Challenges and solutions in drug-target interaction prediction

Ezzat, Ali

2018

Ezzat, A. (2018). Challenges and solutions in drug-target interaction prediction. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/75771>

<https://doi.org/10.32657/10356/75771>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**CHALLENGES AND SOLUTIONS
IN
DRUG-TARGET INTERACTION PREDICTION**

ALI EZZAT

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2018

Acknowledgements

I would like to thank my supervisor, *Assoc. Prof. Li Xiaoli*, from whom I have learned a lot and who was generous with his continuous and kind support throughout the course of my PhD. Similar thanks go to my co-supervisor, *Assoc. Prof. Kwoh Chee Keong*, who regularly assisted with helpful suggestions and guidance. Your efforts with me are deeply appreciated, and I could not thank you both enough.

I would also like to thank *Wu Min* and *Peilin Zhao* from the Institute for Infocomm Research, A*STAR who greatly assisted with my PhD thesis, especially in the early stages. You have my sincere thanks.

Lastly, I would like to thank my family and friends for constantly cheering me on and pushing me to go forward in this long and arduous PhD journey. I could not have done it without you.

Thanks go to you all.

Table of Contents

Chapter 1	Introduction.....	8
1.1	Computational Methods in Drug Discovery	12
1.2	Drug-Target Interaction Prediction.....	13
1.3	Problem Definition	16
1.4	Thesis Roadmap.....	18
Chapter 2	Literature Review	20
2.1	Data: Representation and Types	21
2.1.1	Interaction Data.....	21
2.1.2	Drug and Target Data.....	23
2.1.3	An Important Benchmark Dataset.....	26
2.2	Neighborhood Methods	27
2.2.1	Nearest Profile & Weighted Profile	27
2.2.2	Similarity Rank-based Predictor (SRP).....	28
2.3	Bipartite Local Models	29
2.3.1	SVM-based Bipartite Local Models.....	30
2.3.2	Laplacian Regularized Least Squares (LapRLS)	31
2.3.3	Regularized Least Squares (RLS-avg & RLS-kron)	32
2.3.4	Bipartite Local Models with Neighbor-based Interaction Profile Inferring	33
2.3.5	Regularized Least Squares with Weighted Nearest Neighbors (RLS-WNN).....	34
2.4	Network Diffusion	35
2.4.1	Network-based Inference (NBI).....	35
2.4.2	Heterogeneous Graph-based Inference (HGBI)	35
2.4.3	Network Random Walk with Restart on Heterogeneous Network (NRWRH)	37
2.4.4	Probabilistic Soft Logic (PSL)	38
2.4.5	Determine All Simple Paths, Find Interactions (DASPfind).....	38
2.5	Matrix Factorization	39
2.5.1	Kernelized Bayesian Matrix Factorization with Twin Kernels (KBFM2K).....	40
2.5.2	Probabilistic Matrix Factorization (PMF)	41
2.5.3	Collaborative Matrix Factorization (CMF)	42
2.5.4	Neighborhood Regularized Logistic Matrix Factorization (NRLMF).....	44
2.5.5	Dual-Network Integrated Logistic Matrix Factorization (DNILMF)	46
2.6	Feature-based Classification	48
2.6.1	Incremental and Forward Feature Selection.....	49
2.6.2	Random Forest and Support Vector Machines.....	49
2.6.3	Fuzzy K Nearest Neighbors (Fuzzy KNN)	50
2.6.4	Similarity-based Inference of drug-TARgets (SITAR).....	50
2.6.5	Chemical substructures-protein domains correlation model	51
2.6.6	Support Vector Machines & Minwise Hashing	52
2.7	Summary	53

Chapter 3	Graph Regularized Matrix Factorization	55
3.1	Data	57
3.2	Weighted K Nearest Known Neighbors	57
3.3	Graph-Regularized Matrix Factorization	59
3.3.1	Sparsification of the Similarity Matrices	60
3.3.2	Low-Rank Approximation	61
3.3.3	Regularization	61
3.3.4	Initialization of A and B	63
3.3.5	Alternating Least Squares	64
3.4	Weighted GRMF	64
3.5	Experimental Results	65
3.5.1	Parameter Settings	67
3.5.2	Interaction Prediction under S_2	67
3.5.3	Interaction Prediction under S_3	68
3.5.4	Interaction Prediction with WKNKN	70
3.5.5	Sensitivity Analysis	70
3.5.6	General Comments	74
3.6	Predicting Novel Interactions	75
3.7	Interaction Prediction under S_1	78
3.8	Conclusion	79
Chapter 4	Class Imbalance-Aware Ensemble Learning	81
4.1	Data	82
4.1.1	Drug-Target Interaction Data	82
4.1.2	Data Representation	82
4.2	Methods	84
4.2.1	Our Proposed Algorithm	85
4.2.2	Within-Class Imbalance	87
4.2.3	Between-Class Imbalance	89
4.3	Results and Discussion	90
4.3.1	Experimental Settings	90
4.3.2	Cross Validation Experiments	91
4.3.3	Predicting Interactions for New Drugs and Targets	93
4.4	Conclusion	96
Chapter 5	Dimensionality Reduction and Ensemble Learning	97
5.1	Data	98
5.2	Dimensionality Reduction	99
5.2.1	Singular Value Decomposition (SVD)	100
5.2.2	Partial Least Squares (PLS)	100
5.2.3	Laplacian Eigenmaps (LapEig)	101
5.3	Ensemble of Decision Trees with Different Negative Sets (EnsemDT)	101

5.4	Ensemble of Kernel Ridge Regression Learners (EnsemKRR).....	103
5.4.1	The <i>RLS-avg</i> Base Learner	103
5.4.2	Extending the <i>RLS-avg</i> Base Learner using <i>WNN</i>	104
5.4.3	Overview of <i>EnsemKRR</i>	106
5.5	Experimental Results and Discussion.....	106
5.5.1	Comparison: Feature-based Methods.....	108
5.5.2	Varying Training Sets for EnsemDT's Base Learners	108
5.5.3	Comparison: Similarity-based Methods.....	110
5.5.4	Effect of GIP Kernels and WNN	111
5.5.5	Sensitivity analysis	112
5.5.6	Running times	114
5.5.7	Experiments with Another Dataset	117
5.6	Extension for Feature-based Methods	120
5.6.1	Modification to the Sampling Procedure	122
5.6.2	Experimental Design.....	123
5.6.3	Prediction Results for Feature-based Methods.....	124
5.6.4	Comparison with EnsemKRR	125
5.7	Final Remarks and Conclusion	127
Chapter 6	Conclusion and Outlook	130
6.1	Utilizing More Information.....	134
6.2	Ensemble Learning	136
6.3	Deep Learning.....	137
6.4	Absence of Reliable Negatives	139
6.5	Big Data	140
Appendix A:	List of Publications	142
Appendix B:	Notations & Variables	143
Appendix C:	Bibliography	145

Abstract

When a drug is developed, it is designed so that it interacts with a specific target of interest in order to achieve the desired therapeutic effect. However, it is quite common to later find that the developed drug also interacts with multiple other targets that were not intended during its development. This is interesting because if a drug can interact with multiple targets, then it may have more than one therapeutic effect. Therefore, this provides a clear motivation for discovering new interactions for existing drugs.

In drug discovery, an important task called *drug-target interaction prediction* detects such interactions on a large scale by screening many drugs and targets simultaneously. While there are wet-lab techniques for discovering these interactions, the focus of this thesis is particularly on *computational* drug-target interaction prediction. Specifically, we investigate methods that discover new interactions based on prior knowledge of existing drugs and their experimentally confirmed targets (i.e. machine learning). Throughout this thesis, we identified and addressed 4 outstanding problems in drug-target interaction (DTI) prediction. Having addressed these problems, we were able to enhance the prediction performance and outperform relevant state-of-the-art methods.

Firstly, DTI prediction methods have difficulty predicting interactions involving *new* drugs or targets for which there are no known interactions. To predict interactions, we developed two matrix factorization methods that utilize graph regularization. In addition, considering that many of the non-occurring edges in the bipartite DTI network are actually unknown or missing cases, we developed a preprocessing step to enhance predictions in the “new drug” and “new target” cases by adding edges with intermediate interaction likelihood scores. In our experiments, our methods performed better than the state-of-the-art methods and was found to predict interactions reasonably well.

Secondly, class imbalance is an issue that is prevalent across all DTI datasets. Class imbalance can be divided into two sub-problems, namely *between-class* and *within-class*

imbalance. Between-class imbalance refers to the imbalance ratio between interacting and non-interacting drug-target pairs; this degrades prediction performance due to the bias in prediction results towards the majority class (i.e. the non-interacting pairs), leading to more prediction errors in the minority class (i.e. the interacting pairs). Within-class imbalance refers to the imbalance between the sizes of sub-groups (types) of interactions; this biases the predictions towards the bigger and more well-represented sub-groups, leading to more errors in the smaller groups. Here, we developed an ensemble learning method that incorporates techniques to address the issues of between-class imbalance and within-class imbalance. Experiments show that the proposed method improves results over 4 state-of-the-art methods.

Thirdly, there are DTI datasets where the feature sets for representing the drugs and targets (and, by extension, the drug-target pairs) are of a high dimensionality. High dimensionality of the data may lead to much longer running times for the prediction models. Furthermore, there may be redundancy in the features which may also lead to degradation in prediction performance. In this work, we used dimensionality reduction to deal with both of these issues, and we additionally used ensemble learning to improve the prediction performance further. As base learners for the ensemble, we selected two classifiers, namely *Decision Tree* and *Kernel Ridge Regression*, resulting in two variants of ensemble models, *EnsemDT* and *EnsemKRR*, respectively. Experimental results show that our proposed methods are indeed successful.

Lastly, there is a concept called *differential representation bias* that has an impact on the prediction performance of DTI prediction methods. Specifically, differential representation bias refers to how much a drug (or target) appears in the positive training data as opposed to the negative data. Bearing this concept in mind, we experimented with the way that the negative training data is sampled prior to training the prediction model. We found that our modified sampling procedure produced significant improvements in DTI prediction performance.

Chapter 1

Introduction

Drugs are chemical compounds that bring about desired therapeutic effects via interactions with specific *targets*. Drugs may either directly target the cause of the disease (i.e. disease-associated genes), or they may target other gene products “whose activities can be perturbed to counteract the symptoms of disease-causing proteins” [1].

Developing a drug to treat a disease of interest is a risky endeavor that is so costly and time-consuming [2]. It is a process that involves many steps, including the selection of a proper target (usually a protein whose gene is a disease-associated one) to be targeted by the drug being developed, identifying a lead compound to bind with the selected target, and lead optimization to improve its efficacy, specificity and safety [3]. More importantly, however, drugs must also go through a phase known as the *clinical trials* before they are released into the market. Unfortunately, many drugs do not pass this critical phase. Based on estimates computed from drug development pipelines of many large pharmaceutical companies, the “clinical approval success rate” was only 19% [4]. A brief overview of the drug discovery is given in Figure 1.1 below.

Figure 1.1: Schema of a standard drug development pipeline

In rational drug design, the philosophy of “*one gene–one drug–one disease*” emerged as a result of the increased ability to isolate the disease-associated gene and to create a drug that is maximally selective and binds only with this gene. The rationale behind creating a maximally selective drug that binds to a single target is that, by avoiding unintended off-

targets, undesirable side effects would be reduced, and thus safer, more effective drugs can be realized [5]. However, although the hypothesis is reasonable, the field of drug development has witnessed an increased number of failed endeavors coinciding with the increased adoption of the hypothesis, which suggests a serious flaw in the philosophy.

Moreover, studies of existing drugs – i.e. ones that have successfully reached the market and are deemed safe to use – tell a story that further challenges the above philosophy. It was found that many of the existing drugs, besides their intended targets, interact with other off-targets as well [6]. This drug *promiscuity* (i.e. interaction with multiple targets) has led to the rise of a new trend in drug discovery, namely *drug repositioning*, which is the reuse of existing drugs for new indications. In other words, existing drugs may be used to treat diseases other than those that they were originally developed for [7].

Drug repositioning has been gaining popularity due to its ability to mitigate costs and risks that are inherent in the task of developing a novel drug [8-12]. Repurposing an approved drug is quite convenient as it has already passed clinical trials (i.e. it has been extensively studied and verified as safe to use), which thus accelerates the drug discovery process and increases the chances of success [13].

Drugs that interact with multiple targets are quite common [5, 6], so it is always worth considering to use drug repositioning as a strategy to find treatments for new diseases [14]. One famous example is that of Gleevec (imatinib mesylate) which was originally thought to interact only with the Bcr-Abl fusion gene associated with leukemia; Gleevec was later found to also interact with PDGF and KIT, eventually leading it to be repositioned for treating gastrointestinal stromal tumors as well [15, 16]. This example shows that a drug's *promiscuity* (i.e. interaction with multiple targets) may contribute to its *polypharmacology* (i.e. having multiple therapeutic effects). This is clear motivation for attempting to discover new drug-target interactions for existing drugs.

On a related note, dual-gene knockout experiments have been conducted (where pairs of genes are selected for concurrent deletion) and an interesting finding was revealed; there

happened to be pairs of genes where the isolated deletion of either gene of a pair demonstrated no effect, but the simultaneous deletion of both genes of the same pair caused synthetic lethality or sickness, which implies that the pair's genes compensate for one another [17]. This suggests that if multiple genes were found essential for mediating a certain disease, a drug may possibly have to target all such genes to be effective against this disease.

Amiodarone – known to be the most effective anti-arrhythmic compound – also turned out to be the most promiscuous of all anti-arrhythmics [18], which suggests that drug promiscuity might be a virtue. Going back to the example of Gleevec, it appears that Gleevec's success was not just that it binds with the Bcr-Abl gene associated with leukemia but also that it is promiscuous and binds with at least two other targets [6]. These two examples, along with the results of the dual-gene knockout experiments, lend support to the point that promiscuity is a necessary trait for many drugs to be effective [19, 20]. That said, many of the existing drugs can be expected to be promiscuous, which encourages the investigation of said drugs for undiscovered targets that they may also be binding with; such targets may provide opportunities to reuse drugs for other indications.

Recently, there has been an increasing trend in utilizing drug repositioning in cancer therapy [21]. A common problem in cancer therapy is that of *chemoresistance* where target genes show resistance to the administered chemical compound (or drug), thus negating the intended therapeutic effect [22]. Drug repositioning offers a solution by repurposing old drugs to address the resistant tumor being treated [23]. Many of these old drugs have been repositioned from *non-cancer* indications [24]. This is fortunate since conventional cancer chemotherapy drugs typically have side effects that can substantially reduce the quality of life for cancer patients, whereas most non-cancer drugs relatively have little or tolerable side effects [25], which further promotes drug repositioning for use in future anti-cancer therapy efforts.

While drug repositioning is more commonly applied to successful drugs that have reached the market, it is also possible for a *failed* drug to be repositioned. Thalidomide is one such drug [26]. It was initially developed and prescribed for pregnant women having morning sickness. Unfortunately, women who had this drug during the first trimester of their pregnancy gave birth to babies having severe skeletal defects. Thalidomide was thus removed from the market. Later, however, it was found to be effective in treating a condition called *erythema nodosum laprosum* (ENL). At the present time, Thalidomide is the primary drug used for the treatment of ENL. Regarding women who are diagnosed with ENL, they now go through birth control programs before consuming the drug. This is an example of a drug that has (quite tragically) failed in the past but has since been repositioned to treat a disease that it was not initially intended for. Repositioning of failed drugs is termed *drug rescue* and, in general, it is a strategy that is used only in cases of severe or life-threatening diseases. For example, if the drug being repositioned was originally failed for having side effects that were then deemed unacceptable, these same side effects could be tolerated if using the drug would save the patient's life [27].

Finally, two particular classes of diseases where drug repositioning would be particularly appealing as a drug development strategy are *rare* or *neglected* diseases [28]. These diseases are problematic to develop drugs for as they typically have a low return-on-investment, making *de novo* drug discovery (i.e. developing a novel drug) out of the question. Drug repositioning offers a way to make the development of drugs for such diseases financially feasible. Examples include Tamoxifen [29] (a drug used in treating breast cancer) which was repositioned to treat the rare disease, *leishmaniasis*, and Closantel [11] (a veterinary anthelmintic) which was used to treat the neglected tropical disease, *onchocerciasis* (*river blindness*).

To conclude, a drug repositioning strategy may be employed to find a suitable drug that would bind with the intended targets. However, there is no guarantee that there will be existing drugs on the market that can bind with them. Therefore, drug repositioning cannot be used for all cases of diseases for which drug discovery processes may be

initiated, and traditional de novo drug discovery (using the pipeline in Figure 1.1) would need to be utilized.

1.1 Computational Methods in Drug Discovery

In the field of drug discovery, many different computational methods are being used. Which methods get selected for use depends on a few factors. First, the method being used will heavily depend on the phase of the drug development pipeline that a pharmaceutical company is currently in [3]. In the earlier phases of drug development, efforts would be more inclined towards studying the disease in question; e.g. tasks may be initiated for determining disease-associated genes or observing differences in gene expressions between healthy and infected cells. A later phase in the drug development process will likely involve a search for a suitable lead compound to optimize. With the help of similarity-search algorithms, one may obtain said compound by browsing the PubChem database [30] for results of previously conducted BioAssay experiments between many chemical compounds and targets. After a promising lead compound has been identified, QSAR (Quantitative Structure-Activity Relationships) studies can be initiated to ensure that the lead compound has *drug-like* properties. More precisely, QSAR techniques can help quantify different properties of interest known as *ADMET* properties (Absorption, Distribution, Metabolism, Excretion, Toxicology) which are important to optimize for a drug in order for it to be successful [31].

Another factor that may affect the choice of computational methods used in the drug discovery process is the kinds of information available [32]. For example, if we know a gene associated with a disease, we may opt to use a molecular docking approach if the gene's 3D structure is available [33]. If we have access to gene expression data from prior disease omics experiments (e.g. NCBI-Gene Expression Omnibus, CCLE Cancer Cell Line Encyclopedia [34], etc.), these data may be processed via signature-based methods to discover unknown off-targets or unknown disease mechanisms.

Drug discovery utilizes techniques from both the bioinformatics and cheminformatics realms. Bioinformatics techniques are those that process information from the biological side such as gene expressions, genomic sequences, metabolic pathways, protein-protein interactions and so on. On the other hand, cheminformatics methods deal with information from the chemical side such as the chemical structures of compounds and drugs, their side effects and ADMET properties among others. Since drug discovery projects typically involve both biological and chemical data, it is no surprise that techniques from both fields of bioinformatics and cheminformatics are regularly consulted. As long as advances in both of the two fields continue to take place, drug discovery projects will continue to benefit from them in the form of accelerated, more effective and more cost-efficient drug development pipelines with lower attrition rates.

1.2 Drug-Target Interaction Prediction

Drug-target interaction prediction is a task in drug discovery that consists of simultaneously screening many drugs and targets in search of interactions on a large scale, which is useful in finding new, previously unknown targets for existing drugs. In cases of serendipity, the new targets may result in unexpected therapeutic effects, thus providing excellent opportunities for repositioning of the drugs involved [35]. Alternatively, they may be *off-targets* that are responsible for undesired side-effects [36], which would also be useful information.

Furthermore, the usefulness of drug-target interaction prediction is not exclusive to existing drugs; it can also help discover potential drug candidates among chemical compounds that have not been previously used as drugs. Since drugs are essentially chemical compounds themselves, methods that can predict targets for established drugs may also be used to determine targets for all chemical compounds in general. This is useful and worth noting because there are currently over 90 million chemical compounds in the PubChem database; for most of those compounds (~97%), their interaction

profiles with proteins are unknown [37]. Drug-target interaction prediction can help identify target proteins for these compounds.

Similarly, starting with a target of interest, one can perform drug-target interaction prediction to determine compounds that interact with it, regardless of whether they are existing drugs on the market or chemical compounds that have not been purposed as drugs yet.

Drug-target interaction prediction is typically performed in the lead identification step of the discovery process (see Figure 1.1) when the target for a disease has already been established. Not only will potential lead compounds be identified for the target, but off-targets of these identified compounds will be detected as well.

Prediction of drug-target interactions can be achieved by means of experimental wet-lab techniques. Although reliable in predicting interactions, these techniques involve tedious and time-consuming work. Computational methods, on the other hand, are convenient to use as they can be used to quickly suggest promising leads to follow in the drug discovery process. This substantially narrows down the drug-target interaction space to be investigated by their wet-lab counterparts.

Furthermore, an advantage of computational methods is that they can make use of the information on drugs and targets that is stored in online databases. Such databases include extensive information on known drugs (e.g. chemical structures, side effects, ATC codes, etc.) and their targets (e.g. genomic sequences, gene ontology, etc.) as well as the involved interactions (e.g. permanent or transient, binding affinity, etc.). This wealth of information can be utilized, via computational algorithms, to predict new undiscovered interactions and to gain insights on drug-target interactions in general.

Generally speaking, computational methods for predicting interactions are not meant to replace their wet-lab counterparts, but rather to complement them. For example, computational methods can help detect issues long before they become outstanding problems, thus saving money, time and effort that would have been otherwise wasted.

Alternatively, based on the collective knowledge that computational methods have access to (from online databases and such), they can suggest promising leads to pursue or detect interesting patterns. However, wet-lab techniques will always be ultimately used to verify any findings or suggestions produced by the accompanying computational methods.

Computational methods for predicting drug-target interactions can be divided into three categories of methods:

1. *Ligand-based* approaches depend on the concept that similar molecules tend to share similar properties and usually bind similar proteins [38]. In other words, they predict interactions using the similarity between the proteins' ligand sets [39]. Note that ligands are *small molecules* (same as most drugs), which makes them relevant to drug-target interaction prediction.
2. *Docking* approaches take the 3D structures of a drug and a protein and run a simulation to determine whether they would interact [40, 41].
3. *Chemogenomic* approaches utilize information from both the drug (chemical) and target (genomic) sides simultaneously to perform prediction. They generally use *machine learning* techniques to perform predictions (an illustration of a typical machine learning procedure is given in Figure 1.2).

Chemogenomic approaches for predicting interactions have particularly seen a lot of success, and so many methods of this category have appeared recently (see the literature review in Chapter 2 for more details). Moreover, chemogenomic approaches do not share the disadvantages that are in the traditional ligand-based and docking-based approaches. In ligand-based approaches, when the sets of known ligands for a pair of proteins are of insufficient sizes, prediction results can become unreliable [42]; specifically, the product of the ligand set sizes for a pair of proteins is recommended to be above 100 for the computed similarity between them to be reliable. This is an issue since many proteins have too few (i.e. <5) known ligands [43]. As for docking approaches, they depend on the 3D structures of the proteins being available in order to operate, which is a problem

because there are proteins for which the 3D structure is not known. For example, many drug targets are membrane proteins [1] for which the determination of the 3D structure is still challenging [44].

On the other hand, an advantage of chemogenomic approaches is that they can work with kinds of data that are *widely abundant* to perform prediction. For example, in [45], information used for prediction consisted of chemical structure graphs and genomic sequences for the drugs and targets, respectively, which are kinds of information that are always available and easy to obtain from publicly accessible online databases. As such, this thesis focuses on computational methods that belong to the chemogenomic category.

Figure 1.2: Typical machine learning procedure

1.3 Problem Definition

In this thesis, the *problem* is to **perform computational drug-target interaction prediction on a global scale using chemogenomic approaches**. More specifically, if drugs and their known binding targets are modeled as a bipartite network with edges linking drug-target pairs that are known to interact (see Figure 1.3 for an illustration), then the goal is to predict the missing links in this graph as accurately as possible.

In other words, the problem is to infer novel interactions based on known interactions. That is, based on links (edges) that already exist in the given drug-target interaction (DTI) network, we try to find new links that are not in the network (a problem formally known as the link prediction problem [46]). Note that while interactions (or edges) in the DTI network are true, non-interactions (or non-edges) are not necessarily true. In fact, we are expecting a portion of these non-interactions to be actual interactions, and it is these interactions that DTI prediction methods aim to detect.

Figure 1.3: Bipartite DTI network. Drugs and targets are red circles and blue squares, respectively.

Dashed lines are undiscovered interactions that need to be detected.

Note that drugs are of two types, *small molecules* (i.e. chemically synthesized compounds with low molecular weight) and *biotech drugs* (i.e. drugs manufactured in or extracted from biological sources), while targets could be either *nucleic acids* (i.e. DNA/RNA) or *proteins*. In the context of this thesis, **drugs are small molecules** and **targets are proteins** since they constitute the majority of drugs and targets, respectively.

Having stated the problem of interest, it is now time to move on to the *objective* of the thesis. The aim of this work was to **enhance the prediction performance of computational chemogenomic methods for predicting drug-target interactions**. To do so, we performed a literature review on the current state-of-the-art prediction methods and identified a number of challenging issues that were prevalent in them. We then addressed these issues and were able to improve prediction performance beyond that of the competing methods as demonstrated in the chapters that follow.

One of the challenges in DTI prediction is that true non-interactions (i.e. drug-target pairs that do *not* interact) are not typically reported in the literature [47, 48], so there is no readily available list of confident non-interactions. Another issue in DTI prediction is that there are *orphan* compounds or targets (i.e. those that have no known interactions) that are difficult to predict interactions for [49]. These issues, along with others, have been visited and addressed during our work that is presented in the following chapters.

1.4 Thesis Roadmap

The rest of this document is organized as follows. Chapter 2 includes a comprehensive up-to-date literature review of the current state-of-the-art methods and their different categories. This chapter also gives an overview on the different kinds of data that have been used in these methods as well as where they may be obtained.

In the remaining chapters, a different challenge is addressed in each chapter and a proposed method for addressing this challenge is described.

Chapter 3 presents *WGRMF* (Weighted Graph Regularized Matrix Factorization), a matrix factorization-based method for predicting drug-target interactions. It uses graph regularization terms to learn a manifold for label propagation. In addition, a preprocessing method was developed to enhance predictions in the *new drug* and *new target* cases (i.e. the orphan nodes in the DTI network) by adding edges with intermediate interaction likelihood scores prior to prediction.

Chapter 4 presents an ensemble method that aims to deal with the *class imbalance* that is prevalent in the data. Class imbalance exists in two forms. *Between-class* imbalance refers to the imbalance ratio between interactions and non-interactions where interactions are the minority and non-interactions are the majority. *Within-class* imbalance refers to the imbalance between the sizes of sub-groups (types) of interactions, which biases the predictions towards the bigger and more well-represented sub-groups,

leading to more errors in the smaller groups. The ensemble method presented in this chapter incorporates techniques to address both forms of class imbalance.

Chapter 5 presents methods that use dimensionality reduction to deal with high dimensionality datasets that lead to long running times for prediction models as well as deal with any redundancy in the features that may lead to degradation in prediction performance. Ensemble learning was also leveraged to improve the prediction performance further. The work in this chapter resulted in two different variants of ensemble models, *EnsemDT* and *EnsemKRR*, which use *Decision Tree* and *Kernel Ridge Regression* as base learners, respectively.

The method proposed in Chapter 5 is then extended by an investigation on the concept of *differential representation bias* and how it impacts the prediction performance of DTI prediction methods. Specifically, differential representation bias refers to how much a drug (or target) appears in the positive training data as opposed to the negative data. For example, assuming the training data accurately represents the general population, if an object appears more in the positive training data than in the negative training data, then the prediction method is more likely to correctly predict test pairs involving this object as positive, which is usually true [50]. Bearing this concept in mind, we modified the way that the negative training data are sampled prior to training the prediction model to generally improve the prediction performance.

Finally, we conclude the thesis with Chapter 6 where an outlook is provided for the topic of drug-target interaction prediction and future research directions are discussed.

Chapter 2

Literature Review

This chapter presents an overview on drug-target interaction prediction using computational chemogenomic methods. Many such methods have been developed over the last decade which we briefly describe here and categorize based on the techniques that they use for prediction.

In the upcoming sections, we start by mentioning the data that one would require in order to train the predictive models, and then we provide a survey on the different algorithms for predicting drug-target interactions. An overview of the different categories of prediction methods presented in this chapter is given in Table 2.1 below.

Categories	Category Description
Neighborhood Methods	Neighborhood methods employ relatively simple similarity functions to perform predictions.
Bipartite Local Models	Bipartite local models perform two sets of predictions, one from the drug side and one from the target side, and then aggregate these predictions to give the final prediction scores.
Network Diffusion	Network diffusion methods investigate graph-based techniques to predict new interactions.
Matrix Factorization	Matrix Factorization finds two latent feature matrices that, when multiplied together, reconstruct the interaction matrix.
Feature-based Classification	Feature-based classification methods are those that need the drug-target pairs to be explicitly represented as fixed-length feature vectors.

Table 2.1: The categories of the different methods for predicting drug-target interactions

2.1 Data: Representation and Types

To train a classifier for predicting drug-target interactions, a list of known drug-target interactions is required. In other words, we need to know which drugs and targets interact and which do not. Data for representing the drugs and targets involved are also needed. An overview of a typical drug-target interaction (DTI) prediction task is given in Figure 2.1.

Figure 2.1: Flowchart of a standard DTI prediction task using a chemogenomic prediction method

2.1.1 Interaction Data

Information on known drug-target interactions needs to be gathered since a classifier will be trained on these known interactions to predict the new interactions. Such information can be found in publicly available online databases that store information on drugs and their known targets. Examples of databases that have been used in previous work include KEGG [51], DrugBank [52], ChEMBL [53] and STITCH [54]. For an exhaustive list of such databases, please refer to [55].

Interaction data gathered from these databases are usually formatted into an interaction (adjacency) matrix between drugs and targets (see Section 2.1.3 for an example). This matrix corresponds to a bipartite graph where nodes represent drugs and targets, and edges connect drug-target pairs that interact (e.g. see Figure 1.3).

Below, we present a brief overview for some of the most widely-used repositories for drug-target interactions. Note that the resources described below are publicly available and free to use.

- DrugBank [52] (<https://www.drugbank.ca/>)

DrugBank is advertised as both a bioinformatics and a cheminformatics resource, meaning that it has information on both the targets and drugs. Examples of information supplied for a drug include its chemical structure, its ATC (Anatomical Therapeutic Chemical) codes and indicated diseases/symptoms. As for targets, supplied information include a short summary of its function, gene name and corresponding ID in the Uniprot database (where the protein's sequence and gene ontology information can be accessed among other things). The current version (v5.0.11, released Dec. 2017) contains 11,002 drug entries and 4,910 non-redundant proteins. It is a public resource that is widely used by pharmacists, medicinal chemists and the pharmaceutical industry.

