’CDK1’), subclocation(’
   Cytosol’),
0.7262 :: perturbs(perturbation(’AZD5438’),
   phosphosite(’PKN2(S535)’), onprotein(’PKN2’),
   occupancy(down), direction(down)),
0.9916 :: plocation(protein(’PKN2’), subclocation(’
   Cytosol’).
Overall P=0.71060572
```

The second most influential proof takes into account the following *facts* (with associated probabilities):

```
0.9868 :: klocation(kinase(’CDK1’), subclocation(’
   Cytosol’),
1.0 :: knowntarget(kinase(’CDK1’),
   phosphosite(’NCKAP5L(S436)’)),
0.5076 :: perturbs(perturbation(’AZD5438’),
   phosphosite(’NCKAP5L(S436)’),
onprotein(’NCKAP5L’), occupancy(down), direction(down)),
\+perturbs(perturbation(’AZD5438’), phosphosite(’PKN2(
   S535)’), onprotein(’PKN2’), occupancy(down),
   direction(down)).
Overall P=0.13277925
```

By adding up the individual degrees of belief associated with the above two proofs we get approximately P = 0.8433 which when combined with a further 64 relatively marginal calculated degrees of belief we arrive to the aforementioned figure of of 0.815. Note the inclusion of \+ which corresponds to the negation of the perturbs *fact* found in the previous proof. The probability for this, used in the calculation of the “Overall P”, in our context Degree of Belief, is 1 - the probability reported above.

Looking at it from the perturbagen side, AZD5438 is a known inhibitor of CDK1, CDK2 and CDK9. Out of these three we rank CDK1, as discussed above, and CDK2 however CDK9 only makes it as far as the 2nd iteration of the *rule*. A further insight into Figure 3.12 shows the distribution of Degree

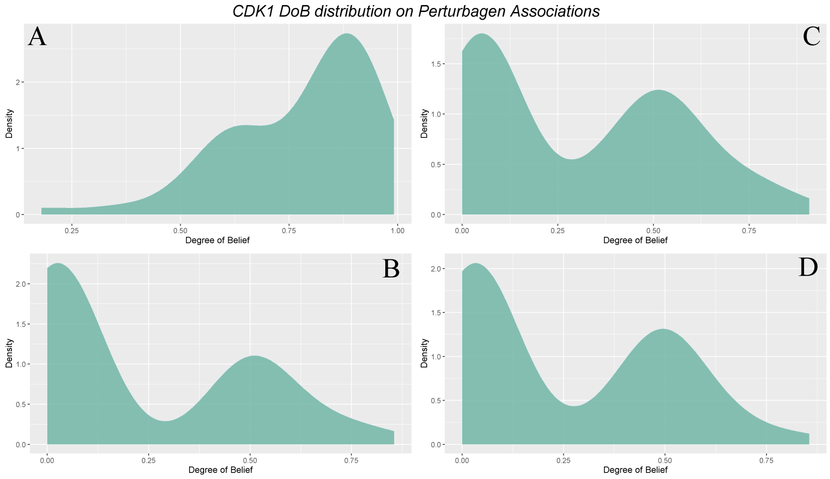


Figure 3.11: Figure includes four separate distributions for the degree of belief predictions for CDK1 kinase for all perturbagens. Panels refer to increasing rule iterations in order. A: Majority of predictions cluster close to 1 (Top 600 predictions between 0.8 and 0.99). B,C,D: Bimodal distribution around 0.5 with majority of predictions exhibiting a degree of belief close to 0.

of Beliefs for AZD5438 for all kinases we can offer a prediction for. Of note are the consistent high number of low value Degree of Belief. This value however is relative to the initial values set for the probabilistic facts and not a measure of likelihood in the traditional sense.

Out of the top 10 ranked (highest Degree of Belief) predicted perturbagen-kinase, CDK1 covers 4 of the (top 5) positions, while the remaining 6 include kinases NEK9, STK17A and CDK7. Specifically, NEK9 and STK17A are of interest as they attain this position (and maintain it for *rule* iteration 4) with a relatively low number of known phosphosite targets (61 for STK17A and 25 for NEK9).

STK17A has four inhibitor associations in our known Knowledge Base, via the discoverX assay. These are GSK233470, CX4945, LY2835219 and AZD5438 (in descending order of activity reduction potency, small range). In our ranking AZD5438 comes first with a Degree of Belief= 0.5663 closely followed by GSK2334470 (Degree of Belief= 0.5534), then CX4945 (Degree

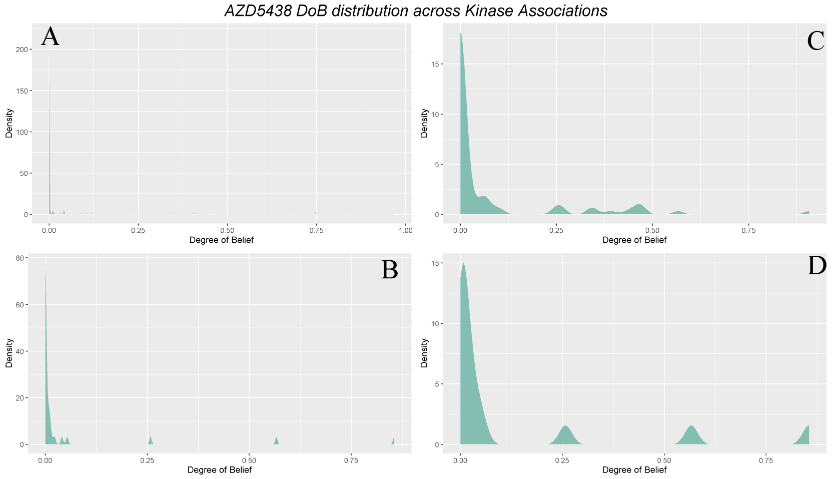


Figure 3.12: Figure includes four separate distributions for the degree of belief predictions for AZD5438 inhibitor for all kinases. Panels refer to increasing rule iterations in order. A-D: Majority of predictions cluster close to 1. C, D: Increase in the frequency of higher Degree of Belief predictions.

of Belief= 0.5454) and finally LY2835219 with Degree of Belief= 0.0154. Apart from LY2835219 our predictions are tightly grouped wrt to Degree of Belief, with the metric representing relative confidence in the prediction. All of the above predictions as well as others included in the results tables (further results lists can be found via the Github repo `https://github.com/Dudelder/Symbolic_AI_Opus`) constitute True Positives (for the known ones in our Knowledge Base), potential novel associations as well as a high number of false positives.

Identifying true and false negatives on the other hand is slightly trickier with this approach. An example would be perturbagen GDC0941 and kinase AAK1. In this case via the discoverX assay GDC0491 reduces the activity of AAK1 by less than 1% and is not predicted in the 3rd iteration of the *rule* (nor is it present in any of the other iterations). For AZD5438 a false negative that comes up and constitutes a glaring omission, is CDK9 which is a vendor verified association that we don't pick up with the 3rd iteration (however it is present in both preceding iterations).

3.4 Discussion

Starting with the first iteration of the rule in the context of the probabilistic logic program implementation with 14640 queries (240 kinases x 61 perturbagens) we were able to offer predictions for 4635 kinase perturbagen pairs. The associated degrees of belief, calculated using the ProbLog engine, ranged from 1.008383e-09 to 0.99. Out of the 4635 predictions 1404 had $\geq$ 0.5 degree of belief and the lowest degree of belief associated with a known kinase perturbagen association was 4.81E-09. This gave us approximately 3000 low confidence predictions within the context of the first iteration of the *rule*. From within our knowledge base of 731 known pairs we were able to cover 235 therefore offering 4400 potentially novel associations of varying degrees of confidence.

In the second iteration the trend continued with less associations present as only the ones for which we were able to offer a prediction from the previous iteration were carried over as a query for subsequent iteration. 4635 individual queries yielded 2365 associations with degrees of belief ranging from 3.99e-09 to 0.906. Within these 415 had a degree of belief $\geq$ 0.5. The overall coverage from within known associations droped from 235 to 180.

For iterations 3 and 4 of the *rule* the results were quite similar. This was mainly due to the fact that the grounded logic programs from which the degrees of belief were extrapolated using the ProbLog machine were similar. The main difference was in the number of queries posed in order to arrive to these results. For rule 3, the number of queries was 2365 as this was the number of associations from rule 2 that yielded a positive degree of belief prediction. Of these, 901 pairs had an above 0 degree of belief with 95 of these being over 0.5 For rule 4 on the other hand the number stood at 907 queries which yielded 903 pairs with a degree of belief $\geq$ 0.5. The number of associations with above 0.5 reduced slightly to 93.

The above results point to an increase of confidence in the decreased number of predictions being made. The degree of belief is not an absolute metric of strength in the association across all iterations therefore should be considered for each rule iteration separately. The overall number of pairs that we were able to offer predictions for decreased as the stringency of the *rule* iterations increased. Between the third and fourth iterations however the differences were marginal therefore the overall best performing rule iteration, considering trust in predictions and associated degrees of belief is the third iteration.

Within the predicted novel associations, of praticular interest are those that partain to lesser studied kinases. Specifically, NEK9 for which knockdown has been shown to reduce proliferation in p53 mutated cancer cells (Kurioka et al. 2014). Its top predicted inhibitors are Go6976, Tak715,TBCA and AZD8055. Alongside NEK9 and CDK1, the next most prevalent kinase with perturbagen associations, by degree of belief is STK17A. This has been associated with cisplatin resistance in p53 mutated cells (Mao et al. 2011). AZD5363, Tofacitinib and DNAPK were its top three predicted inhibitors.

Overall, our coverage of known perturbagen kinase associations is lacking. However, within this context, predictions were made with a significantly low overlap between experimentally described phosphosite *facts* and those that are within our background knowledge base. Specifically, even with the new set of computationally derived *facts*, from the same experimental data as were used in the Prolog chapter (chapter 2), the coverage was low. All above predictions were ultimately based on 1492 phosphosites for which we have known kinase targeting information. That is out of 24688 associations in omnipath and 13424 individual phosphosites in the observation *facts*. This poor overlap extends also to the perturbagen kinase level of associations within our knowledge base. 17 of the tested inhibitors have no known kinase targets (from three different sources that fed into our database, namely the Kuster study (Brand et al. 2015), vendor defined associations and discoverX assay results as reported in (Hijazi et al. 2020)). Notably this lack of overlap, was picked up by our predictions with AZ20 being predicted to target MTOR, high degree of belief as well as SP600125 targeting CDK1 with a degree of belief second in rank only to AZD5438. All these were known associations for these highly studied kinases.

This page is intentionally left blank

Chapter 4

Implementing methodology in *Saccharomyces cerevisiae* gene perturbation dataset

4.1 Introduction

The previous chapters focused on evaluation of symbolic AI in the context of phosphoproteomics data from cell lines treated with small molecule kinase inhibitors. The effects of these inhibitors on individual kinases formed part of the background knowledge for those studies, either as Boolean facts (in Chapter 2) or with associated degrees of belief (in Chapter 3). For this chapter the choice was made to step away from cancer cell lines and seek a different organism to serve as testing ground for our approach. *Saccharomyces cerevisiae*, brewer's yeast or budding yeast (named after the way it reproduces (Herskowitz 1988)) was chosen as the appropriate species. In theory this should pose less of a challenge than human cell lines due to yeast's smaller signalling network and the ability to reliably silence individual kinases via large scale gene knockout experiments.

As with any organism, so with *S. cerevisae* protein phosphorylation plays a crucial role in various signalling pathways that control cellular activities such as gene expression, protein synthesis, DNA replication, RNA processing, cell cycle regulation, metabolism, transport of vesicles and other organelles, response to environmental stress and nutrients, and cell differentiation. The specificities of kinases and substrates are largely determined by protein interactions, which are affected by phosphorylation (Ptacek et al. 2005).

It is also known that many of these signalling pathways are conserved across different species, so yeast can be used as a model organism to study phosphorylation networks. Yeast has 159 genes that encode protein kinases and phosphatases, and 136 of these have counterparts in humans. Therefore any insight in this model organism could potentially offer indications of mechanisms of action in the human kinome.

Ours is not the first application of symbolic AI in the study of yeast and its pathways more specifically. As mentioned earlier in the thesis, a major inspiration for applying techniques that fall under the symbolic framework came from the extensive work done by Prof R.King. Particularly, work of the type found in this study by A.Clare and R.King (Clare & King 2002) showed that it is possible to build a knowledge base via the use of a declarative logic programming language. The seminal work on Robot Scientist Project also offered insight in how Prolog can be used to model the metabolism of yeast and how deletion in associated genes can affect the processes associate (King et al. 2004). Further use of a Prolog based approach can be found in (Whelan & King 2008) where a Flux Balance Analysis model was combined with genome related information from the KEGG database to predict the growth patterns of yeast. More recently, network reconstruction as well as new structure hypotheses were made using ProbLog, with the aim of providing new insights to experimental biologists (Goncalves et al. 2014).

However, all of the above approaches have focused on the use of gene-based background knowledge and experimentally derived facts to reach their conclusions. These lack the granularity of a proteomics and even more so a phosphoproteomics based approach. Therefore, and continuing the work done in the previous chapters, we decided to apply our methodology to a similar and appropriate dataset, in the context of *S. cerevisae* as a model organism. Specifically, the aim was to see whether logic-programming based models could correctly identify which kinases had been impacted by a gene knockout, whether there was any compensatory effect from other kinases and which kinases were responsible such effects.

This chapter includes a brief description of the study from which we derived our data, the background knowledge databases of associations we used to populate our knowledge base as well as how we applied our methodology and subsequent results.

4.2 Methods

4.2.1 Toy Model development

The implementation of our methodology on the yeast dataset began in a similar fashion to the previous two (Prolog, ProbLog). Initially a toy model version was developed, with a few key differences. Firstly the perturbations were in this instance gene deletions and the question thus became whether silencing (or deletion) of a given kinase has actually worked. Within the context of the study chosen (to be described in detail in subsequent section), specific gene deletions were confirmed by either proteomics or PCR assays, therefore and due to the nature of gene deletions in yeast it was unnecessary to identify off targets or known targets of a given perturbation, as was in the case with small molecule inhibitors (Chapters 2 and 3).

In Figure 4.1 the overall structure of our yeast Toy Model can be seen. As with the previous Toy Models, edges represent either perturbations (deletions in this case), observational *facts*, presence of a phosphosite on a kinase and kinase phosphosite associations. Nodes represent either phosphosites, perturbations, kinases or other proteins.

To determine whether a kinase has been successfully silenced, development of the *rule* iterations began with the following:

```
doesSilofKwork(deletionOF((kinase(K))) :-
    phosphorylates(kinase(K),phosphosite(S),protein(P)),
    perturbs(deletionOF((kinase(K))), phosphosite(S),
       occupancy(down),_).
```

The *rule* `doesSilofKwork(deletionOF(kinase(K)))` determines the success of deleting a kinase **K**. It checks two conditions: the presence of a known phosphosite **S** on a protein **P** where **K** phosphorylates **S** on **P**, and confirmation from mass spectrometry that occupancy at **S** decreases after deleting **K**. When both conditions are met, the interpreter confirms the successful deletion of the gene encoding kinase **K** based on experimental evidence.

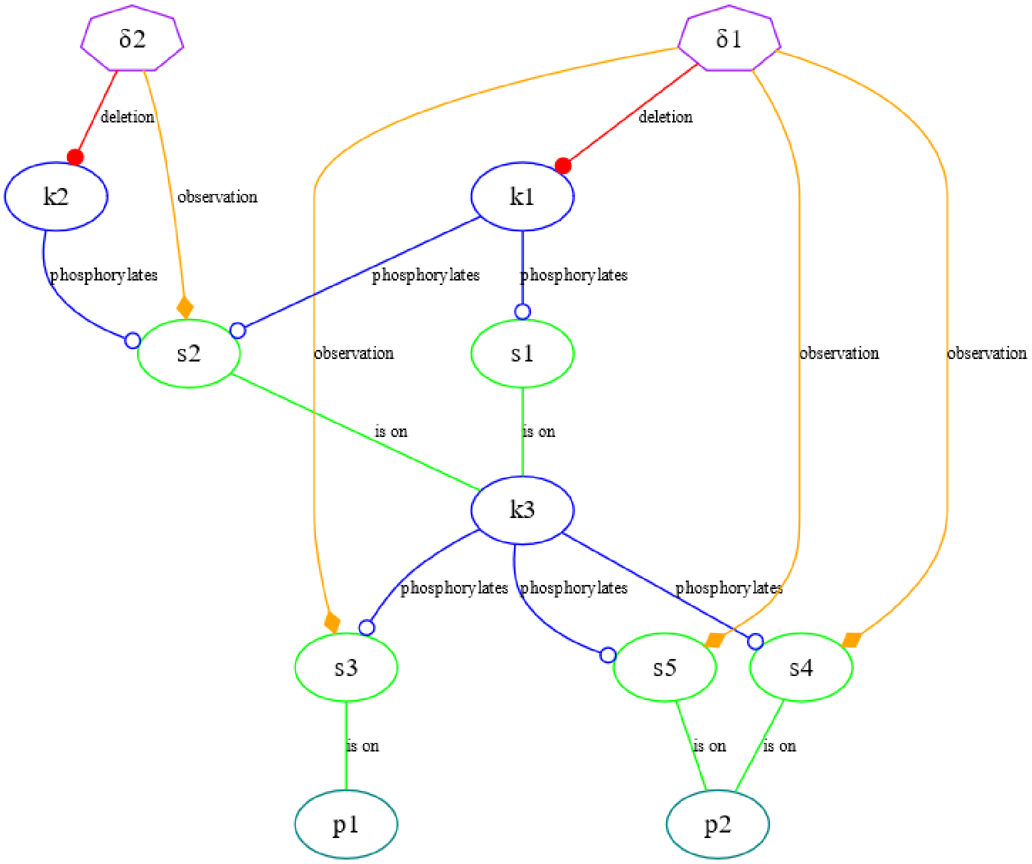


Figure 4.1: A Toy Model illustrates the minimal number of interactions needed to understand a biological scenario involving kinases (**k**), proteins (**p**), and their phosphosites (**s**) in response to an initial perturbation (δ).In this case, in contrast with the Prolog implementation, the perturbation refers to a gene encoding a specific kinase being deleted. Our background information in this implementation has levels of granularity, in this case the kinase targeting a phosphosite on a protein version is depicted.

The next iteration involves a "uniqueness" check that, similarly to previous implementations, checks whether a phosphosite **S** is uniquely targeted by a kinase **K**. This is achieved by establishing a *rule* `shared-target(kinase(K), kinase(_K2), phosphosite(S))` and checking for its negation as part of the 2nd iteration. In Prolog notation:

```
doesSilofKwork2(deletionOF((kinase(K))) :-
    phosphorylates(kinase(K),phosphosite(S),protein(P)),
    \+ sharedtarget(kinase(K), kinase(_K2),
       phosphosite(S)),
    perturbs(deletionOF((kinase(K))),phosphosite(S),
       occupancy(down),_).
```

The *rule* sharedtarget(kinase(K), kinase(_K2), phosphosite(S)) includes two goals matching phosphorylation *facts* on shared phosphosites then checking that the two phosphorylating are not the same.

```
sharedtarget(kinase(K), kinase(K2), phosphosite(S)) :-
    (phosphorylates(kinase(K),phosphosite(S),kinase(P)),
    phosphorylates(kinase(K2),phosphosite(S),kinase(P)),
    K1 \= K2).
```

In the toy model, the responses to these two iterations were appropriate. However, there is still a concern that when applied to the larger dataset with more than five perturbation-affected phosphosites, the true responses will be based on only one phosphosite instance. Therefore the addition of a majority check (similar to previous implementations) was decided. The final iteration of the *rule* took the following form:

```
doesSilofKwork3(deletionOF((kinase(K))) :-
    phosphorylates(kinase(K), phosphosite(S), kinase(P),
    \+ sharedtarget(kinase(K), kinase(_K2), phosphosite(
       S)),
    findall((phosphosite(S),
    perturbs(deletionOF((kinase(K))),phosphosite(S),
       occupancy(up),_),
    perturbs(deletionOF((kinase(K))),phosphosite(S),
       occupancy(unaffected),_),
    List1)),
    findall((phosphosite(S),
    perturbs(deletionOF((kinase(K))),phosphosite(S),
       occupancy(down),_),
    List2)))),
    length(List1, Length1), length(List2, Length2),
    compare( '>', Length2, Length1).
```

This *rule* introduces the concept of a "majority check" to the previous iteration. It accomplishes this by creating two distinct lists: one containing all the known targets of the kinase that are either labelled as "up" or "unaffected", and another containing those described as "down". The *rule* then compares the lengths of these lists to determine whether the majority of the known targets were labelled as "down", thus confirming the silencing of the kinase. Unlike the previous version in Chapter 2, where separate *rules* were called for the majority check in both the toy model logic program and full-scale logic program, this version combines all the steps into one concise representation.

4.2.2 Experimental facts and background knowledge sources

The first challenge was to identify a study that looked at the effect of kinase perturbation at the phosphoproteomic level. Recently a study was published that fits this criteria very closely. In (Li et al. 2019), a systems-level proteomic and phosphoproteomic analysis was carried out on 110 yeast single-kinase or phosphatase deletion strains under standard growth conditions. The 110 were split in 84 kinases and 26 phosphatases. They employed various methods, including traditional enrichment analysis, δgene-δgene correlation networks, and molecular covariance networks in order to analyse the functional relationships between these active proteins.

In total their chosen deletions, which will also be referred to as perturbations in this section cover 82% of all possible yeast and phosphatase deletion strains. Through their experimental and mass spectrometry workflow they were able to identify more than 4,600 and 13,000 phosphosites and proteins, respectively. Of interest is their finding that they were able to capture a large part of regulated phosphorylation events as well as 30% of those being newly captured. This was entirely attributed to their ability to normalise their data with protein abundance. This is also the data that we elected to include in our experimental 'perturbs/n' predicate format.

As is evident in the previous chapters describing the Prolog and ProbLog implementations in the MCF-7 cancer cell line, key constituents of our background knowledge base are the associations between kinases and their targets both at the protein and phosphosite levels. This was also the case when we had to implement a logic program in the yeast dataset. For this purpose, two main sources were chosen, namely the Yeast Kinase Interaction Database (Sharifpoor et al. 2011) (accessible at `http://www.moseslab.csb.utoronto.ca/KID/`) and the Yeast Kinome database (Breitkreutz et al. 2010) (accessible at `https://thebiogrid.org/project/2`).

The Yeast Kinase Interaction Database (KID) is a resource that contains data on phosphorylation events from various high- and low-throughput experiments. It includes a total of literature-curated 6,225 low-throughput and 21,990 high-throughput interactions, resulting from over 35,000 experiments. It includes 517 high-confidence (or 853 low-confidence) kinase-substrate pairs depending on the cutoffs used for the metric provided in the study. On the other hand the Yeast Kinome database contains 1,844 interactions observed by mass-spectrometry based analysis of protein complexes. It is updated monthly and has been subsumed as part of the BioGRID project which, in turn is a broad interaction database including more than 1.7m individual interactions (Oughtred et al. 2021).

Both these contain kinase (and phosphatase) target associations with the latter (Yeast Kinome db) also containing these at the phosphosite level. As will be shown below however, the issue of overlap between our experimental *facts* and background knowledge was present here as well, therefore limiting the predictions that could be made.

Facts List	Source	Size (Number of facts)
Dgene-Dgene phosphosite phosphosite observations	Perturbation Study	685,358
Kinase Substrate Associations	Yeast Kinome Database / bioGRID	7589
Kinase Protein Associations	Yeast Kinome Database / bioGRID	1558
Kinase Protein Associations	Kinase Interraction Database (lenient)	853
Kinase Protein Associations	Kinase Interraction Database (Strict	517
	Total =	695,875

Table 4.1: Table listing the title, sources and associated number of *Facts* that make up the Background Knowledge section of the probabilistic logic model. The total reported is for the strict cut-off of the Kinome Interaction Database. For the lenient cutt-off the total is 696,211. Specific sources referenced (Sharifpoor et al. 2011), (Li et al. 2019), (Breitkreutz et al. 2010)

4.2.2.1 Yeast perturbation observation *facts* and background knowledge sources

From the perturbation study chosen, a total of 685,358 *facts* were extracted containing the effect of the aforementioned 110 kinase and phosphatase deletions over 13,258 individual phosphosites. They were extracted from a .csv file containing the log2fold change recorded for each phosphosite between wild type and δgene strains of yeast. For most phosphosites there were two replicates reported. Not all phosphosites had both replicates for all δgene combinations. Only phosphosites containing two replicates were considered and the mean value between the two was taken for those that did. The code determined whether the average of the log2 fold change values is less than or equal to -0.5, greater than or equal to 0.5, or between those two values. Depending on the value of the average, the code sets the value of a variable called " change" to "down", "up", or "unaffected", respectively. The *fact* took the following form:

```
perturbs('YCK2', 'ENO2_pT324', 'ENO2', unaffected, 0.0475).