- KEGG [51] (<http://www.kegg.jp/>)

KEGG (short for Kyoto Encyclopedia of Genes and Genomes) is a database resource that is geared towards understanding biological systems (e.g. cell, organism, ecosystem, etc.) from molecular-level information that is generated from high-throughput experimental technologies (e.g. genome sequencing). A section of this database, KEGG DRUG, is concerned with drugs. Each record in the KEGG DRUG database has information on a specific drug (e.g. chemical structure, indicated diseases, efficacy, etc.) including the list of targets that it binds to. For more information on the drug's targets, the user may consult the KEGG GENES section of the database. Currently, KEGG contains 10,490 drugs. As for the targets, it contains protein and RNA records for many organisms (e.g. mouse, *c. elegans*, etc.). At this time, *human (homo sapien)* proteins and RNAs are numbered at 20,380 and 19,172, respectively. Note that not all of these

proteins and RNAs are drug targets since KEGG also has objectives other than supporting drug discovery (such as assisting studies in systems biology and functional genomics).

- ChEMBL [53] (<https://www.ebi.ac.uk/chembl/db/>)

ChEMBL is an open database that contains bioactivity information (i.e. binding, functional and ADMET information) for a large number of drug-like compounds. Unlike DrugBank which focuses only on drugs, ChEMBL stores bioactivity information on compounds in general. At present, ChEMBL stores extensive information on 1,735,442 compounds and 11,538 targets.

- STITCH [54] (<http://stitch.embl.de/>)

STITCH (short for Search Tool for Interacting Chemicals) is a collection of interactions between proteins and chemicals. These interactions are either obtained via computational prediction, knowledge transfer between organisms or aggregated from other sources (e.g. databases, literature, etc.). Furthermore, these interactions are either direct (physical) or indirect (functional) associations. Currently, STITCH has a total of 9,643,763 proteins from 2,031 organisms as well as over 430,000 compounds.

2.1.2 Drug and Target Data

Types of data that are available for drugs and may be used for training DTI classifiers include – but are not limited to – graphical representations of drugs' chemical structures [56], side effects [57], ATC (Anatomical Therapeutic Chemical) codes [58] and gene expression responses to drugs [59]. Other forms of data may further be extracted from the chemical structure graphs of drugs including substructure fingerprints as well as constitutional, topological and geometrical descriptors among other molecular properties (e.g. via the Rcp1 [60], PyDPI [61] or Open Babel [62] packages).

As for targets, available data that can be obtained include genomic sequences [63], Gene Ontology (GO) information [64], gene expression profiles [65], disease associations [66] and protein-protein interaction network information [67, 68] among others. Further information may also be extracted from protein sequences including amino acid composition, CTD (composition, transition and distribution) and autocorrelation descriptors (e.g. via the PROFEAT web server [69]).

Below, we describe some of the publicly available databases and packages that one may use to supplement their DTI prediction efforts with more drug and target data:

- SIDER [57] (<http://sideeffects.embl.de/>)

SIDER (short for Side Effect Resource) is an online resource that contains information on the adverse drug reactions (ADRs) of drugs that have already reached the market. Information available on ADRs include their frequency as well as their severity. Links to other information are also included (e.g. the drug-target interactions involved in the ADRs). SIDER currently stores information on 1430 drugs and 5868 ADRs, involving 139,756 drug-ADR pairs.

- Connectivity Map [59] (<https://clue.io/cmap>)

Connectivity Map (CMap) is a resource that houses over 1 million gene expression responses of cultured human cells to bioactive small molecules. The original motivation of the CMap project was that highly similar gene expressions may lead to discovering unexpected relationships such as between two small molecules that are structurally dissimilar but found to exhibit the same response. Even highly dissimilar responses may be connected as having related physiological effects on the cell. Furthermore, another goal of CMap was to use gene signatures of disease states in order to pair them with small molecules that may be able to suppress them. At the present time, CMap contains 1,319,138 gene expression profiles generated from 42,080 perturbagens (19,811 small molecule compounds, 18,493 shRNAs, 3,462 cDNAs, and 314 biologics) [70].

- Uniprot [63] (<http://www.uniprot.org/>)

Uniprot (short for Universal Protein Resource) is a comprehensive online resource for storing protein sequences and relevant annotations. For every protein record in Uniprot, examples of information that one may access include the protein's sequence, what protein family it belongs to, what domains it consists of, its gene ontology tags, a link to its structure (if available) and its functional description. At present, it contains 556,825 manually annotated and reviewed protein records, forming a set of records collectively known as Swiss-Prot. Uniprot also contains another set of ~109 million automatically annotated protein records (which are not manually reviewed) known as TrEMBL.

- Protein-protein interaction databases

There exist many databases that store protein-protein interactions (PPIs). Examples of these databases include DIP (Database of Interacting Proteins) [71], BioGrid (Biological General Repository for Interaction Datasets) [72], MINT (Molecular Interaction Database) [73] and IntAct [74]. While many such databases store PPIs for multiple different organisms, there are those that specifically focus on human PPIs such as HPRD (Human Protein Reference Database) [75]. For a recent survey on PPI databases, please refer to [76].

- Rcp1 [60], PROFEAT [69] and other packages

It has been mentioned earlier that drugs' chemical structures and proteins' sequences can be used as input to available software packages to generate extra features for them. For example, Rcp1 can be used to extract descriptors for a drug from its chemical structure. An example from the target side is PROFEAT which can be used to generate features from a protein's sequence. Table 2.2 shows a list of packages that have been previously used for extracting additional features for drugs and targets.

Package	For	Link
ChemCPP	Drug	http://chemcpp.sourceforge.net
RDKit	Drug	http://www.rdkit.org/
PyDPI [61]	Drug	https://sourceforge.net/projects/pydpicao/
Open Babel [62]	Drug	http://openbabel.org/
Rcpi [60]	Drug	http://bioconductor.org/packages/release/bioc/html/Rcpi.html
Rchemcpp [77]	Drug	http://shiny.bioinf.jku.at/Analoging/
KeBABS [78]	Target	http://www.bioinf.jku.at/software/kebabs/
PROFEAT [69]	Target	http://bidd2.nus.edu.sg/cgi-bin/profeat2016/main.cgi

Table 2.2: Software packages to compute features for drugs and targets

2.1.3 An Important Benchmark Dataset

A lot of the DTI prediction methods that have been developed over the years have been frequently using a certain benchmark dataset that was introduced in [45]. That said, the input data for the methods described in this chapter, unless otherwise stated, is from this dataset which consists of:

- An interaction matrix, $Y \in \mathbb{R}^{n \times m}$, showing which of the n drugs and m targets interact with each other where $Y_{ij} = 1$ if drug d_i and target t_j interact, and $Y_{ij} = 0$ otherwise. Rows and columns of Y represent the interaction profiles of the drugs and targets, respectively. Later in this text, the interaction profile for a drug d_i or target t_j is denoted by $Y(d_i)$ or $Y(t_j)$, respectively.
- A drug similarity matrix, $S_d \in \mathbb{R}^{n \times n}$, where drug pairwise chemical structure similarities are computed using SIMCOMP [79], a graph algorithm that computes a similarity score between two drugs based on the *maximal common subgraph* between their molecular graphs. Given two drugs d and d' , the chemical similarity between them can be roughly described as

$$S_d(d, d') = \frac{|d \cap d'|}{|d \cup d'|} = \frac{|d \cap d'|}{|d| + |d'| - |d \cap d'|} \quad (2.1)$$

where $|d|$ and $|d'|$ are the cardinalities of the molecular graphs of d and d' , respectively, and $|d \cap d'|$ is the cardinality of the maximal common subgraph determined between the graphs of d and d' .

- A target similarity matrix, $S_t \in \mathbb{R}^{m \times m}$, where target pairwise genomic sequence similarities are computed using a normalized version of the Smith-Waterman sequence alignment algorithm [80]. Given two proteins t and t' , the genomic similarity between them (i.e. the normalized Smith-Waterman score) is

$$S_t(t, t') = \frac{SW(t, t')}{\sqrt{SW(t, t)}\sqrt{SW(t', t')}} \quad (2.2)$$

where $SW(\cdot, \cdot)$ is the original Smith-Waterman score.

Most of the DTI prediction methods in the following sections use this dataset (or other similar datasets) as input, and they all have the same goal of obtaining a prediction scores matrix, $\hat{Y} \in \mathbb{R}^{n \times m}$, that contains the new predicted interactions.

2.2 Neighborhood Methods

Neighborhood methods are those that employ relatively simple similarity functions to perform predictions. More precisely, a *new* drug or target has its interaction profile predicted using its similarities to other drugs or targets, respectively; a new drug is one that has no known targets and, similarly, a new target is one that has no known interactions with any drugs.

2.2.1 Nearest Profile & Weighted Profile

Nearest profile and *weighted profile* are two methods that were introduced in [45]. *Nearest profile* infers the interaction profile of a new drug or target from its nearest

neighbor (i.e. the drug or target most similar to it). For example, the nearest profile for a new drug d_i is computed as

$$\hat{Y}(d_i) = S_d(d_i, d_{nearest}) \times Y(d_{nearest}) \quad (2.3)$$

where $d_{nearest}$ is the drug most similar to d_i , and $Y(d_i)$ is the interaction profile of drug d_i . On the other hand, *weighted profile* performs something like a weighted average using the similarities to all the other drugs or targets. The weighted profile for d_i is computed as

$$\hat{Y}(d_i) = \frac{\sum_{j=1}^n S_d(d_i, d_j) \times Y(d_j)}{\sum_{j=1}^n S_d(d_i, d_j)}. \quad (2.4)$$

In both of these methods, predictions from both the drug and target sides are averaged to obtain the final predictions. Of the two methods, *weighted profile* is the variant that is likely to produce more accurate prediction results since it predicts interactions for a drug or target by incorporating information from the interaction profiles of all the other drugs or targets, respectively, rather than only the nearest drug or target.

2.2.2 Similarity Rank-based Predictor (SRP)

SRP [81] computes two tendency indices for each drug-target pair; one for the likelihood that it would interact and one for the likelihood that it would not. Given a drug d_i and a target t_j , its “tendency-to-interact” index is computed as

$$TI^+(d_i, t_j) = \sum_{p \in P^+(t_j)} \frac{S_d(d_i, d_p)}{R_d(d_i, d_p)} \quad (2.5)$$

where $P^+(t_j) \subset \{d_1, d_2, \dots, d_n\}$ is the set of drugs that are known to interact with t_j , and $R_d(d_i, d_p)$ is the similarity *rank* of drug d_i to d_p among the n drugs. Specifically, the rank of the nearest drug is 1, that of the second nearest drug is 2 and so on. The ranks are used as a sort of enforced *decay* for the more dissimilar drugs to lessen their contribution to the value of $TI^+(d_i, t_j)$. Since the term, $TI^+(d_i, t_j)$, is a measure of how likely it is that

drug d_i would interact with target t_j , it is desired for the drugs most similar to d_i to contribute more to $TI^+(d_i, t_j)$ than those that are less similar to d_i . Another “tendency-to-not-interact” index, $TI^-(d_i, t_j)$, is similarly computed as

$$TI^-(d_i, t_j) = \sum_{q \in Q^-(t_j)} \frac{S_d(d_i, d_q)}{R_d(d_i, d_q)} \quad (2.6)$$

where $Q^-(t_j)$ is the set of drugs known to *not* interact with t_j . An interaction score is then computed for (d_i, t_j) as

$$C(d_i, t_j) = \frac{TI^+(d_i, t_j)}{TI^-(d_i, t_j)}. \quad (2.7)$$

In addition to the above score which was computed using S_d , a similar corresponding score is also obtained using S_t , and then the two scores are averaged to give the final prediction score.

SRP and *weighted profile* are both non-parametric methods that involve predicting interactions for a drug or target via collective inference from the interaction profiles of other drugs or targets, respectively, based on pairwise similarities. However, *SRP* is a step up from *weighted profile* due to the inclusion of the rank terms (e.g. $R_d(d_i, d_p)$) that lessen the contribution of the less similar drugs or targets that are being inferred from, thus improving prediction results.

2.3 Bipartite Local Models

Bipartite local models perform two sets of predictions, one from the drug side and one from the target side, and then aggregates these predictions to give the final prediction scores for the potential interaction candidates.

2.3.1 SVM-based Bipartite Local Models

This pioneering effort [49] introduced the concept of bipartite local models (BLM) where a local model is trained for each drug (or target) to predict which targets (or drugs) would interact with it. In the case of [49], the local models were *SVM* (Support Vector Machines) classifiers. Predictions from the drug and target sides are then averaged to get the final results.

Prior to this work, there were other efforts involving the use of *SVM* to predict drug-target interactions [42, 82]. However, these efforts considered drug-target pairs as objects and constructed a *global SVM* with these drug-target pairs as training examples. The number of training examples would then be $n \times m$ where n and m are the numbers of drugs and targets in the data, respectively. This means that the computational complexity of *SVM* would then be $O((nm)^2)$, which presents computational difficulties for large-scale problems.

On the other hand, in bipartite local models, the number of training examples in any of the trained models is either n or m examples. More precisely, there are n classifiers each trained with m training examples, and m classifiers each trained with n training examples. This collectively results in a computational complexity of $O(nm^2 + mn^2)$ for the *SVM*-based BLM, which is markedly more manageable than *global SVM*'s $O((nm)^2)$.

Specifically, assuming a bipartite DTI network, the algorithm tries to predict whether the edge e_{ij} exists between drug d_i and target t_j . The following steps are performed:

1. Ignoring t_j , a classifier is trained for d_i using the list of its known interactions with other targets (positive examples) as well as the list of targets not known to interact with d_i (negative examples). Interactions are labeled as +1, whereas non-interactions are labeled as -1. The trained classifier is used to predict e_{ij} .
2. Ignoring d_i , a classifier is trained for t_j using the list of its known interactions with other drugs as well as the list of drugs not known to interact with t_j .

Interactions are labeled as +1, whereas non-interactions are labeled as -1. The trained classifier then predicts e_{ij} .

3. Predictions from both the drug and target sides (i.e. from both classifiers) are aggregated using the $\max(\cdot, \cdot)$ function.

2.3.2 Laplacian Regularized Least Squares (LapRLS)

LapRLS [83] is another algorithm that is based on the BLM concept. The local models in *LapRLS* use regularized least squares to minimize an objective function that includes an error term as well as a graph regularization term. From the drug side, the objective function to be minimized is

$$\min_{\alpha_d} \|Y - S_d \alpha_d\|_F^2 + \beta_d \text{Tr}(\alpha_d^T S_d L_d S_d \alpha_d) \quad (2.8)$$

where $\|\cdot\|_F$ is the Frobenius norm, L_d is the normalized Laplacian obtained using S_d , and β_d is a parameter. Note that the trace of a given matrix A is $\text{Tr}(A) = \sum_i A_{ii}$, and the expression $\text{Tr}(\alpha_d^T S_d L_d S_d \alpha_d)$ is a graph regularization term which helps model the *manifold* – a low-dimensional surface embedded in a high-dimensional space – that is assumed to underlie the data. The manifold assumption (i.e. that data points lie on a low-dimensional non-linear manifold) is one that is found to be usually true [84-86] and, therefore, modeling the manifold would be beneficial to the prediction performance. After obtaining α_d^* that minimizes the above function, the prediction matrix from the drug side is obtained as $\hat{Y}_d = S_d \alpha_d^*$. A similar objective function is minimized from the target side to obtain $\hat{Y}_t$, and then the final prediction matrix is obtained as

$$\hat{Y} = \frac{\hat{Y}_d + \hat{Y}_t}{2}. \quad (2.9)$$

It is important to emphasize that the non-interactions in the data – i.e. the drug-target pairs represented as 0's in Y – are not necessarily true, confident non-interactions. The interactions in the data (i.e. the 1's in Y) are those that have been reported *so far*, which means that the 0's may contain undiscovered interactions among them. With that in

mind, note that the *SVM*-based BLM by Bleakley et al. [49] used the 0's in the data as negative examples. Although there are likely more than a few undiscovered interactions in the negative data, the prediction performance was reasonable. On the other hand, *LapRLS* additionally models the manifold assumption into the trained prediction model, thus mitigating the impact of utilizing the 0's as negatives and improving predictions.

2.3.3 Regularized Least Squares (RLS-avg & RLS-kron)

RLS-avg [87] uses *kernel ridge regression* to perform prediction. Furthermore, *Gaussian interaction profile* (GIP) kernels are used to compute network similarity matrices for drugs and targets from the interaction matrix Y ; network similarity between two drugs d_i and d_j is computed as $GIP_d(d_i, d_j) = \exp(-\gamma \|Y(d_i) - Y(d_j)\|^2)$ where γ is a parameter, and $Y(d_i)$ and $Y(d_j)$ are the interaction profiles of d_i and d_j , respectively. These network similarity matrices are then merged with S_d and S_t as

$$K_d = \alpha S_d + (1 - \alpha) GIP_d \quad (2.10)$$

$$K_t = \alpha S_t + (1 - \alpha) GIP_t \quad (2.11)$$

where α is a parameter such that $0 \leq \alpha \leq 1$. The authors of this work hypothesized that incorporating network information (i.e. interaction information from the DTI network) into the prediction process as indicated above would lead to better prediction performance. Next, the prediction scores matrix is obtained as

$$\hat{Y} = \frac{1}{2} (K_d (K_d + \sigma I)^{-1} Y) + \frac{1}{2} (K_t (K_t + \sigma I)^{-1} Y^T)^T \quad (2.12)$$

where σ is a regularization parameter. Note that the above equation shows predictions from both the drug and target sides being averaged to give the final scores.

RLS-avg is similar to *LapRLS* in terms of the objective function. The only difference is that *RLS-avg* uses a Tikhonov regularization term, while *LapRLS* uses a manifold regularization term. Besides the objective function, another difference is that *RLS-avg*

additionally generates the network similarity matrices, GIP_d and GIP_t , and merges them with the similarity matrices, S_d and S_t , via linear combination.

Moreover, another algorithm was also introduced in [87], named *RLS-kron*, where the drug and target sides of the prediction have been merged into one by using the Kronecker product. Given that $K = K_d \otimes K_t$ is a kernel over drug-target pairs, the prediction scores matrix is obtained as

$$vec(\hat{Y}^T) = K(K + \sigma I)^{-1}vec(Y^T) \quad (2.13)$$

where $vec(Y^T)$ is a column vector created by stacking the columns of Y^T . Since the matrix K would require too much memory and the computation of its inverse would be computationally intensive, the authors use a more efficient implementation [88] that is based on eigen decompositions.

RLS-kron is similar to the *global SVM* mentioned in section 2.3.1 in that the training examples are the drug-target pairs; note that K from Equation (2.13) is a kernel over drug-target pairs. According to [89], least squares has the same classification accuracy as *SVM* when a good kernel is supplied. This means that *RLS-kron* is at least as good as *global SVM* when it comes to prediction performance. At the same time, *RLS-kron* is not as computationally intensive as *global SVM* thanks to the use of the matrix tricks based on eigen decompositions, thus circumventing the computational complexity problem of *global SVM*.

2.3.4 Bipartite Local Models with Neighbor-based Interaction Profile Inferring

BLM algorithms that exploit local models achieved decent performance for DTI prediction. However, they had an outstanding issue that they are not able to train local models for drugs or targets that do not have any known interactions (i.e. novel drugs or targets). To address this issue, *BLM-NII* [90], which is based on *RLS-avg*, introduces a

preprocessing method denoted as *NII* (Neighbor-based Interaction profile Inferring) to infer temporary interaction profiles for those novel drugs or targets.

Specifically, a local model is trained for each drug d_i . However, if drug d_i happens to have an empty interaction profile, a temporary interaction profile is inferred for it prior to training as

$$Y(d_i) = \frac{\sum_{j=1}^n S_d(d_i, d_j) \times Y(d_j)}{\sum_{j=1}^n S_d(d_i, d_j)} \quad (2.14)$$

after which it is normalized via min-max normalization to obtain

$$\tilde{Y}(d_i) = \frac{Y(d_i) - \min(Y(d_i))}{\max(Y(d_i)) - \min(Y(d_i))}. \quad (2.15)$$

Now that drug d_i does not have an empty profile, classifier training and prediction may proceed as per normal. The *NII* procedure is similarly applied to the target side wherever applicable, and predictions are obtained from the target side as well. Predictions from the drug and target sides are then aggregated as is typical in algorithms from the category of bipartite local models. The *NII* preprocessing procedure was found to improve prediction performance.

2.3.5 Regularized Least Squares with Weighted Nearest Neighbors (RLS-WNN)

Another method based on *RLS-kron* [87] was introduced in [91] where *RLS-kron* was augmented with a preprocessing method, *WNN* (Weighted Nearest Neighbor), that has the same purpose as *NII*. For every *new* drug d_i , *WNN* is used to infer an interaction profile for it as

$$Y(d_i) = \sum_{j=1}^n w_j Y(d_j) \quad (2.16)$$

where d_1 to d_n are drugs sorted in descending order based on their similarity to d_i , and $w_j = \eta^{j-1}$ where η is a decay term with $\eta \leq 1$. The same is done from the target side, and then *RLS-kron* proceeds as per normal. As with *NII*, applying *WNN* has also resulted in improvements in the prediction performance, which confirms that such preprocessing methods are indeed successful.

2.4 Network Diffusion

The *network diffusion* category of methods includes those that investigate graph-based techniques to predict new interactions; the network diffusion technique is predominant in this category, which is why it is named as such.

2.4.1 Network-based Inference (NBI)

To perform prediction, *NBI* [92] applies network diffusion on the DTI bipartite network corresponding to the interaction matrix Y . Network diffusion is performed according to

$$\hat{Y} = WY \quad (2.17)$$

where $W \in \mathbb{R}^{n \times n}$ is the weight matrix defined as

$$W_{ij} = \frac{1}{\Gamma_{(i,j)}} \sum_{l=1}^m \frac{Y_{il}Y_{jl}}{k(t_l)} \quad (2.18)$$

where Γ is the diffusion rule, and $k(x)$ is the degree of node x in the DTI bipartite network. In case of *NBI*, the Γ rule is given by $\Gamma = k(d_j)$.

2.4.2 Heterogeneous Graph-based Inference (HGBI)

HGBI is another method presented in [93] that extends *NBI*. In place of the basic bipartite network, network diffusion is performed on a heterogeneous network (as illustrated in Figure 2.2). The heterogeneous network augments the basic bipartite one by adding, between all pairs of drugs or targets, edges whose weights correspond to the

pairwise similarities as indicated in S_d or S_t , respectively. Network diffusion in this method is done using the equation,

$$Y^{(i+1)} = \alpha S_d Y^{(i)} S_t + (1 - \alpha) Y^{(0)}, \quad (2.19)$$

where $Y^{(i)}$ is the prediction scores matrix at time step i , $Y^{(0)} = Y$, and α is an adjustable parameter. To ensure that the above formula would converge, S_d and S_t are normalized beforehand as

$$\tilde{S}_d(d_i, d_j) = \frac{S_d(d_i, d_j)}{\sqrt{\sum_{k=1}^n S_d(d_i, d_k) \sum_{k=1}^n S_d(d_k, d_j)}}, \quad (2.20)$$

$$\tilde{S}_t(t_i, t_j) = \frac{S_t(t_i, t_j)}{\sqrt{\sum_{k=1}^m S_t(t_i, t_k) \sum_{k=1}^m S_t(t_k, t_j)}}. \quad (2.21)$$

According to [94], if Equation (2.19) can be rewritten as $Y^{(i+1)} = \alpha C Y^{(i)} + (1 - \alpha) Y^{(0)}$, then normalizing C as $\tilde{C} = D^{-1/2} C D^{-1/2}$ can guarantee convergence towards a solution (where D is a diagonal matrix such that D_{ii} is the sum of the i -th row of C). Based on this information, the authors have deduced the above equations that guarantee convergence. For more details, please refer to the appendix in [93].

Figure 2.2: (a) Bipartite drug-target interaction network, (b) Heterogeneous network that additionally includes drug and target pairwise similarities (the dashed lines).

2.4.3 Network Random Walk with Restart on Heterogeneous Network (NRWRH)

NRWRH [95] uses a heterogeneous network as well, and it predicts interactions via random walk on it. To perform random walk, *NRWRH* uses the transition matrix,

$$M = \begin{bmatrix} M_{DD} & M_{DT} \\ M_{TD} & M_{TT} \end{bmatrix}, \quad (2.22)$$

where M_{DD} and M_{TT} are transition matrices between drugs themselves and targets themselves, respectively, and M_{TD} and M_{DT} are transition matrices from targets to drugs and from drugs to targets, respectively. Note that $M_{DD} = S_d$, $M_{TT} = S_t$, $M_{DT} = Y$ and $M_{TD} = Y^T$. The predicted matrix at step $i + 1$ is modeled as

$$A^{(i+1)} = (1 - r)M^T A^{(i)} + rA^{(0)} \quad (2.23)$$

where $A^{(0)} \in \mathbb{R}^{(n+m) \times (n+m)}$ is the adjacency matrix between the nodes (including the n drugs and m targets in both the rows and the columns), and r is the restart probability. By running the above equation multiple times until convergence, the matrix A would contain the final prediction scores, and the prediction scores matrix $\hat{Y} \in \mathbb{R}^{n \times m}$ is then extracted from A .

NRWRH does not treat unlabeled drug-target pairs as negative examples while training the prediction model. *NRWRH* is run on the heterogeneous network and, using each drug and target as a starting point for a random walk, probabilities on where these random walks would end are computed.

Finally, *NRWRH* is better than *NBI* in predicting interactions for *orphan* drugs or targets (i.e. those that have no known interactions). The reason for this is that *NRWRH* uses a heterogeneous network (figure 2.2b), whereas *NBI* uses the basic bipartite DTI network (figure 2.2a). *HGBI* (which extends *NBI*) does not have this issue as it uses the heterogeneous network like *NRWRH*.

2.4.4 Probabilistic Soft Logic (PSL)

PSL [96] also utilizes a heterogeneous network similar to the one shown in Figure 2.2. As its name suggests, *PSL* uses *probabilistic soft logic* to perform prediction; i.e. it involves applying rules that use logical connectives such as $\wedge$ (*and*), $\vee$ (*or*), $\neg$ (*not*), etc. Specifically, to determine if a drug and target interact, *triad* and *tetrad* relations (i.e. paths of length 3 and 4, respectively) involving the drug and target are searched for in the heterogeneous network and used for prediction of the potential interaction. Triad rules take the form of

$$\text{SimilarTarget}(t_1, t_2) \wedge \text{Interacts}(d, t_1) \rightarrow \text{Interacts}(d, t_2) \quad (2.24)$$

$$\text{SimilarDrug}(d_1, d_2) \wedge \text{Interacts}(d_1, t) \rightarrow \text{Interacts}(d_2, t) \quad (2.25)$$

while tetrad rules take the form of

$$\text{SimilarDrug}(d_1, d_2) \wedge \text{SimilarTarget}(t_1, t_2) \wedge \text{Interacts}(d_1, t_1) \rightarrow \text{Interacts}(d_2, t_2). \quad (2.26)$$

To predict new interactions, the above rules are applied wherever applicable on the DTI network (i.e. each of the rules is applied to its corresponding relations that exist in the network).

Furthermore, to avoid investigating the large number of all possible such relations, a technique called *blocking* is used beforehand where edges between pairs of drugs or targets are removed from the network if their weights are below some predefined threshold.

2.4.5 Determine All Simple Paths, Find Interactions (DASPfind)

DASPfind [97] predicts an interaction between a drug d and a target t by finding all simple paths (i.e. that have no cycles) connecting them on the heterogeneous network. For the found simple paths, each path p has its score s_p computed by multiplying the

weights on its constituting edges. Finally, the scores are summed up to give the final prediction score for (d, t) as per the equation,

$$score(d, t) = \sum_{p=1}^z \left(\prod s_p \right)^{\alpha \times len(p)} \quad (2.27)$$

where z is the number of simple paths between drug d and target t , α is an adjustable decay parameter, and $len(p)$ is the length of path p (i.e. longer paths will have less of a contribution to the prediction score). Note that $len(p) \leq 3$. Similar to *PSL*, the blocking procedure (i.e. eliminating edges with weights below a certain threshold) is also used here prior to prediction.

When *DASPfind* traverses each path using depth-first search, the visited nodes are marked to prevent the traversal from cycling back to it. Once the path is traversed, the marks are removed before traversing the next path and so on. This ensures that the determined paths are simple paths that do not have cycles in them (i.e. paths with no repeated nodes). In contrast, *NRWRH* and *HGBI* do not avoid cycles in their paths since they use transition matrices for traversal.

DASPfind managed to produce prediction results that are comparable to those of other methods of the *network diffusion* category. Moreover, its ability to predict new interactions was tested by applying *DASPfind* on the dataset from [45]. The newly obtained interactions were then investigated and checked against online repositories of drug-target interactions, and indeed many top-ranked ones were confirmed to be true.

2.5 Matrix Factorization

Matrix Factorization takes an input matrix and tries to find two other matrices that, when multiplied together, approximate the input matrix. In the case of DTI prediction, the interaction matrix $Y \in \mathbb{R}^{n \times m}$ is factorized into two matrices $A \in \mathbb{R}^{n \times k}$ and $B \in \mathbb{R}^{m \times k}$ such that $AB^T \approx Y$. k is an adjustable parameter corresponding to the number of latent features in A and B , and $k \ll n, m$.

Matrix factorization identifies latent features of drugs and targets in an unsupervised fashion, which is useful for *collaborative filtering*. For example, if the latent vectors of two drugs turn out to be quite similar, then it is likely that these drugs share many of the same interactions, thus allowing the transfer of interactions between them.

Since we are looking for missing interactions in the matrix Y , matrix factorization can be used as a matrix completion technique (i.e. the above-mentioned transfer of interactions between drugs themselves and targets themselves), which makes it a good fit for the DTI prediction problem. An illustration of matrix factorization is given in Figure 2.3.

Figure 2.3: Illustration of matrix factorization. The goal is to find two low-rank latent feature matrices, A and B , that reconstruct the interaction matrix Y when multiplied together.

2.5.1 Kernelized Bayesian Matrix Factorization with Twin Kernels (KBFM2K)

KBMF2K [98] is, to our knowledge, the first in a number of methods that used matrix factorization for predicting drug-target interactions. It uses a Bayesian probabilistic formulation along with the concept of matrix factorization to perform prediction. Worded differently, it uses variational approximation to perform non-linear dimensionality reduction, thus improving efficiency in terms of computation time.

Since there are too many algorithmic details to mention, we only provide a minimal overview of the algorithm here. Please observe the figure below which is **inspired from Figure 1 of [98]**.

Figure 2.4: Minimal representation of the KBMF2K algorithm which predicts drug-target interactions from drug and target kernels, S_d and S_t .

Assuming R is the chosen subspace dimensionality, $P_d \in \mathbb{R}^{n \times R}$ contains projection parameters and Λ_d contains the corresponding priors. With the projection matrix P_d , the drug kernel matrix S_d is used to project the interactions (more precisely, the drug-target pairs) to a low-dimensional space (called the *pharmacological space*). This results in G_d which consists of the low-dimensional representations of drugs in this space. The same is done to obtain G_t (using a projection matrix $P_t \in \mathbb{R}^{m \times R}$) which consists of lower-dimensional representations of targets in that same space. Having obtained lower-dimensional representations of both drugs and targets in the same unified space, a prediction scores matrix F is obtained and presented as the interaction matrix $\hat{Y}$.

2.5.2 Probabilistic Matrix Factorization (PMF)

PMF [99] is another matrix factorization method that uses probabilistic formulations as well. Specifically, it models interactions via “probabilistic linear models with Gaussian noise”. Unlike *KBMF2K*, however, it does not depend on or utilize similarity matrices between drugs or targets while performing prediction, and thus it achieves relatively lower performance than other matrix factorization techniques introduced here.

To explain the general idea behind *PMF*, suppose we have two matrices A and B containing latent feature vectors for the drugs and targets, respectively, that construct the interaction matrix Y as $AB^T = Y$. The conditional probability over observed interactions in Y is given by

$$p(Y|A, B, \sigma^2) = \prod_{i=1}^n \prod_{j=1}^m [f(Y_{ij}|a_i b_j^T, \sigma^2)]^{I_{ij}} \quad (2.28)$$

where n and m are the numbers of drugs and targets, respectively, $f(x|\mu, \sigma^2)$ is the Gaussianly distributed probability density function for x with mean μ and variance σ^2 , and I_{ij} is an indicator function that is equal to 1 if Y_{ij} is known and 0 otherwise. Assuming zero-mean, spherical Gaussian priors on the latent vectors of A and B , the formula of the log-likelihood of A and B is derived using Bayes' Rule as

$$\ln(p(A, B|Y, \sigma^2, \sigma_A^2, \sigma_B^2)) = -\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^m I_{ij} (Y_{ij} - a_i b_j^T)^2 - \frac{1}{2\sigma_A^2} \sum_{i=1}^n a_i a_i^T - \frac{1}{2\sigma_B^2} \sum_{j=1}^m b_j b_j^T. \quad (2.29)$$

The first term on the right-hand side of the above equation is the squared-error function to be minimized, while the last two terms are extra Tikhonov regularization terms that are added to help avoid overfitting by preventing the latent features of A and B from assuming large values. The goal here is to find the two latent matrices A and B that maximize the log-likelihood presented above. Finally, the prediction scores matrix is obtained as $\hat{Y} = AB^T$.

2.5.3 Collaborative Matrix Factorization (CMF)

CMF [100] uses collaborative filtering for prediction. In addition to the standard goal of matrix factorization to find two matrices A and B where $AB^T \approx Y$, *CMF* proposes regularization terms to ensure that $AA^T \approx S_d$ and $BB^T \approx S_t$. *CMF*'s objective function is given as

$$\begin{aligned} \min_{A, B} & \|W \odot (Y - AB^T)\|_F^2 \\ & + \lambda_l (\|A\|_F^2 + \|B\|_F^2) \\ & + \lambda_d \|S_d - AA^T\|_F^2 \\ & + \lambda_t \|S_t - BB^T\|_F^2 \end{aligned} \quad (2.30)$$

where $\|\cdot\|_F$ is the Frobenius norm, $\odot$ is the element-wise product, λ_l , λ_d and λ_t are parameters, and $W \in \mathbb{R}^{n \times m}$ is a weight matrix where $W_{ij} = 0$ for unknown drug-target pairs (i.e. the test set instances) so that they would not contribute to the estimation of A and B . The first line is the weighted low-rank approximation term that tries to find the latent feature matrices A and B that reconstruct Y . The second line is the Tikhonov regularization term that prevents large values to be assumed by A and B , which thus promotes simpler solutions and helps avoid overfitting. The third and fourth lines are regularization terms that require latent feature vectors of similar drugs/targets to be similar *and* latent feature vectors of dissimilar drugs/targets to be dissimilar, respectively.

Furthermore, denoting the objective function in the above equation as $\mathbf{L}$ and letting a_i and b_j be the i^{th} and j^{th} row vectors of A and B , respectively, two alternative update rules (one for updating a_i and one for updating b_j) were derived by setting $\frac{\partial \mathbf{L}}{\partial a_i} = 0$ and $\frac{\partial \mathbf{L}}{\partial b_j} = 0$. Alternating least squares is then used to run the update rules alternately until convergence. Finally, the predicted matrix for drug-target interactions is then obtained as $\hat{Y} = AB^T$.

Unlike *KBMF2K* and *PMF*, *CMF* does not make statistical assumptions about the variables (such as when *PMF* assumes zero-mean, spherical Gaussian priors on the latent vectors of A and B). Furthermore, *CMF* makes use of *weighted* low-rank approximation; i.e. the W in Equation (2.30). The advantage of W is that it allows the exclusion of selected instances (drug-target pairs) from contributing towards the solution of the objective function in Equation (2.30). That is, given some drug-target pairs for which we wish to determine whether they interact or not, they may be excluded from the training of the *CMF* model by setting their corresponding values in W as $W_{ij} = 0$. Otherwise, not excluding such drug-target pairs would lead to them contributing to the training process as non-interactions (i.e. negative instances), which may not be correct.

Another variant for *CMF*, named *MSCMF*, involves utilizing multiple similarities for both the drugs and targets. Besides the chemical structure similarity and genomic sequence similarity that are typically used for the drugs and targets, respectively, ATC (Anatomical Therapeutic Chemical) similarity is also used for drugs, and GO (Gene Ontology) and PPI (protein-protein interaction network) similarities are used for targets. The objective function for *MSCMF* is given as

$$\begin{aligned}
& \min_{A, B, \omega_d, \omega_t} \|W \odot (Y - AB^T)\|_F^2 \\
& + \lambda_l (\|A\|_F^2 + \|B\|_F^2) \\
& + \lambda_d \left\| \sum_{k=1}^{M_d} \omega_d^k S_d^k - AA^T \right\|_F^2 \\
& + \lambda_t \left\| \sum_{k=1}^{M_t} \omega_t^k S_t^k - BB^T \right\|_F^2 \\
& + \lambda_\omega (\|\omega_d\|_F^2 + \|\omega_t\|_F^2) \\
& s. t. \quad |\omega_d| = |\omega_t| = 1
\end{aligned} \tag{2.31}$$

where M_d and M_t are the numbers of drug and target similarity matrices, respectively, and λ_ω is a parameter. ω_d and ω_t are weight vectors for linearly combining the drug and target similarity matrices, respectively. The fifth line in the above equation includes Tikhonov regularization terms for ω_d and ω_t , while the sixth line is a constraint that ensures the weights in each of ω_d and ω_t sum up to 1. In addition to the update rules obtained for updating A and B , additional update rules are obtained for updating ω_d and ω_t . Alternating least squares is then applied to run the update rules until convergence.

2.5.4 Neighborhood Regularized Logistic Matrix Factorization (NRLMF)

NRLMF [101] performs prediction via utilizing the idea of logistic matrix factorization (LMF) [102]; i.e. it models the probability of an interaction between drug d_i and target t_j as the logistic function,

$$p_{ij} = \frac{\exp(a_i b_j^T)}{1 + \exp(a_i b_j^T)}, \quad (2.32)$$

where a_i and b_j are the latent feature vectors of d_i and t_j , respectively. Drug-target pairs are more likely to interact (i.e. p_{ij} tends to 1) on higher values of $a_i b_j^T$. Moreover, to prevent overfitting the training data, the model being trained is regularized by placing spherical zero-mean Gaussian priors on the latent vectors of all drugs and targets. Finally, the model is further regularized using the local neighborhoods of the drugs and targets via graph regularization. The objective function to be minimized is given as

$$\begin{aligned} \min_{A,B} \sum_{i=1}^n \sum_{j=1}^m (1 + cY_{ij} - Y_{ij}) \ln[1 + \exp(a_i b_j^T)] - cY_{ij} a_i b_j^T \\ + \lambda_d \|A\|_F^2 + \lambda_t \|B\|_F^2 \\ + \alpha \text{Tr}(A^T L_d A) + \beta \text{Tr}(B^T L_t B) \end{aligned} \quad (2.33)$$

where c , λ_d , λ_t , α and β are parameters. The first line of the above equation is the LMF expression, which is augmented by Tikhonov and graph regularization terms in the second and third lines, respectively. Tikhonov regularization terms prevent overfitting by favoring simpler solutions with smaller values, while graph regularization terms implicitly learn the underlying manifold in the data to encourage more accurate label propagation within the interaction matrix Y .

Graph regularization encourages latent vectors of similar drugs or targets (according to S_d or S_t , respectively) to be similar. When the latent vectors of two objects (either two drugs or two targets) are similar, then they will also have similar interaction profiles in the predicted matrix $\hat{Y} = AB^T$. This makes sense based on the widely-accepted similarity principle which states that similar drugs bind the same targets and vice versa [103].

2.5.5 Dual-Network Integrated Logistic Matrix Factorization (DNILMF)

DNILMF [104] can be considered an extension of *NRLMF*. *DNILMF* additionally incorporates network-based similarity in a way that is similar to how it is done in *RLS-avg* and *RLS-kron*. Unlike *RLS-avg* and *RLS-kron*, however, all kernels (i.e. similarity matrices) undergo a kernel diffusion step beforehand. For a drug (or target) kernel, a *local* similarity matrix is generated by keeping the similarities to the nearest k neighbors for each drug (or target) while discarding the rest, then the local similarity matrix is diffused with the *global* similarity matrix over a number of iterations.

Suppose that we are given the drug chemical similarity matrix S_d and the target sequence similarity matrix S_t , and that the drug and target network similarity matrices GIP_d and GIP_t have been computed from the interaction matrix Y as explained in Section 2.3.3. Each of these four matrices is then normalized (by dividing values of each row by the row's sum) and symmetrized. Taking the target matrices S_t and GIP_t as an example (with the drug matrices following the same process), *local* matrices L_t and $L_{t,GIP}$ are generated as

$$L_t(i, j) = \begin{cases} \frac{S_t(i, j)}{\sum_{r \in N_i} S_t(i, r)} & , \quad j \in N_i \\ 0 & , \quad otherwise \end{cases} \quad (2.34)$$

$$L_{t,GIP}(i, j) = \begin{cases} \frac{GIP_t(i, j)}{\sum_{r \in N_i} GIP_t(i, r)} & , \quad j \in N_i \\ 0 & , \quad otherwise \end{cases} \quad (2.35)$$

where N_i denotes the indices of the nearest neighbors of target t_i , and $|N_i| = k$ where k is a parameter that specifies the number of nearest neighbors to be considered. According to the equations above, similarities to targets outside the list of nearest neighbors are set to 0. The local matrices, L_t and $L_{t,GIP}$, are then used to update the *global* matrices, S_t and GIP_t , as

$$S_t^{(h+1)} = (L_t)GIP_t^{(h)}(L_t)^T \quad (2.36)$$

$$GIP_t^{(h+1)} = (L_{t,GIP})S_t^{(h)}(L_{t,GIP})^T \quad (2.37)$$

where $S_t^{(h+1)}$ and $GIP_t^{(h+1)}$ are the current matrices after h iterations. In the above equations, two interchanging diffusion operations are occurring in parallel. After a sufficient number of iterations, the final target similarity matrix, K_t , is obtained by averaging $S_t^{(h+1)}$ and $GIP_t^{(h+1)}$. The final drug similarity, K_d , is obtained using the same procedure.

Predictions are obtained using the objective function:

$$\begin{aligned} \min_{A,B} \sum_{i=1}^n \sum_{j=1}^m & (1 + cY_{ij} - Y_{ij}) \ln[1 + \exp(\alpha AB^T + \beta K_d AB^T + \gamma AB^T K_t)] \\ & - cY_{ij}(\alpha AB^T + \beta K_d AB^T + \gamma AB^T K_t) \\ & + \frac{\lambda_d}{2} \|A\|_F^2 + \frac{\lambda_t}{2} \|B\|_F^2 \end{aligned} \quad (2.38)$$

which is based on the modified logistic function:

$$p_{ij} = \frac{\exp(\alpha AB^T + \beta K_d AB^T + \gamma AB^T K_t)}{1 + \exp(\alpha AB^T + \beta K_d AB^T + \gamma AB^T K_t)}, \quad (2.39)$$

where α , β , γ , λ_d and λ_t are parameters. Note that, in contrast to *NRLMF*'s logistic function from Equation (2.32), the above logistic function incorporates information from the similarity matrices K_d and K_t (which were obtained via kernel diffusion). The matrices, A and B , that minimize the objective function in Equation (2.38) are used to obtain the final predictions matrix as $\hat{Y} = AB^T$.

As both *NRLMF* and *DNILMF* are based on logistic matrix factorization, their objective functions [from Equations (2.33) and (2.38)] resemble one another. However, while *NRLMF* uses graph regularization to make use of the similarity matrices S_d and S_t in prediction, *DNILMF* instead obtains the diffused kernels K_d and K_t , incorporates them into the logistic function and uses them in the objective function from Equation (2.38).

2.6 Feature-based Classification

Feature-based classification methods are those that need drug-target pairs to be explicitly represented as fixed-length feature vectors. Given a drug feature vector $d = [d_1, d_2, \dots, d_p]$ and a target feature vector $t = [t_1, t_2, \dots, t_q]$, the drug-target pair would typically be represented by the concatenated feature vector $d \oplus t = [d_1, d_2, \dots, d_p, t_1, t_2, \dots, t_q]$. In addition to the feature vector, each drug-target pair has a label to show whether it is a known interaction (i.e. positive class) or a non-interaction (i.e. negative class). With the feature vectors and labels, various supervised machine learning methods can thus be developed for predicting DTI interactions as illustrated in Figure 2.5.

Figure 2.5: Illustration of how feature-based prediction models are created. In the training phase, feature vectors for the training instances (i.e. the drug-target pairs) are generated by concatenating the feature vectors of the involved drugs and targets. Along with their labels, the training instances are used to train the prediction model. In the testing phase, feature vectors are generated for the testing instances, and the prediction model (from the training phase) is used to predict for the testing instances.

Note that it is more accurate to call non-interactions as *unlabeled* pairs as we do not know for sure whether these pairs are true non-interactions or not. Despite this detail, however, methods of this category commonly treat unlabeled pairs as if they are, in fact, true non-interactions.

2.6.1 Incremental and Forward Feature Selection

In [105], drugs were represented by a number of common functional groups that are found in drugs’ chemical structures, while targets were represented by pseudo amino acid composition. An innovative feature selection procedure was additionally introduced in this work for the sake of improving the prediction performance by using a better feature set.

The feature selection procedure starts by ranking features using the *mRMR* (minimum Redundancy Maximum Relevance) algorithm [106]. *Incremental feature selection* is then applied on the ranked features; i.e. the ranked features are added to the selected feature set in order, one by one, until the prediction performance on a temporary validation set stops improving. Finally, the set of selected features is further filtered by applying *forward feature selection* to it. After the feature selection phase is complete, a nearest neighbor algorithm is then applied to obtain final predictions.

2.6.2 Random Forest and Support Vector Machines

Random Forest and *SVM* models were proposed for predicting interactions in [107]. Assuming that the data consists of n_d drugs and n_t targets, this means that there are $n_d \times n_t$ drug-target pairs in total. When the dimensionality of the data is high (i.e. drug-target pairs are represented by many features), it becomes very challenging, if not infeasible, to use the entire dataset of all drug-target pairs as training data to train a classification model. Therefore, the set of non-interactions, which is far bigger than the set of known interactions, is undersampled until its size is equal to that of the set of interactions.

Compared with the method in the previous section (2.6.1), this work uses a more sophisticated prediction model based on *Random Forest* or *SVM* rather than *Nearest Neighbor*. In addition, this work uses undersampling to reduce the computational complexity of using the entire dataset.

Drug and target features were generated using the DRAGON (<http://www.taletе.mi.it/>) and PROFEAT [69] packages, respectively. Drug features generated by DRAGON include constitutional and topological descriptors, eigenvalue-based indices and 2D autocorrelations among others. On the other hand, target features generated by PROFEAT include CTD (Composition, Transition, Distribution) and autocorrelation descriptors, amino acid composition and so on.

2.6.3 Fuzzy K Nearest Neighbors (Fuzzy KNN)

Fuzzy KNN [108] models each training instance as belonging to two classes (i.e. positive and negative classes) with different membership values. For each test instance, its membership value to each of the two classes is computed by a kind of weighted average of its similarities to its nearest K neighbors, and the higher of the two values decides which class it belongs to. The drugs were represented by FP2 fingerprints that were generated using the Open Babel [62] package, while the targets were represented using pseudo amino acid composition.

2.6.4 Similarity-based Inference of drug-TARgets (SITAR)

Unlike typical feature-based classification methods which concatenate both drug and target feature vectors to represent drug-target pairs, *SITAR* [109] represents each instance (drug-target pair) as a vector of its similarities to the positives in the data. The similarity between two drug-target pairs (d, t) and (d', t') is computed using the geometric mean as

$$S((d, t), (d', t')) = S_d(d, d')^r \cdot S_t(t, t')^{(1-r)} \quad (2.40)$$

where r is an adjustable parameter. This results in feature vectors whose length is equal to the number of known interactions. After the feature vectors are generated, logistic regression is then used to perform prediction.

2.6.5 Chemical substructures-protein domains correlation model

In [110], drugs are represented as PubChem fingerprints (binary vectors indicating the absence/presence of 881 common chemical substructures), while targets are represented as domain fingerprints (binary vectors indicating the absence/presence of 876 protein domains obtained from the Pfam database [111]).

SCCA (Sparse canonical correspondence analysis) is then applied for the extraction of drug and target features that, when occurring together, would indicate the existence of an interaction between the drug and target involved. *SCCA* extends ordinary *CCA* by adding L_1 norm regularization terms to ensure that the learned weight vectors are sparse. The objective function that *SCCA* attempts to minimize is given as

$$\max_{\alpha, \beta} \alpha^T D^T Y T \beta \quad \text{subject to} \quad (2.41)$$

$$\|\alpha\|_2^2 \leq 1, \quad \|\beta\|_2^2 \leq 1, \quad \|\alpha\|_1 \leq c_1 \sqrt{u}, \quad \|\beta\|_1 \leq c_2 \sqrt{v}$$

where $D \in \mathbb{R}^{n \times u}$ and $T \in \mathbb{R}^{m \times v}$ are the drug and target feature matrices, respectively, c_1 and c_2 are parameters that are used to control the sparsity level where $0 < c_1 < 1$ and $0 < c_2 < 1$.

When used to predict drugs-target interactions, *SCCA* produced results that are comparable to those of *SVM*. However, unlike *SVM* which is focused only on prediction, *SCCA* is an *interpretable* classifier that, having been trained, can be inspected for learned rules that may contain useful insights. As stated above, *SCCA* emphasizes learning sparse weight vectors, which makes it possible to inspect these weight vectors for biological

insights; the non-zero elements in the learned weight vector would correspond to the most significant chemical structures and protein domains that govern drug-target interactions.

This work is similar to the work by He et al. presented in section 2.6.1 in that they both had a goal of coming up with biological insights. While this work focused on learning sparse weight vectors that can be later inspected, the work in section 2.6.1 inspected the list of features that remained after a feature selection step had been performed. These efforts study drug-target interactions from different perspectives and can thus be considered *complementary*; this work performed a study to determine which patterns of chemical structures and protein domains tend to co-exist in drug-target interactions, whereas He et al. observed other features (pseudo amino acid composition and common chemical functional groups) to see which features are the most important in determining whether or not a drug-target pair interacts.

2.6.6 Support Vector Machines & Minwise Hashing

In [112], drugs were represented as PubChem fingerprints (881 chemical substructures) and proteins were represented as domain fingerprints (4,137 Pfam domains). Given a drug vector $\Phi(C)$ and a protein vector $\Phi(P)$, a compound-protein pair fingerprint $\Phi(C, P)$ is then obtained by the tensor product of $\Phi(C)$ and $\Phi(P)$ as

$$\Phi(C, P) = \Phi(C) \otimes \Phi(P) \quad (2.42)$$

resulting in a binary vector that is 3,644,697 elements long. Dimensionality reduction is then achieved by applying *minwise hashing* [113] to the compound-protein fingerprints in order to convert them to *compact fingerprints* to make the algorithm scalable to large datasets.

After the dimensionality of the data has been reduced via minwise hashing, a *linear SVM* classifier is trained on the data and then used for prediction. Two variants have been considered: one with an L_2 regularization term (*MH-L2SVM*) and another with an L_1

regularization term (*MH-L1SVM*). The two variants were found to produce similar prediction performance. However, the learned weight vector from the *MH-L1SVM* is more interesting because the number of features extracted was much smaller than that of *MH-L2SVM* (i.e. less features to inspect for insights).

Other than predicting new interactions, one may wish to observe the learned weight vector for meaningful insights. However, note that the learned weight vector corresponds to the *dimensionality-reduced* data; i.e. the length of the weight vector is equal to that of the compact fingerprints. This means that the weight vector is not useful in its current form. As such, the inverse of the minwise operation mentioned earlier is used to convert the learned weight vector into a final weight vector that corresponds to the original fingerprints. This final weight vector is then inspected for biological interpretation.

Note that *MH-L1SVM* is similar to *SCCA* (from Section 2.6.5) in that they both use the same data as input (i.e. chemical substructures and protein domains) and they are both *interpretable* classifiers whose learned weight vectors can be inspected for interesting findings. However, with the help of dimensionality reduction via minwise hashing, *MH-L1SVM* is more suitable for larger datasets. That the minwise hashing technique has an inverse operation helps preserve the interpretability of *MH-L1SVM* by applying said operation on the weight vector that is learned on the dimensionality-reduced data. This makes *MH-L1SVM* both interpretable *and* scalable.

2.7 Summary

We presented a summary of the chemogenomic DTI prediction methods that have been developed in the last decade, provided brief descriptions for them and divided them into meaningful categories. Table 2.3 lists the different DTI prediction methods that were discussed in this chapter and the categories that they belong to. Note that there are efforts that were not covered in the above literature review such as those utilizing deep learning in their prediction of drug-target interactions [114-119]. These efforts will be

elaborated upon (among other things) in the last chapter of this thesis document when we discuss future research directions in drug-target interaction prediction.

Categories	Methods
Neighborhood	<i>Nearest Profile</i> and <i>Weighted Profile</i> [45], <i>SRP</i> [81]
Bipartite Local Models	<i>SVM-based BLM</i> [49], <i>LapRLS</i> [83], <i>RLS-avg</i> and <i>RLS-kron</i> [87], <i>BLM-NII</i> [90], <i>RLS-WNN</i> [91]
Network Diffusion	<i>NBI</i> [92], <i>HGBI</i> [93], <i>NRWRH</i> [95], <i>PSL</i> [96], <i>DASPfind</i> [97]
Matrix Factorization	<i>KBMF2K</i> [98], <i>PMF</i> [99], <i>CMF</i> [100], <i>NRLMF</i> [101], <i>DNILMF</i> [104]
Feature-based Classification	He et al. [105], Yu et al. [107], <i>Fuzzy KNN</i> [108], <i>SITAR</i> [109], <i>SCCA</i> [110], <i>MH-L1SVM</i> [112]

Table 2.3: The categories of the different methods for predicting DTIs

Chapter 3

Graph Regularized Matrix

Factorization

Predicting interactions for *new* drugs or targets is common in drug discovery. New drugs and targets are those with unknown interaction profiles; that is, no targets are known for new drugs, and new targets are not known to interact with any drugs. Predicting interactions for *new drugs* is important; when a novel drug is developed, one needs to predict targets for chemical compounds since they are potential drug candidates that have not been used as drugs previously. Determining drugs that bind *new targets* is also beneficial because one may wish to target a protein that has not been targeted previously by other drugs, which is a common scenario in drug repositioning.

However, it turns out that predicting interactions for *new* drugs or targets is a challenging task. While some existing methods have been proposed to improve the accuracy of predictions for new drugs or targets, their results show that there is still room for improvement [90, 91].

Given a drug-target interaction network (as in Figure 1.3), we assume that a portion of the missing edges (or non-edges) in the network are actual, unknown interactions. Based on this assumption, we designed a preprocessing step that adds edges with intermediate interaction likelihood scores to assist with prediction. Additional motivation for this preprocessing step was provided by [120] where the importance of nearest-neighbor information in drug-target interaction prediction was emphasized—the authors were interested in predicting ligands for orphan (i.e. new) targets, and their experiments have shown that ligands of orphan targets could be predicted with reasonable accuracy

whenever close neighbors (with known ligands) of those orphan targets were available, regardless of what kernels were used for ligands or targets. Note that, like ligands, most drugs are small molecules, so this applies to drug-target interactions as well.

Furthermore, previous work has shown that data usually lie on (or near to) a non-linear manifold with lower dimensionality than that of the original data [84-86]. The term, *manifold*, refers to a low-dimensional surface embedded in a high-dimensional space (see Figure 3.1 for an example). Modeling such a manifold may potentially lead to improvements in prediction performance; geodesic distances from the determined manifold are likely more accurate than distances computed in the original high-dimensional space. Furthermore, that the manifold is of lower dimensionality may reduce noise in the data, thus leading to more robust predictions. As such, we additionally propose *GRMF* (Graph Regularized Matrix Factorization), a matrix factorization method that uses graph regularization to implicitly learn the underlying manifolds in order to further improve prediction performance. The work presented in this chapter was recently published in [121].

Figure 3.1: The swiss roll dataset. The distance between two arbitrary points in the original high-dimensional space (dashed line) may not be accurate as the geodesic distance along the manifold (solid line) in representing the similarity/difference between them. **This figure is borrowed from [84].**

3.1 Data

We use the same four datasets introduced in [45] which correspond to four different target protein types, namely nuclear receptors (NR), G protein-coupled receptors (GPCR), ion channels (IC) and enzymes (E). The following table contains some simple statistics for the four datasets.

Datasets	NR	GPCR	IC	E
Drugs	54	223	210	445
Targets	26	95	204	664
Interactions	90	635	1,476	2,926

Table 3.1: Drugs, targets, and interactions in each dataset

Each dataset contains three matrices: $Y \in \mathbb{R}^{n \times m}$, $S_d \in \mathbb{R}^{n \times n}$, and $S_t \in \mathbb{R}^{m \times m}$. As mentioned earlier in Section 2.1.3, the matrix Y is the adjacency matrix encoding the drug-target interactions with n drugs as rows and m targets as columns, where $Y_{ij} = 1$ if drug d_i and target t_j are known to interact and $Y_{ij} = 0$ otherwise. The matrix S_d represents the drug pairwise chemical structure similarities and the matrix S_t denotes the target pairwise genomic sequence similarities. Drug similarities were obtained using SIMCOMP [79], and the target similarities are normalized Smith-Waterman scores [80]. As indicated in Section 2.1.3, this is a famous benchmark dataset that has been used in many previous efforts in drug-target interaction, so we use it here to demonstrate the prediction performance of our proposed methods against other state-of-the-art methods.

3.2 Weighted K Nearest Known Neighbors

The given drug-target matrix $Y \in \mathbb{R}^{n \times m}$ has n drug rows and m target columns. The i^{th} row in Y , denoted as $Y(d_i)$, is the interaction profile for drug d_i . Similarly, the j^{th} column in Y , denoted as $Y(t_j)$, is the interaction profile for target t_j . A drug (or target) being *known* means that it has at least one interaction in its profile, while it being *new* means

that it has no interactions in its profile. Some of the non-interactions (or 0's) in Y are unknown cases that could potentially be true interactions (i.e. they are false negatives). Thus, we propose *WKNKN* as a preprocessing step to estimate the interaction likelihoods for these unknown cases based on their *known* neighbors. That is, assuming Y_{ij} equals 0, *WKNKN* replaces it by a continuous value in the range 0 to 1 as follows:

- 1) *Horizontal direction update*: Get weighted average of the corresponding values in the profiles of the K known drugs nearest to d_i (the *weights* are the similarities of d_i to these nearest neighbors).
- 2) *Vertical direction update*: Get weighted average of the corresponding values in the profiles of the K known targets nearest to t_j (the *weights* are the similarities of t_j to these nearest neighbors).
- 3) *Final update*: Replace the $Y_{ij} = 0$ by taking the average of the above two values, representing the overall score that d_i and t_j interact.

Figure 3.2 contains the pseudocode that describes the above procedure in detail. η is a decay term that the *weights* mentioned above are multiplied by where $\eta \leq 1$, and *KNearestKnownNeighbors()* returns the K nearest known neighbors in descending order based on their similarities to d_i or t_j .

Note that to infer the interaction likelihood for drug-target pairs, *WKNKN* does not simply use the K nearest neighbors; it uses the K nearest *known* neighbors, which is reasonable since *known* neighbors, having additional interaction information, would contribute more than *unknown* neighbors whose interaction profiles are all 0's. According to [120], the main factor affecting the accuracy of predicting interactions for orphan drugs and targets is the quality of the nearest neighbors being inferred from. By excluding the unknown neighbors that have all-0's interaction profiles, we can ensure (to some extent) that the neighbors being inferred from are of reasonable quality. This point is important since interaction prediction for orphan drugs and targets is a main focus of this work.

```

function WKNKN( $Y, S_d, S_t, K, \eta$ )
   $Y_d = Y_t = 0$ 
  for  $d \leftarrow 1$  to  $n$  do
     $dnn = KNearestKnownNeighbors(d, S_d, K)$ 
    for  $i \leftarrow 1$  to  $K$  do
       $w_i = \eta^{i-1} S_d(d, dnn_i)$ 
    end for
     $Z_d = \sum_{i=1}^K S_d(d, dnn_i)$  // normalization term
     $Y_d(d) = \frac{1}{Z_d} \sum_{i=1}^K w_i Y(dnn_i)$ 
  end for
  for  $t \leftarrow 1$  to  $m$  do
     $tnn = KNearestKnownNeighbors(t, S_t, K)$ 
    for  $j \leftarrow 1$  to  $K$  do
       $w_j = \eta^{j-1} S_t(t, tnn_j)$ 
    end for
     $Z_t = \sum_{j=1}^K S_t(t, tnn_j)$  // normalization term
     $Y_t(t) = \frac{1}{Z_t} \sum_{j=1}^K w_j Y(tnn_j)$ 
  end for
   $Y_{dt} = (Y_d + Y_t)/2$ 
   $Y = \max(Y, Y_{dt})$ 
  return  $Y$ 
end function

```

Figure 3.2: WKNKN algorithm

3.3 Graph-Regularized Matrix Factorization

Manifold learning depends on the *local invariance assumption* which states that points close to each other in the original data should also be close to each other in the learned manifold. In other words, the manifold to which the data will be mapped is learned such that the local geometries in the original data are preserved [122]. In this section, we describe our proposed method, *GRMF*, which learns manifolds implicitly via graph regularization terms that have been added to a collaborative filtering objective function.