```

The background knowledge base includes *facts* that describe the relationships between kinases, proteins and phosphosites at both the kinase-protein and kinase-phosphosite level. The first set of *facts* describing kinase - protein level information came from the Kinome Interaction Database and the Yeast KINOME database, while the kinase - phosphosite level information came from the KINOME/bioGRID database alone. They were transcribed into *facts* with the following format:

Protein level information: `knownKtarget('ARK1', 'PAN1').`
Phosphosite level information: `knowntarget('ATG1','ALY1_pS813').`

It is important to note here, that during the above process of transcribing the experimental and background knowledge sources into *facts* it was decided that this implementation would take place with a Prolog interpreter rather than with a ProbLog system. More specifically, only the known kinase target associations and of those only the ones mined from the Kinase Interaction database had an associated degree of belief-esque value associated with them. For the perturbation phosphosite observations, due to the fact that only two replicates were reported, and in some cases only one, a p-value or p-value based degree of belief could not be established.

4.2.2.2 Overlap between experimental observations and background knowledge

The problem of overlap (or lack thereof) was significant with this implementation as it was for the human kinase studies (Chapter 2 & 3). Starting from the kinase-phosphosite level, looking at the overlap between the observational *facts* containing individual phosphosites and those present in the background knowledge base, only 667 individual phosphosites from the total are present in the kinase-phosphosite associations knowledge base Figure 4.2 A, approximately 5%. At the protein level things improve slightly depending on the data source and cutoff used (for Kinase Interaction Database). For the Kinome database (middle of Figure 4.2) when looking at the Kinase Protein pairs information the overlap stands at 72 deleted kinases and 58 deleted kinases when extracting the information from the Kinase Protein Phosphosite level triples. The disparity between the two is because the database reports these in different files which possibly contain an error with respect to formatting or ommission of a subset of data. (n.b. The database curators have been contacted directly but no response was given at the time of writing).

Looking at the Kinase Interaction database as a source, the overlap depends on the cutoff used. Cutoffs of 6.73 and 4.72 correspond to a P value of less than 0.01 and 0.05, respectively, for strict and lenient lists of kinase-protein pairs as were suggested in the article detailing the database (Sharifpoor et al. 2011). Accordingly, the overlap between the deleted kinases and the two lists of Kinase and their target Protein pairs can be seen in Figure 4.3. As will be demonstrated in the following part of this chapter, the poor overlap was addressed by considering combinations of background knowledge sources.

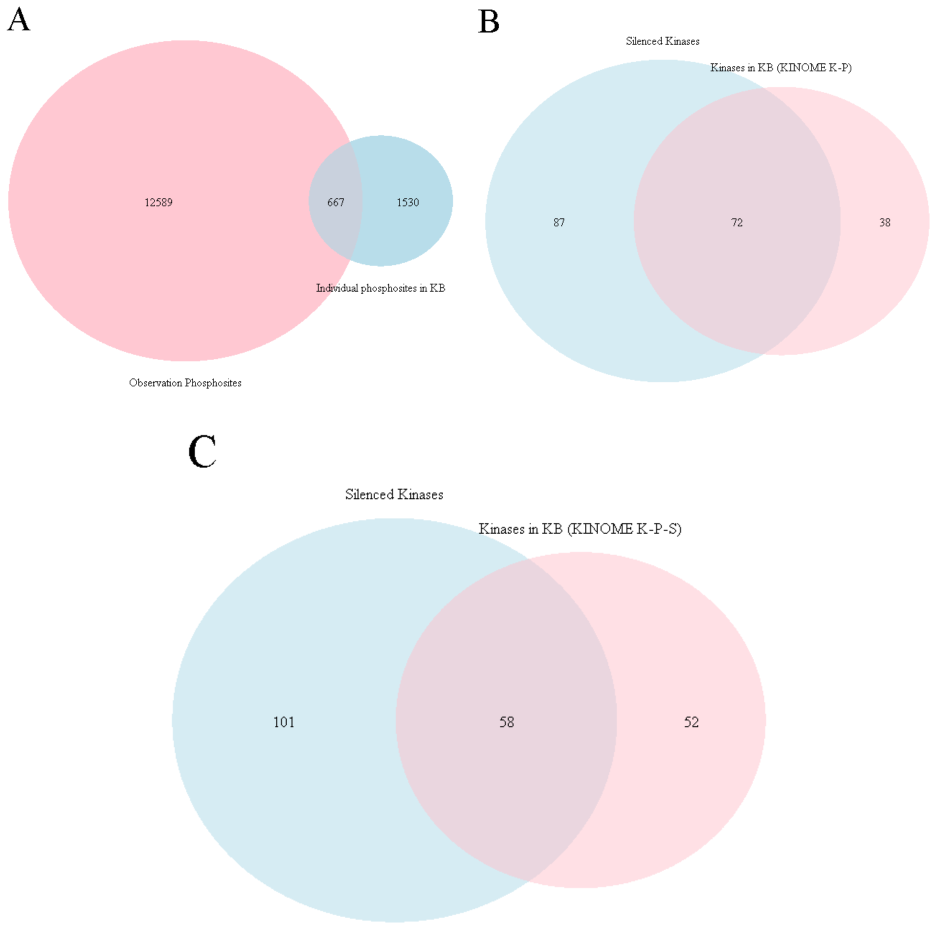


Figure 4.2: Venn Diagrams depicting the overlap between our experimental data and background knowledge sourced from Kinome Database / bioGRID (Breitkreutz et al. 2010). In A) total number of phosphosites is 13,256. B: Overlap between kinases in KinomeDB/bioGRID and deleted kinases at the kinase-protein level of association. C: Overlap between kinases in KinomeDB/bioGRID and deleted kinases at the kinase-phosphosite level of association.

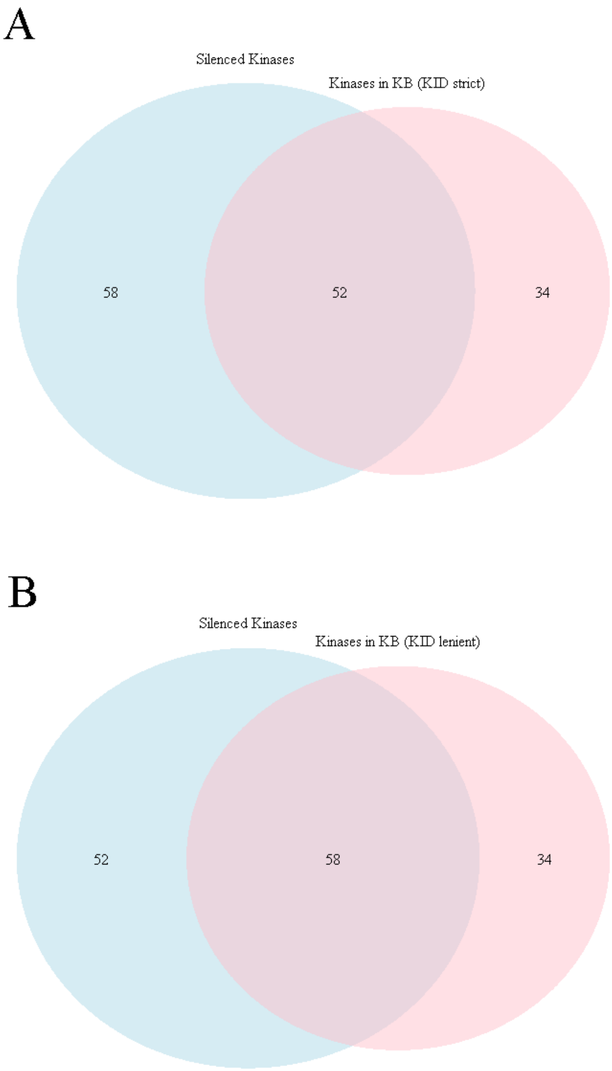


Figure 4.3: Venn Diagrams depicting the overlap between our experimental data and background knowledge sourced from Kinase Interaction database (Sharifpoor et al. 2011). These Venn diagrams A) Strict and B) Lenient represent the overlap between our knowledge base and deleted kinases. The association is at the kinase protein level.

4.3 Results

4.3.1 Silence-of-K Rule iterations and Kinase Compensation Rule

Collecting the results was accomplished following a similar pipeline to that described in Chapter 2. The Prolog interpreter was used to perform queries based on certain combinations, and the resulting output was stored in lists. These lists were then parsed by Python scripts to perform additional visualisation and analysis in RStudio. This centred around the number of deletions that were predicted to have worked. Additionally, RStudio was used to create the kinase compensation networks, as described below in this section.

Using kinase-phosphosite level associations, the first iteration of the *rule* proved to exhibit the best recall, confirming 21 kinase gene deletions having taken place, thus confirming the ground truth established by PCR, as reported in the study. Successive iterations of the *rule*, 2nd (checking uniqueness) and 3rd (checking majority of known phosphosites) were less accurate, both confirming only 5 kinases as being silenced. Using kinase-protein level associations the results improved overall depending on the source of kinase protein pairs used. For the Kinome Database from bioGrid, 33, 7 and 3 kinase deletions were confirmed for *rule* iterations 1,2 and 3, respectively. Kinase Interaction Database (KID) with a lenient cut-off yielded 26, 9 and 5, whereas with a strict cut-off the results were 21, 8 and 4. Results summarised in Table 4.2.

Rule Iteration & Background Knoweldge Level (Source where appropriate)	True Positive	False Negative
First Iteration & Phosphosite level	21	63
Second Iteration & Phosphosite level	5	79
Third Iteration & Phosphosite level	5	79
First Iteration & Protein level (bioGrid)	33	51
Second Iteration & Protein level (bioGrid)	7	77
Third Iteration & Protein level (bioGrid)	3	81
First Iteration & Protein level (KID Strict)	21	63
Second Iteration & Protein level (KID Strict)	8	76
Third Iteration & Protein level (KID Stict)	4	80
First Iteration & Protein level (KID Lenient)	26	68
Second Iteration & Protein level (KID Lenient)	9	75
Third Iteration & Protein level (KID Lenient)	5	89

Table 4.2: Table listing the True Positive and False Negative counts for each of the rule iterations. The False Negatives refer to kinase gene deletions we were not able to pick up due to either poor background information overlap or *rule* iterations not performing as intended.

As the overlap with a ground truth (the deletion of a kinase encoding gene) was poor, it was imperative to look at ways of improving this. First off combining the outputs of the two levels of background knowledge considered was the most straightforward approach. Collecting the data involved a relatively complex query which took the following form:

```
findall(A, (yeastkinase(A),doesSilofKworkProtlvl(A)),L),
sort(L,N),
findall(B, (yeastkinase(B), doesSilofKwork(B)), Q),
sort(Q,T),
append(T,N,W), length(W,K).
```

The first 'findall/3' predicate collects all instances of `true` groundings of the protein level while the second collects the groundings of the phosphosite level. The 'yeastkinase/1' predicate makes sure that only kinases from the list of deleted kinases is checked. Finally the sort/2 predicates remove duplicates from the above two lists, the 'append/3' predicate concatenates the sorted lists and the 'length/2' outputs the count of each grounded instance within the concatenated list. As above, starting with the Kinome Database as a Kinase-Protein association source, 54, 12 and 8 ground truths were captured with each successive iteration of the *rule.* From the Kinase Interaction Database, the lenient cut off gave 47, 14 and 10, and with the strict cut off 42, 13 and 9. Overall we noted a significant improvement which was mainly evident in the 1st iteration of the rule, picking up increasingly more of the ground truth. These can be considered True Positive results.