Before moving on, we briefly describe here how the use of graph regularization leads to learning the manifold underlying the data. First, note that matrix factorization decomposes the matrix Y into two matrices A and B where $Y = AB^T$. The rows of A and B are the latent feature vectors corresponding to the drugs (rows) and targets (columns) of Y , respectively. According to the local invariance assumption mentioned above, we wish

the latent vectors of neighboring objects (i.e. drugs or targets) to be similar; that is, if we have prior knowledge that two objects are neighbors (i.e. they are sufficiently similar) then the objects' latent vectors that are derived from matrix factorization should be similar as well. Graph regularization helps achieve this by finding the nearest neighbors for each object and encouraging its derived latent vector to be similar to those of its nearest neighbors.

Moreover, we remind the reader that enhancing predictions for new (or orphan) drugs and targets is a main focus of this work. That only the nearest neighbors are used per drug or target is desirable since, as mentioned earlier, prediction performance on orphan objects is mainly determined by nearest-neighbor relations [120], which motivated the use of graph regularization in this work. Note that this contrasts with *CMF* (from Section 2.5.3) where the latent feature matrices are determined in a way such that their inner products fully reconstruct the similarity matrices S_d and S_t (i.e. all the objects are used and not just the near neighbors).

3.3.1 Sparsification of the Similarity Matrices

Sparsification of similarity matrices is a technique that has been used before with graph regularization [122]. In this work, we derived a p -nearest neighbor graph from each of the drug and target similarity matrices, S_d and S_t . That is, given the drug similarity matrix S_d , a p -nearest neighbor graph is generated as

$$\forall i, j, \quad N_{ij} = \begin{cases} 1 & d_j \in \mathbf{NN}_p(d_i) & d_i \in \mathbf{NN}_p(d_j) \\ 0 & d_j \notin \mathbf{NN}_p(d_i) & d_i \notin \mathbf{NN}_p(d_j) \\ 0.5 & \text{otherwise} \end{cases} \quad (3.1)$$

where $\mathbf{NN}_p(d_i)$ and $\mathbf{NN}_p(d_j)$ are the sets of p nearest neighbors to drugs d_i and d_j , respectively. In other words, if d_i and d_j are in each other's nearest-neighbor lists, then $N_{ij} = 1$; if neither is in the other's list, then $N_{ij} = 0$; otherwise, $N_{ij} = 0.5$. N is then used to sparsify the similarity matrix S_d as

$$\forall i, j, \quad (\hat{S}_d)_{ij} = N_{ij}(S_d)_{ij} \quad (3.2)$$

where $(S_d)_{ij} = S_d(d_i, d_j)$. This results in a sparse similarity matrix for drugs. The same procedure is done for the target similarity matrix S_t . $\hat{S}_d$ and $\hat{S}_t$ now hold the local similarities of the drugs and targets to their nearest neighbors, respectively, and they will be used in Section 3.3.3 to model the previously mentioned local invariance assumption to determine the manifold.

3.3.2 Low-Rank Approximation

GRMF depends on the basic idea of low rank approximation (LRA) which decomposes the drug-target matrix $Y \in \mathbb{R}^{n \times m}$ into two low-rank latent feature matrices $A \in \mathbb{R}^{n \times k}$ (for drugs) and $B \in \mathbb{R}^{m \times k}$ (for targets) which minimize the LRA objective

$$\min_{A, B} \|Y - AB^T\|_F^2 \quad (3.3)$$

where $\|\cdot\|_F$ is the Frobenius norm and k is a parameter indicating the number of latent features in A and B . Typically, $k \ll n, m$. Also note that k here is different from K of *WKNKN*.

3.3.3 Regularization

To prevent overfitting and increase generalization capability, Tikhonov and graph regularization terms are added, giving *GRMF*'s objective function

$$\begin{aligned} \min_{A, B} & \|Y - AB^T\|_F^2 \\ & + \lambda_l (\|A\|_F^2 + \|B\|_F^2) \\ & + \lambda_d \sum_{i, r=1}^n (\hat{S}_d)_{ir} \|a_i - a_r\|^2 \\ & + \lambda_t \sum_{j, q=1}^m (\hat{S}_t)_{jq} \|b_j - b_q\|^2 \end{aligned} \quad (3.4)$$

where λ_l , λ_d and λ_t are positive parameters, a_i and b_j are the i^{th} and j^{th} rows of A and B , respectively, and n and m are the numbers of drugs and targets, respectively. The first term requires the model to approximate the matrix Y . The second term is the Tikhonov regularization which minimizes the norms of both A and B , which helps prevent overfitting. The third term is for drug graph regularization which minimizes the distance between latent feature vectors of two neighboring drugs. The final term is for target graph regularization. The above equation can be rewritten as:

$$\begin{aligned}
& \min_{A,B} \|Y - AB^T\|_F^2 \\
& + \lambda_l (\|A\|_F^2 + \|B\|_F^2) \\
& + \lambda_d \text{Tr}(A^T L_d A) \\
& + \lambda_t \text{Tr}(B^T L_t B)
\end{aligned} \tag{3.5}$$

where $\text{Tr}(\cdot)$ is the trace of a matrix, $L_d = D_d - \hat{S}_d$ and $L_t = D_t - \hat{S}_t$ are the graph Laplacians for $\hat{S}_d$ and $\hat{S}_t$, respectively, and $(D_d)_{ii} = \sum_r (\hat{S}_d)_{ir}$ and $(D_t)_{jj} = \sum_q (\hat{S}_t)_{jq}$ are diagonal matrices. For more details on the rewriting of the graph regularization terms, please refer to [123].

Furthermore, since normalized graph Laplacians are known to perform better than their unnormalized versions in many cases [124], the graph Laplacians L_d and L_t are replaced by their normalized counterparts $\tilde{L}_d = (D_d)^{-1/2} L_d (D_d)^{-1/2}$ and $\tilde{L}_t = (D_t)^{-1/2} L_t (D_t)^{-1/2}$ giving

$$\begin{aligned}
& \min_{A,B} \|Y - AB^T\|_F^2 \\
& + \lambda_l (\|A\|_F^2 + \|B\|_F^2) \\
& + \lambda_d \text{Tr}(A^T \tilde{L}_d A) \\
& + \lambda_t \text{Tr}(B^T \tilde{L}_t B).
\end{aligned} \tag{3.6}$$

Intuitively, minimizing *GRMF*'s objective function would rank the interactions above the non-interactions by forcing their prediction scores to tend to 1 and 0, respectively.

However, since some of the 0's are likely interactions that are not discovered yet, this may lead to unsatisfactory results. In this work, *WKNKN* is applied to the datasets prior to running *GRMF* to address this problem in advance; applying *WKNKN* replaces the 0's with intermediate interaction values in the range $[0,1]$.

3.3.4 Initialization of A and B

The first step in the *GRMF* procedure (see Figure 3.3) is to initialize A and B . To do so, we adapted a method inspired by the procedure used in [125]. Specifically, we decomposed $Y \in \mathbb{R}^{n \times m}$ into $U \in \mathbb{R}^{n \times k}$, $S_k \in \mathbb{R}^{k \times k}$ and $V \in \mathbb{R}^{m \times k}$ such that US_kV^T is the closest k -rank approximation to Y where U and V are matrices having orthonormal columns, and S_k is a diagonal matrix containing the k largest singular values. The maximum possible number of singular values in Y is $\min(n, m)$, thus $k_{max} = \min(n, m)$. Finally, we get the square root of S_k and let $A = US_k^{1/2}$ and $B = VS_k^{1/2}$.

While other possible initialization methods for A and B exist [126], we found this method to be quite robust in that results obtained by it were highly reproducible relative to other methods. For example, one of the other initialization methods involves randomly initializing A and B with values from 0 to 1; using this initialization method, different initializations can have very different end results for *GRMF* and can take varying numbers of iterations for *GRMF*'s alternating least squares to converge. In contrast, our selected method was found to produce similar end results for *GRMF* and a few iterations were required each time.


```

function GRMF( $Y, S_d, S_t, K, \eta, k, \lambda_l, \lambda_d, \lambda_t$ )
   $Y = WKNKN(Y, S_d, S_t, K, \eta)$ 
   $[U, S_k, V] = SVD(Y, k)$ 
   $A = US_k^{1/2}$ 
   $B = VS_k^{1/2}$ 
  Compute  $\hat{S}_d, \hat{S}_t$  from  $S_d, S_t$ 
  Compute  $\tilde{L}_d, \tilde{L}_t$  from  $\hat{S}_d, \hat{S}_t$ 
  repeat
     $A = (YB - \lambda_d \tilde{L}_d A)(B^T B + \lambda_l I_k)^{-1}$ 
     $B = (Y^T A - \lambda_t \tilde{L}_t B)(A^T A + \lambda_l I_k)^{-1}$ 
  until convergence
   $\hat{Y} = AB^T$ 
  return  $\hat{Y}$ 
end function

```

Figure 3.3: GRMF algorithm

3.3.5 Alternating Least Squares

Next, alternating least squares is used to obtain the solution. Denoting the objective function in the Equation 3.6 as $\mathbf{L}$, we derive the following two alternative update rules (by setting $\frac{\partial \mathbf{L}}{\partial A} = 0$ and $\frac{\partial \mathbf{L}}{\partial B} = 0$) which are run alternately until convergence

$$A = (YB - \lambda_d \tilde{L}_d A)(B^T B + \lambda_l I_k)^{-1}, \quad (3.7)$$

$$B = (Y^T A - \lambda_t \tilde{L}_t B)(A^T A + \lambda_l I_k)^{-1}. \quad (3.8)$$

3.4 Weighted GRMF

WGRMF is another variant of *GRMF* that has a weight matrix W . The weight matrix W prevents unknown instances (for which interaction information is not available) from contributing to the determination of the latent feature matrices A and B which reconstruct the drug-target matrix Y . Including W makes the objective function in Equation 3.6 look as follows

$$\begin{aligned}
& \min_{A, B} \|W \odot (Y - AB^T)\|_F^2 \\
& + \lambda_l (\|A\|_F^2 + \|B\|_F^2) \\
& + \lambda_d \text{Tr}(A^T \tilde{L}_d A) \\
& + \lambda_t \text{Tr}(B^T \tilde{L}_t B)
\end{aligned} \quad (3.9)$$

where $\odot$ is the element-wise product. Denoting the objective function in the above equation as $\mathbf{L}$ and letting a_i and b_j be the i^{th} and j^{th} row vectors of A and B , respectively, we set $\frac{\partial \mathbf{L}}{\partial a_i} = 0$ and $\frac{\partial \mathbf{L}}{\partial b_j} = 0$ to derive the following two update rules that are run alternately until convergence

$$\forall i = 1 \dots n, \quad a_i = \left(\sum_{j=1}^m W_{ij} Y_{ij} b_j - \lambda_d (\tilde{L}_d)_{i*} A \right) \left(\sum_{j=1}^m W_{ij} b_j^T b_j + \lambda_l I_k \right)^{-1}, \quad (3.10)$$

$$\forall j = 1 \dots m, \quad b_j = \left(\sum_{i=1}^n W_{ij} Y_{ij} a_i - \lambda_t (\tilde{L}_t)_{j*} B \right) \left(\sum_{i=1}^n W_{ij} a_i^T a_i + \lambda_l I_k \right)^{-1}. \quad (3.11)$$

where $(\tilde{L}_d)_{i*}$ and $(\tilde{L}_t)_{j*}$ are the i^{th} and j^{th} row vectors of $\tilde{L}_d$ and $\tilde{L}_t$, respectively. Note that the above update rules differ from *GRMF*'s update rules (Equations 3.7 and 3.8); *GRMF*'s rules are *matrix-wise* update rules, while those of *WGRMF* are *row-wise* update rules.

3.5 Experimental Results

We performed experiments to compare our proposed methods with the existing techniques *BLM-NII* [90], *RLS-WNN* [91] and *CMF* [100] (refer to Chapter 2 for more details on these methods). Specifically, we conducted five repetitions of 10-fold cross validation (CV) for each of the methods, both with and without *WKNKN* as a preprocessing step. In each repetition of 10-fold CV, Y was divided into 10 folds and each fold, in turn, was left out as the test set while the remaining nine folds were treated as the training set. In previous studies (e.g. [45, 90, 91]), the Area Under the Precision-Recall curve (AUPR) [127] was employed as the main metric for performance evaluation. As such, AUPR was also used in this work as our evaluation metric. Furthermore, AUPR heavily penalizes non-interactions that are highly ranked, which is desirable here because, in practice, we do not want incorrect predictions to be recommended by the prediction algorithm (i.e. the AUPR metric heavily punishes highly ranked false positives).

[127]). In our experiments, we calculated an AUPR score for each 10-fold CV repetition, and the final AUPR score was the average over five such repetitions.

To test different aspects of the prediction methods, we performed CV under two scenarios described in [128]:

- 1) S2, where entire drug interaction profiles are left out to be used as the test set;
- 2) S3, where entire target interaction profiles are left out to be used as the test set.

Given an interaction prediction method, S2 tests its ability to predict interactions for new drugs while S3 tests its ability to predict interactions for new targets. Although the focus of this work is to specifically improve predictions for the new drug and target cases (for which the S2 and S3 CV settings are adequate for testing), we added a section at the end of this chapter containing CV results under S1 which is the standard CV setting where random drug-target pairs are left out as the test set. Figure 3.4 illustrates the different CV settings: S1, S2 and S3.

Figure 3.4: The different cross validation settings: (a) S1 involves leaving out random drug-target pairs from the interaction matrix Y to use as the test set, (b) S2 is the setting where entire drug profiles are left out, and (c) S3 leaves out entire target profiles. Gray boxes represent left-out test instances.

Tables 3.2 and 3.3 contain the results from the above-mentioned cross validation scenarios, and these results are explained in the coming subsections. Corresponding plots of the precision-recall curves are also provided in GRMF's website: <http://www.comp.nus.edu.sg/~lixl/GRMF/index.html>

3.5.1 Parameter Settings

We performed cross validation on the training set for setting *GRMF*'s parameters, namely k (rank of matrices A and B), λ_l , λ_d and λ_t . Using grid search, the best parameter combination is obtained from the values: $k \in [50, 100]$, $\lambda_l \in [2^{-2}, 2^{-1}, 2^0, 2^1]$, $\lambda_d/\lambda_t \in [0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$. Note that, as mentioned in Section 3.3.4, k could not be more than $\min(n, m)$; e.g. if the value being tested for k is 50 and $\min(n, m) < 50$, we then set $k = \min(n, m)$. *WKNKN*'s parameters (K and η) were also set using grid search. As for p , it was set to $p = 5$ as it was found to produce reasonable results.

In the case of *WGRMF*, for instances (d_i, t_j) in Y that were in the test set, we set $W_{ij} = 0$ in the weight matrix W , which means that such instances would not contribute to the updating of the latent feature matrices A and B which will be used to reconstruct the final predictions matrix $\hat{Y}$.

For the other methods, we set the parameters to their optimal values wherever possible. In the end, all parameters were set to their default values (which are already optimal) with the exception of η which was set as 0.7 in *RLS-WNN*. For *CMF*, the parameters were obtained as mentioned in [100].

3.5.2 Interaction Prediction under S2

Results under S2 are shown in Table 3.2. In the NR dataset, *GRMF* and *WGRMF* are comparable with *RLS-WNN*. While *CMF* is better than *GRMF* in the other three datasets, *WGRMF* is consistently better than *CMF* in all datasets, though the results are almost the same in the GPCR dataset. Without the use of *WKNKN* as a preprocessing step, *WGRMF* is the superior method.

Furthermore, the additional weight matrix W in *WGRMF* provided a good boost to the prediction performance as compared with *GRMF* (except in the NR dataset). In addition, thanks to the graph regularization term in *WGRMF*, it displayed an obvious improvement over *CMF* overall, supporting the usefulness of manifold learning.

Methods	NR	GPCR	IC	E
BLM-NII	0.410 (0.043)	0.233 (0.010)	0.201 (0.009)	0.167 (0.011)
RLS-WNN	0.519 (0.025)	0.363 (0.008)	0.319 (0.011)	0.386 (0.014)
CMF	0.482 (0.034)	0.406 (0.008)	0.350 (0.008)	0.375 (0.007)
GRMF	0.517 (0.025)	0.369 (0.011)	0.341 (0.016)	0.349 (0.012)
WGRMF	0.520 (0.025)	0.408 (0.010)	0.364 (0.018)	0.404 (0.014)
WKNKN	0.529 (0.015)	0.399 (0.011)	0.352 (0.014)	0.388 (0.011)
WKNKN+BLM-NII	0.514 (0.023)	0.386 (0.005)	0.350 (0.014)	0.385 (0.005)
WKNKN+RLS-WNN	0.523 (0.030)	0.395 (0.008)	0.352 (0.014)	0.385 (0.007)
WKNKN+CMF	0.515 (0.032)	0.409 (0.010)	0.350 (0.015)	0.385 (0.004)
WKNKN+GRMF	0.542 (0.028)	0.404 (0.011)	0.356 (0.014)	0.390 (0.010)
WKNKN+WGRMF	0.528 (0.033)	0.410 (0.012)	0.369 (0.017)	0.401 (0.013)

Table 3.2: AUPR Results for Interaction Prediction Under S2

Best AUPR result in each column is **bold**. Standard deviations are given in (parentheses).

After applying WKNKN, in all datasets, all of *GRMF*'s results greatly improved after applying *WKNKN*. On the other hand, the improvements in *WGRMF* after applying *WKNKN* were not as pronounced; there was some improvement in the NR dataset and slight improvements in the GPCR and IC datasets, while the AUPR score actually decreased in the E dataset. In the end, *WGRMF* is still the superior method. In addition, *GRMF* generally displayed comparable results to *WGRMF*, and in the NR dataset, *GRMF* did better than *WGRMF*.

3.5.3 Interaction Prediction under S3

Results under S3 are shown in Table 3.3. The first thing that can be observed is that AUPR scores under S3 are generally higher than those under S2. That is, with absent target interaction profiles, the different methods can still achieve good prediction performance, whereas their performance is much decreased if drug interaction profiles are hidden instead. On first glance, this observation seems to suggest that target sequence similarity is more reliable and informative than drug chemical similarity. However, although this conclusion was also stated in previous work [49, 87], there may be an additional factor in play that is causing the generally lower AUPRs under S2, namely that the average number of interactions per drug is lower than that per target. In other words, drugs have fewer interactions to infer from than targets, which could be contributing to the lower AUPRs under S2.

In the NR dataset, *RLS-WNN* is the best method, while in the rest of the datasets, *WGRMF* is the superior method. As in the S2 results, due to the additional weight matrix W , *WGRMF* is better than *GRMF* on all datasets (except on the NR dataset where the results are the same). Furthermore, *WGRMF* achieved better results than *CMF* thanks to the graph regularization terms.

Methods	NR	GPCR	IC	E
BLM-NII	0.418 (0.022)	0.447 (0.024)	0.634 (0.008)	0.583 (0.021)
RLS-WNN	0.468 (0.030)	0.547 (0.025)	0.746 (0.004)	0.761 (0.015)
CMF	0.379 (0.020)	0.540 (0.028)	0.751 (0.014)	0.740 (0.014)
GRMF	0.423 (0.032)	0.567 (0.027)	0.745 (0.008)	0.763 (0.020)
WGRMF	0.423 (0.017)	0.574 (0.027)	0.801 (0.008)	0.801 (0.018)
WKNKN	0.465 (0.034)	0.572 (0.027)	0.787 (0.010)	0.792 (0.016)
WKNKN+BLM-NII	0.460 (0.032)	0.607 (0.027)	0.794 (0.005)	0.814 (0.017)
WKNKN+RLS-WNN	0.471 (0.028)	0.603 (0.020)	0.806 (0.007)	0.809 (0.017)
WKNKN+CMF	0.434 (0.029)	0.557 (0.021)	0.742 (0.015)	0.772 (0.014)
WKNKN+GRMF	0.500 (0.028)	0.615 (0.023)	0.815 (0.010)	0.807 (0.016)
WKNKN+WGRMF	0.446 (0.015)	0.585 (0.027)	0.799 (0.007)	0.798 (0.018)

Table 3.3: AUPR Results for Interaction Prediction Under S3

Best AUPR result in each column is **bold**. Standard deviations are given in (parentheses).

After applying *WKNKN*, in the NR, GPCR and IC datasets, *GRMF* is the superior method. In contrast to *GRMF* whose results greatly improved after applying *WKNKN*, the AUPR values did not increase as much for *WGRMF* (in fact, there were decreases in the IC and E dataset). In the GPCR, IC and E datasets, the performances of *BLM-NII* and *RLS-WNN* are comparable to that of *GRMF*, especially in the E dataset.

Interestingly, with *WKNKN* as a preprocessing method, *GRMF* did better than *WGRMF* on all datasets, especially on the NR dataset. Note that *WGRMF*'s weight matrix W forces the instances of the test set to not contribute in the updating of the latent feature matrices A and B (that are used to obtain predictions later); this weight matrix W indeed improves the results when *WKNKN* is not being used. However, after applying *WKNKN* which updates the drug-target matrix Y , the weight matrix W also forces the test set instances (despite their *updated* values) to not contribute to the predictions in *WGRMF*. On the other hand, *GRMF* does not have a weight matrix W so these instances (and their updated values) were not ignored. Since target information is more important than drug

information under S3 and is generally more reliable, this has made *GRMF*'s results better than those of *WGRMF*.

3.5.4 Interaction Prediction with WKNKN

In Tables 3.2 and 3.3, *WKNKN* was tested by itself and added as a baseline method. Interestingly, *WKNKN* produced results that are quite competitive; under both S2 and S3, *WKNKN* set a challenging baseline that was difficult for most methods to surpass. Only *WGRMF* was able to consistently produce results that are better than those of *WKNKN* (except in the NR dataset). That *WGRMF* and *WKNKN* are the superior methods under S2 and S3 (as shown in the upper halves of Tables 3.2 and 3.3) can be attributed to a trait that they have in common, namely that they both focus on only the nearest neighbors (via the parameters p and K , respectively), which is important for improved predictions on orphan drugs (S2) and targets (S3) [120].

Furthermore, while *WKNKN* generally improved the results of the different methods (as shown in the last five rows in Tables 3.2 and 3.3), many of the results were still not better than those of *WKNKN* itself. However, for *WKNKN+GRMF* and *WKNKN+WGRMF*, improvements over *WKNKN* itself were realized in all datasets (as shown in the last two rows of each table).

3.5.5 Sensitivity Analysis

The optimal parameter values for k , λ_l , λ_d and λ_t are determined automatically via cross validation on the training set during the execution of *GRMF* (refer to Section 3.5.1); these values may differ from one fold to another depending on the training set used in each, which complicates the task of plotting sensitivity analysis results for the parameters.

In order to study the roles of λ_l , λ_d and λ_t , we compared *GRMF*'s results from Tables 3.2 and 3.3 against variants of *GRMF* where we set $\lambda_l = 0$ for the first variant, $\lambda_d = 0$ for the second variant and $\lambda_t = 0$ for the third. Results for these *GRMF* variants under S2 and S3 are reported in Tables 3.4 and 3.5, respectively. It was found that setting $\lambda_l = 0$ negatively

impacts the performance under both S2 and S3. As for λ_d , setting it to 0 negatively impacts results under S2, but not so much under S3. Vice versa for λ_t , setting it to 0 negatively impacts results under S3, but not so much under S2. This means that λ_d is important under S2, while λ_t is important under S3.

Methods	NR	GPCR	IC	E
GRMF	0.517	0.369	0.341	0.349
GRMF ($\lambda_l = 0$)	0.105	0.293	0.263	0.264
GRMF ($\lambda_d = 0$)	0.122	0.049	0.034	0.011
GRMF ($\lambda_t = 0$)	0.516	0.369	0.341	0.349

Table 3.4: AUPR Results for GRMF Variants under S2

Best AUPR result in each column is **bold**.

Methods	NR	GPCR	IC	E
GRMF	0.423	0.567	0.745	0.763
GRMF ($\lambda_l = 0$)	0.094	0.417	0.603	0.671
GRMF ($\lambda_d = 0$)	0.411	0.560	0.741	0.761
GRMF ($\lambda_t = 0$)	0.089	0.039	0.037	0.011

Table 3.5: AUPR Results for GRMF Variants Under S3

Best AUPR result in each column is **bold**.

As for k (the number of latent features), it was found that the higher the value of k , the better the prediction performance. However, due to computational complexity reasons, we set the value of k to a maximum of 100. In addition, the results were found not to improve much beyond $k = 100$.

On the other hand, the parameters K and η (from *WKNKN*) are fixed before running *GRMF*. Sensitivity analyses are provided for these two parameters in Figures 3.5, 3.6, 3.7 and 3.8. For all datasets under both S2 and S3, it appears that prediction performance does not improve much for values of $K > 5$. As for η , increasing its value generally improves the prediction performance until $\eta = 0.7$ after which the AUPR decreases, but with two exceptions. The first exception is the NR dataset (under both S2 and S3) where the AUPR starts decreasing from $\eta = 0.5$, showing a preference for lower values of η . The low value of η (i.e. high decay rate) may be explained by the smaller number of reliable neighbors to infer from (refer to Table 3.1 for numbers of drugs and targets in each

dataset). This contrasts with the second exception, namely the E dataset under S3 where the AUPR keeps improving all the way till $\eta = 1$. This may be due to the abundance of reliable neighbors to infer from; note that the number of targets (which are more important under S3) is greater than the number of drugs in the E dataset (see Table 3.1).

Figure 3.5: Sensitivity analysis for K under S2

Figure 3.6: Sensitivity analysis for η under S2

Figure 3.7: Sensitivity analysis for K under $S3$

Figure 3.8: Sensitivity analysis for η under $S3$

3.5.6 General Comments

Following are some remaining findings and conclusions reached from our CV experiments:

- Contrary to the other datasets, the results of the NR dataset under S3 are lower than those under S2. In [128], this was concluded to be due to the small size of the NR dataset (see Table 3.1) which is causing the results to be unstable. We complement this by saying that target information is more important for inference under S3, and since the number of targets in the NR dataset is very small, this means that there are few reliable targets to infer from (as opposed to S2 where there are relatively more drugs to infer from). Thus, the prediction performance is negatively affected by the lack of a sufficient number of neighboring targets.
- While the prediction scores may differ between datasets, this may not be solely due to the type of interactions in each dataset (each dataset contains a different type of interactions involving a specific kind of target proteins) but also due to the number of drugs and targets in each dataset and the reliability of the drug and target similarities used as well as the cross validation scenario being performed. For example, in both the NR and GPCR dataset, the drug-to-target ratio is higher than in the IC and E datasets (see Table 3.1); we believe this is why all methods achieved higher AUPR for the NR and GPCR datasets than for the IC and E datasets under S2 (where drug information is more important). In addition, it seems that the same drug-to-target ratio caused the opposite to take place under S3 (where target information is more important)—all methods achieved higher AUPR for the IC and E datasets than for the NR and GPCR datasets. In conclusion, it is important to take into account the drug-to-target ratio while studying the prediction performance of any method on the different datasets.

- By observing the results of our proposed methods in Tables 3.2 and 3.3, modeling the manifold structures of the drug and target spaces (via the drug and target graph regularization terms, respectively) was shown to improve prediction performance in terms of AUPR, indicating the effectiveness of the proposed graph regularization.
- Finally, an advantage of *GRMF* over *WGRMF* that is not apparent in the results is that *GRMF* runs much faster than *WGRMF* because *GRMF* uses *matrix-wise* update rules for A and B (Equations 3.7 and 3.8), while *WGRMF* is forced to use *row-wise* update rules (Equations 3.10 and 3.11) due to the weight matrix W . Equations 3.7, 3.8, 3.10 and 3.11 all contain matrix multiplication and inversion operations. Equations 3.7 and 3.8 are each run only once. As for Equations 3.10 and 3.11, the matrix inversion operation (the rightmost term in the equations) is run n and m times, respectively. This means that Equations 3.10 and 3.11 perform $(n + m - 2)$ more inversion operations than Equations 3.7 and 3.8. Since matrix inversion, at best, can have a computational complexity of about $O(k^{2.4})$ for inverting a $k \times k$ matrix, the *added* complexity from the extra $(n + m - 2)$ operations is $O((n + m)k^{2.4})$ where k is the number of latent features in A and B .

3.6 Predicting Novel Interactions

In this section, we simulate some cases where an arbitrary drug or target is left out (i.e. its interactions are removed from the original dataset) to see if its interactions would be predicted successfully. We particularly focused on drugs and targets for which the similarity to the nearest neighbor (according to S_d and S_t , respectively) is low as they represent the tougher cases to predict interactions for; according to [120], nearest neighbor information has high influence on the prediction performance, so if the nearest known neighbor to a new drug or target is not so near, it may be difficult to accurately predict interactions for it.

From the IC dataset, we left out the target protein—*transient receptor potential cation channel, TRPV6* (KEGG id: hsa55503)—for which there are nine known interactions in *Y*. After *GRMF* was run on the modified dataset, all drugs were sorted in descending order of how likely they would interact with *TRPV6*; the top 20 predicted interactions for *TRPV6* are given in Table 3.6. The nine interactions of *TRPV6* were predicted successfully in the top 20. *RLS-WNN* and *CMF* were also able to predict these interactions in their top 20. As for *BLM-NII*, it was able to predict seven of the nine interactions, and the predicted interactions were not as highly ranked as in the other methods’ predictions.

Rank	Drug	Drug KEGG id
1	Verapamil	D02356
2	Gabapentin	D00332
3	Phenytoin	D00512
4	Nisoldipine	D00618
5	Nifedipine	D00437
6	Nitrendipine	D00629
7	Cinnarizine	D01295
8	Pimozide	D00560
9	Halothane	D00542
10	Diazoxide	D00294
11	Nimodipine	D00438
12	Verapamil hydrochloride	D00619
13	Terfenadine	D00521
14	Isradipine	D00349
15	Felodipine	D00319
16	Methoxyflurane	D00544
17	L-Proline	D00035
18	Amiloride hydrochloride	D00649
19	Mibefradil dihydrochloride	D05024
20	Penfluridol	D02630

Table 3.6: Predicted Drugs for *TRPV6*, IC Dataset

Known interaction are in **bold**.

The same procedure was performed for the drug—*Hexobarbital* (KEGG id: D01071)—by leaving it out of the E dataset before running *GRMF*; the top 20 predictions are given in Table 3.7. 19 out of the top 20 predicted interactions were known interactions in *Y*. A

similar prediction performance was displayed by *RLS-WNN*, while both *BLM-NII* and *CMF* were each able to predict only one known interaction in their top 20. Tables of predicted interactions for *TRPV6* and *Hexobarbital* using *BLM-NII*, *RLS-WNN* and *CMF* are provided as supplementary material at *GRMF*'s website.

Rank	Target	Target KEGG id
1	cytochrome P450 CYP19A1	hsa1588
2	cytochrome P450 CYP4Z1	hsa199974
3	cytochrome P450 CYP4X1	hsa260293
4	cytochrome P450 CYP4B1	hsa1580
5	cytochrome P450 CYP2J2	hsa1573
6	hydroxymethylglutaryl-CoA reductase HMGCR	hsa3156
7	cytochrome P450 CYP2S1	hsa29785
8	cytochrome P450 CYP2F1	hsa1572
9	cytochrome P450 CYP2B6	hsa1555
10	cytochrome P450 CYP1B1	hsa1545
11	cytochrome P450 CYP3A43	hsa64816
12	cytochrome P450 CYP2A6	hsa1548
13	cytochrome P450 CYP3A5	hsa1577
14	cytochrome P450 CYP3A7	hsa1551
15	cytochrome P450 CYP2C18	hsa1562
16	cytochrome P450 CYP2A13	hsa1553
17	cytochrome P450 CYP2E1	hsa1571
18	cytochrome P450 CYP17A1	hsa1586
19	cytochrome P450 CYP1A1	hsa1543
20	cytochrome P450 CYP2C19	hsa1557

Table 3.7: Predicted Targets for Hexobarbital, E Dataset

Known interaction are in **bold**.

We remind the reader that the above two cases (i.e. *TRPV6* and *Hexobarbital*) are considered difficult cases. Specifically, the similarity of *TRPV6* to its nearest neighboring target (according to S_t) is as low as 0.05, while the similarity of *Hexobarbital* to its nearest neighboring drug (according to S_d) is 0.35 which is also quite low. Note that the average drug similarity to its nearest neighbor according to the E dataset is 0.61. According to these cases, it was shown that *GRMF* performs reasonably well, which was also confirmed by other experiments that have been conducted (results added to *GRMF*'s website). In conclusion, *GRMF* is generally able to predict challenging interactions.

3.7 Interaction Prediction under S1

For the sake of completion, another set of cross validation experiments was performed based on a different scenario, S1, in which random drug-target pairings (rather than entire drugs or targets) are left out. Note that S1 corresponds to the case where a drug repositioning strategy is used to address a certain disease; i.e. we already know some targets of existing drugs and we wish to see whether these drugs also interact with other targets of interest (e.g. disease-associated genes). This is in contrast to, say, S2 which corresponds to the traditional drug development strategy where a novel compound (i.e. one whose interaction profile is initially unknown) is identified as the lead compound to be investigated and optimized. Results from averaging 5 repetitions of 10-fold CV are given in Table 3.8.

Methods	NR	GPCR	IC	E
BLM-NII	0.707 (0.006)	0.755 (0.007)	0.903 (0.002)	0.862 (0.002)
RLS-WNN	0.649 (0.027)	0.753 (0.003)	0.923 (0.002)	0.900 (0.002)
CMF	0.587 (0.015)	0.754 (0.003)	0.935 (0.002)	0.885 (0.002)
GRMF	0.651 (0.028)	0.698 (0.012)	0.885 (0.002)	0.852 (0.001)
WGRMF	0.655 (0.022)	0.738 (0.008)	0.932 (0.003)	0.892 (0.003)
WKNKN	0.613 (0.033)	0.660 (0.003)	0.838 (0.004)	0.847 (0.002)
WKNKN+BLM-NII	0.622 (0.023)	0.732 (0.004)	0.907 (0.002)	0.888 (0.001)
WKNKN+RLS-WNN	0.649 (0.027)	0.753 (0.003)	0.923 (0.002)	0.900 (0.002)
WKNKN+CMF	0.582 (0.015)	0.746 (0.005)	0.933 (0.002)	0.888 (0.001)
WKNKN+GRMF	0.613 (0.025)	0.718 (0.004)	0.909 (0.002)	0.882 (0.001)
WKNKN+WGRMF	0.590 (0.015)	0.739 (0.004)	0.923 (0.002)	0.878 (0.003)

Table 3.8: AUPR Results for Interaction Prediction Under S1

Best AUPR result in each column is **bold**. Standard deviations are given in (parentheses).

Firstly, we observe that the AUPR scores under S1 are better than those under S2 and S3, which is expected because S1 is the more informative case of the three CV scenarios – that is, for each left-out interaction, there are more neighboring interactions to infer an interaction from.

Secondly, while *WGRMF* was not the superior method in any of the datasets, it performed reasonably well, and its results were comparable to those of the best method for each dataset. This is in contrast to the S2 and S3 settings where *WGRMF* was the

superior method in most cases (as shown in the upper halves of Tables 3.2 and 3.3). Note that S1 does not leave out entire drug or target interaction profiles like S2 and S3, so there were less orphan drugs and targets being predicted for. *WGRMF* outperformed the other state-of-the-art methods when there were many orphan drugs (S2) or targets (S3). Under S1, however, the other methods were able to produce competitive results.

Thirdly, in many of the results, the AUPR decreased after using *WKNKN* as a preprocessing method. Originally, the intention behind *WKNKN* was to reinforce the learning process by adding artificial interactions when there is a considerable lack of neighboring information to use in inferring new interactions. However, under S1, there was already abundant neighbor information, so the generated artificial interactions (that are generally less reliable than the real interactions already in Y) actually hurt the predictions because they were not really needed in the first place.

Finally, we wish to mention that, in the new drug case (S2) and new target case (S3) from Tables 3.2 and 3.3, the manifold that was learned (using the graph regularization terms) allowed *GRMF* and *WGRMF* to compensate for the lack of sufficient neighbor information and produce better results than the competing methods. However, in the pair prediction case S1, the lack of neighbor information was not as serious, so the advantage from the learned manifold was less in this setting.

3.8 Conclusion

In this chapter, we presented two matrix factorization methods, *GRMF* and *WGRMF*, for drug-target interaction prediction. Both of them implicitly perform manifold learning via graph regularization. *WGRMF* differs from *GRMF* in that it additionally has a weight matrix W that prevents unknown instances (i.e. drug-target pairs for which interaction information is not available) from contributing to the final predictions. Experiments were conducted using two different types of cross validations, namely S2 (drug side CV) and S3 (target side CV), to compare our methods with three other competing state-of-the-art methods and specifically test how they perform in the *new* drug and target cases.

In most of the cases, the top results belonged to either *GRMF* or *WGRMF*, showing that manifold learning resulted in improvements in prediction performance. As such, we conclude that the manifold assumption that points lie on or near to a low-dimensional manifold is successful in improving DTI prediction performance.

Moreover, we developed a preprocessing step, *WKNKN*, which transforms all the 0's in the given drug-target matrix into interaction likelihood values. This is important as many 0's in the matrix actually correspond to unknown cases or missing values rather than confirmed non-interactions. It was observed that *WKNKN* improves results dramatically for all methods but with exceptions for those methods that contain a weight matrix W (i.e. *CMF* and *WGRMF*). Furthermore, when *WKNKN* is applied, results of *GRMF* became comparable to those of *WGRMF*, which is an added bonus because *GRMF* actually runs much faster due to its *matrix-wise* update rules.

Next, we performed a couple of experiments to investigate *GRMF*'s ability to predict novel interactions for new drugs and targets. Specifically, the entire profile of a known drug or target was left out, and then *GRMF* was run to see how many of its known interactions would get predicted. *GRMF*'s ability to predict new interactions in such cases was successfully confirmed in the experiments.

We then ended the chapter with a prediction performance comparison under the S1 cross validation setting to see how well our methods perform in the standard pair prediction case. While not the best, our methods still were able to produce results that are comparable to the best competing state-of-the-art methods.

As future work, other sources of information may be used to further improve prediction performance. For example, instead of using just one kernel (or similarity matrix) for drugs and one for targets, multiple kernels may be used for either drugs or targets in a multiple kernel learning algorithm [129]. We elaborate more on this in Chapter 6 wherein potential ways of further improving performance of drug-target interaction prediction are discussed.

Chapter 4

Class Imbalance-Aware

Ensemble Learning

Many existing methods treat drug-target interaction prediction as a binary classification problem where the *positive* class and the *negative* class consist of interacting and non-interacting drug-target pairs, respectively. Clearly, there exists a *between-class* (or inter-class) imbalance as the number of the non-interacting drug-target pairs (or majority negative class instances) far exceeds that of the interacting drug-target pairs (or minority positive class instances). This results in biasing the existing prediction methods towards classifying instances into the majority class to minimize the classification errors [130]. Unfortunately, minority class instances are the ones of interest to us. A common solution that was used in previous studies (e.g. [107]) is to perform random sampling from the majority class until the number of sampled majority class instances matches that of the minority class instances. While this considerably mitigates the bias problem, it inevitably leads to discarding of useful information (from the majority class) whose inclusion may lead to better predictions.

The other kind of class imbalance that also degrades prediction performance, but has not been receiving enough attention, is the *within-class* (or intra-class) imbalance which takes place when *rare* cases are present in the data [131]. In our case, there are multiple different types of drug-target interactions in the positive class, but some of them are represented by relatively fewer members than others and can be considered as less well-represented interaction groups (also known as *small concepts* or *small disjuncts*). If not processed well, they are a source of errors because predictions would be biased towards the well-represented interaction types in the data and ignore these specific small

concepts. It has been empirically shown that ignoring these small concepts leads to an overall increase in the error rate, which makes it an unfavorable strategy [132].

In this work, which we published in [133], we propose a simple method that addresses the imbalance problems stated above. Firstly, we provide a solution for the high imbalance ratio between the minority and majority classes while greatly decreasing the amount of information discarded from the majority class. Secondly, our method also deals with the within-class imbalance prevalent in the data by balancing the ratios between the different concepts inside the minority class. Particularly, we first perform clustering to detect homogenous groups where each group corresponds to one specific concept. Since the interactions within smaller groups are relatively easier to be incorrectly classified, we thus artificially enhance small groups via oversampling, which essentially helps our classification model focus on these small concepts to minimize classification errors.

4.1 Data

4.1.1 Drug-Target Interaction Data

We collected the interaction data used in this study from the DrugBank database [52] (version 4.3, released on 17 Nov. 2015). Some statistics regarding the collected interaction data are given below.

Drugs	Targets	Interactions
5,877	3,348	12,674

Table 4.1: Statistics of the interaction dataset used in this study

4.1.2 Data Representation

After having obtained the interaction data, we generated features for the drugs and targets respectively. Particularly, descriptors for drugs were calculated using the *Rcpi* [60] package. Examples of drug features include constitutional, topological and geometrical descriptors among other molecular properties. Note that *biotech* drugs have

been excluded from this study as *Rcpi* could only generate such features for *small-molecule* drugs. The statistics given in Table 4.1 reflect our final dataset after the removal of these biotech drugs.

Now, we describe how target features were obtained. Since it is generally assumed that the complete information of a target protein is encoded in its sequence [105], it may be intuitive to represent targets by their sequences. However, representing the targets this way is not suitable for machine learning algorithms because the length of the sequence varies from one protein to another. To deal with this issue, an alternative to using the raw protein sequences is to compute (from these same sequences) a number of different descriptors corresponding to various protein properties. The list of computed features is intended to be as comprehensive as possible so that it may, as much as possible, convey all the information available in the genomic sequences that they were computed from. Computing this list of features for each of the targets lets them be represented using fixed-length feature vectors that can be used as input to machine learning methods. In our work, the target features were computed from their genomic sequences with the help of the *PROFEAT* [69] web server.

The features that have been used to represent targets in this work are descriptors related to amino acid composition; dipeptide composition; autocorrelation; composition, transition and distribution; quasi-sequence-order; amphiphilic pseudo-amino acid composition and total amino acid properties. Note that a similar list of features was used previously in [107]. Subsets of these features have also been used in other previous studies concerning drug-target interaction prediction [105, 120].

After generating features for drugs and targets, there were features that had constant values among all drugs (or targets). Such features were removed as they would not contribute to the prediction of drug-target interactions. Furthermore, there were other features that had missing values for some of the drugs (or targets). For each of these features, the missing values were replaced by the mean of the feature over all drugs (or targets). In the end, 193 and 1290 features remained for drugs and targets, respectively.

Next, every drug-target pair is represented by feature vectors that are formed by concatenating the feature vectors of the corresponding drug and target involved. For example, a drug-target pair (d, t) is represented by the feature vector,

$$[d_1, d_2, \dots, d_{193}, t_1, t_2, \dots, t_{1290}]$$

where $[d_1, d_2, \dots, d_{193}]$ is the feature vector corresponding to drug d , and $[t_1, t_2, \dots, t_{1290}]$ is the feature vector corresponding to target t . Hereafter, we also refer to these drug-target pairs as *instances*. Finally, to avoid potential feature bias in the original feature values, all features were normalized to the range $[0,1]$ using min-max normalization before performing drug-target interaction prediction as follows

$$\forall i = 1 \dots 193, \quad d_i = \frac{d_i - \min(d_i)}{\max(d_i) - \min(d_i)} \quad (4.1)$$

$$\forall j = 1 \dots 1290, \quad t_j = \frac{t_j - \min(t_j)}{\max(t_j) - \min(t_j)} . \quad (4.2)$$

4.2 Methods

The proposed method was developed with an intention to deal with two key imbalance issues, namely the between-class imbalance and the within-class imbalance. Here, we describe in detail how each of these imbalance issues was handled. For notation, we use P to refer to the set of *positive* instances (i.e. the *known* experimentally verified drug-target interactions) and use N to refer to the remaining *negative* instances (consisting of all other drug-target pairs that do not occur in P).

Technically speaking, these remaining instances should be called *unlabeled* instances as they have not been experimentally verified to be true non-interactions. In fact, we believe that some of the instances in N are actually true drug-target interactions that have not been discovered yet. Nevertheless, to simplify the discussion, we refer to them as *negative* instances since we assume the proportion of non-interactions in N to be quite high.

4.2.1 Our Proposed Algorithm

The base of our method is an ensemble learning method where the prediction results of the different base learners are aggregated to produce the final prediction scores. For base learners, our ensemble method uses decision trees which are popularly used in ensemble methods (e.g. *Random Forest* [134]). Decision trees are known to be *unstable learners*, meaning that their prediction results are easily perturbed by modifying the training set, making them a good fit with ensemble methods which make use of the diversity in their base learners to improve prediction performance [135].

It is generally known that an ensemble learning method improves prediction performance over any of its constituent base learners only if they are uncorrelated. Intuitively, if the base learners of an ensemble method were identical, then there would be no gain in prediction performance at all. As such, adding diversity to the base learners is important.

One way of introducing diversity to the base learners that is used in our method is supplying each base learner with a different training set. Another way of adding diversity that we also employ here is *feature subsampling*; that is, for each of the base learners, we represent the instances using a different subset of the features. More precisely, for each base learner, we randomly select two thirds of the features to represent the instances. Feature subsampling is applicable here since the feature vectors are not overly sparse, and the features themselves share some correlation with one another; after all, the features have all been obtained from the same two sources: drugs’ chemical structures and targets’ protein sequences. Therefore, the risk of eliminating particularly important and unique (non-redundant) features is minimal.

Feature subsampling is a special case of what is more generally termed *multi-view learning* [136, 137]. Multi-view learning is about simultaneously utilizing multiple views of the same data (i.e. more alternate feature spaces) for training prediction models. The rationale for doing so is that, if done properly, training with more data views should lead

to a better trained model with improved prediction performance. In the case of DTI prediction, targets can be viewed from multiple perspectives including genomic sequences, gene ontology or known protein-protein interactions, while drugs can be represented by their chemical structures, ATC codes or side effects. Multiple kernel learning [129] and stacking-based ensembles [135] are two examples of how multiple data views can be simultaneously used to train prediction models. As for feature subsampling, it artificially generates multiple views of the data by randomly sampling a different feature subset for each base learner of an ensemble. Similar to bagging which injects diversity into an ensemble by randomly sampling instances from a training set, feature subsampling may also improve an ensemble’s prediction performance via the diversity it induces by randomly sampling features from a feature set for each base learner.

Figure 4.1 shows our pseudocode for the overall architecture of our proposed method where the specific steps for handling the two imbalance issues are discussed in the following subsections. Following is a summary of the method given an input consisting of a positive set P , a negative set N , and a user-defined parameter M indicating the number of base learners to be trained in the ensemble.

- M base learners (i.e. decision trees) are trained, and their prediction results are then aggregated by simple averaging to give the final prediction scores.
- For each decision tree, $tree_i$:
 - Randomly select a subset of the features, F_i
 - Obtain P_i by feature subsampling on P using F_i
 - Oversample P_i
 - Randomly sample N_i from N such that $|N_i| = |P_i|$
 - Remove instances of N_i from N
 - Perform feature subsampling on N_i using F_i
 - Train $tree_i$ using $P_i \cup N_i$ as the training set

```

Input:  $P$  = positive instances,
          $N$  = negative instances,
          $M$  = number of base learners.

Result: ensemble = trained ensemble.

Begin
  for  $i \leftarrow 1$  to  $M$  do
     $F_i$  = randomly selected feature subset
     $P_i = P(F_i)$  //feature subsampling

    //for within-class imbalance
     $P_i = \text{OVERSAMPLE}(P_i)$ 

    //for between-class imbalance
    repeat
      Randomly sample  $N_i \in N$ 
    until  $|N_i| = |P_i|$ ;
     $N = N - N_i$ 

     $N_i = N(F_i)$  //feature subsampling
     $tree_i$  = train decision tree using  $P_i \cup N_i$ 
  end for
  return  $ensemble = \frac{1}{M} \sum_{i=1}^M tree_i$ 

```

Figure 4.1: Pseudocode of proposed method

4.2.2 Within-Class Imbalance

We are now ready to explain the $\text{OVERSAMPLE}(P_i)$ in Figure 4.1. As mentioned at the beginning of this chapter, *within-class* imbalance refers to the presence of specific types of interactions in the positive set P that are underrepresented in the data as compared to other interaction types. Such cases are referred to as *small concepts*, and they are a source of errors because prediction algorithms are typically biased in that they favor the better represented interaction types in the data so as to achieve better generalization performance on unseen data [131].

To deal with this issue, we use the *K-means++* clustering method [138] to cluster the data into K homogenous clusters (K is a parameter) where each cluster corresponds to one specific concept. This results in interaction groups/clusters of different sizes. The assumption here is that the small clusters (i.e. those that contain few members) correspond to the rare concepts (or small disjuncts) that we are concerned about. Supposing that the size of the biggest cluster is *maxClusterSize*, all clusters are

resampled until their sizes are equal to $maxClusterSize$. This way, all concepts become represented by the same number of members and are consequently treated equally in training our classifier. Essentially, this is similar in spirit to the idea of boosting [139] where examples that are incorrectly classified have their weights increased so that classification methods will focus on the hard-to-classify examples to minimize the classification errors.

Oversampling of small disjuncts prevents them from being ignored as a result of the generality bias of machine learning algorithms, which may help improve prediction performance [140]. However, at the same time, it is widely assumed that oversampling can lead to overfitting since it duplicates existing instances without introducing new information [141, 142]. Fortunately, feature subsampling mitigates this issue of overfitting since the varying feature subsets of the base learners cause the clustering step to produce a different set of clusters for each learner. The different cluster sets of the base learners add to the diversity of the ensemble which helps counteract the overfitting.

Figure 4.2 shows the pseudocode for the oversampling procedure. P_i is first clustered into K clusters of different sizes. After determining the size of the biggest of these clusters, $maxClusterSize$, all clusters are re-sampled until their sizes are equal to $maxClusterSize$. The resampled clusters are then assigned to P_i before returning it to the main algorithm in Figure 4.1.

```

Input:  $P_i$  = positive instances.
Result:  $P_i$  = oversampled positive instances.
Begin
  Cluster  $P_i$  into  $K$  clusters:  $C_1 \dots C_K$ 
   $maxClusterSize = \max_k size(C_k)$ 
   $P_i = \emptyset$ 
  for  $j \leftarrow 1$  to  $K$  do
    repeat
      Re-sample  $C_j$ 
    until  $size(C_j) = maxClusterSize$ ;
     $P_i = P_i \cup members(C_j)$ 
  end for
  return  $P_i$ 

```

Figure 4.2: Oversampling procedure

The data used in this study was obtained from DrugBank [52], and since the data stored there is regularly curated by experts, we have high confidence in the interactions observed in our dataset. In other words, the interactions (or positive instances) are quite reliable and are expected to contain little to no noise. On the other hand, the negative instances are expected to contain noise since, as mentioned earlier, these negative instances are actually unlabeled instances that likely contain interactions that have not been discovered yet. Here, we only amplify the importance of small-concept data from the positive set (i.e. the set of known drug-target interactions). Since the positive instances being emphasized are highly reliable, the potential impact of noise on the prediction performance is minimal.

4.2.3 Between-Class Imbalance

Between-class imbalance refers to the bias in the prediction results towards the majority class, leading to errors where minority examples are classified into the majority class. We wanted to ensure that predictions are not biased towards the majority class while, at the same time, decrease the amount of useful majority class information being discarded. To that end, a different set of negative instances N_i is randomly sampled from N for each base learner i such that $|N_i| = |P_i|$. The 1:1 ratio of the sizes of P_i and N_i eliminates the bias of the prediction results towards the majority class. Moreover, whenever a set of negative instances N_i is formed for a base learner, its instances are excluded from consideration when we perform random sampling from N for future base learners. The different non-overlapping negative sets that are formed for the base learners lead to better coverage of the majority class in training the ensemble classifier.

Note that, to improve coverage of the majority class in training, the value of the parameter M needs to be increased where M is the number of base learners in the ensemble method, which also determines the number of the times that we want to draw instances from the negative set N . In general, with the increase of the value of M , more useful information from the majority class will be incorporated to build our final classification model.

4.3 Results and Discussion

In this section, we have performed comprehensive experiments in which we compare our proposed technique with 4 existing methods. Below, we first elaborate on our experimental settings. Next, we provide details of our cross validation experiments and comparison results. Finally, we focus on predicting interactions for new drugs and new targets, which is crucial for both novel drug design and drug repositioning tasks.

4.3.1 Experimental Settings

To evaluate our proposed method, we conducted an empirical comparison with 2 state-of-the-art methods and 2 baseline methods. Particularly, *Random Forest* and *SVM* are existing state-of-the-art methods that were both used in a recent work for predicting drug-target interactions [107]. Note that the parameters for these 2 methods were set to the default optimal values supplied in [107]. We also included two baseline methods, namely *Decision Tree* and *Nearest Neighbor*. For *Decision Tree*, we employed the *fitctree* built-in package in MATLAB and used the default parameter values as they were found to produce reasonable results. As for *Nearest Neighbor*, it produces a prediction score for every test instance a by computing its similarity to the nearest neighbor b from the minority class P (which contains the known interacting drug-target pairs) based on the following equations,

$$score_a = \max_b(sim(a, b)), \quad b \in P \quad (4.3)$$

$$sim(a, b) = \exp\left(-\frac{\|a - b\|^2}{|F|}\right) \quad (4.4)$$

where $|F|$ is the number of features.

For the above 4 competing methods, they all used P as the positive set, while the negative training data was sampled randomly from N until its size reached $|P|$. In contrast, our method oversampled P for each base learner i , giving P_i , and a negative set N_i was sampled from N for each base learner i such that $|N_i| = |P_i|$. Note that different base learners have used different negative sets in our proposed method. In addition, the parameters K and M for our method were set to 100 and 500, respectively, to generate sufficient homogenous clusters and leverage more negative data.

4.3.2 Cross Validation Experiments

To study the prediction performance of our proposed method, we performed a standard 5-fold cross validation and computed the AUC for each method (i.e. the area under the ROC curve). More precisely, for each of the methods being compared, 5 AUC scores were computed (one for each fold) and then averaged to give the final overall AUC score. Note that AUC is known to be insensitive to skewed class distributions [143]. Considering that the drug-target interaction dataset used in this study is highly imbalanced (we have much more negatives than positives), AUC score is thus a suitable metric for evaluation of the different computational methods.

Figure 4.3 shows the ROC curves for various methods. It is obvious that the ROC curve for our proposed method dominates those for the other methods, implying that it has a higher AUC score. Moreover, Table 4.2 shows the AUC scores for the different methods. Our proposed method achieves an AUC of 0.900 and performs significantly better than other existing methods.

Figure 4.3: Plot of ROC curves of the different methods. ROC curves for the different methods are plotted together, providing a visual comparison between their prediction performances.

Decision Tree	0.760 (0.004)
SVM	0.804 (0.004)
Nearest Neighbor	0.814 (0.003)
Random Forest	0.855 (0.006)
Proposed Method	0.900 (0.006)

Table 4.2: AUC Results of cross validation experiments.

Standard deviations are included between parentheses. Best AUC is indicated in **bold**.

As shown in Table 4.2, the second best method is *Random Forest*. Moreover, our method is similar to *Random Forest* in that they are both ensembles of decision trees with feature subsampling. Both our proposed method and *Random Forest* perform very well in drug-target interaction prediction, showing that ensemble methods are indeed superior

to achieve good prediction performance. However, our method differs from *Random Forest* in two aspects. Firstly, *Random Forest* performs bagging on a single sampled negative set for each base learner, while our method leverages multiple non-overlapping negative sets for different base learners. Secondly, our method also oversamples the positive set in a way that is intended to deal with the within-class imbalance, while *Random Forest* does not. Due to these 2 differences, our method achieved an AUC of 0.900, which is 4.5% higher than *Random Forest* with an AUC of 0.855. This supports our claim that dealing with class imbalance in the data is important for improving the prediction performance.

4.3.3 Predicting Interactions for New Drugs and Targets

A scenario that occurs in drug discovery is that we may have a target protein of interest for which no information on interacting drugs is available. This is typically a more challenging case than if we had information on drugs that the target protein is already known to interact with. A similar scenario that occurs frequently in practice is that we have new compounds (potential drugs) for which no interactions are known yet, and we want to determine candidate target proteins that they may interact with. When there is no interaction information on a drug or target, they are referred to as a *new drug* or a *new target*.

To test the ability of our method to correctly predict interactions in these challenging cases, we simulated the cases of new drugs and targets by leaving them out of our dataset, training with the rest of the data and then obtaining predictions for these new drugs and new targets. In our case studies, we ranked the predicted interactions and investigated the top 20 interactions. In particular, we investigated two drugs, *Aripiprazole* and *Theophylline*, and two targets, *Glutamate receptor ionotropic, kainate 2* and *Xylose isomerase*. Tables 4.3 and 4.4 show the top 20 predictions for these drugs and targets.

In our dataset, *Aripiprazole* and *Theophylline* are known to interact with 25 and 8 targets, respectively. Out of the top 20 predicted targets for *Aripiprazole*, 19 were correctly predicted as shown in Table 4.3. For *Theophylline*, all of its 8 interactions were highly ranked in its top 20 list.

<i>Aripiprazole</i>		<i>Theophylline</i>	
Rank	Target	Rank	Target
1	5-hydroxytryptamine receptor 2A	1	cAMP-specific 3',5'-cyclic phosphodiesterase 4A
2	Alpha-1B adrenergic receptor	2	Histone deacetylase 2
3	Muscarinic acetylcholine receptor M2	3	Adenosine receptor A2a
4	5-hydroxytryptamine receptor 2C	4	Adenosine receptor A1
5	D(1) dopamine receptor	5	cGMP-inhibited 3',5'-cyclic phosphodiesterase A
6	Alpha-2C adrenergic receptor	6	cAMP-specific 3',5'-cyclic phosphodiesterase 4B
7	Histamine H1 receptor	7	Adenosine receptor A2b
8	Muscarinic acetylcholine receptor M3	8	cGMP-specific 3',5'-cyclic phosphodiesterase
9	D(2) dopamine receptor	9	Adenosine receptor A3
10	Muscarinic acetylcholine receptor M1	10	Thymidylate synthase
11	5-hydroxytryptamine receptor 1B	11	Histone deacetylase 1
12	Delta-type opioid receptor	12	Cyclin-dependent kinase 2
13	D(4) dopamine receptor	13	Reverse transcriptase/RNaseH
14	D(3) dopamine receptor	14	Cap-specific mRNA (nucleoside-2'-O-)-methyltransferase
15	5-hydroxytryptamine receptor 1D	15	Multi-sensor signal transduction histidine kinase
16	Alpha-1 adrenergic receptor	16	Alpha-1 adrenergic receptor
17	Muscarinic acetylcholine receptor M5	17	Serine/threonine-protein kinase pim-1
18	Muscarinic acetylcholine receptor M4	18	Serine-protein kinase ATM
19	Alpha-2B adrenergic receptor	19	Proto-oncogene tyrosine-protein kinase Src
20	5-hydroxytryptamine receptor 1A	20	Phosphatidylinositol 4,5-bisphosphate 3-kinase catalytic subunit alpha isoform

Table 4.3: Top 20 targets predicted for *Aripiprazole* and *Theophylline*.

Targets in **bold** are the true known targets of the drugs

Moreover, *Glutamate receptor ionotropic, kainate 2* and *Xylose isomerase* have 20 and 7 interacting drugs in our dataset, respectively. Out of the top 20 predicted drugs for *Glutamate receptor ionotropic, kainate 2*, 17 were successfully predicted as shown in Table 4.4. For *Xylose isomerase*, all its 7 drugs were predicted in the top 20. These promising results show that our method is indeed reliable for predicting interactions in the cases of new drugs or targets.

Finally, we investigated the possibility that some of the unconfirmed interactions in Tables 4.3 and 4.4 might be true. For example, we observed that *Delta-type opioid receptor* is indeed a target for *Aripiprazole*, which was confirmed from the T3DB online database [144]. We have also confirmed, using the STITCH online database [54], that *Adenosine receptor A3* and *Histone deacetylase 1* are true targets of *Theophylline* as well. These findings suggest that the unconfirmed interactions in Tables 4.3 and 4.4 may be true interactions that have not been discovered yet.