The next step was to combine all of the known background sources whilst removing duplicates to avoid clashes. This yielded a Kinase-Protein pair background knowledge base which covered (with at least one association) 79/82 of the deleted kinases in our dataset. This was the largest overall coverage and yielded the best result, confirming 62 individual deletions as seen in Figure 4.4. In contrast, iterations 2 and 3 yielded worse results, 13 and 8, respectively. From the above the best combinations between knowledge base sources (KID: strict/lenient and KINOME:bioGRID), level of background knowledge considered (kinase-phosphosite, kinase-protein or both) for each rule iteration were:

1. Rule 1: All BK, both levels: 62 TP / 17 FN
2. Rule 2: KID Lenient cut-off, both levels: 14 TP / 46 FN
3. Rule 3: KID Lenient cut-off, both levels: 10 TP / 50 FN

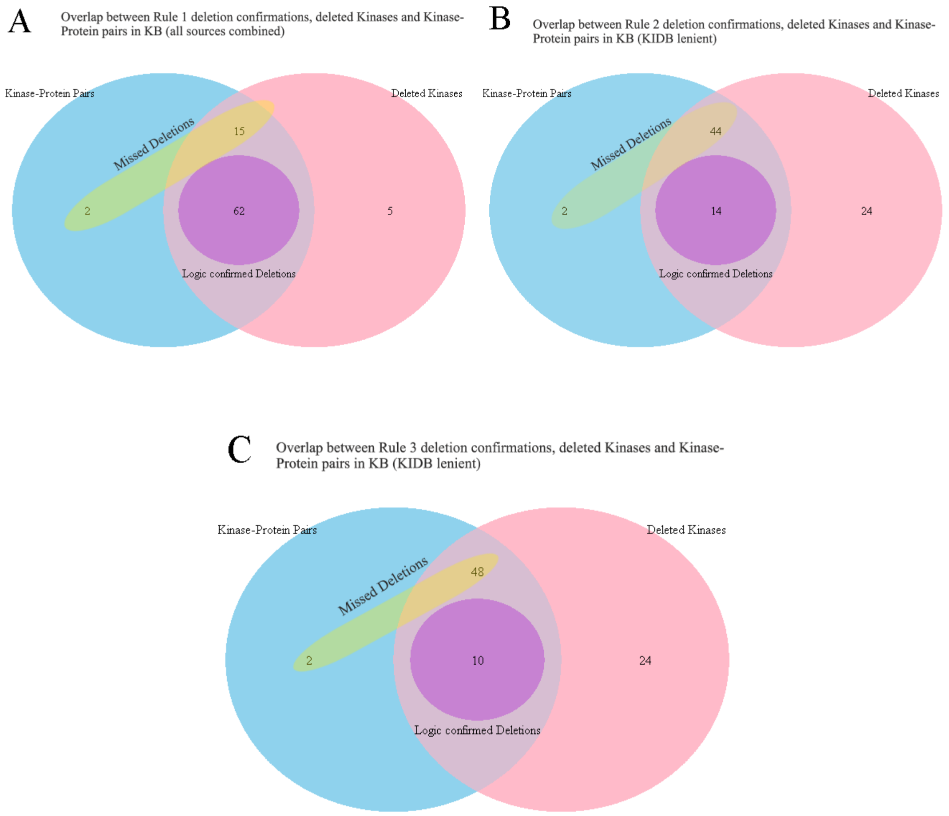

A
Overlap between Rule 1 deletion confirmations, deleted Kinases and Kinase-Protein pairs in KB (all sources combined)
Kinase-Protein Pairs
Deleted Kinases
Missed Deletions
15
2
62
5
Logic confirmed Deletions
B
Overlap between Rule 2 deletion confirmations, deleted Kinases and Kinase-Protein pairs in KB (KIDB lenient)
Kinase-Protein Pairs
Deleted Kinases
Missed Deletions
44
2
14
24
Logic confirmed Deletions
C
Overlap between Rule 3 deletion confirmations, deleted Kinases and Kinase-Protein pairs in KB (KIDB lenient)
Kinase-Protein Pairs
Deleted Kinases
Missed Deletions
48
2
10
24
Logic confirmed Deletions

Figure 4.4: Figure containing Venn Diagrams depicting the ground truth (True Positives) and False negatives captured by the top performing *rule* iterations and background knowledge levels considered. In all three Venn diagrams (A,B and C) the colour scheme is as follows: Purple for True Positives, light yellow for False Negatives, light blue for background knowledge kinase protein target pairs and light pink for gene deletion kinases. A: *Rule* iteration 1 with background knowledge at the protein level and from all sources combined had the best recall with 62 True Positives identified out of a potential 78. B: Rule iteration 2, Kinase Interaction database lenient cut-off and phosphosite level information with 14 True Positives and 46 False Negatives out of 60 potential. C: The third iteration of the rule performed best when the background knowledge was sourced from KID with lenient cutoff at 10 True Positives, 50 False Negatives out of a potential 64. The difference in potential totals comes can be attributed to how many of the deleted kinases were present and therefore queriable in each instance of background knowledge source and protein or phosphosite level consideration.

4.3.2 Explaining the 17 missed deletions with a new rule

From the above it was therefore decided to further focus on explaining the results obtained from *rule* iteration 1, with the results of both levels included and all background knowledge sources combined. However, as can be seen in the triple Venn diagram of Figure 4.4 A, there were 17 kinase deletions the *rule* iteration was not able to identify. As this iteration did not exclude at shared targets between kinases a possible hypotheses is that other kinases are acting in a compensatory manner. For this the 'ykinasecompcheck/2' *rule* was developed.

In brief, it checks whether a kinase **K2** increases its activity when a different kinase **DelK1** is silenced, in order to compensate for the loss of **DelK1's** activity and maintain the overall activity of the signalling pathway. Specifically the Prolog notation the *rule* took the following form:

```
ykinasecompcheck(DelK1, K2) :-
    (perturbs(DelK1,_Pst,Tprot1,down,_) ;
    perturbs(DelK1,_Pst,Tprot1,unaffected,_)),
    sharedKinasetarget(DelK1,K2,Tprot1),
    knowntarget(K2, Pst1, Tprot1),
    (perturbs(DelK2,Pst1,Tprot1,up,_),
    (DelK2 \= K2, DelK2 \= DelK1)).
```

Specifically, the first goal calls the 'perturbs/5' predicate that holds if the first argument (**DelK1**) perturbs the fourth argument (**Tprot1**) in the direction specified by the fifth argument (down or unaffected). 'sharedKinasetarget/3' is a predicate that holds if the first argument (**DelK1**) and the second argument (**K2**) share the third argument (Tprot1) as a target. The information used for this, in order to maintain consistent background knowledge was the combination of all kinase protein background knowledge sources. 'knowntarget/3' is a predicate that holds if **Tprot1** is a known target of the **K2** at the phosphosite specified by the second argument (**Pst1**). Finally, 'perturbs/5' is a predicate that holds if the first argument (**DelK2**) perturbs the second argument (**Pst1**) in the direction specified by the fifth argument (up). The goal is true if **DelK2** is different from both **K2** and **DelK1**. The last goal ensures that kinase **K2** indeed phosphorylates phosphosite **Pst1** (in the condition it, itself is not deleted).

Using the above rule and the list of 17 missed deleted Kinases, for each, a further list of compensatory kinases was collated using the *rule* described above. These lists where then used to build compensatory kinase interaction networks in order to visualise the effect of kinase deletions on known kinase interaction networks. Predicted or compensatory pathways were compared with pathways reconstructed from literature (Sharifpoor et al. 2011). Visualisations were carried out in Rstudio using the igraph package (Csardi & Nepusz 2006). Specifically, the graph development algorithm chosen was Fruchterman-Reingold (Fruchterman & Reingold 1991).

From the 17 kinases, we were able to place at least one within known pathways, such as HOG1, Cell Wall Integrity, Meiosis, DNA damage repair, cell growth, glucose based signalling and cell cycle. Cell cycle had the largest number (4) of kinases from the list of 17 whereas none of the false negatives (kinase deletions not present in the *rule* predictions) were present in other pathways such as those associated with endocytosis, transcription response and mating. In Figure 4.5 A, Cell Cycle pathway is highlighted with a red border, the (High-osmorality glucose) HOG pathway with a blue border in Figure 4.5 B), red border in Figure 4.5 C and meiosis pathway with a fuchsia border (Figure 4.5 C). They are all connected canonically, showing the relative accuracy of the *rule* described above in picking up the shift in pathway activity to account for the deletion of a given kinase within the pathway. For the highlighted pathways the majority of their constituent kinases are present, deletions of which the first iteration of the *rule* did not pick up. This indirectly suggests that in response to certain key deletions these highlighted pathways are able to rewire themselves in an effort to counteract the perturbation.

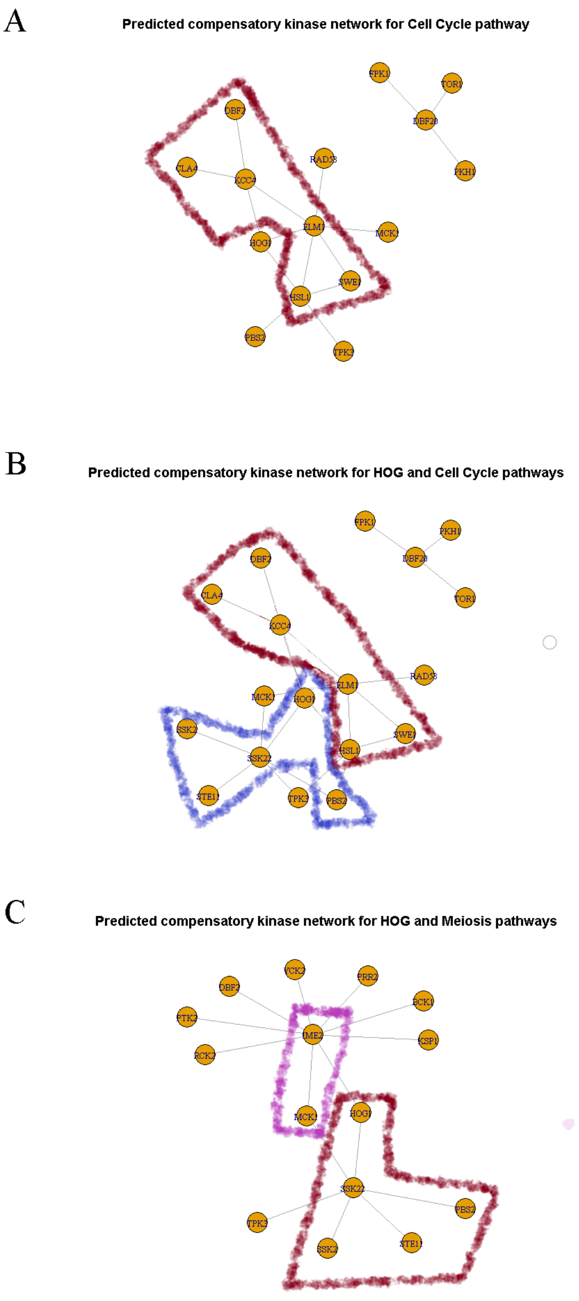


Figure 4.5: Predicted manner of kinase network rewiring following gene deletions. Kinases within pathways act in a compensatory manner. A) Cell Cycle pathway, predicted pathway rewiring. B) High-osmorality glucose and cell cycle pathway rewiring. C) High-osmorality and Meiosis pathway rewiring. Highlighted are the kinases belonging to the respective pathways that include False Negative i.e. missed gene deletions within our predictions.

4.4 Discussion

Applying the methodology directly to a similar dataset but from a different organism was not as straightforward as initially envisaged. The ever present issue of overlap was a challenge here also. Despite this being one of the most well studied model organisms poor coverage between kinase - phosphosite associations and observational data taken from the gene knockout study is lacking.

The approach that yielded the most accurate results, in terms of recall, with the simplest iteration of the *rule*. The issue of poor overlap was also partially overcome by combining background knowledge sources and levels of granularity. Increasing complexity, as was the case with applying the methodology in the cancer cell line (Chapters 2 and 3) did not yield better results.

Delving deeper into the False Negatives, i.e. kinases from verified deleted genes which we were unable to pick up, yielded interesting results. The interconnected nature of kinase interaction pathways was captured by the 'ykinasecompcheck/2' *rule*. Using the 17 kinases as a base we were able to identify and contextualise the majority of major large Yeast pathways, namely HOG and cell cycle as well as smaller one such as Meiosis. For the remaining pathways, their constituents were correctly captured as having been silenced. Combining the outputs of the individual silenced gene queries offers an insight into potential yeast signalling rewiring following strong perturbation such as silencing of a group of kinases.

Addressing the issue of overlap requires initiatives like bioGRID and other datasets that maintain consistent information for both protein and phosphosite interactions. Ensuring naming conventions are uniform is even more crucial in the context of the human kinome and proteome. For genome-related studies the HUGO Gene Nomenclature Committee (HGNC) provides standardised resources and techniques, alleviating this concern significantly.

The HGNC assigns unique symbols to genes, reflecting their functions or attributes. In the context of the yeast kinome, combining resources, while a somewhat rudimentary approach, appeared to partially mitigate this issue.

Next steps in applying this methodology would initially involve extending the knowledge base to include protein subcellular locations. Additionally, augmenting facts with degrees of belief, as described in Chapter 3 could further increase confidence in results. Both of these require the mining of appropriate and well curated information in order to prove viable.

This page is intentionally left blank

Chapter 5

Conclusions and Future work

The overall aim of this thesis was the evaluation of Symbolic AI methodologies as a tool to analyse cell signalling. The two main methodologies considered were Prolog and its probabilistic extension ProbLog. These were initially applied to a phosphoproteomics dataset based on a cancer cell line model, with the former (Prolog) also being applied to a similar dataset from a yeast model.