<i>Glutamate receptor ionotropic, kainate 2</i>		<i>Xylose isomerase</i>	
Rank	Drug	Rank	Drug
1	Metharbital	1	D-Xylitol
2	Butabarbital	2	alpha-D-Xylopyranose
3	Pentobarbital	3	L-Xylopyranose
4	Thiopental	4	beta-D-Ribopyranose
5	Butethal	5	D-Sorbitol
6	Secobarbital	6	D-Xylulose
7	Talbutal	7	Vitamin C
8	Hexobarbital	8	2-Methylpentane-1,2,4-Triol
9	Barbital	9	Tris-Hydroxymethyl-Methyl-Ammonium
10	Amobarbital	10	(4r)-2-Methylpentane-2,4-Diol
11	Phenobarbital	11	Ethanol
12	Butalbital	12	Beta-D-Glucose
13	Aprobarbital	13	D-Allopyranose
14	Methylphenobarbital	14	2-Deoxy-Beta-D-Galactose
15	Primidone	15	Tris
16	Lysine Nz-Carboxylic Acid	16	3-O-Methylfructose in Linear Form
17	Domoic Acid	17	Dithioerythritol
18	Heptabarbital	18	(2s,3s)-1,4-Dimercaptobutane-2,3-Diol
19	Vitamin A	19	1,4-Dithiothreitol
20	Mephenytoin	20	Glycerol

Table 4.4: Top 20 drugs predicted for *Glutamate receptor ionotropic, kainate 2* and *Xylose isomerase*.

Drugs in **bold** are the true known drugs of the targets

4.4 Conclusion

We proposed a simple yet effective ensemble method for predicting drug-target interactions. This method includes techniques for dealing with two types of class imbalance in the data, namely between-class imbalance and within-class imbalance. In our experiments, our method has demonstrated significantly better prediction performance than that of the state-of-the-art methods via cross validation.

In addition, we simulated new drug and new target prediction cases to evaluate our method's performance under such challenging scenarios. Our experimental results show that our proposed method was able to highly rank true known interactions, indicating that it is reliable in predicting interactions for new compounds or previously untargeted proteins. This is particularly important in practice for both identifying new drugs and detecting new targets for drug repositioning.

Chapter 5

Dimensionality Reduction and Ensemble Learning

Ensemble learning techniques that aim to integrate multiple base classifiers/learners for robust and accurate predictions have been widely used to improve the classification performance. In Chapter 4, we developed an ensemble learning method for predicting drug-target interactions while dealing with a prevalent issue in the data, namely *class imbalance*.

However, we observed that the information for drugs and targets as well as the number of known drug-target interactions keep on increasing. Ensemble learning models, especially with a large number of base learners, would become inefficient for the prediction tasks. To address this issue, we propose to apply different types of dimensionality reduction techniques to remove those noisy, redundant, or irrelevant information so that it can effectively reduce the data size to a manageable level before we build our machine learning models.

In this work, which was published in [145], we propose a framework for drug-target interaction prediction that uses *ensemble learning* and *feature dimensionality reduction*. Firstly, we conducted feature subsampling to inject diversity into the classifier ensemble. Secondly, we investigated three different dimensionality reduction methods, namely *Singular Value Decomposition*, *Partial Least Squares* and *Laplacian Eigenmaps*, and applied them to the subsampled features (obtained by feature subsampling) to reduce the number of features. Finally, we trained homogeneous base learners with the reduced features and then derived the final predictions by aggregating their scores. In

general, any classification method can be applied as a base learner in our framework. In our work, we selected two efficient base learners, namely *Decision Tree* [135] and *Kernel Ridge Regression* [146], and presented two variants of ensemble models, denoted as *EnsemDT* and *EnsemKRR*, respectively.

5.1 Data

In this work, the main dataset used is from [133] which includes the drug-target interaction data as well as the features used to represent drugs and targets—it is also the dataset introduced in Chapter 4 of this thesis document. Statistics for this dataset are shown in Table 5.1 below. The features of the drugs and targets have been calculated using the *Rcpi* [60] package and the *PROFEAT* [69] web server, respectively.

Examples of drug features used include constitutional, topological and geometrical descriptors, while target features include amino acid composition, pseudo-amino acid composition and CTD (composition, transition, distribution) descriptors among others. In addition, the features have been properly normalized before they are used in all the machine learning methods in this work.

Drugs	Targets	Interactions
5,877	3,348	12,674

Table 5.1: Statistics of the main dataset used in this work

For notation, let $Y \in \mathbb{R}^{n \times m}$ be the drug-target interaction matrix having n drug rows and m target columns where $Y_{ij} = 1$ if drug d_i and target t_j interact and 0 otherwise. In addition, let $D \in \mathbb{R}^{n \times f_d}$ and $T \in \mathbb{R}^{m \times f_t}$ be the feature matrices for the drugs and targets, respectively, where f_d and f_t are the numbers of drug and target features, respectively.

5.2 Dimensionality Reduction

Dimensionality reduction projects our training data into a feature space with a lower dimension, and thus reduces the memory requirement and decreases the time complexity for feature-based methods. Next, we briefly introduce three different dimensionality reduction techniques to be investigated in this paper, namely *Singular Value Decomposition*, *Partial Least Squares* and *Laplacian Eigenmaps*. Note that we have used k below ($k \ll f_d$, $k \ll f_t$) as an adjustable parameter for controlling the dimensionality of the reduced feature space obtained from these techniques.

Before moving on, we give a brief mention of a special category of dimensionality reduction methods known as *feature selection* methods. These methods attempt to select a subset of relevant features that enhance the generalization (and, in turn, the prediction performance) of the prediction model being trained [147]. Unlike other dimensionality reduction methods that transform the features (i.e. any information that existed in the original features is lost), feature selection produces a subset of the full feature set containing features that were deemed most relevant to the prediction task at hand. Hence, this reduced feature set is interpretable and can be inspected for insights.

However, feature selection can be too computationally intensive. As an example, *forward feature selection* starts by passing over all the features to determine which feature gives the best predictions. Specifically, a prediction model is trained using each feature separately, and the model with the best performance determines which feature gets added to the feature subset being formed. Next, another pass over the remaining features is performed to see which of them improves predictions the most when added to the current feature subset and so on. This would result in the machine learning algorithm being run many times, with a theoretical upper bound of $f(f + 1)/2$ times where f is the number of features. The problem is further compounded if ensemble learning is used as the running time would dramatically increase with there being many base learners in the ensemble. As such, feature selection was not used for the dimensionality reduction of the data.

Since the improvement of the running time is a priority in this work, we opted to use *SVD*, *PLS* and *Laplacian Eigenmaps*. Each of these techniques *transforms* the features, which means that the information in the original features is lost. However, they also have some advantages. Compared with feature selection, such techniques execute relatively quickly (though some of them may not scale well to very large datasets). Moreover, these techniques extract patterns that may be useful. For example, *Laplacian Eigenmaps* learns the manifold in the data by explicitly transforming the data in a way that preserves the local geometries of the instances. Finally, some of these techniques can reduce the noise in the data as a side effect; e.g. *SVD* converts the original features into linearly uncorrelated ones via an orthogonal transformation and then retains the ones with the most variance (which have a higher signal-to-noise ratio than the rest).

5.2.1 Singular Value Decomposition (SVD)

In our work, we use *Truncated SVD* which is an economical variant of standard *SVD*. It takes a given matrix $G \in \mathbb{R}^{n \times u}$ and decomposes it into $U \in \mathbb{R}^{n \times k}$, $S \in \mathbb{R}^{k \times k}$ and $V \in \mathbb{R}^{u \times k}$ such that $G = USV^T$. S is a diagonal matrix containing the largest k eigenvalues of G . The reduced matrix can then be obtained as $\tilde{G} = US$. This procedure is applied to both D and T to obtain $\tilde{D} \in \mathbb{R}^{n \times k}$ and $\tilde{T} \in \mathbb{R}^{m \times k}$, respectively. For the remainder of this chapter, we refer to *Truncated SVD* as *SVD*.

5.2.2 Partial Least Squares (PLS)

PLS [148] is a supervised dimensionality reduction technique that takes a matrix of predictor variables $G \in \mathbb{R}^{n \times u}$ and a response matrix $H \in \mathbb{R}^{n \times v}$ as input. After first centering G and H (by subtracting off column means to get centered variables), *PLS* then builds the model

$$G = UJ^T + E \tag{5.1}$$

$$H = VQ^T + F \tag{5.2}$$

where $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{n \times k}$ are projections of G and H , respectively. Here, J and Q are loading matrices, and E and F are error terms that are assumed to be i.i.d. random normal variables. The model implicitly aims to maximize the covariance between U and V . We reduce the dimensionality of the drug matrix D by replacing G and H with D and Y in the above equation, respectively. The target matrix T can be processed in a similar manner by replacing G and H with T and Y^T , respectively.

5.2.3 Laplacian Eigenmaps (LapEig)

Laplacian Eigenmaps [86] is a manifold learning algorithm which performs non-linear dimensionality reduction. Given a feature matrix $X \in \mathbb{R}^{n \times u}$, Laplacian Eigenmaps constructs a t -nearest neighbor graph W where $W_{ij} = \exp(-\|x_i - x_j\|^2)$ if x_i and x_j are neighbors and 0 otherwise. It then finds a k -dimensional representation of X , $Z \in \mathbb{R}^{n \times k}$, by minimizing the following objective function,

$$Z = \underset{Z'}{\operatorname{argmin}} \sum_{ij} W_{ij} \|z'_i - z'_j\|^2 \quad (5.3)$$

where neighboring observations z'_i and z'_j are encouraged to be similar in the transformed space if their corresponding x_i and x_j are also similar to each other in the original feature space as indicated by W_{ij} .

5.3 Ensemble of Decision Trees with Different Negative Sets (EnsemDT)

Figure 5.1 presents an overview for our ensemble learning method *EnsemDT* that employs *Decision Tree* as the base learner. For each base learner, the minority class instances (or the positive instances) are included in the training set along with a randomly sampled set of negative instances from the majority class. The number of negative instances sampled per base learner is controlled by the parameter, *npRatio* (short for *negative-to-positive ratio*). For example, if *npRatio* = 5, then the number of

sampled negative instances is 5 times the number of positive instances in the training set. Note that in the ensemble method introduced in Chapter 4, for each base learner, negative examples were sampled until their size was equal to that of the positive training data (i.e. $npRatio = 1$). However, we decided to investigate what would happen if we were to increase the value of $npRatio$ beyond $npRatio = 1$.

After that, feature subsampling is performed (i.e. a different subset of features is randomly selected per base learner) to inject diversity into the ensemble – diversity is generally known to be beneficial to the prediction performance of ensembles [135]. This is then followed by dimensionality reduction to reduce the size of the data; that is, using a parameter r (where $0 < r \leq 1$), a subset of the features is randomly sampled for each base learner. More precisely, the number of randomly selected features per base learner is $r \times f$ where f is the total number of features in the full feature set. After training and predicting with M base learners, the results are aggregated by simple averaging.

```

Input:   $P$  = positive instance set,
           $N$  = negative instance set,
           $D$  = feature matrix for the drugs,
           $T$  = feature matrix for the targets,
           $npRatio$  = negative-to-positive ratio,
           $r$  = dimensionality reduction parameter,
           $M$  = number of base learners in the ensemble.

Result:  $ensemble$  = trained ensemble.

Begin
  for  $i \leftarrow 1$  to  $M$  do
    Randomly sample  $N_i \in N$  until  $|N_i| = npRatio \times |P|$ 
     $TrainingSet = P \cup N_i$ 
     $D_i$  = randomly selected feature subset (of size  $r \times f_d$ )
     $D_i = DimRed(D_i)$  // dimensionality reduction
     $T_i$  = randomly selected feature subset (of size  $r \times f_t$ )
     $T_i = DimRed(T_i)$  // dimensionality reduction
     $TrainingSet = TrainingSet(D_i, T_i)$  // form instance vectors
     $tree_i$  = train decision tree model using  $TrainingSet$ 
  end for
return  $ensemble = \frac{1}{M} \sum_{i=1}^M tree_i$ 

```

Figure 5.1: *EnsemDT – Decision Tree Ensemble*

Note that we derive the feature vectors for the instances (i.e. drug-target pairs) by concatenating the feature vectors of the involved drugs and targets. For example, an

instance (d_i, t_j) would have a feature vector $[D_{i,*}, T_{j,*}]$, where $D_{i,*}$ and $T_{j,*}$ are the row vectors in D and T that correspond to drug d_i and target t_j , respectively. Each base learner (i.e. decision tree) will take as input the instance vectors after having been processed by both feature subsampling and feature dimensionality reduction as shown in Figure 5.1.

5.4 Ensemble of Kernel Ridge Regression Learners (EnsemKRR)

EnsemKRR is an ensemble method for which the base learner is *RLS-avg*. Unlike *EnsemDT*, however, *EnsemKRR* does not have the first step of undersampling the majority class as this step is not required for the *RLS-avg* base learners.

Computational speed is important for a base learner in an ensemble because many base learners will be trained whose predictions will be aggregated to give the final prediction scores. *RLS-avg* is a similarity-based method that is known for having competitive performance. At the same time, it is a relatively fast algorithm; this is because *RLS-avg* is a BLM method (refer to Section 2.3.3 for details). As such, it was selected as the base learner for use in *EnsemKRR*. Next, we describe the *RLS-avg* model which was proposed in [87], followed by an overview of *EnsemKRR*.

5.4.1 The *RLS-avg* Base Learner

While *RLS-avg* was described earlier in Section 2.3.3 of Chapter 2, we describe it here in more detail since it is central to our proposed method, *EnsemKRR*. The first step in *RLS-avg* is to compute kernel matrices for the drugs and targets from D and T , respectively. For example, for a pair of drugs d_i and d_r , the drug similarity matrix K_d is computed as

$$K_d(d_i, d_r) = \exp\left(-\frac{\|D_{i,*} - D_{r,*}\|^2}{f_d}\right) \quad (5.4)$$

where $D_{i,*}$ and $D_{r,*}$ are the feature vectors for drugs d_i and d_r , respectively, and f_d is the number of drug features in D . The target similarity matrix K_t is similarly obtained. Furthermore, two more *Gaussian Interaction Profile* (*GIP*) kernels for drugs and targets are computed from the drug-target interaction matrix Y . Specifically, for a pair of drugs d_i and d_r , the drug GIP kernel is computed as

$$GIP_d(d_i, d_r) = \exp\left(-\frac{\|Y(d_i) - Y(d_r)\|^2}{\gamma}\right) \quad (5.5)$$

where $Y(d_i)$ and $Y(d_r)$ are the rows in the interaction matrix Y that correspond to the interaction profiles of drug d_i and d_r , respectively, and

$$\gamma = \frac{1}{n} \sum_{i=1}^n |Y(d_i)| \quad (5.6)$$

where n is the number of drugs. GIP_t is similarly obtained. GIP_d is then merged with K_d via linear combination as

$$\tilde{K}_d = \alpha K_d + (1 - \alpha) GIP_d \quad (5.7)$$

where α is an adjustable parameter. $\tilde{K}_t$ is similarly obtained. The final least-squares solution is then given by

$$\hat{Y} = \frac{1}{2} \left(\tilde{K}_d (\tilde{K}_d + \sigma I)^{-1} Y \right) + \frac{1}{2} \left(\tilde{K}_t (\tilde{K}_t + \sigma I)^{-1} Y^T \right)^T \quad (5.8)$$

where σ is a (Tikhonov) regularization parameter.

5.4.2 Extending the *RLS-avg* Base Learner using *WNN*

In the procedure of *RLS-avg* described above, each drug is used to train a classifier and each target is used to train a classifier as well, resulting in $f_d + f_t$ trained models (where f_d and f_t are the numbers of drugs and targets, respectively). However, when the interaction profile of a drug or target is empty (i.e. it is a new drug or target that has no known interaction in Y), this results in a useless model that does not help with

prediction. This motivated us to augment the original *RLS-avg* model with a preprocessing procedure called *Weighted Nearest Neighbor* (*WNN*) which was introduced in [91] to infer initial interaction profiles for new drugs or targets.

Note that in [91], *WNN* was used to augment a different method, *RLS-kron*. However, we opted to use *RLS-avg* as the base learner for *EnsemKRR* because it is relatively faster than *RLS-kron*, which makes it a better choice for a base learner in the ensemble. It also turns out that *RLS-avg* generally produces better prediction performance than *RLS-kron* as shown in Section 5.5.3. Since *RLS-avg* was never augmented with *WNN* previously, we do so here to assist each *RLS-avg* base learner with its prediction.

For each new drug d_i , *WNN* infers its interaction profile as

$$\tilde{Y}(d_i) = \sum_{u=1}^n w_u Y(d_u) \quad (5.9)$$

where d_1 to d_n are the drugs sorted in descending order based on their similarity to d_i , and $w_i = \eta^{i-1}$ where η is a decay term with $\eta \leq 1$. Similarly, every new target t_j has its interaction profile inferred by *WNN* as

$$\tilde{Y}(t_j) = \sum_{v=1}^m w_v Y(t_v) \quad (5.10)$$

As an intermediate interaction profile is inferred for a new drug d_i or a new target t_j , the decay term gives more weight to its nearer neighbors than the further neighbors. That is, the parameter η is used to control how much the contribution of the neighbors to the profile inferring decreases (or *decays*) as we go further down the neighbor list. The decaying contribution is desirable because the further a neighbor is, the less relevant it is to the object having its profile inferred. Otherwise, if there is no decay (i.e. $\eta = 1$), the contribution of the few nearest (and most relevant) neighbors will be dominated by the contribution of the numerous further (and less relevant) neighbors.

5.4.3 Overview of *EnsemKRR*

EnsemKRR uses the extended *RLS-avg* base learner described above. Many base learners are trained, and their prediction scores are averaged to give the final predictions.

Figure 5.2 presents an overview for our proposed *EnsemKRR* method.

```
Input:   $D$  = feature matrix for the drugs,  
          $T$  = feature matrix for the targets,  
          $r$  = dimensionality reduction parameter,  
          $M$  = number of base learners in the ensemble.  
  
Result:  $\hat{Y}$  = predicted interaction matrix.  
  
Begin  
  for  $i \leftarrow 1$  to  $M$  do  
     $D_i$  = randomly selected feature subset (of size  $r \times f_d$ )  
     $D_i = \text{DimRed}(D_i)$  // dimensionality reduction  
     $T_i$  = randomly selected feature subset (of size  $r \times f_t$ )  
     $T_i = \text{DimRed}(T_i)$  // dimensionality reduction  
     $\hat{Y}_i = \text{RLS-avg}(D_i, T_i)$   
  end for  
  return  $\hat{Y} = \frac{1}{M} \sum_{i=1}^M \hat{Y}_i$ 
```

Figure 5.2: *EnsemKRR* – Kernel Ridge Regression Ensemble

5.5 Experimental Results and Discussion

To demonstrate the prediction performance of *EnsemDT* and *EnsemKRR* as compared with other state-of-the-art methods, we performed a 5-fold cross validation experiment where prediction performance was measured in terms of AUC (Area Under the ROC Curve). More precisely, the interaction matrix Y was divided into 5 folds and these folds had turns being left out to act as a test set while the rest were treated as the training set. This procedure led to 5 AUC scores being computed which were then averaged to give the final score. In fact, AUC is insensitive to the high class imbalance present in the data [143], and thus it is widely used for measuring the performance of various prediction methods. Furthermore, we also conducted sensitivity analyses for the parameters of *EnsemDT* and *EnsemKRR*.

When dimensionality reduction is used, each fold separately undergoes a dimensionality reduction step *before* running the prediction method, unless the prediction method used is *EnsemDT* or *EnsemKRR*. In the case when either *EnsemDT* or *EnsemKRR* is used, dimensionality reduction is done *after* the feature subsampling step in these methods.

In this work, we make a distinction between two types of prediction methods. The first type is the *feature-based* methods which take their inputs in the form of feature vectors, representing a set of instances (i.e. drug-target pairs) along with their corresponding class labels (i.e. binary values indicating whether or not an interaction exists). *Decision Tree (DT)*, *Random Forest (RF)* [134] and *Support Vector Machines (SVM)* [149] are typical feature-based methods that build classification models based on the labeled feature vectors. In particular, *RF* and *SVM* are applied for predicting drug-target interactions in [107]. In these machine learning methods, known interactions represent positive instances and non-interactions denote negative instances. To be more precise, negative instances here include both non-interactions and unknown drug-target interactions (false negatives). Due to the huge number of possible negative instances, to balance the positive and negative training data, feature-based methods usually undersample the negative data prior to training and prediction; i.e. negative instances are randomly sampled to have the same size as the positive instances.

The second type of methods is the *similarity-based* methods which take their inputs explicitly in the form of two similarity matrices for the drugs and targets, respectively, along with an interaction matrix that indicates which pairs of drugs and targets interact. Looking back at the different categories of prediction methods in Chapter 2 of this thesis document, the category of similarity-based methods would contain all the categories except that of the feature-based methods. The similarity-based methods that we have chosen to include in the comparison are *Nearest Profile (NP)* and *Weighted Profile (WP)* [45], *Network-based Inference (NBI)* [92], *RLS-avg* and *RLS-kron* [87] (where *RLS* is short for *Regularized Least Squares*).

5.5.1 Comparison: Feature-based Methods

We compare between the different feature-based methods: *DT*, *RF*, *SVM* and *EnsemDT*. Table 5.2 shows the comparison among these feature-based methods in terms of AUC. Default parameter values were used for *RF* and *SVM* as defined in MATLAB's *TreeBagger* and *fitcsvm*, respectively. For *EnsemDT*, the parameter values were set as $M = 50$, $npRatio = 5$ and $r = 0.2$.

We first observe that both *RF* and *EnsemDT* have benefited from all dimensionality reduction techniques. This is quite satisfactory since dimensionality reduction also considerably decreased the running time for *RF* and *EnsemDT* (see Section 5.5.6). The highest AUC results for *EnsemDT* and *RF* were obtained using the *PLS* dimensionality reduction technique, with *SVD* and *Laplacian Eigenmaps* producing results comparable to those of *PLS*. As for *SVM*, it benefited from *SVD*, whereas it did not benefit from either *PLS* or *Laplacian Eigenmaps* as the prediction performance actually degraded. These results suggest that *PLS* and *Laplacian Eigenmaps* are not a good fit with *SVM*. In the end, *EnsemDT* is the best out of the four feature-based methods.

Methods	No Dim. Reduction	SVD	PLS	LapEig
DT	0.760	0.746	0.757	0.764
RF	0.855	0.876	0.880	0.874
SVM	0.804	0.818	0.643	0.642
EnsemDT	0.882	0.899	0.902	0.901

Table 5.2: AUC Results for Feature-based Methods

5.5.2 Varying Training Sets for EnsemDT's Base Learners

As shown in Table 5.2, *EnsemDT* produced better results than *RF*. In fact, there are two main differences between *EnsemDT* and *RF*. First, *EnsemDT* randomly samples a different negative set for each base learner. This is unlike *RF* which performs bagging on the same negative set. Second, *EnsemDT* includes all of the positive instances in every

base learner, while *RF* performs bagging on the positive instances along with the negative instances. That is, each decision tree in *RF* uses only a subset of the positive instances.

We further created different variants of *EnsemDT* by using different samples to train the base learners. The first variant randomly samples a negative set (of instances), merges it with the positive set to form a training set, and uses that same training set in all base learners of the ensemble. The second variant also uses the same training set, but additionally does bagging on it for each base learner. The third variant is similar to the second variant, but instead of bagging on this training set entirely, bagging is performed only on its negative instances while retaining all the positive instances. Finally, the fourth variant is the originally proposed variant of *EnsemDT* that is included in Table 5.2. Like the third variant, it uses all the positive instances in every base learner but, instead of bagging on the same negative instances, samples a different negative set for each base learner. The results of the different variants are given in Table 5.3. Note that no dimensionality reduction was employed in these variants as we wanted the results to be free from any influence that dimensionality reduction may have on them. This way, the conclusions drawn would be more reliable.

Methods	AUC
Same Training Set	0.874
Same Training Set + Bagging	0.867
Entire Positive Set + Bagging on Negatives only	0.876
Entire Positive Set + Different Negative sets	0.882

Table 5.3: AUC Results for Variants of *EnsemDT*

The first variant (i.e. same training set in all learners) provides us with a baseline against which to compare the other variants. The second variant (i.e. bagging on the same training set for all learners) produced a result that is lower than the first variant, which is a surprising observation since bagging is a mechanism that is typically employed in ensembles to improve the prediction performance [135]. However, we can identify the cause of this observation by looking at the result of the third variant (i.e. same training

set with bagging only on negative instances). By avoiding bagging on the positive instances, the result improved as compared to that of the second variant. As such, we conclude that the prediction performance is very sensitive to any positive instances being left out. That is, it appears to be a good strategy to include all positive instances in all base learners of *EnsemDT*. The last variant (i.e. all positive instances and different negative sets for the learners) gives the best result out of all the variants as it incorporates more data into the training of the ensemble.

5.5.3 Comparison: Similarity-based Methods

We then compare between the different similarity-based methods: *NP*, *WP*, *NBI*, *RLS-avg*, *RLS-kron* and *EnsemKRR*. Table 5.4 contains the AUC scores for various similarity-based methods. *NP*, *WP* and *NBI* do not have any parameters to tune. As for *RLS-avg* and *RLS-kron*, default values for α and σ are used as specified in [87], while η (of *WNN*) was set to 0.7 as it was determined in [121] to be a good default value. For *EnsemKRR*, the parameter values were set as $M = 50$ and $r = 0.2$.

Methods	No Dim. Reduction	SVD	PLS	LapEig
NP	0.679	0.679	0.670	0.626
WP	0.793	0.793	0.792	0.791
NBI	0.606	0.601	0.613	0.618
RLS-avg	0.925	0.912	0.915	0.909
RLS-kron	0.915	0.900	0.904	0.894
EnsemKRR	0.943	0.942	0.941	0.941

Table 5.4: AUC Results for Similarity-based Methods

Clearly, *RLS-avg* and *RLS-kron* performed much better than *NP*, *WP* and *NBI*. As an ensemble made of *RLS-avg* learners, *EnsemKRR* produced better results than all the other methods. By simply performing feature subsampling for each base learner, it was able to produce a satisfactory AUC result of 94.3%.

An important observation here is that similarity-based methods generally do not benefit from dimensionality reduction. This is attributed to the way the input data is presented to these methods, which is in the form of similarity or kernel matrices for the drugs and targets. Dimensionality reduction techniques retain the similarity between instances in the reduced feature space. Therefore, similarity matrices before and after dimensionality reduction are expected to be similar, leading to no improvement for these similarity-based methods after dimensionality reduction.

5.5.4 Effect of GIP Kernels and WNN

We noticed from the results in Table 5.4 that *NP*, *WP* and *NBI* are not doing as well as *RLS-avg* and *RLS-kron*. A potential factor that may be leading to this performance gap is that *NP*, *WP* and *NBI* use drug and target kernels K_d and K_t (obtained from Equation 5.4), whereas *RLS-avg* and *RLS-kron* additionally make use of GIP kernels (see Equation 5.5) to merge with K_d and K_t prior to prediction. As such, we studied the effect of GIP kernels on all similarity-based methods. Table 5.5 shows the results for the different methods with and without the GIP kernels (no dimensionality reduction applied). Note that when GIP kernels are used, *WNN* is also applied to Y beforehand.

Methods	No WNN, No GIP	WNN & GIP
NP	0.679	0.690
WP	0.793	0.878
NBI	0.606	0.870
RLS-avg	0.873	0.925
RLS-kron	0.786	0.915
EnsemKRR	0.866	0.943

Table 5.5: AUC Results for Interaction Prediction with and without WNN & GIP

It is obvious from the results of all the methods that GIP kernels and *WNN* have a large positive impact on the prediction performance. For example, *WP* and *NBI* now have results that are comparable to those of feature-based methods from Table 5.2. An interesting observation here is *RLS-avg* is better than *EnsemKRR* when neither *WNN*

nor GIP is used. It seems that feature subsampling did not help improve the prediction performance in this case. It is possible that the value of r (portion of randomly selected features per base learner) simply needs to be adjusted.

We further compared *RLS-avg* and *EnsemKRR* with and without the use of *WNN* or GIP kernels as shown in Table 5.6. Results in Table 5.6 show that prediction performance improves when *WNN* is applied prior to prediction whether or not GIP kernels are being computed. However, when the computation of the GIP kernels is attempted without having applied *WNN* beforehand, prediction performance is seriously affected. This is due to the presence of *empty* interaction profiles in Y ; that is, there are new drugs and targets that have no known interactions in Y . As mentioned in Section 5.4.1, creating GIP kernels involves computing similarities between interaction profiles of pairs of drugs or pairs of targets. The problem is that when some of these interaction profiles are empty, the resulting computed similarity can be very misleading because these empty profiles will be incorrectly considered identical, while in reality they are cases of missing information. The presence of such cases leads to improper GIP kernels that worsen the prediction performance.

Methods	No WNN, No GIP	WNN	GIP	WNN & GIP
RLS-avg	0.873	0.908	0.795	0.925
EnsemKRR	0.866	0.919	0.771	0.943

Table 5.6: AUC Results for Interaction Prediction with and without WNN and/or GIP

5.5.5 Sensitivity analysis

Here, we provide sensitivity analysis for two parameters M (number of base learners) and r (portion of features to randomly select per base learner) under the two proposed methods *EnsemDT* and *EnsemKRR*. Figures 5.3 and 5.4 contain the sensitivity analysis for M and r , respectively. Regarding M , the AUC for *EnsemDT* seems to improve quickly from $M = 1$ until $M = 50$, after which it still steadily improves but at a slower rate. As for *EnsemKRR*, it does not improve much beyond $M = 20$. Regarding r , *EnsemDT* seems to

be quite robust to the value of r till $r = 0.8$, after which the AUC decreases. As for *EnsemKRR*, it shows a steady decrease in AUC as the value of r increases, meaning that it performs best at $0.1 \leq r \leq 0.2$. As for *npRatio*, we found that the AUC of *EnsemDT* does not improve beyond *npRatio* = 5 (which we found to be the optimal value for *npRatio*).

Figure 5.3: Sensitivity Analysis for M

Figure 5.4: Sensitivity Analysis for r

5.5.6 Running times

The cross validation experiments have been performed on an Intel Xeon CPU (E5-1620 0 @ 3.60 GHz). For feature-based methods, dimensionality reduction helped to significantly decrease the running time. Table 5.7 shows running times for the feature-based methods.

Methods	No Dim. Reduction	SVD	PLS	LapEig
DT	5	1	4	4
RF	921	190	208	211
SVM	2636	84	113	117
EnsemDT	86	26	43	163

Table 5.7: Running Time for Various Feature-based Methods (mins)

As shown in Table 5.7, dimensionality reduction greatly decreases the running times. SVM in particular was suffering from the high dimensionality of the full feature set. Note that for *EnsemDT*, the dimensionality reduction is done for every base learner (on its subspace feature set), while it is done only once prior to prediction for the rest of the methods. This explains why the running time improvement for *EnsemDT* is not as strong as that for, say, *RF* when each is compared to its own running time without dimensionality reduction. For example, when *SVD* is used, *EnsemDT* gains a speedup of 3.31, while *RF* gains a higher speedup of 4.85.

An additional note that we wish to mention is that *RLS-avg* and, by extension, *EnsemKRR* run reasonably fast. Without dimensionality reduction, their running times are 2 and 68 min, respectively. We believe that this is a byproduct of its base learners being bipartite local models that separate the prediction problem into drug and target sides (i.e. the main prediction problem is divided into two smaller sub-problems) that are solved separately and then have their results merged to give the final result.

It is also worth noting that none of the similarity-based methods benefitted from dimensionality reduction in terms of running time. Specifically, if a similarity-based method is preceded by an additional dimensionality reduction step, its running time will

actually *increase*. This is due to the initial step that is common to all the similarity-based methods that are included in this study, namely that similarity matrices S_d and S_t are computed from the features that are then used by their respective algorithms. More precisely, regardless of whether the matrices are obtained from reduced or non-reduced features, the generated similarity matrices S_d and S_t will still have the same size ($S_d \in \mathbb{R}^{n \times n}$, $S_t \in \mathbb{R}^{m \times m}$) and hence the computational complexity of the similarity-based algorithm (and its running time) will remain unaffected.

Finally, we performed experiments to test how the parameter r (the portion of randomly sampled features) affects the running time. Figures 5.5 and 5.6 below show the results of these experiments for *EnsemDT* and *EnsemKRR*, respectively. To clarify, the number of features used per base learner is $r \times f$ where f is the total number of features in the full feature set. Note that $f = 1,483$ features (see section 4.1.2 for more details on the features). Furthermore, the remaining parameters are set as $M = 50$ and $npRatio = 5$ (which are the default values stated in Sections 5.5.1 and 5.5.3).

Figure 5.5: *EnsemDT* – Running time against r

In Figure 5.5, the running time for *EnsemDT* (without dimensionality reduction) appears to steadily increase with the value of r . When either *SVD* or *Laplacian Eigenmaps* is used for dimensionality reduction, the running time does not change much with the

value of r . However, the same cannot be said when *PLS* is used where the running time is observed to substantially increase with the value of r .

Note that, regardless of the number of input features, the dimensionality reduction step produces the same fixed number of reduced features for use in training the base learners. This means that, when dimensionality reduction is performed, the time taken to train the base learners is constant (and is not affected by r), and any observed increase in running time against r is due to the preceding dimensionality reduction step alone.

That said, *EnsemDT+PLS* displayed a higher rate of increase in running time than either of *EnsemDT+SVD* or *EnsemDT+LapEig*. Moreover, when the value of r is low, the running time for *EnsemDT+PLS* is close to that of *EnsemDT+SVD*. Conversely, when the value of r is high, the running time for *EnsemDT+PLS* approaches that of *EnsemDT+LapEig*. In fact, at $r = 1$ (i.e. no feature subsampling, and all the 1,483 features are used), the running time of *EnsemDT+PLS* exceeds that of *EnsemDT+LapEig*. In conclusion, *PLS* might not be suitable as a dimensionality reduction technique if the dataset is of much higher dimensionality than the one used in this work.

Figure 5.6: *EnsemKRR* – Running time against r

In Figure 5.6, as r increases, there is a modest increase in the running time of *EnsemKRR* (without dimensionality reduction). This modest increase is from the step where the

similarity matrices S_d and S_t are generated from the features; as the number of features (or r) increases, the time taken to generate similarity matrices from said features also increases. However, when dimensionality reduction is used, the time taken by the generation of these matrices is fixed since the number of reduced features received from the dimensionality reduction is also fixed.

As can be seen from Figure 5.6, the use of dimensionality reduction does not improve the running time for *EnsemKRR* (unlike *EnsemDT* whose running time generally benefitted from dimensionality reduction). *EnsemKRR+SVD* displayed running times that are very similar to those of *EnsemKRR* when dimensionality reduction is not used; that is, in *EnsemKRR+SVD*, whatever time was gained in the similarity matrix generation step (due to the reduced dimensionality) was wasted by the dimensionality reduction step itself. Moreover, the time taken by dimensionality reduction via *PLS* increases with the increase in r , whereas the time taken by *SVD* and *Laplacian Eigenmaps* is roughly constant (same as observed from Figure 5.5). Finally, although the running time for *EnsemKRR+LapEig* does not change much with the increase in r , its running time is already much worse than base *EnsemKRR* (even at $r = 0.1$).

5.5.7 Experiments with Another Dataset

Here, we present experimental results for running the prediction methods covered in this work on another dataset that was presented in [150]. The motivation for doing so is that the features used for representing the drugs and targets are different in nature from the ones in the main dataset, and we were interested how the different methods – especially our proposed ones – would perform on such a dataset. Statistics for this dataset are shown in Table 5.8 below. Drugs in this dataset are represented as PubChem fingerprints (i.e. binary vectors where each element indicates the presence or absence of one of 881 known chemical substructures). On the other hand, targets are represented as fingerprints that indicate the presence or absence of 876 different protein domains that are obtained from the Pfam database [151].

Drugs	Targets	Interactions
1,862	1,554	4,809

Table 5.8: Statistics of Dataset 2

The results of this experiment are given in Table 5.9. Before commenting on the results, note that in this experiment, the value of the r parameter (i.e. the portion of randomly selected features per base learner) was altered to optimize the prediction performance for *EnsemDT* on this dataset; the new value is $r = 0.9$ (i.e. feature subsampling is almost turned off). There is a characteristic in this dataset that forced this modification of the r parameter, namely the very sparse nature of the domain fingerprints representing the targets. Indeed, the average number of domains per target is 1.51 (out of 876 domains). Due to that sparse nature of the target feature vectors, feature subsampling was decreased to avoid the loss of what little information there is available for each target.

Methods	No Dim. Reduction	SVD	PLS	LapEig
DT	0.808	0.781	0.801	0.791
RF	0.762	0.891	0.891	0.893
SVM	0.733	0.733	0.697	0.627
EnsemDT	0.886	0.914	0.898	0.914
NP	0.700	0.638	0.682	0.628
WP	0.822	0.868	0.818	0.817
NBI	0.582	0.579	0.687	0.632
RLS-avg	0.928	0.899	0.918	0.916
RLS-kron	0.909	0.873	0.913	0.874
EnsemKRR	0.933	0.931	0.930	0.930

Table 5.9: AUC Results using Dataset 2

Interestingly, in contrast to *EnsemDT*, the optimal value of r when using *EnsemKRR* on this dataset is $r = 0.1$. In *RLS-avg*, the base learner used in *EnsemKRR*, the similarity matrices for the drugs and targets are first generated from their feature vectors. As a result of the target feature vectors being very sparse, the target similarity matrix is not

expected to change much with feature subsampling applied. As per Equation 5.4, all pairwise similarities are expected to be close to 1 because all pairs of targets will have most of the protein domains *commonly absent* (with or without feature subsampling). Therefore, the optimal value of r is not affected by the target side of *RLS-avg*. As for the drug similarity matrix, doing feature subsampling on the drug features beforehand would help inject diversity into the ensemble, thus improving prediction performance. This is why the optimal value is as low as $r = 0.1$ for *EnsemKRR* which is similar to the optimal value obtained for r ($r = 0.2$) on the original dataset. This is in contrast to *EnsemDT* where r is greatly affected by the sparsity of the target vectors, which led the optimal value of r to be $r = 0.9$ (as opposed to $r = 0.2$ on the original dataset); i.e. *EnsemDT* is able to make better use of the information in the sparse target vectors, which is achievable via a high value for r .

After having adjusted the value of r for optimal performance and looking at the results in Table 5.9, the conclusions are almost the same as those drawn from the results in Tables 5.2 and 5.4. However, there are two main differences.

The first difference is that *RF* produces a lower AUC than *DT* when dimensionality reduction is not performed (the first column of Table 5.9). This can be explained by the fact that the target feature vectors are too sparse and, as such, the prediction performance is hurt by the feature subsampling that is available in *RF* by default. However, this phenomenon does not happen when dimensionality reduction is used because, in case of *RF*, dimensionality reduction is done before running the *RF* algorithm on the reduced (and, thus, non-sparse) features.

The other difference is that *PLS* is no longer the superior dimensionality reduction technique for use with *EnsemDT*. *PLS* is generally suited for cases where there is multicollinearity in the features [152], which is not the case in the sparse domain fingerprints of the targets.

In the end, we conclude by saying that the conclusions that were drawn earlier in this chapter are generalizable to other datasets. The differences in results that were seen on Dataset 2 (as opposed to Dataset 1) are probably consequences of a characteristic that is unique to it, namely the excessive sparseness of the target feature vectors.

5.6 Extension for Feature-based Methods

So far, we presented the work that has been published in [145]. In this section, we extend the work presented in this chapter so far by modifying the procedure by which the negative training data is sampled from the negative set. We provide motivation and results of this modification below.

In Chapters 4 and 5, we addressed issues that are prevalent in feature-based methods, namely class imbalance and high space/time complexity that comes with high data dimensionality. While our contributions did improve prediction performance for feature-based methods, we found that they generally did not perform as well as some of the similarity-based methods. This is especially surprising for *EnsemDT*, a feature-based ensemble method, which paled in comparison to the similarity-based ensemble method, *EnsemKRR*. Since the difference in prediction performance was not small ($\sim 4\%$ AUC), there was likely another factor at play that is degrading the performance of *EnsemDT*.

In [50], the authors discuss factors affecting prediction performance of *pair-input methods*. Pair-input methods are computational prediction methods that operate on pairs of objects by considering features of each. Examples of pair-input methods include methods that perform prediction of protein-protein interactions, drug-target interactions, drug-disease associations, etc. In the case of drug-target interaction prediction, the *objects* are the drugs and targets, and the object *pairs* each consists of one drug and one target (i.e. drug-target pairs).

The authors conducted experiments to test prediction performance on three cases of test pairs:

- i. C_1 where both objects of the test pair appear in the training data
- ii. C_2 where only one of the test pair's objects appears in the training data
- iii. C_3 where neither of the test pair's objects appears in the training data

According to the experimental results, it was found that predictions for test pairs are more accurate when their involved objects appear in the training data. In contrast, it is more difficult to accurately classify test pairs having objects that do not appear in the training data at all. More precisely, prediction performance for test pairs in C_1 was better than that of test pairs in C_2 which, in turn, was better than that of test pairs belonging to C_3 . This indicates the importance of having the training data include, as best as possible, the objects that appear in the test set.

Moreover, the authors conducted experiments to investigate a concept called *differential representation bias* which refers to how much each object is represented in the positive class (interactions) as opposed to the negative class (non-interactions). Generally speaking, differential representation bias was found to have a positive impact on the prediction performance in the case of pair-input methods. For example, assuming the training data accurately represents the general population, if an object appears (or is *represented*) in the positive training data more than in the negative training data, then the prediction method is more likely to correctly predict test pairs involving this object as positive, which is usually true [50]. Conversely, if the negative data contains more appearances of an object than the positive data, then the prediction method will likely predict pairs involving this object as negative.

That said, since drug-target interaction data has substantially more non-interactions than interactions, having more negative examples than positive examples in the training data would hypothetically lead to better prediction performance as that would preserve the differential bias that is prevalent in the full data. In the case of drug-target interaction prediction, there is a severe between-class imbalance that we addressed in the previous chapters. In Chapter 5, we found the optimal value of *npRatio* (i.e. the negative-to-

positive ratio in the training set) to be $npRatio = 5$. That is, having $npRatio = 5$ mitigates the between-class imbalance in the data while, at the same time, retaining some of the differential representation bias (from the full dataset) that leads to better prediction performance. This supports the claim that differential representation bias is beneficial for prediction performance.

However, there is an extreme case of differential representation bias that occurs frequently in our data; *the training data contains drugs and targets that exist only in the negative class*. We hypothesize that the prediction model trained by these data will always tend to predict test pairs involving such drugs and targets as negative regardless of the test pairs' true labels. This is due to the bias in prediction (as negative) induced by how these drugs and targets are differentially represented in the training data. As such, to avoid this issue, we propose a modified procedure for sampling negative training data from the negative set, thus improving the prediction performance considerably.

5.6.1 Modification to the Sampling Procedure

Many of the drugs and targets in the dataset having only a single known interaction, and when such interactions get left out (in cross validation experiments), it renders their involved drugs or targets unavailable in the positive training data. In other words, there will always be drugs and targets that are covered in the negative training data but *not* covered in the positive training data. This skews the representation bias heavily in the negative direction for these drugs and targets. Therefore, any interactions in the test set involving these drugs and targets (that are missing from the positive set) will likely not be predicted correctly. As such, we propose the following modification in the procedure of sampling negative training data from the negative set:

When sampling negative examples, ignore those that involve drugs or targets that do not appear in the positive training data.

By doing so, we can guarantee that all drugs and targets that are in the negative training data are also in the positive training data and vice versa. We expect this to rectify any bias of predicting as negative any test pairs with objects that do not appear in the positive training data. Furthermore, with the above modification, predictions for any test pair involving a drug and/or a target missing from the training data will depend on (the representation bias of) their neighboring drugs and targets that exist in the training data.

From now on, we will be referring to the above modification as the *sampling modification*.

5.6.2 Experimental Design

We performed experiments to compare between the performance of feature-based methods before and after the sampling modification. Specifically, we conduct 5-fold cross validation (CV) experiments under the following settings:

- S1: where random drug-target pairs are left out for use as the test set
- S2: where entire drug interaction profiles are left out for use as the test set
- S3: where entire target interaction profiles are left out for use as the test set

Note that we also consider an additional setting, S4 (also known as 3×3-fold CV) which was suggested in [128], where drugs and targets in the test set do not appear in the training set and vice versa. More precisely, the drugs in the dataset are divided into 3 partitions, and the targets are divided into 3 partitions as well. Obtaining all possible combinations of the drug and target partitions gives us $3 \times 3 = 9$ different folds. When each of these folds is left out as the test set, any drugs and targets in the test set are *not* to be included in the training set. Out of the CV settings, this is the most challenging one. An illustration of the S4 cross validation setting is given in Figure 5.5.

Figure 5.5: Illustration of the S4 cross validation setting

(Yellow and red partitions denote the test and training sets, respectively. Gray partitions are ignored)

5.6.3 Prediction Results for Feature-based Methods

Here, we present the results for all the CV settings. Experiments were run using the main dataset mentioned in Section 5.1. Table 5.10 shows the results for the feature-based methods before and after the sampling modification. Note that two variants of *EnsemDT* are included in the comparison, one with 50 base learners (i.e. decision trees) and one with 500 base learners. To simplify discussion, we do not perform any of the dimensionality reduction techniques mentioned in the previous chapter. By looking at the results in Table 5.10, it is easy to see that the sampling modification is successful because, under all CV settings, all results consistently improved after applying it (with the single exception of *RF* under S4).

As expected, AUC scores under S1 are higher than those under S2 and S3 which are, in turn, higher than those under S4. Surprisingly, however, scores under S3 are much lower than those under S2. After some investigation, we found that the average number of known interactions per target (3.79) is higher than that per drug (2.16). Under S2 and S3, when a left-out object (drug or target) is of a larger degree, it is more difficult to rank all of its interactions highly. This is especially true since it will be depending on training pairs involving *neighboring* objects (rather than on training pairs involving the left-out object itself) to perform accurate predictions. Thus, left-out objects with a larger degree (targets under S3) would have a lower AUC score overall.

Methods	S1		S2		S3		S4	
	Before	After	Before	After	Before	After	Before	After
DT	0.760	0.767	0.747	0.754	0.564	0.593	0.552	0.558
RF	0.855	0.883	0.842	0.872	0.659	0.665	0.639	0.638
SVM	0.804	0.849	0.800	0.844	0.676	0.679	0.640	0.648
EnsemDT (50 learners)	0.882	0.906	0.856	0.886	0.684	0.689	0.633	0.642
EnsemDT (500 learners)	0.886	0.917	0.861	0.897	0.716	0.726	0.665	0.673

Table 5.10: AUC Results for Feature-based Methods before/after the Sampling Modification

(Results after the sampling modification are in **bold**)

Another observation from the results showing the benefit of the suggested sampling modification is that, after the sampling modification, the increase in the number of base learners in *EnsemDT* from 50 to 500 learners led to a more pronounced improvement in the prediction performance.

In the following section, we use the version of *EnsemDT* that uses the sampling modification and has 500 base learners. It will be referred to as *EnsemDT+*.

5.6.4 Comparison with EnsemKRR

Results for *EnsemKRR* under all CV settings are shown in Table 5.11 as compared to the *EnsemDT+*. While *EnsemKRR* does not outperform *EnsemDT+* under S2 and S4, it still does fairly better than *EnsemDT* under S1 and S3, which means that *EnsemKRR* is still the better method overall.

Methods	S1	S2	S3	S4
EnsemDT+	0.917	0.897	0.726	0.673
EnsemKRR	0.943	0.898	0.816	0.673

Table 5.11: AUC Results Comparing between *EnsemDT+* and *EnsemKRR*

Firstly, we look at the big difference in prediction performance between *EnsemDT+* and *EnsemKRR* under S1. Note that *EnsemDT+*, for computational efficiency as well as class imbalance issues, performs sampling on the negative set for each base learner, which means that every base learner is trained with only a subset of the entire negative data.

This is in contrast with *EnsemKRR* where the entire interaction matrix Y is used for prediction. However, we believe that the main reason for *EnsemKRR*'s competitive performance is its use of the *WNN* preprocessing procedure as well as the incorporation of network similarity via GIP kernels as illustrated in Table 5.5 (with *WNN* and GIP, the AUC score of *EnsemKRR* increased from 0.866 to 0.943). Otherwise, *EnsemDT+* would have been the superior method under S1.

Ignoring the results under S3 for now, *EnsemDT+* managed to achieve AUC scores that match those of *EnsemKRR* under S2 and S4. This is interesting as it confirms the correctness of our hypothesis that led to the suggestion of the sampling modification; when basic random undersampling of the majority (negative) class is used, some objects (drugs or targets) were present only in the negative training data and not in the positive data, and we hypothesized that this causes test instances containing said objects to be too biased towards being predicted as negative, thus degrading the prediction performance. The sampling modification was proposed to specifically mitigate this bias. According to the results in Tables 5.10 and 5.11, addressing this bias has generally improved the prediction results, which confirms the stated hypothesis. At the same time, however, the result of *EnsemDT+* under S1 was well behind that of *EnsemKRR*. We can only assume that *EnsemKRR* is better able to make use of the differential representation bias in the training data under the S1 setting, thus leading to its superior performance; as mentioned in the previous paragraph, it does not need to sample a subset of the negative data for each base learner like *EnsemDT+*.

Getting back to the results under S3, it is surprising to see that the AUC score for *EnsemDT+* is very far behind that of *EnsemKRR*, especially since its AUC scores matched those of *EnsemKRR* under both S2 and S4. We believe that it is related to the relatively higher degree of targets in the data as opposed to those of the drugs; in our dataset, drugs have degrees that range from 1 to 25, while there are targets with degrees up to 132. To explain, suppose we have a target with a very high degree (i.e. it appears many times in the positive training data). When sampling negative examples randomly for the base

learners, the negative examples containing that high-degree target will likely be much fewer in number than the positive examples containing that same target. This is a situation that is *artificially* caused by the nature of sampling; we remind the reader that *EnsemKRR*, on the other hand, does not perform sampling on the negative data and uses the entire data.

5.7 Final Remarks and Conclusion

Based on the experiments conducted earlier in this chapter, *EnsemDT* was the superior method among the feature-based methods, while *EnsemKRR* was the superior method among the similarity-based methods. However, a simple comparison between feature-based methods and similarity-based methods shows that similarity-based methods are generally hold more promise. For example, *WP* and *NBI* (after inclusion of GIP kernels) produced results that easily exceed that of *SVM* and that is comparable to that of *RF*. In addition, Stand-alone *RLS-avg* (augmented with *WNN*) produced results that are superior to those of the best feature-based method, *EnsemDT*. After using *RLS-avg* as a base learner in an ensemble, its results improved even further.

We believe the good performance of *RLS-avg* is due to three reasons. The first is that it is a bipartite local model, which relieves it from having to figure out a suitable way to merge the drug and target information in order to form feature vectors for the instances (as opposed to concatenating drug and target feature vectors to form instances in feature-based methods). Secondly, it makes good use of an extra source of information, the network similarity derived from the interaction matrix Y (in the form of GIP kernels), by merging it with the drug and target similarity matrices that were derived from the initial drug and target features to assist with prediction. Thirdly, there was the subtle addition that we added to *RLS-avg* that led to the satisfactory prediction performance that was reached by *EnsemKRR* in our experiments; that is, the applying of the *WNN* procedure [91] to the interaction matrix Y prior to prediction. Furthermore, it was found to be quite

important when considering to make use of generated GIP kernels (i.e. network similarity matrices) from the data.

As for dimensionality reduction, only the feature-based methods seemed to benefit from it. While the main benefit was the reduction of the running time (which was especially evident in *SVM*), the prediction performance of the feature-based methods (with some exceptions) actually improved. However, for similarity-based methods, no significant improvements in prediction performance were observed, which we concluded is due to the nature of how the training data is received as input by these methods, namely that the drug and target features are converted into similarity matrices for them to operate on. As for space complexity, there is no difference in the input given to the similarity-based methods; the similarity matrices computed to be given as input would still be in the same sizes regardless of whether dimensionality reduction is employed or not.

Experiments of *EnsemDT* have also shown an interesting observation. That is, performing bagging on the positive instances was found to degrade the prediction performance, which suggests that leaving out some of the positive instances per base learner is not a good idea, and all the positive examples should be fully leveraged for building the classification model. In addition, we increased the size of the negative training data by setting $npRatio = 5$ (as opposed to the method in Chapter 4 where $npRatio$ was set to 1). This has resulted in a significant improvement, which means that incorporating more negative data per base learner is beneficial.

At the time, we inferred that the some of the *unlabeled* instances that get sampled as negative instances are actually positive and that these incorrect training data are what is causing the prediction performance of feature-based methods (including *EnsemDT*) to fall short of their similarity-based counterparts. However, we discovered in later investigations that a concept called *differential representation bias* has a significant impact on the prediction performance of feature-based methods in particular.

Based on our understanding of the concept of differential representation bias, we presented a modification to the sampling procedure of the negative set for each base learner in *EnsemDT*. The modification was successful in significantly enhancing prediction performance of the feature-based methods in general (and *EnsemDT* in particular), and consequently decreasing the performance gap between feature-based and similarity-based methods for predicting drug-target interactions.

Chapter 6

Conclusion and Outlook

In this thesis, we aimed to enhance the prediction performance beyond what has been reached by the state-of-the-art methods that are used for predicting drug-target interactions on a large scale. In particular, we were interested in chemogenomic methods that make use of information from both the drug and target sides simultaneously for prediction. Furthermore, the drugs and targets that we were concerned with were small molecules and proteins, respectively. To that goal, we surveyed the literature in Chapter 2 to determine outstanding challenges that have not yet been resolved or handled directly so that we may work on them to improve prediction performance.

In Chapter 3, we presented a preprocessing method, *WKNKN*, that helps improve predictions for the specific cases of *new* drugs and *new* targets for which no prior information on their interaction profiles is known. In addition, to improve the prediction performance further, we sought to utilize the manifold assumption that data usually lie on or nearby a non-linear manifold of lower dimensionality. Thus, we developed two variants of a matrix factorization method, *GRMF* and *WGRMF*, that incorporate graph regularization terms for implicitly learning the manifold, which indeed led to performance improvements.

In Chapter 4, we presented an ensemble method that deals with the class imbalance prevalent in the data in its two forms, *between-class* imbalance and *within-class* imbalance. For the between-class imbalance, we presented a modification to how negative training data is sampled from the entire negative set such that more of the negative set is incorporated into the training of the base learners of our ensemble method; this was done by sampling different negative data for each base learner (as opposed to sampling negative training data once and bagging on it for each base learner

as in *Random Forest*). As for the within-class imbalance, so-called *small concepts* were detected in the positive set which were then reinforced by oversampling to mitigate the errors caused by inductive biases of algorithms towards the larger and more well-represented concepts.

In Chapter 5, we utilized dimensionality reduction to improve the computational efficiency of the prediction algorithms in cases of high-dimensional data. While it did improve the running times of feature-based methods in particular, improvements in the prediction performance were also observed. Ensemble learning was also utilized to enhance the prediction performance that was obtained, and two variants of ensemble methods were proposed, one with a feature-based method (*Decision Tree*) as a base learner and the other with a similarity-based one (*Kernel Ridge Regression* or *RLS-avg*). In general, similarity-based methods were found to be more promising than feature-based methods for prediction of interactions. However, there was a phenomenon affecting the overall performance of the feature-based methods which was later addressed.

Near the end of Chapter 5, we investigated the impact of *differential representation bias* on prediction results of feature-based methods, especially in light of the class imbalance that is prevalent in the data. We suggested a modification to the sampling procedure from the negative set that would ensure that objects (drug and targets) that appear in the positive training data would also appear in the negative training data and vice versa. This modification, though a simple one, resulted in significant and consistent improvements over all feature-based methods under all cross validation settings (with the single exception of *Random Forest* under S4). In addition, as a result of this modification, *EnsemDT* (our *feature-based* ensemble method) was finally able to produce results that are comparable to those of *EnsemKRR* (our *similarity-based* ensemble method) which is the top competing method for drug-target interaction prediction. This concludes our work on drug-target interaction prediction during the course of this PhD thesis.

Based on the experience gained from working on this topic, similarity-based methods are generally the better choice when it comes to prediction performance. However, an important thing to note is that the datasets used in this thesis are generally small in size. If the dataset used becomes much larger, the use of similarity-based methods may be infeasible due to their memory complexities; the memory complexity for similarity matrices S_d and S_t is $O(n^2)$ and $O(m^2)$, respectively, where n and m are the numbers of drugs and targets, respectively. In such a case, only feature-based methods can be used for prediction; note that feature-based methods also have the option to undersample the data (particularly, the majority class). Furthermore, since ensemble learning generally enhances prediction performance, it is always better to use an ensemble method whenever possible. In conclusion, it is best to use an ensemble where the base learner is a similarity-based method (e.g. *EnsemKRR*) and, when this is not possible due to memory issues, use a feature-based ensemble (e.g. *EnsemDT*).

It is also worth noting that similarity-based methods have the issue known as the *cold start* problem, which is due to their dependence on matrix multiplications to perform predictions. As an example, suppose a similarity-based method was trained using instances (drug-target pairs) involving n drugs or m targets. The trained method would not be able to perform predictions for new instances involving drugs or targets that did not appear in its training set. More precisely, we would have to add the new drugs and targets to the interaction matrix Y , recalculate the similarity matrices S_d and S_t with these new drugs and targets, and *retrain* the similarity-based method again. Feature-based methods, on the other hand, do not have this issue as trained models can be used to predict interactions for drugs and targets that were not previously seen in the training data. This counts as a strong point in favor of using feature-based methods for drug-target interaction prediction, especially if datasets become much larger than the ones used throughout this thesis.

Moreover, there is the issue of *small disjuncts* that exists in drug-target interaction data. Addressing small disjuncts (or concepts) is an avenue of improvement that is worth

pursuing since it has been empirically shown – via experiments on 30 real-world datasets – that most of the prediction errors are concentrated in these small disjuncts [153, 154]. *Data-level* approaches (such as the basic random oversampling used in Chapter 4) can help improve prediction performance on these small disjuncts. Alternatively, there are also *algorithm-level* approaches that aim to mitigate the generality bias that is hurting the prediction performance on instances belonging to small disjuncts [155]; e.g. by using an inductive bias that is more suitable for imbalanced data.

Although oversampling of small disjuncts did help improve the prediction performance, the technique used in Chapter 4 (i.e. basic random oversampling) is one that is widely assumed to cause overfitting due to the duplication of existing instances without adding any new information [141, 142]. To avoid overfitting, it is possible to use other methods that perform oversampling but without instance duplication (e.g. SMOTE [156] generates new artificial instances between pairs of neighboring instances). However, according to [157], data-level approaches only mask the underlying issue, and the real solution would come from algorithm-level approaches. The reasoning here is that while data-level approaches can potentially improve predictions, they are known to have drawbacks; e.g. oversampling can cause overfitting, undersampling may lead to loss of relevant information, etc. In addition, the reason that these data-level approaches were needed in the first place is that the typical generality bias of machine learning algorithms is unable to adequately handle skewed data distributions (i.e. concepts of varying sizes) and generally favors larger concepts, leading to more prediction errors in smaller ones. As such, it is likely better to adjust the inductive bias of the learning algorithm – which is the source of the problem – via an algorithm-level approach than to modify the training data via a data-level approach.

Finally, the methods proposed in this thesis are computational methods for drug-target interaction prediction that can be used to complement their wet-lab counterparts. Specifically, as computational methods, they may be used to swiftly narrow down the search space and recommend interactions for further validation via wet-lab experiments.

In other words, the methods presented here may be used to guide the wet-lab methods, thus saving valuable time and effort. This, in turn, would accelerate drug discovery processes since they typically involve a step where a search is made for a compound or drug that interacts with a target of interest (see Figure 1.1). As such, we believe that our proposed methods would be beneficial to use in drug discovery pipelines in general.

For the remainder of this chapter, we present an overview of potential future research directions in this field as well as alternative research trends that are related to the line of research pursued in this thesis.

6.1 Utilizing More Information

As mentioned in Section 2.1.2 of Chapter 2, there are multiple information sources that can possibly be used for DTI prediction. These sources represent different aspects of the drugs and targets involved and can help improve prediction performance if used concurrently. We provide some examples below of previous work that used more than one source of information at once.

One such work was [158] where many drug and target kernels were used, and a multiple kernel learning method was developed to take them as input and determine how best to merge them to provide the best predictions. An interesting thing about this work is that some of the kernels were produced from the same information source. From target protein sequences, normalized Smith-Waterman, mismatch and spectrum kernels were created (via the *KeBABS* package [78]). From the drug side, the *Rchemcpp* [77] package was used to obtain spectrum, Lambda-k, Marginalized, Min-Max and Tanimoto kernels from drugs' chemical structures. For more on multiple kernel learning in general, the reader is referred to [129]. Furthermore, Table 6.1 provides a list of software packages that exist for extracting drug and target features from their chemical structures and genomic sequences, respectively.