In Chapter 2, initially we showed that Symbolic AI and more specifically purely Prolog based methodologies are able to handle this scale of *facts* knowledge bases and domain specific *rules* to reason over them without facing an explosion in computational time or resources. We showed that an entirely Prolog based approach was able to capture a significant amount of known perturbagen kinase associations whilst also providing novel combinations. The stringency of the "doesDinhibitKinC/3" *rule* (which included checks for known kinase substrate associations, subcellular colocalisation and kinase cross-talk) resulted in more confidence. Specific combinations of *rules* that made up part of the aforementioned "top-level" rule, proved more efficient in capturing the complexities and uncertainties inherent in cell signalling than others.

In Chapter 3, the augmentation of the methodology with degrees of belief allowed us to rank predictions accordingly. Additionally, an increase in the overall number of *facts* due to the consideration of other background knowledge sources as well as the adoption of a new pipeline for processing of the raw MS data improved coverage. This was also done to in order to address the augmented nature of ProbLog *facts* (in this case with a contextual degree of belief). An issue that arose however was the exponential increase in computational resources needed to perform certain parts of the ProbLog pipeline, specifically revolving around availability of RAM. This was addressed by making use of the High Performance Computing Cluster of QMUL, Apocrita (King et al. 2017). Ultimately, this allowed us to output ranked lists of perturbagen kinase predictions which included, amongst a number of false positives, already known associations as well as potential novel kinase targets for inhibitors.

In Chapter 4, the methodology was applied to an entirely different organism, specifically the model organism *Saccharomyces cerevisiae* or budding yeast. The experimental dataset chosen was also phosphoproteomics based, similar to the cancer cell line dataset, however the perturbation in this case was deletion of kinase encoding genes. This provided reliable ground truths by removing the ambiguity of small molecule inhibitor off target interactions. We were able to confirm the majority of verified gene deletions with the simplest iteration of the context specific *rule*. This was achieved by combining different background knowledge sources and from within these looking at both protein and phosphosite level kinase pairs. For the silenced kinases that we were unable to confirm, we were able to provide a prediction for the overall rewiring of known kinase pathways.

Across all these applications of both Prolog and ProbLog, the major obstacle faced was poor overlap between our experimental data and the background knowledge base. A number of steps were taken to address this in all implementations however there is still much room for improvement on this front. Initially this can be addressed by re-applying parts of the methodology, specifically for the cancer cell line model, in a manner consistent with generating further background knowledge, while maintaining phosphoproteomics coverage. Another way to enhance the methodology's granularity is by considering additional factors, such as the sign (inhibitory or excitatory effect) of phosphorylation on specific sites within proteins and kinases. Moreover, incorporating information on overall cell activities associated with proteins and kinases can provide further context to predictions and novel associations. By implementing these improvements, we aim to overcome the challenge posed by the limited overlap and enhance the reliability and comprehensiveness of the models. The issue of overlap between newly proposed and established associations has been highlighted previously (Hijazi et al. 2020).

When it comes to increasing the complexity of *rules*, careful consideration is essential. On one hand, it is crucial to avoid over-modelling or creating context-specific rules that may obscure novel associations and hinder the overall explainability and interpretability of the approach. It is vital to strike a balance to maintain the model's explainability. A well-written and structured model within a logical formalism framework ensures readability and interpretability, contributing to better understanding and explanation of the model's behaviour. As has been demonstrated in the chapters before, even the transition from a Toy Model to a full fledged Logic Program decreased the explainability inherent in Symbolic AI approaches.

Additional work which mainly arose via collaborations, that was ultimately not included in this thesis and was mainly related to Chapter 2 Prolog implementation, was the development of an explanation based "meta-interpreter" with an associated website. The approach resulted in ~ 3.7 million "paths" with associated explanations, for which the website served as another layer of meta-interpretation. While this was intended to explain the reasoning behind the Prolog generated outputs, the number of "paths" made its use intractable for its intended purpose. Nonetheless, there is potential for improvement by refining the predicted "paths" into natural language arguments. A brief description of the "paths" based "meta-interpreter" as well as the setup of the website can be found in Appendix 1.

Our ultimate goal in this research and application of Symbolic AI is to create predictions and novel hypotheses that are trusted because they are explainable, auditable, easy to understand and therefore verify or debunk. In order to achieve this, we must be clear in using our domain expertise and adhere to the principle of elegance when writing the rules for the application side of these methodologies. This will help ensure that the outputs are reliable and can be trusted.

In summary, the work in this thesis has demonstrated that symbolic AI methods can be applied to the large and complex datasets produced by phosphoproteomics studies, in the context of the heterogeneous and sometimes uncertain background knowledge of cell signalling. This in itself is a useful finding, given that symbolic approaches are largely ignored by bioinformatics community, which tends towards "reinventing the wheell" with every new tool as well as not being trained or even aware of logical formalism. This, in turn almost always leaves the crucial logical reasoning part of scientific discovery entirely to human scientists. However, it is clear from the above discussion that much needs to be done before symbolic AI is fully ready for practical application, at least in the field of cell signalling research.

This page is intentionally left blank

Appendix A

Appendices

A.1 Detailed description of in house pipeline for new log 2 fold changes of phosphosites following perturbation

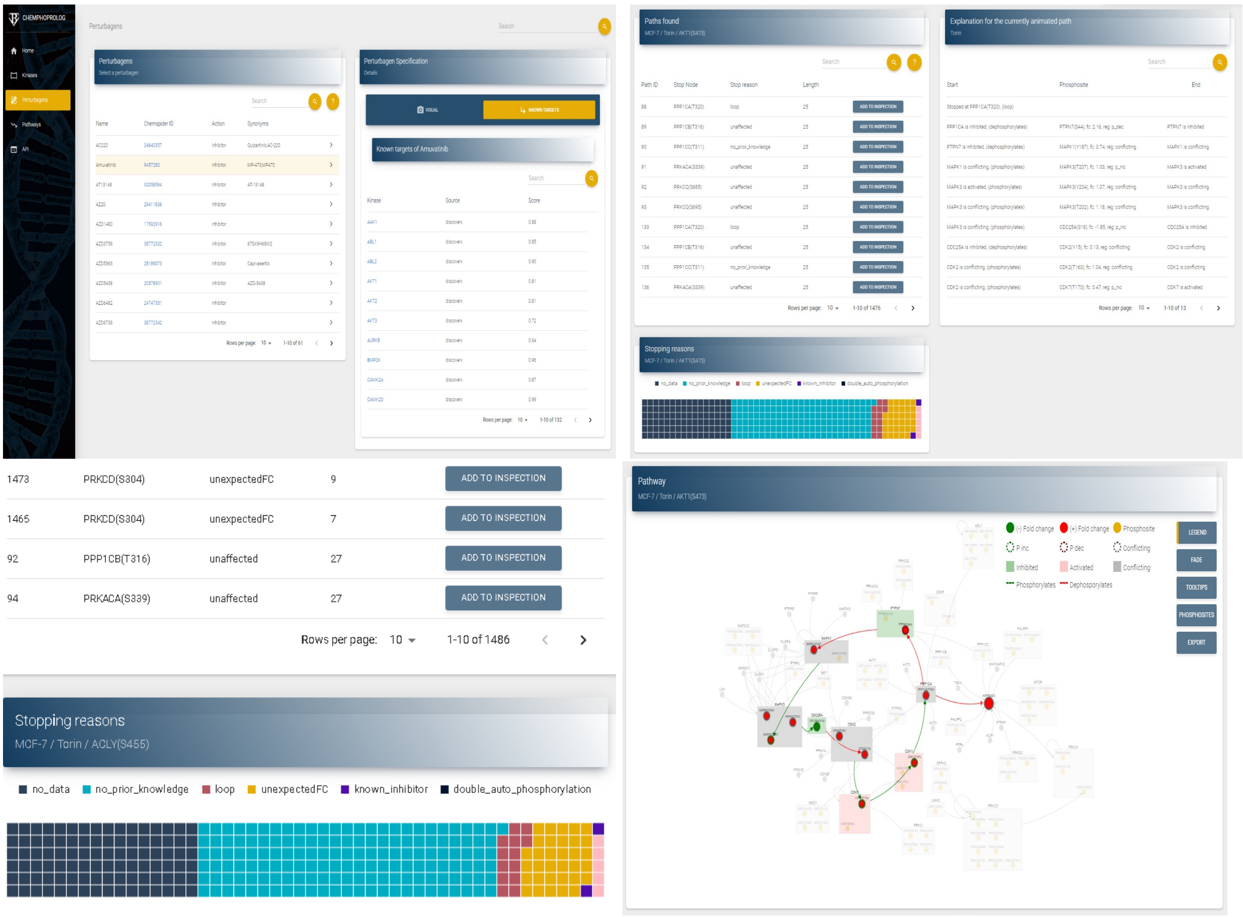
The experimental data for this study is a publicly available LC-MS/MS-based phosphoproteomics dataset obtained from (Hijazi et al. 2020), which contains phosphopeptide abundance data for three cell lines treated with 61 kinase inhibitors in biological and technical duplicates. The focus of the study is on the cell line MCF-7, providing quantified peak areas for 14,448 phosphosites measured in 61 treatments, for a total of 277 samples. Reprocessing and differential phosphorylation analysis were performed with an in-house pipeline, including log2 transformation and normalization by median scaling. Filtering was performed to remove one sample with less than 60% quantifiable phosphosites and 128 phosphosites with poor quantification across all samples. Differential phosphorylation analysis was performed using the R package limma, with the experimental batch included as a blocking variable. Log2 fold changes were calculated by setting contrasts between each kinase inhibitor treatment and DMSO vector control. Missing fold changes were imputed and signal intensity-dependent p-values were calculated for all phosphorylation states using an approach that borrows information from other samples. n order to impute missing fold changes, the signal intensity distribution for each phosphosite is determined across all samples. A Z-test is used to estimate the distance of a measurement from the distribution of the respective phosphosite. Z-scores are transformed so that an intensity value close to the distribution mean corresponds to a fold change of 1, a negative Z-score corresponds to a fold change between 0 and 1, and a positive Z-score corresponds to a fold change between 1 and 2 when the control is missing. When a measurement is missing from a treatment, fold changes are multiplied by -1 to reflect under-phosphorylation with respect to the control. A confidence score is established based on the signal intensity at which phosphorylation measurements were observed. The signal dependent variance is quantified by calculating hypothetical fold changes between control samples for each phosphosite, and then sorting and fitting models to the means and standard deviations. This allows for the determination of an error distribution for the signal intensity of a fold change, which is used to perform a Z-test and determine a p-value. Phosphosites observed at a lower signal require a higher fold change to be considered significant.

A.2 ChemPhoProlog Description

This following work was carried out in collaboration with M.Huebner and implemented as improvements to the website by O.Ozcan. The current pathway analysis is done with Prolog, which outputs valid paths for a given cell line, substrate and perturbagen combination in a text format. These paths are generated with the aim of explaining the overall effect of a given perturbagen on the cell lines phosphoproteome. A path is initiated from a phosphosite and built "upwards" towards a perturbagen or top level kinase. At every step, using prior knowledge, the algorithm checks whether a set of conditions is met. Often the paths are long with explanations embedded in the results, therefore hard to interpret in text format. Additionally they "fail", i.e. stop prematurely or wrongly due to a number of reasons. These include lack of observational data (no_data), lack of prior knowledge (no_prior_knowledge) and auto-phosphorylation loops (loops, double_auto_phosphorylation) (Figure A.1 iii). To visualise this process as well as our curated knowledge base, we significantly improved our web based platform (`https://chemphoprolog.herokuapp.com/home`). Site users can visualise their selected paths allowing easier exploration of the analysis and the ability to compare and contrast. Moreover, the website contains other relevant interactions, that make up the knowledge base, such as known perturbagen kinase interactions from other publicly available databases.

Chemphoprolog is developed with React.js on the frontend, Node.js on the backend with a PostgreSQL database. A modern frontend library such as React.js allows excellent performance with intractability via efficient DOM updates and built-in extendability via its inherent component based architecture. Node.js acts as a bridge between the PostgreSQL database and the front-end where we also host the Chemphoprolog API. Specifically, the design of the backend allows us to scale the database accordingly as well as providing an amenable point of access to the all or parts of the stored data. An overview of the site features can be seen in Figure A.1. On the left pane (i) the user can navigate the knowledge base and access information about Perturbagens and (ii) Kinases. All information displayed is interactive while seamlessly allowing the user to navigate between different choices. Additionally, selected kinases of interest are displayed on the left pane (not shown here) for faster access. The final screencap represents a still from the animation which was developed using parts of the Cytoscape Js software Franz et al. (2016). Once a user selects a phosphosite or kinase of interest, the animation can be started to provide a visual representation of how the pathway is built.

We envision the website and its functionalities providing biologists with a rich and user friendly resource. Through the website they can currently access detailed visualisations of the paths as well as elementary associated explanations. These well be supplemented by the interpretable reanalysis of the derived paths which will also include phenotypic associations. Currently, it can be considered to stand alongside ChemPhopro (`http://chemphopro.org`), however future plans include combining the two thus offering a more complete picture of the EBDT paper data.



Perturbagens

Perturbagens
Select a perturbagen

Name	Chemspider ID	Action	Synonyms
AC220	24640357	inhibitor	Quizartinib,AC-220
Amuvatinib	9457280	inhibitor	MP-470,MP470
AT13148	32058594	inhibitor	AT-13148
AZ20	29411836	inhibitor	
AZD1480	17592916	inhibitor	
AZD3759	38772332	inhibitor	67SX9H68W2
AZD5363	28189073	inhibitor	Capivasertib
AZD5438	20578901	inhibitor	AZD-5438
AZD6482	24747351	inhibitor	
AZD6738	38772342	inhibitor	

Rows per page: 10 1-10 of 61

Perturbagen Specification
Details

Known targets of Amuvatinib

Kinase	Source	Score
AAK1	discoverx	0.88
ABL1	discoverx	0.85
ABL2	discoverx	0.90
AKT1	discoverx	0.81
AKT2	discoverx	0.81
AKT3	discoverx	0.72
AURKB	discoverx	0.64
BMP2K	discoverx	0.96
CAMK2A	discoverx	0.87
CAMK2D	discoverx	0.99

Rows per page: 10 1-10 of 132

Paths found
MCF-7 / Torin / AKT1(S473)

Path ID	Stop Node	Stop reason	Length	
88	PPP1CA(T320)	loop	25	ADD TO INSPECTION
89	PPP1CB(T316)	unaffected	25	ADD TO INSPECTION
90	PPP1CC(T311)	no_prior_knowledge	25	ADD TO INSPECTION
91	PRKACA(S339)	unaffected	25	ADD TO INSPECTION
92	PRKCQ(S685)	unaffected	25	ADD TO INSPECTION
93	PRKCQ(S695)	unaffected	25	ADD TO INSPECTION
133	PPP1CA(T320)	loop	25	ADD TO INSPECTION
134	PPP1CB(T316)	unaffected	25	ADD TO INSPECTION
135	PPP1CC(T311)	no_prior_knowledge	25	ADD TO INSPECTION
136	PRKACA(S339)	unaffected	25	ADD TO INSPECTION

Rows per page: 10 1-10 of 1476

Explanation for the currently animated path
Torin

Start	Phosphosite	End
Stopped at PPP1CA(T320), (loop)		
PPP1CA is inhibited, (dephosphorylates)	PTPN7(S44), fc: 2.16, reg: p_dec	PTPN7 is inhibited
PTPN7 is inhibited, (dephosphorylates)	MAPK1(Y187), fc: 0.74, reg: conflicting	MAPK1 is conflicting
MAPK1 is conflicting, (phosphorylates)	MAPK3(T207), fc: 1.03, reg: p_inc	MAPK3 is activated
MAPK3 is activated, (phosphorylates)	MAPK3(Y204), fc: 1.07, reg: conflicting	MAPK3 is conflicting
MAPK3 is conflicting, (phosphorylates)	MAPK3(T202), fc: 1.18, reg: conflicting	MAPK3 is conflicting
MAPK3 is conflicting, (phosphorylates)	CDC25A(S18), fc: -1.85, reg: p_inc	CDC25A is inhibited
CDC25A is inhibited, (dephosphorylates)	CDK2(Y15), fc: 0.13, reg: conflicting	CDK2 is conflicting
CDK2 is conflicting, (phosphorylates)	CDK2(T160), fc: 1.04, reg: conflicting	CDK2 is conflicting
CDK2 is conflicting, (phosphorylates)	CDK7(T170), fc: 0.47, reg: p_inc	CDK7 is activated

Rows per page: 10 1-10 of 13

Stopping reasons
MCF-7 / Torin / AKT1(S473)

■ no_data ■ no_prior_knowledge ■ loop ■ unexpectedFC ■ known_inhibitor ■ double_auto_phosphorylation

1473	PRKCD(S304)	unexpectedFC	9	ADD TO INSPECTION
1465	PRKCD(S304)	unexpectedFC	7	ADD TO INSPECTION
92	PPP1CB(T316)	unaffected	27	ADD TO INSPECTION
94	PRKACA(S339)	unaffected	27	ADD TO INSPECTION

Rows per page: 10 1-10 of 1486

Stopping reasons
MCF-7 / Torin / ACLY(S455)

■ no_data ■ no_prior_knowledge ■ loop ■ unexpectedFC ■ known_inhibitor ■ double_auto_phosphorylation

Pathway
MCF-7 / Torin / AKT1(S473)

Figure A.1: From left to right i) Example selection from perturbagen Knowledge base ii) Recursion derived path and associated explanations with metrics. Below iii) Stoppage reasons distribution iv) Pathway animation example

This page is intentionally left blank

References

Alaa, A. M. & Schaar, M. V. D. (2019).

Axtman, A. D. (2021), 'Characterizing the role of the dark kinome in neurodegenerative disease – a mini review', *Biochimica et Biophysica Acta (BBA) - General Subjects* **1865**(12), 130014.
URL: *https://www.sciencedirect.com/science/article/pii/S0304416521001732*

Baader, F., Bürckert, H.-J., Heinsohn, J., Hollunder, B., Muller, J., Nebel, B., Nutt, W. & Profitlich, H.-J. (1970), 'Terminological knowledge representation: A proposal for a terminological logic'.

Bateman, A., Martin, M. J., O'Donovan, C., Magrane, M., Alpi, E., Antunes, R., Bely, B., Bingley, M., Bonilla, C., Britto, R., Bursteinas, B., Bye-AJee, H., Cowley, A., Da Silva, A., De Giorgi, M., Dogan, T., Fazzini, F., Castro, L. G., Figueira, L., Garmiri, P., Georghiou, G., Gonzalez, D., Hatton-Ellis, E., Li, W., Liu, W., Lopez, R., Luo, J., Lussi, Y., MacDougall, A., Nightingale, A., Palka, B., Pichler, K., Poggioli, D., Pundir, S., Pureza, L., Qi, G., Rosanoff, S., Saidi, R., Sawford, T., Shypitsyna, A., Speretta, E., Turner, E., Tyagi, N., Volynkin, V., Wardell, T., Warner, K., Watkins, X., Zaru, R., Zellner, H., Xenarios, I., Bougueleret, L., Bridge, A., Poux, S., Redaschi, N., Aimo, L., ArgoudPuy, G., Auchincloss, A., Axelsen, K., Bansal, P., Baratin, D., Blatter, M. C., Boeckmann, B., Bolleman, J., Boutet, E., Breuza, L., Casal-Casas, C., De Castro, E., Coudert, E., Cuche, B., Doche, M., Dornevil, D., Duvaud, S., Estreicher, A., Famiglietti, L., Feuermann, M., Gasteiger, E., Gehant, S., Gerritsen, V., Gos, A., Gruaz-Gumowski, N., Hinz, U., Hulo, C., Jungo, F., Keller, G., Lara, V., Lemercier, P., Lieberherr, D., Lombardot, T., Martin, X., Masson, P., Morgat, A., Neto, T., Nouspikel, N., Paesano, S., Pedruzzi, I., Pilbout, S., Pozzato, M., Pruess, M., Rivoire, C., Roechert, B., Schneider, M., Sigrist, C., Sonesson, K., Staehli, S., Stutz, A., Sundaram, S., Tognolli, M., Verbregue, L., Veuthey, A. L., Wu, C. H., Arighi, C. N., Arminski, L., Chen, C., Chen, Y., Garavelli, J. S., Huang, H., Laiho, K., McGarvey, P., Natale, D. A., Ross, K., Vinayaka, C. R., Wang, Q., Wang, Y., Yeh, L. S. & Zhang, J. (2017), 'UniProt: The universal protein knowledgebase', *Nucleic Acids Research* **45**(D1), D158–D169.

Berggren, K., Xia, Q., Likharev, K. K., Strukov, D. B., Jiang, H., Mikolajick, T., Querlioz, D., Salinga, M., Erickson, J. R., Pi, S., Xiong, F., Lin, P., Li, C., Chen, Y., Xiong, S., Hoskins, B. D., Daniels, M. W., Madhavan, A., Liddle, J. A., McClelland, J. J., Yang, Y., Rupp, J., Nonnenmann, S. S., Cheng, K. T., Gong, N., Lastras-Montantilde;o, M. A., Talin, A. A., Salleo, A., Shastri, B. J., Lima, T. F. D., Prucnal, P., Tait, A. N., Shen, Y., Meng, H., Roques-Carmes, C., Cheng, Z., Bhaskaran, H., Jariwala, D., Wang, H., Shainline, J. M., Segall, K., Yang, J. J., Roy, K., Datta, S. & Raychowdhury, A. (2020), 'Roadmap on emerging hardware and technology for machine learning', *Nanotechnology* **32**, 012002.
URL: *https://iopscience.iop.org/article/10.1088/1361-6528/aba70f/meta*

Bergman, M., Paavola, S., Pietarinen, A.-V. & Rydenfelt, H. (2010), 'Nordic studies in pragmatism 1'.

Brand, A., Allen, L., Altman, M., Hlava, M. & Scott, J. (2015), 'Beyond authorship: Attribution, contribution, collaboration, and credit', *Learned Publishing* **28**(2), 151–155.

Breitkreutz, A., Choi, H., Sharom, J. R., Boucher, L., Neduva, V., Larsen, B., Lin, Z. Y., Breitkreutz, B. J., Stark, C., Liu, G., Ahn, J., Dewar-Darch, D., Reguly, T., Tang, X., Almeida, R., Qin, Z. S., Pawson, T., Gingras, A. C., Nesvizhskii, A. I. & Tyers, M. (2010), 'A global protein kinase and phosphatase interaction network in yeast', *Science (New York, N.Y.)* **328**, 1043–1046.
URL: *https://pubmed.ncbi.nlm.nih.gov/20489023/*

Calegari, R., Ciatto, G., Denti, E. & Omicini, A. (2020), 'Logic-based technologies for intelligent systems: State of the art and perspectives', *Information (Switzerland)* **11**.

Casado, P. & Cutillas, P. R. (2011), 'A self-validating quantitative mass spectrometry method for assessing the accuracy of high-content phosphoproteomic experiments', *Molecular and Cellular Proteomics* **10**(1).
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013456/*

Chan, C. Y. X., Gritsenko, M. A., Smith, R. D. & Qian, W. J. (2016), 'The current state of the art of quantitative phosphoproteomics and its applications to diabetes research'.

Cheng, H. C., Qi, R. Z., Paudel, H. & Zhu, H. J. (2011), 'Regulation and function of protein kinases and phosphatases', *Enzyme Research* **2011**.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3238372/*

Cintula, P., Fermüller, C. G. & Noguera, C. (2021), Fuzzy Logic, *in* E. N. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', Winter 2021 edn, Metaphysics Research Lab, Stanford University.

Clare, A. & King, R. D. (2002), 'Data mining the yeast genome in a lazy functional language'.
URL: *http://www.aber.ac.uk/compsci/Research/bio/dss/polyfarm/*

Colmerauer, A. & Roussel, P. (1992), 'The birth of prolog'.

Consuelo, G. & Faustino, M. (2002), 'Biological activities, mechanisms of action and biomedical prospect of the antitumor ether phospholipid et-18-och(3) (edelfosine), a proapoptotic agent in tumor cells', *Current drug metabolism* **3**, 491–525.
URL: *https://pubmed.ncbi.nlm.nih.gov/12369895/*

Crabbé, J. & Schaar, M. V. D. (2022), 'Label-free explainability for unsupervised models'.

Csardi, G. & Nepusz, T. (2006), 'The igraph software package for complex network research', *InterJournal* **Complex Systems**, 1695.
URL: *https://igraph.org*

Darwiche, A. (2002), 'A compiler for deterministic, decomposable negation normal form'.

Darwiche, A. (2009*a*), *The Complexity of Probabilistic Inference*, Cambridge University Press, p. 270–286.

Darwiche, A. (2009*b*), *Propositional Logic*, Cambridge University Press, p. 13–26.

Das, A., Member, G. S., Rad, P. & Member, S. (2022).

Davies, V., Wandy, J., Weidt, S., Hooft, J. J. V. D., Miller, A., Daly, R. & Rogers, S. (2021), 'Rapid development of improved data-dependent acquisition strategies', *Analytical Chemistry* **93**, 5676–5683.
URL: *https://pubs.acs.org/doi/full/10.1021/acs.analchem.0c03895*

Dinkel, H., Chica, C., Via, A., Gould, C. M., Jensen, L. J., Gibson, T. J. & Diella, F. (2011), 'Phospho.elm: a database of phosphorylation sites—update 2011', *Nucleic Acids Research* **39**, D261.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013696/*

Douven, I. (2021), Abduction, *in* E. N. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', Summer 2021 edn, Metaphysics Research Lab, Stanford University.

Dries, A., Kimmig, A., Meert, W., Renkens, J., Broeck, G. V. D., Vlasselaer, J. & Raedt, L. D. (2015), 'Problog2: Probabilistic logic programming'.
URL: *https://dtai.cs.kuleuven.be/problog*

Dung, P. M. (1992), 'On the relations between stable and well-founded semantics of logic programs', *Theoretical Computer Science* **105**, 7–25.

Dunham, I. (2018), 'Human genes: Time to follow the roads less traveled?', *PLoS Biology* **16**, e3000034.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6175530/*

Duong-Ly, K. C. & Peterson, J. R. (2013), 'The human kinome and kinase inhibition as a therapeutic strategy', *Current protocols in pharmacology / editorial board, S.J. Enna (editor-in-chief) ... [et al.]* **0 2**, Unit2.9.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4128285/*

Eduati, F., Jaaks, P., Wappler, J., Cramer, T., Merten, C. A., Garnett, M. J. & Saez-Rodriguez, J. (2020), 'Patient-specific logic models of signaling pathways from screenings on cancer biopsies to prioritize personalized combination therapies', *Molecular Systems Biology* **16**, e8664.
URL: *https://onlinelibrary.wiley.com/doi/full/10.15252/msb.20188664*

Eid, S., Turk, S., Volkamer, A., Rippmann, F. & Fulle, S. (2017), 'Kinmap: a web-based tool for interactive navigation through human kinome data', *BMC bioinformatics* **18**.
URL: *https://pubmed.ncbi.nlm.nih.gov/28056780/*

Essegian, D., Khurana, R., Stathias, V. & Schürer, S. C. (2020), 'The clinical kinase index: A method to prioritize understudied kinases as drug targets for the treatment of cancer', *Cell Reports Medicine* **1**(7), 100128.
URL: *https://www.sciencedirect.com/science/article/pii/S2666379120301701*

Franz, M., Lopes, C. T., Huck, G., Dong, Y., Sumer, O. & Bader, G. D. (2016), 'Cytoscape.js: a graph theory library for visualisation and analysis', *Bioinformatics* **32**, 309–311.
URL: *https://academic.oup.com/bioinformatics/article/32/2/309/1744007*

Fruchterman, T. M. J. & Reingold, E. M. (1991), 'Graph drawing by force-directed placement', **21**, 1129–1164.

Garson, J. (2021), Modal Logic, *in* E. N. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', Summer 2021 edn, Metaphysics Research Lab, Stanford University.

Gelder, A. V., Ross, K. A. & Schlipf, J. S. (1991), 'The well-founded semantics for general logic programs', *Journal of the ACM (JACM)* **38**, 619–649.

Goncalves, A., Ong, I., Lewis, J. A. & Costa, V. S. (2014), 'Towards using probabilities and logic to model regulatory networks', *Proceedings - IEEE Symposium on Computer-Based Medical Systems* pp. 239–242.

Gunning, D., Vorm, E., Wang, J. Y. & Turek, M. (2021), 'Darpa 's explainable ai (xai) program: A retrospective', *Applied AI Letters* **2**.

Herskowitz, I. (1988), 'Life cycle of the budding yeast saccharomyces cerevisiae', *MICROBIOLOGICAL REVIEWS* pp. 536–553.
URL: *https://journals.asm.org/journal/mr*

Hijazi, M., Smith, R., Rajeeve, V., Bessant, C. & Cutillas, P. R. (2020), 'Reconstructing kinase network topologies from phosphoproteomics data reveals cancer-associated rewiring', *Nature Biotechnology* **38**(4), 493–502.

Hill, A., Gotham, D., Fortunak, J., Meldrum, J., Erbacher, I., Martin, M., Shoman, H., Levi, J., Powderly, W. G. & Bower, M. (2015), 'Target prices for mass production of tyrosine kinase inhibitors for global cancer treatment background: Tkis have proven survival benefits in the'.
URL: *http://bmjopen.bmj.com/*

Hirai, H., Sootome, H., Nakatsuru, Y., Miyama, K., Taguchi, S., Tsujioka, K., Ueno, Y., Hatch, H., Majumder, P. K., Pan, B. S. & Kotani, H. (2010), 'Mk-2206, an allosteric akt inhibitor, enhances antitumor efficacy by standard chemotherapeutic agents or molecular targeted drugs in vitro and in vivo', *Molecular cancer therapeutics* **9**, 1956–1967.
URL: *https://pubmed.ncbi.nlm.nih.gov/20571069/*

Horn, H., Schoof, E. M., Kim, J., Robin, X., Miller, M. L., Diella, F., Palma, A., Cesareni, G., Jensen, L. J. & Linding, R. (2014), 'KinomeXplorer: An integrated platform for kinome biology studies'.

Humphrey, S. J., Yang, G., Yang, P., Fazakerley, D. J., Stöckli, J., Yang, J. Y. & James, D. E. (2013), 'Dynamic adipocyte phosphoproteome reveals that akt directly regulates mTORC2', *Cell Metabolism* **17**(6), 1009–1020.

King, R. (2017), The Adam and Eve Robot Scientists for the Automated Discovery of Scientific Knowledge, *in* 'APS March Meeting Abstracts', Vol. 2017 of *APS Meeting Abstracts*, p. X49.001.

King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G., Bryant, C. H., Muggleton, S. H., Kell, D. B. & Oliver, S. G. (2004), 'Functional genomic hypothesis generation and experimentation by a robot scientist', *Nature* **427**(6971), 247–252.

King, T., Butcher, S. & Zalewski, L. (2017), 'Apocrita - high performance computing cluster for queen mary university of london'.
URL: *https://zenodo.org/record/438045*

Klaeger, S., Heinzlmeir, S., Wilhelm, M., Polzer, H., Vick, B., Koenig, P. A., Reinecke, M., Ruprecht, B., Petzoldt, S., Meng, C., Zecha, J., Reiter, K., Qiao, H., Helm, D., Koch, H., Schoof, M., Canevari, G., Casale, E., Depaolini, S. R., Feuchtinger, A., Wu, Z., Schmidt, T., Rueckert, L., Becker, W., Huenges, J., Garz, A. K., Gohlke, B. O., Zolg, D. P., Kayser, G., Vooder, T., Preissner, R., Hahne, H., Tõnisson, N., Kramer, K., Götze, K., Bassermann, F., Schlegl, J., Ehrlich, H. C., Aiche, S., Walch, A., Greif, P. A., Schneider, S., Felder, E. R., Ruland, J., Médard, G., Jeremias, I., Spiekermann, K. & Kuster, B. (2017), 'The target landscape of clinical kinase drugs', *Science (New York, N.Y.)* **358**.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6542668/*

Koundouros, N., Karali, E., Tripp, A., Valle, A., Inglese, P., Perry, N. J., Magee, D. J., Virmouni, S. A., Elder, G. A., Tyson, A. L., Dória, M. L., van Weverwijk, A., Soares, R. F., Isacke, C. M., Nicholson, J. K., Glen, R. C., Takats, Z. & Poulogiannis, G. (2020), 'Metabolic fingerprinting links oncogenic pik3ca with enhanced arachidonic acid-derived eicosanoids', *Cell*

181, 1596–1611.e27.
URL: *https://pubmed.ncbi.nlm.nih.gov/32559461/*

Koundouros, N. & Poulogiannis, G. (2018), 'Phosphoinositide 3-kinase/akt signaling and redox metabolism in cancer', *Frontiers in oncology* **8**.
URL: *https://pubmed.ncbi.nlm.nih.gov/29868481/*

Kowalski, R. (1974), 'Predicate logic as a programming language'.

Kurioka, D., Takeshita, F., Tsuta, K., Sakamoto, H., Watanabe, S. I., Matsumoto, K., Watanabe, M., Nakagama, H., Ochiya, T., Yokota, J., Kohno, T. & Tsuchiya, N. (2014), 'Nek9-dependent proliferation of cancer cells lacking functional p53', *Scientific Reports 2014 4:1* **4**, 1–8.
URL: *https://www.nature.com/articles/srep06111*

Kustatscher, G., Collins, T., Gingras, A. C., Guo, T., Hermjakob, H., Ideker, T., Lilley, K. S., Lundberg, E., Marcotte, E. M., Ralser, M. & Rappsilber, J. (2022), 'Understudied proteins: opportunities and challenges for functional proteomics', *Nature Methods 2022 19:7* **19**, 774–779.
URL: *https://www.nature.com/articles/s41592-022-01454-x*

Lee, J., Ko, Y. U., Chung, Y., Yun, N., Kim, M., Kim, K. & Oh, Y. J. (2018), 'The acetylation of cyclin-dependent kinase 5 at lysine 33 regulates kinase activity and neurite length in hippocampal neurons', *Scientific Reports 2018 8:1* **8**, 1–19.
URL: *https://www.nature.com/articles/s41598-018-31785-9*

Li, J., Paulo, J. A., Nusinow, D. P., Huttlin, E. L. & Gygi, S. P. (2019), 'Investigation of proteomic and phosphoproteomic responses to signaling network perturbations reveals functional pathway organizations in yeast', *Cell reports* **29**, 2092.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7382779/*

Li, Y. F., Arnold, R. J., Tang, H. & Radivojac, P. (2010), 'The importance of peptide detectability for protein identification, quantification, and experiment design in ms/ms proteomics', *Journal of proteome research* **9**(12), 6288–6297.

Linding, R., Jensen, L. J., Ostheimer, G. J., van Vugt, M. A., Jørgensen, C., Miron, I. M., Diella, F., Colwill, K., Taylor, L., Elder, K., Metalnikov, P., Nguyen, V., Pasculescu, A., Jin, J., Park, J. G., Samson, L. D., Woodgett, J. R., Russell, R. B., Bork, P., Yaffe, M. B. & Pawson, T. (2007), 'Systematic discovery of in vivo phosphorylation networks', *Cell* **129**, 1415–1426.

Lu, Z. & Hunter, T. (2009), 'Degradation of activated protein kinases by ubiquitination', *Annual review of biochemistry* **78**, 435.
URL: */pmc/articles/PMC2776765/ /pmc/articles/PMC2776765/?report=abstract https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2776765*

López-Otín, C. & Hunter, T. (2010), 'The regulatory crosstalk between kinases and proteases in cancer', *Nature Reviews Cancer 2010 10:4* **10**, 278–292.
URL: *https://www.nature.com/articles/nrc2823*

MacCoss, M. J., Wu, C. C. & Yates, J. R. (2002), 'Probability based validation of protein identifications using a modified sequest algorithm', *Analytical Chemistry* **74**, 5593–5599.

Maeyer, D. D., Renkens, J., Cloots, L., Raedt, L. D. & Marchal, K. (2013), 'Phenetic: network-based interpretation of unstructured gene lists in e. coli', *Molecular BioSystems* **9**, 1594–1603.
URL: *https://pubs.rsc.org/en/content/articlehtml/2013/mb/c3mb25551d https://pubs.rsc.org/en/content/articlelanding/2013/mb/c3mb25551d*

Manning, G., Whyte, D. B., Martinez, R., Hunter, T. & Sudarsanam, S. (2002), 'The protein kinase complement of the human genome', *Science (New York, N.Y.)* **298**, 1912–1934.
URL: *https://pubmed.ncbi.nlm.nih.gov/12471243/*

Mao, P., Hever, M. P., Niemaszyk, L. M., Haghkerdar, J. M., Yanco, E. G., Desai, D., Beyrouthy, M. J., Kerley-Hamilton, J. S., Freemantle, S. J. &

Spinella, M. J. (2011), ‘Serine/threonine kinase 17a is a novel p53 target gene and modulator of cisplatin toxicity and reactive oxygen species in testicular cancer cells’, *The Journal of Biological Chemistry* **286**, 19381.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3103316/*

Martinez, R., Defnet, A. & Shapiro, P. (2020), ‘Avoiding or co-opting atp inhibition: Overview of type iii, iv, v, and vi kinase inhibitors’, *Next Generation Kinase Inhibitors* p. 29.
URL: */pmc/articles/PMC7359047/ /pmc/articles/PMC7359047/?report=abstract https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7359047*

Mccarthy, J. (1986), ‘Circumscription-a form of nonmonotonic reasoning’.
URL: *http://www-formal.stanford.edu/jmc/1986*

McClendon, C. L., Kornev, A. P., Gilson, M. K. & Taylora, S. S. (2014), ‘Dynamic architecture of a protein kinase’, *Proceedings of the National Academy of Sciences of the United States of America* **111**, E4623–E4631.
URL: *https://www.pnas.org/doi/abs/10.1073/pnas.1418402111*

Medicine, T. L. R. (2018), ‘Opening the black box of machine learning’, *The Lancet Respiratory* **6**, 801.
URL: *https://doi.org/10.1016/S2213-*

Miller, M. L., Jensen, L. J., Diella, F., Jørgensen, C., Tinti, M., Li, L., Hsiung, M., Parker, S. A., Bordeaux, J., Sicheritz-Ponten, T., Olhovsky, M., Pasculescu, A., Alexander, J., Knapp, S., Blom, N., Bork, P., Li, S., Cesareni, G., Pawson, T., Turk, B. E., Yaffe, M. B., Brunak, S. & Linding, R. (2008), ‘Linear motif atlas for phosphorylation-dependent signaling’, *Science Signaling* **1**.
URL: *https://www.science.org/doi/10.1126/scisignal.1159433*

Montoya, A., Beltran, L., Casado, P., Rodríguez-Prados, J. C. & Cutillas, P. R. (2011), ‘Characterization of a tio2 enrichment method for label-free quantitative phosphoproteomics’, *Methods* **54**, 370–378.

Muise, C., McIlraith, S. A., Beck, J. C. & Hsu, E. I. (2012), 'Dsharp: Fast d-dnnf compilation with sharpsat', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **7310 LNAI**, 356–361.
URL: *https://link.springer.com/chapter/10.1007/978-3-642-30353-1*$_3$6

Needham, E. J., Parker, B. L., Burykin, T., James, D. E. & Humphrey, S. J. (2019), 'Illuminating the dark phosphoproteome', *Science Signaling* **12**, 8645.
URL: *https://www.science.org/doi/10.1126/scisignal.aau8645*

Ochoa, D., Jarnuczak, A. F., Viéitez, C., Gehre, M., Soucheray, M., Mateus, A., Kleefeldt, A. A., Hill, A., Garcia-Alonso, L., Stein, F., Krogan, N. J., Savitski, M. M., Swaney, D. L., Vizcaíno, J. A., Noh, K. M. & Beltrao, P. (2020), 'The functional landscape of the human phosphoproteome', *Nature Biotechnology* **38**(3), 365–373.

O'Keefe, R. A. (1990), *The Craft of Prolog*, MIT Press, Cambridge, MA, USA.

Orre, L. M., Vesterlund, M., Pan, Y., Arslan, T., Zhu, Y., Fernandez Woodbridge, A., Frings, O., Fredlund, E. & Lehtiö, J. (2019), 'SubCellBarCode: Proteome-wide Mapping of Protein Localization and Relocalization', *Molecular Cell* **73**(1), 166–182.e7.

Oughtred, R., Rust, J., Chang, C., Breitkreutz, B. J., Stark, C., Willems, A., Boucher, L., Leung, G., Kolas, N., Zhang, F., Dolma, S., Coulombe-Huntington, J., Chatr-aryamontri, A., Dolinski, K. & Tyers, M. (2021), 'The biogrid database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions', *Protein science : a publication of the Protein Society* **30**, 187–200.
URL: *https://pubmed.ncbi.nlm.nih.gov/33070389/*

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner,

B., Fang, L., Bai, J. & Chintala, S. (2019), Pytorch: An imperative style, high-performance deep learning library, *in* 'Advances in Neural Information Processing Systems 32', Curran Associates, Inc., pp. 8024–8035.
URL: *http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf*

Pawson, T. & Scott, J. D. (n.d.), 'Protein phosphorylation in signaling - 50 Years and counting', *Trends in Biochemical Sciences* (6), 286–290.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011), 'Scikit-learn: Machine learning in python', *Journal of machine learning research* **12**(Oct), 2825–2830.

Ptacek, J., Devgan, G., Michaud, G., Zhu, H., Zhu, X., Fasolo, J., Guo, H., Jona, G., Breitkreutz, A., Sopko, R., McCartney, R. R., Schmidt, M. C., Rachidi, N., Lee, S. J., Mah, A. S., Meng, L., Stark, M. J., Stern, D. F., Virgilio, C. D., Tyers, M., Andrews, B., Gerstein, M., Schweitzer, B., Predki, P. F. & Snyder, M. (2005), 'Global analysis of protein phosphorylation in yeast', *Nature 2005 438:7068* **438**, 679–684.
URL: *https://www.nature.com/articles/nature04187*

Raedt, L. D., Kimmig, A. & Toivonen, H. (2015), 'Problog: A probabilistic prolog and its application in link discovery'.
URL: *www.ncbi.nlm.nih.gov/Entrez/*

Remy, E., Rebouissou, S., Chaouiya, C., Zinovyev, A., Radvanyi, F. & Calzone, L. (2015), 'A modeling approach to explain mutually exclusive and co-occurring genetic alterations in bladder tumorigenesis', *Cancer research* **75**, 4042–4052.
URL: *https://pubmed.ncbi.nlm.nih.gov/26238783/*

Riley, N. M. & Coon, J. J. (2016), 'Phosphoproteomics in the Age of Rapid and Deep Proteome Profiling', *Analytical Chemistry* **88**(1), 74–94.

Rodgers, G., Austin, C., Anderson, J., Pawlyk, A., Colvis, C., Margolis, R. & Baker, J. (2018), 'Glimmers in illuminating the druggable genome', *Nature Reviews Drug Discovery 2018 17:5* **17**, 301–302.
URL: *https://www.nature.com/articles/nrd.2017.252*

Roper, K., Abdel-Rehim, A., Hubbard, S., Carpenter, M., Rzhetsky, A., Soldatova, L. & King, R. D. (2022), 'Testing the reproducibility and robustness of the cancer biology literature by robot', *Journal of the Royal Society Interface* **19**.
URL: *https://royalsocietypublishing.org/doi/10.1098/rsif.2021.0821*

Roskoski, R. (2016), 'Classification of small molecule protein kinase inhibitors based upon the structures of their drug-enzyme complexes', *Pharmacological Research* **103**, 26–48.

Rudolph, J. D., de Graauw, M., van de Water, B., Geiger, T. & Sharan, R. (2016), 'Elucidation of Signaling Pathways from Large-Scale Phosphoproteomic Data Using Protein Interaction Networks', *Cell Systems* **3**(6), 585–593.e3.

Saez-Rodriguez, J., Simeoni, L., Lindquist, J. A., Hemenway, R., Bommhardt, U., Arndt, B., Haus, U. U., Weismantel, R., Gilles, E. D., Klamt, S. & Schraven, B. (2007), 'A logical model provides insights into t cell receptor signaling', *PLOS Computational Biology* **3**, e163.
URL: *https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.0030163*

Sato, T. (1995).

Sato, T. (2009), 'Generative modeling by prism'.
URL: *http://sato-www.cs.titech.ac.jp/*

Sato, T. & Kameya, Y. (2011), 'Parameter learning of logic programs for symbolic-statistical modeling', *Journal Of Artificial Intelligence Research* **15**, 391–454.
URL: *http://arxiv.org/abs/1106.1797*

Sharifpoor, S., Ba, A. N. N., Young, J. Y., van Dyk, D., Friesen, H., Douglas, A. C., Kurat, C. F., Chong, Y. T., Founk, K., Moses, A. M. & Andrews, B. J. (2011), 'A quantitative literature-curated gold standard for kinase-substrate pairs', *Genome Biology* **12**, R39.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3218865/*

Sinha, S., Eisenhaber, B., Jensen, L. J., Kalbuaji, B., Eisenhaber, F., Sinha, S., Eisenhaber, B., Kalbuaji, B., Eisenhaber, F. & Jensen, L. J. (2018), 'Darkness in the human gene and protein function space: Widely modest or absent illumination by the life science literature and the trend for fewer protein function discoveries since 2000', *PROTEOMICS* **18**, 1800093.
URL: *https://onlinelibrary.wiley.com/doi/full/10.1002/pmic.201800093 https://onlinelibrary.wiley.com/doi/abs/10.1002/pmic.201800093 https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/10.1002/pmic.201800093*

Sokol, K. & Flach, P. (2022), 'Explainability is in the mind of the beholder: Establishing the foundations of explainable artificial intelligence; explainability is in the mind of the beholder: Establishing the foundations of explainable artificial intelligence'.

Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K. P., Kuhn, M., Bork, P., Jensen, L. J. & Mering, C. V. (2015), 'String v10: protein-protein interaction networks, integrated over the tree of life', *Nucleic acids research* **43**, D447–D452.
URL: *https://pubmed.ncbi.nlm.nih.gov/25352553/*

Terfve, C. D., Wilkes, E. H., Casado, P., Cutillas, P. R. & Saez-Rodriguez, J. (2015), 'Large-scale models of signal propagation in human cells derived from discovery phosphoproteomic data', *Nature Communications* **6**(1), 1–11.

Tognetti, M., Gabor, A., Yang, M., Cappelletti, V., Windhager, J., Rueda, O. M., Charmpi, K., Esmaeilishirazifard, E., Bruna, A., de Souza, N., Caldas, C., Beyer, A., Picotti, P., Saez-Rodriguez, J. & Bodenmiller, B.

(2021), 'Deciphering the signaling network of breast cancer improves drug sensitivity prediction', *Cell Systems* **12**, 401–418.e12.

Traynard, P., Tobalina, L., Eduati, F., Calzone, L. & Saez-Rodriguez, J. (2017), 'Logic modeling in quantitative systems pharmacology', *CPT: Pharmacometrics Systems Pharmacology* **6**, 499–511.
URL: *https://onlinelibrary.wiley.com/doi/full/10.1002/psp4.12225*

Turdo, A., D'Accardo, C., Glaviano, A., Porcelli, G., Colarossi, C., Colarossi, L., Mare, M., Faldetta, N., Modica, C., Pistone, G., Bongiorno, M. R., Todaro, M. & Stassi, G. (2021), 'Targeting phosphatases and kinases: How to checkmate cancer', *Frontiers in Cell and Developmental Biology* **9**.
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8581442/*

Türei, D., Valdeolivas, A., Gul, L., Palacio-Escat, N., Ivanova, O., Gábor, A., Módos, D., Korcsmáros, T. & Saez-Rodriguez, J. (2020), 'Integrated intra- And intercellular signaling knowledge for multicellular omics analysis', p. 2020.08.03.221242.
URL: *https://doi.org/10.1101/2020.08.03.221242*

Usman, M. W., Gao, J., Zheng, T., Rui, C., Li, T., Bian, X., Cheng, H., Liu, P. & Luo, F. (2018), 'Macrophages confer resistance to pi3k inhibitor gdc-0941 in breast cancer through the activation of nf-b signaling', *Cell Death Disease 2018 9:8* **9**, 1–12.
URL: *https://www.nature.com/articles/s41419-018-0849-6*

Vella, V., Giamas, G. & Ditsiou, A. (2021), 'Diving into the dark kinome: lessons learned from lmtk3', *Cancer Gene Therapy 2021 29:8* **29**, 1077–1079.
URL: *https://www.nature.com/articles/s41417-021-00408-3*

Whelan, K. E. & King, R. D. (2008), 'Using a logical model to predict the growth of yeast', *BMC Bioinformatics* **9**, 1–16.
URL: *https://link.springer.com/article/10.1186/1471-2105-9-97*

Wielemaker, J., Schrijvers, T., Triska, M. & Lager, T. (2012), 'Swi-prolog', *Theory and Practice of Logic Programming* **12**, 67–96.
URL: *https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/abs/swiprolog/1A18020C8CA2A2EE389BE6A714D6A148*

Yueh, C., Rettenmaier, J., Xia, B., Hall, D. R., Alekseenko, A., Porter, K. A., Barkovich, K., Keseru, G., Whitty, A., Wells, J. A., Vajda, S. & Kozakov, D. (2019), 'Kinase atlas: Druggability analysis of potential allosteric sites in kinases'.
URL: *https://kinase-atlas.bu.edu*