Package	Link
<i>ChemCPP</i>	http://chemcpp.sourceforge.net
<i>RDKit</i>	http://www.rdkit.org/
<i>PyDPI</i>	https://sourceforge.net/projects/pydpicao/
<i>OpenBabel</i>	http://openbabel.org/
<i>Rcpi</i>	http://bioconductor.org/packages/release/bioc/html/Rcpi.html
<i>Rchemcpp</i>	http://shiny.bioinf.jku.at/Analoging/
<i>KeBABS</i>	http://www.bioinf.jku.at/software/kebabs/
<i>PROFEAT</i>	http://bidd2.nus.edu.sg/cgi-bin/profeat2016/main.cgi

Table 6.1: Software packages to compute features for drugs and targets.

Another work that used multiple kernels was [159]. Drugs were represented as FP2 fingerprints. As for targets, different representations were generated including those based on autocovariance, entropy, discrete wavelet and substitution matrices among several others. An ensemble of *SVM* classifiers was trained; one classifier for each target descriptor type. Each drug-target pair was represented by concatenating the FP2 fingerprint of the drug with the target’s descriptor. Predictions from the different *SVM* classifiers were summed up to give the final predictions.

Secondary structure information of proteins [160] is something that has not been used often. It is a type of information that can be extracted from protein sequences, which is a good thing since genomic sequences are always available for proteins. Known drug-disease and protein-disease associations may also be utilized as auxiliary sources of information [161]; however, this data has also not been used frequently for DTI prediction.

Determination of 3D structures of membrane proteins (via wet-lab techniques) is becoming more feasible over time [44], so we may eventually witness the use of *protein structure* information in *global-scale* DTI prediction. This would possibly trigger the emergence of software packages that would routinely be used to generate descriptors for a protein from its structure (instead of from its sequence). These features may yield

better prediction performance since it is widely accepted that the protein’s structure is what dictates its function.

Finally, for a comprehensive overview of the different ways to integrate multiple information sources simultaneously for improving prediction performance, we refer the reader to [162].

6.2 Ensemble Learning

There are two types of ensemble methods: *heterogeneous* ensembles and *homogeneous* ensembles. Heterogeneous ensembles consist of different learners that have different induction biases. These learners are typically trained using the same data. A procedure known as *stacking* is usually used where the results from the different learners are concatenated to form feature vectors that are then used to train yet another meta-learner which gives the final predictions [135]. The improvement in prediction performance is intended to be obtained from the diversity induced by the different inductive biases of the learners constituting the ensemble.

A heterogeneous ensemble for DTI prediction was previously developed [163] that consisted of four methods: *weighted profile*, *RLS-avg*, *LapRLS* and *NBI*. After predicting with these methods, an *SVM* meta-learner is trained with their results and then used to give the final prediction. The ensemble showed improved prediction performance over all the constituent methods. Another heterogeneous ensemble, *DrugE-Rank* [164], also uses a number of different learners as in [163], but instead of the *SVM* meta-learner, it uses a ranking algorithm, *LambdaMART* [165], to give the final predictions.

In terms of possible future work regarding heterogeneous ensemble methods, a technique that has not been used in previous work is *ensemble pruning*. That is, a *subset* of the base learners is used to constitute the ensemble. This would lead to smaller ensembles and, subsequently, to better computational efficiency due to the lower number of base learners performing predictions. In addition, it was shown that these smaller

ensembles obtained via ensemble pruning can also have better generalization performance [166].

Homogeneous ensembles, on the other hand, consist of learners of the same type. For example, *Random Forest* [134] is a homogeneous ensemble method that consists of many decision trees. To obtain improved prediction performance, the diversity that would help achieve this can come from different sources. An example is *bagging* which induces diversity by randomly sampling with replacement; for each learner, a subset of the training examples is randomly sampled to train it (i.e. each learner uses a different training set). Another example is *feature subsampling* which induces diversity by randomly sampling a subset of the features for each learner, and so on.

A homogeneous ensemble method based on decision trees was introduced in [167]. It randomly projects the features matrix (representing the different drug-target pairs) into a lower dimensionality matrix. This reduced the dimensionality of the data (thus improving computational speed) as well as injected diversity into the ensemble leading to gains in prediction performance. An example of another homogeneous ensemble is [107] which has been described earlier in Section 2.6 of Chapter 2.

Besides bagging and feature subsampling, there are other ways to generate diversity in the base learners that have not yet been employed in homogeneous ensemble methods for DTI prediction. One such way is to use *different parameter settings* for each of the base learners [135]. Another way is to randomly *flip the labels* of some training examples (i.e. convert from 1 to 0 or vice versa) [168]. These tricks may be used to enhance the prediction performance further.

6.3 Deep Learning

The use of deep learning has been steadily increasing in drug discovery [169, 170]. The reason for this is that deep learning has the potential to build complex models that are able to learn difficult concepts and thus outperform other competing methods. In

addition, since it has the ability to extract useful features from the input features, we believe that deep learning methods would especially shine when it comes to merging different sources of information. The two main conditions that must be met for deep learning to be utilized are: (1) a lot of training data is needed to train the complex model being generated, and (2) a lot of computational power is needed to perform the training. As time goes on, these two issues are becoming less of drawbacks due to the accumulation of more data to work with as well as the emerging of more high-performance computing resources.

In [128], the authors suggested the use of the cross validation setting, S4, where drugs and targets used in training do not appear in the test set, and it is known to be a challenging setting indeed. In the experiments conducted in [128], only very trivial interactions were predicted successfully under S4. We believe that deep learning – with its ability to obtain useful deep representations of the drugs, targets and interactions – has the potential to do better than other state-of-the-art methods in predicting interactions under S4. This is yet to be confirmed in future work.

A number of efforts regarding DTI prediction have made use of deep learning to improve prediction performance. Deep learning techniques that have been used in DTI prediction include restricted Boltzmann machines [114], deep neural networks [115-117], stacked auto-encoders [118] and deep belief networks [119].

As of yet, none of the deep learning methods developed for DTI prediction have attempted to *simultaneously utilize multiple heterogeneous sources of drug and target information*. Deep learning is known for achieving great prediction performance from complex data. Enabling deep learning models to work with heterogeneous data may improve the prediction performance further [171]. Deep learning has already been applied to heterogeneous data in other research domains [172, 173], and it would be interesting to see similar efforts in the field of DTI prediction in the future. In the efforts that consider applying deep learning to heterogeneous data sources, some of the main

challenges would be: (1) the heterogeneous data are not all of the same quality, (2) there will surely be missing data as not all kinds of data will be available for all the drugs, targets and interactions, and (3) depending on the types of data being considered, the data volume can be quite huge (e.g. gene expressions [59], functional linkage networks [174], etc.).

6.4 Absence of Reliable Negatives

A prevalent issue in DTI prediction is the absence of a list of reliable negatives; i.e. there are no confident non-interactions. Unfortunately, reporting such non-interactions is not something that researchers routinely do. However, researchers have done efforts to deal with this problem.

Biased SVM is a variant of SVM that was used in [175] to give different weights to the positive and negative classes in the data. Positive examples, being more reliable, are given higher weights than the negative examples. The weights are tuned to give the best possible prediction performance.

PUDT [176] is a DTI prediction method that uses positive unlabeled learning to deal with the issue of unconfident negatives. Sets of unlabeled drug-target pairs are labeled as reliable negative and likely negative. An *SVM* classifier is then trained where, similar to *Biased SVM*, weights are given to these negative classes (along with the positive class) and are tuned to give a good prediction performance.

In [47], multiple drug and target similarities were obtained and then merged together via the following equation,

$$S_{ij} = 1 - \prod_n (1 - S_{ij}^{(n)}). \quad (6.1)$$

After that, predictions are made using a simple network-based method. From the set of predictions, a subset is taken as the reliable negative set that can be used later with any prediction method.

In [48], the BioLip [177] and BindingDB [178] databases are searched for interactions with a binding affinity $< 10\mu M$ to be used as negatives.

Rather than using binary values to represent interactions and non-interactions, the use of datasets where continuous values correspond to drug-target affinities has been previously proposed [128]. Examples of such datasets include those introduced in [179] and [180] which, respectively, contain kinase disassociation constants and kinase inhibition constants – constants with lower values correspond to higher affinities and vice versa. It is suggested in [128] that such datasets be used as benchmarking datasets in future DTI prediction efforts. It is an idea worth considering since these datasets provide a more accurate representation of reality than traditional binary-valued datasets. Using such datasets would also implicitly eliminate the issue of reliable negatives discussed above. We suspect that this is a trend that will increase in the future as data of this kind become more abundant.

6.5 Big Data

Over 90 million chemical compounds are currently stored in PubChem [181], while the conducted BioAssays have only covered about 2.4 million compounds (i.e. targets are now known for these compounds) [37]. It is unlikely that future BioAssay experiments will cover the remaining compounds anytime in the foreseeable future. Virtual screening of these compounds is thus inevitable. However, since the size of the data is exceptionally large, big-data technologies (e.g. cloud computing) will need to be utilized.

However, adjustments to algorithms (or, possibly, novel algorithms altogether) will also need to be made in order to handle data of such size. For example, the work done in [112] is a step in this direction – minwise hashing was used to obtain a compact representation for the drug-target pairs, which reduces the data dimensionality, and the reduced dimensionality helps lower both the space and time complexities. Another work that aims for scalability is [182] where a memory-efficient tree structure is developed to query large databases for similar drug-target pairs.

On a related note, a recent survey on rational drug discovery [183] makes the case that, in this era of big data, deep learning is a good fit for the kinds of challenges that are met in the data. Specifically, big data is typically characterized by four attributes, namely *volume* (size of the data), *velocity* (rate of growth), *variety* (diversity) and *veracity* (certainty), which are collectively known as the four V's. Fortunately, deep learning has the ability to automatically select the proper representation from the given data which is high-dimensional, raw, and heterogeneous in nature [184]. As such, deep learning is a suitable fit for big data-related challenges that are prevalent in the field of drug discovery.

Recently, new technologies for dealing with big data have been emerging and it is becoming easier to process huge amounts of data. *Spark*, for example, is one such technology that can distribute the computational tasks over a cluster of computers, leading to faster processing of the data. It would be interesting to see Spark being used to detect new interactions over large numbers of proteins and compounds and, possibly, utilize such detected interactions to guide the BioAssay experiments mentioned earlier so that they may discover higher numbers of compound-protein interactions.

Appendix A: List of Publications

My work at Nanyang Technological University led to the following list of publications.

Note that Chapters 3, 4 and 5 are based on them:

❖ **Drug-target interaction prediction with graph regularized matrix factorization**

Ali Ezzat, Peilin Zhao, Min Wu, Xiao-Li Li, Chee Keong Kwoh

IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2016

❖ **Drug-target interaction prediction via class imbalance-aware ensemble learning**

Ali Ezzat, Min Wu, Xiao-Li Li, Chee Keong Kwoh

BMC Bioinformatics, 15th International Conference on Bioinformatics, 2016

❖ **Drug-target interaction prediction using ensemble learning and dimensionality reduction**

Ali Ezzat, Min Wu, Xiao-Li Li, Chee Keong Kwoh

Methods (El-Sevier), 2017

In addition, the following survey paper was published based on my work in this thesis:

❖ **Computational prediction of drug–target interactions using chemogenomic approaches: An empirical survey**

Ali Ezzat, Min Wu, Xiao-Li Li, Chee Keong Kwoh

Briefings in Bioinformatics, 2018

Appendix B: Notations & Variables

Following is a list of notations that are used in this thesis as well as variables that are used throughout the different chapters.

Mathematical Notations

$\mathbb{R}$	Set of real numbers
$\ \cdot\ _F$	Frobenius norm
$\ \cdot\ _1$	L_1 norm
$\ \cdot\ $	L_2 norm, can also be written as $\ \cdot\ _2$
$A \otimes B$	Kronecker product of matrices A and B
$A \odot B$	Hadamard (element-wise) product of matrices A and B
$Tr(A)$	Trace of matrix A
A^T	Transpose of matrix A

General

n	Number of drugs
m	Number of targets
Y	Interaction (or adjacency) matrix, $Y \in \mathbb{R}^{n \times m}$
Y_{ij}	$Y_{ij} = 1$ if drug d_i and target t_j interact, and $Y_{ij} = 0$ otherwise
$Y(d_i)$	Interaction profile for drug d_i (i.e. i^{th} row in Y)
$Y(t_j)$	Interaction profile for target t_j (i.e. j^{th} column in Y)
S_d	Drug similarity matrix, $S_d \in \mathbb{R}^{n \times n}$
S_t	Target similarity matrix, $S_t \in \mathbb{R}^{m \times m}$
$S_t(t_j, t_q)$	Similarity between targets t_j and t_q , can also be written as $(S_t)_{jq}$
$\hat{Y}$	Prediction scores matrix, $\hat{Y} \in \mathbb{R}^{n \times m}$
P	Set of positive instances
N	Set of negative instances

Chapter 3

K	Number of nearest neighbors to infer from
η	Decay rate
k	Number of latent features
A	Drug latent feature matrix, $A \in \mathbb{R}^{n \times k}$
B	Target latent feature matrix, $B \in \mathbb{R}^{m \times k}$
p	Number of nearest neighbors in p -nearest neighbor graph
λ_l	Parameter for Tikhonov regularization term
λ_d	Parameter for drug graph regularization term
λ_t	Parameter for target graph regularization term
W	Weight matrix (used in $WGRMF$), $W \in \mathbb{R}^{n \times m}$

Remaining Chapters

M	Number of base learners in the ensemble
r	Portion of randomly selected features per base learner
D	Drug feature matrix, $D \in \mathbb{R}^{n \times f_d}$
T	Target feature matrix, $T \in \mathbb{R}^{m \times f_t}$
f_d	Number of drug features in D
f_t	Number of target features in T
$npRatio$	Negative-to-positive ratio

Appendix C: Bibliography

1. Yildirim MA, Goh K-I, Cusick ME, Barabási A-L, Vidal M: **Drug-target network**. *Nature Biotechnology* 2007, **25**(10):1119-1126.
2. Paul SM, Mytelka DS, Dunwiddie CT, Persinger CC, Munos BH, Lindborg SR, Schacht AL: **How to improve R&D productivity: The pharmaceutical industry's grand challenge**. *Nature Reviews in Drug Discovery* 2010, **9**(3):203-214.
3. Yao L, Evans JA, Rzhetsky A: **Novel opportunities for computational biology and sociology in drug discovery**. *Trends in Biotechnology* 2010, **28**(4):161-170.
4. DiMasi JA, Feldman L, Seckler A, Wilson A: **Trends in risks associated with new drug development: Success rates for investigational drugs**. *Clinical Pharmacology & Therapeutics* 2010, **87**(3):272-277.
5. Hopkins AL: **Network pharmacology: The next paradigm in drug discovery**. *Nature Chemical Biology* 2008, **4**(11):682-690.
6. Imming P, Sinning C, Meyer A: **Drugs, their targets and the nature and number of drug targets**. *Nature Reviews in Drug Discovery* 2006, **5**(10):821-834.
7. Ashburn TT, Thor KB: **Drug repositioning: Identifying and developing new uses for existing drugs**. *Nature Reviews in Drug Discovery* 2004, **3**(8):673-683.
8. Pepin J, Guern C, Milord F, Schechter PJ: **Difluoromethylornithine for arseno-resistant Trypanosoma brucei gambiense sleeping sickness**. *The Lancet* 1987, **330**(8573):1431-1433.
9. Chong CR, Chen X, Shi L, Liu JO, Sullivan DJ: **A clinical drug library screen identifies astemizole as an antimalarial agent**. *Nature Chemical Biology* 2006, **2**(8):415-416.

10. Chow WA, Jiang C, Guan M: **Anti-HIV drugs for cancer therapeutics: back to the future?** *The Lancet Oncology* 2009, **10**(1):61-71.
11. Gloeckner C, Garner AL, Mersha F, Oksov Y, Tricoche N, Eubanks LM, Lustigman S, Kaufmann GF, Janda KD: **Repositioning of an existing drug for the neglected tropical disease Onchocerciasis.** *Proceedings of the National Academy of Sciences* 2010, **107**(8):3424-3429.
12. Aronson JK: **Old drugs–new uses.** *British Journal of Clinical Pharmacology* 2007, **64**(5):563-565.
13. Novac N: **Challenges and opportunities of drug repositioning.** *Trends in Pharmacological Sciences* 2013, **34**(5):267-272.
14. Haupt VJ, Schroeder M: **Old friends in new guise: repositioning of known drugs with structural bioinformatics.** *Briefings in Bioinformatics* 2011.
15. Frantz S: **Drug discovery: Playing dirty.** *Nature* 2005, **437**(7061):942-943.
16. McLean SR, Gana-Weisz M, Hartzoulakis B, Frow R, Whelan J, Selwood D, Boshoff C: **Imatinib binding and cKIT inhibition is abrogated by the cKIT kinase domain I missense mutation Val654Ala.** *Molecular Cancer Therapeutics* 2005, **4**(12):2008-2015.
17. Ooi SL, Pan X, Peyser BD, Ye P, Meluh PB, Yuan DS, Irizarry RA, Bader JS, Spencer FA, Boeke JD: **Global synthetic-lethality analysis and yeast functional profiling.** *Trends in Genetics* 2006, **22**(1):56-63.
18. Roden DM: **Antiarrhythmic Drugs: Past, Present, and Future.** *Pacing and Clinical Electrophysiology* 2003, **26**(12):2340-2348.
19. Hopkins AL, Mason JS, Overington JP: **Can we rationally design promiscuous drugs?** *Current Opinion in Structural Biology* 2006, **16**(1):127-136.
20. Hopkins AL: **Predicting promiscuity.** *Nature* 2009, **462**:167.

21. Li YY, Jones SJ: **Drug repositioning for personalized medicine.** *Genome Medicine* 2012, **4**(3):27.
22. Abdullah LN, Chow EK-H: **Mechanisms of chemoresistance in cancer stem cells.** *Clinical and Translational Medicine* 2013, **2**(1):3.
23. Yoshida GJ: **Therapeutic strategies of drug repositioning targeting autophagy to induce cancer cell death: from pathophysiology to treatment.** *Journal of Hematology & Oncology* 2017, **10**(1):67.
24. Würth R, Thellung S, Bajetto A, Mazzanti M, Florio T, Barbieri F: **Drug-repositioning opportunities for cancer therapy: novel molecular targets for known compounds.** *Drug Discovery Today* 2016, **21**(1):190-199.
25. Shim JS, Liu JO: **Recent advances in drug repositioning for the discovery of new anticancer drugs.** *International journal of biological sciences* 2014, **10**(7):654.
26. Anderson KC: **Dark Remedy: The Impact of Thalidomide and Its Revival as a Vital Medicine.** *Nature Medicine* 2001, **7**:275.
27. Hurle MR, Yang L, Xie Q, Rajpal DK, Sanseau P, Agarwal P: **Computational drug repositioning: From data to therapeutics.** *Clinical Pharmacology & Therapeutics* 2013, **93**(4):335-341.
28. Haupt VJ, Schroeder M: **Old friends in new guise: Repositioning of known drugs with structural bioinformatics.** *Briefings in Bioinformatics* 2011, **12**(4):312-326.
29. Miguel DC, Yokoyama-Yasunaka JKU, Andreoli WK, Mortara RA, Uliana SRB: **Tamoxifen is effective against Leishmania and induces a rapid alkalization of parasitophorous vacuoles harbouring Leishmania (Leishmania) amazonensis amastigotes.** *Journal of Antimicrobial Chemotherapy* 2007, **60**(3):526-534.

30. Chen B, Wild D, Guha R: **PubChem as a source of polypharmacology.** *Journal of Chemical Information and Modeling* 2009, **49**(9):2044-2055.
31. Roy K, Kar S, Das RN: **Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment:** Academic press; 2015.
32. Jin G, Wong STC: **Toward better drug repositioning: Prioritizing and integrating existing methods into efficient pipelines.** *Drug Discovery Today* 2014, **19**(5):637-644.
33. Cheng AC, Coleman RG, Smyth KT, Cao Q, Soulard P, Caffrey DR, Salzberg AC, Huang ES: **Structure-based maximal affinity model predicts small-molecule druggability.** *Nature Biotechnology* 2007, **25**(1):71-75.
34. Barretina J, Caponigro G, Stransky N, Venkatesan K, Margolin AA, Kim S, Wilson CJ, Lehár J, Kryukov GV, Sonkin D *et al*: **The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity.** *Nature* 2012, **483**:603.
35. Barratt MJ, Frail DE: **Drug repositioning: Bringing new life to shelved assets and existing drugs:** John Wiley & Sons; 2012.
36. Keiser MJ, Setola V, Irwin JJ, Laggner C, Abbas AI, Hufeisen SJ, Jensen NH, Kuijter MB, Matos RC, Tran TB *et al*: **Predicting new molecular targets for known drugs.** *Nature* 2009, **462**(7270):175-181.
37. Wang Y, Bryant SH, Cheng T, Wang J, Gindulyte A, Shoemaker BA, Thiessen PA, He S, Zhang J: **PubChem BioAssay: 2017 update.** *Nucleic Acids Research* 2017, **45**(D1):D955-D963.
38. Johnson MA, Maggiora GM: **Concepts and applications of molecular similarity:** Wiley; 1990.

39. Keiser MJ, Roth BL, Armbruster BN, Ernsberger P, Irwin JJ, Shoichet BK: **Relating protein pharmacology by ligand chemistry.** *Nature Biotechnology* 2007, **25**(2):197-206.
40. Li H, Gao Z, Kang L, Zhang H, Yang K, Yu K, Luo X, Zhu W, Chen K, Shen J *et al*: **TarFisDock: A web server for identifying drug targets with docking approach.** *Nucleic Acids Research* 2006, **34**(suppl_2):W219-W224.
41. Pujadas G, Vaque M, Ardevol A, Blade C, Salvado M, Blay M, Fernandez-Larrea J, Arola L: **Protein-ligand docking: A review of recent advances and future perspectives.** *Current Pharmaceutical Analysis* 2008, **4**(1):1-19.
42. Jacob L, Vert J-P: **Protein-ligand interaction prediction: an improved chemogenomics approach.** *Bioinformatics* 2008, **24**(19):2149-2156.
43. Keiser MJ, Hert J: **Off-target networks derived from ligand set similarity.** In: *Chemogenomics: Methods and Applications*. Edited by Jacoby E. Totowa, NJ: Humana Press; 2009: 195-205.
44. Opella SJ: **Structure determination of membrane proteins by nuclear magnetic resonance spectroscopy.** *Annual Review of Analytical Chemistry* 2013, **6**:305-328.
45. Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M: **Prediction of drug–target interaction networks from the integration of chemical and genomic spaces.** *Bioinformatics* 2008, **24**(13):i232-i240.
46. Lü L, Zhou T: **Link prediction in complex networks: A survey.** *Physica A: Statistical Mechanics and its Applications* 2011, **390**(6):1150-1170.
47. Liu H, Sun J, Guan J, Zheng J, Zhou S: **Improving compound–protein interaction prediction by building up highly credible negative samples.** *Bioinformatics* 2015, **31**(12):i221-i229.

48. Coelho ED, Arrais JP, Oliveira JL: **Computational discovery of putative leads for drug repositioning through drug-target interaction prediction.** *PLOS Computational Biology* 2016, **12**(11):e1005219.
49. Bleakley K, Yamanishi Y: **Supervised prediction of drug–target interactions using bipartite local models.** *Bioinformatics* 2009, **25**(18):2397-2403.
50. Park Y, Marcotte EM: **Flaws in evaluation schemes for pair-input computational predictions.** *Nature Methods* 2012, **9**(12):1134-1136.
51. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M: **KEGG for integration and interpretation of large-scale molecular data sets.** *Nucleic Acids Research* 2012, **40**(D1):D109-D114.
52. Knox C, Law V, Jewison T, Liu P, Ly S, Frolkis A, Pon A, Banco K, Mak C, Neveu V *et al*: **DrugBank 3.0: A comprehensive resource for ‘Omics’ research on drugs.** *Nucleic Acids Research* 2011, **39**(suppl 1):D1035-D1041.
53. Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, Light Y, McGlinchey S, Michalovich D, Al-Lazikani B *et al*: **ChEMBL: A large-scale bioactivity database for drug discovery.** *Nucleic Acids Research* 2012, **40**(D1):D1100-D1107.
54. Kuhn M, Szklarczyk D, Pletscher-Frankild S, Blicher TH, von Mering C, Jensen LJ, Bork P: **STITCH 4: Integration of protein–chemical interactions with user data.** *Nucleic Acids Research* 2014, **42**(D1):D401-D407.
55. Chen X, Yan CC, Zhang X, Zhang X, Dai F, Yin J, Zhang Y: **Drug–target interaction prediction: Databases, web servers and computational models.** *Briefings in Bioinformatics* 2016, **17**(4):696-712.
56. Weininger D: **SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules.** *Journal of Chemical Information and Computer Sciences* 1988, **28**(1):31-36.

57. Kuhn M, Campillos M, Letunic I, Jensen LJ, Bork P: **A side effect resource to capture phenotypic effects of drugs.** *Molecular Systems Biology* 2010, **6**(1):343.
58. Skrbo A, Begović B, Skrbo S: **Classification of drugs using the ATC system (Anatomic, Therapeutic, Chemical Classification) and the latest changes.** *Med Arh* 2004, **58**(1 Suppl 2):138-141.
59. Lamb J: **The Connectivity Map: A new tool for biomedical research.** *Nature Reviews in Cancer* 2007, **7**(1):54-60.
60. Cao D-S, Xiao N, Xu Q-S, Chen AF: **Rcpi: R/Bioconductor package to generate various descriptors of proteins, compounds and their interactions.** *Bioinformatics* 2015, **31**(2):279-281.
61. Cao D-S, Liang Y-Z, Yan J, Tan G-S, Xu Q-S, Liu S: **PyDPI: Freely available Python package for chemoinformatics, bioinformatics, and chemogenomics studies.** *Journal of Chemical Information and Modeling* 2013, **53**(11):3086-3096.
62. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR: **Open Babel: An open chemical toolbox.** *Journal of Cheminformatics* 2011, **3**(1):33.
63. Jain E, Bairoch A, Duvaud S, Phan I, Redaschi N, Suzek BE, Martin MJ, McGarvey P, Gasteiger E: **Infrastructure for the life sciences: Design and implementation of the UniProt website.** *BMC Bioinformatics* 2009, **10**(1):136.
64. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT *et al*: **Gene Ontology: Tool for the unification of biology.** *Nature Genetics* 2000, **25**(1):25-29.
65. Emig D, Ivliev A, Pustovalova O, Lancashire L, Bureeva S, Nikolsky Y, Bessarabova M: **Drug target prediction and repositioning using an integrated network-based approach.** *PLOS ONE* 2013, **8**(4):e60618.

66. Zong N, Kim H, Ngo V, Harismendy O: **Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations.** *Bioinformatics* 2017:btx160.
67. Cannataro M, Guzzi PH, Veltri P: **Protein-to-protein interactions: Technologies, databases, and algorithms.** *ACM Computing Surveys* 2010, **43**(1):1-36.
68. Klingström T, Plewczynski D: **Protein–protein interaction and pathway databases, a graphical review.** *Briefings in Bioinformatics* 2011, **12**(6):702-713.
69. Li Z-R, Lin HH, Han LY, Jiang L, Chen X, Chen YZ: **PROFEAT: A web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence.** *Nucleic Acids Research* 2006, **34**(suppl 2):W32-W37.
70. Subramanian A, Narayan R, Corsello SM, Peck DD, Natoli TE, Lu X, Gould J, Davis JF, Tubelli AA, Asiedu JK *et al*: **A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles.** *Cell* 2017, **171**(6):1437-1452.e1417.
71. Xenarios I, Salwinski L, Duan XJ, Higney P, Kim S-M, Eisenberg D: **DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions.** *Nucleic Acids Research* 2002, **30**(1):303-305.
72. Stark C, Breitkreutz B-J, Chatr-aryamontri A, Boucher L, Oughtred R, Livstone MS, Nixon J, Auken KV, Wang X, Shi X *et al*: **The BioGRID Interaction Database: 2011 update.** *Nucleic Acids Research* 2011, **39**(Database issue):D698-D704.
73. Licata L, Briganti L, Peluso D, Perfetto L, Iannuccelli M, Galeota E, Sacco F, Palma A, Nardozza AP, Santonico E *et al*: **MINT, the molecular interaction database: 2012 update.** *Nucleic Acids Research* 2012, **40**(D1):D857-D861.

74. Kerrien S, Aranda B, Breuza L, Bridge A, Broackes-Carter F, Chen C, Duesbury M, Dumousseau M, Feuermann M, Hinz U *et al*: **The IntAct molecular interaction database in 2012**. *Nucleic Acids Research* 2012, **40**(Database issue):D841-D846.
75. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, Telikicherla D, Raju R, Shafreen B, Venugopal A *et al*: **Human Protein Reference Database—2009 update**. *Nucleic Acids Research* 2009, **37**(suppl_1):D767-D772.
76. Szklarczyk D, Jensen LJ: **Protein-Protein Interaction Databases**. In: *Protein-Protein Interactions: Methods and Applications*. Edited by Meyerkord CL, Fu H. New York, NY: Springer New York; 2015: 39-56.
77. Klambauer G, Wischenbart M, Mahr M, Unterthiner T, Mayr A, Hochreiter S: **Rchemcpp: A web service for structural analoging in ChEMBL, Drugbank and the Connectivity Map**. *Bioinformatics* 2015, **31**(20):3392-3394.
78. Palme J, Hochreiter S, Bodenhofer U: **KeBABS: An R package for kernel-based analysis of biological sequences**. *Bioinformatics* 2015, **31**(15):2574-2576.
79. Hattori M, Okuno Y, Goto S, Kanehisa M: **Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways**. *Journal of the American Chemical Society* 2003, **125**(39):11853-11865.
80. Smith TF, Waterman MS: **Identification of common molecular subsequences**. *Journal of Molecular Biology* 1981, **147**(1):195-197.
81. Shi J-Y, Yiu S-M: **SRP: A concise non-parametric similarity-rank-based model for predicting drug-target interactions**. In: *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM): 9-12 Nov. 2015* 2015. 1636-1641.

82. Faulon J-L, Misra M, Martin S, Sale K, Sapra R: **Genome scale enzyme-metabolite and drug-target interaction predictions using the signature molecular descriptor.** *Bioinformatics* 2008, **24**(2):225-233.
83. Xia Z, Wu L-Y, Zhou X, Wong ST: **Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces.** *BMC systems biology* 2010, **4**(Suppl 2):S6.
84. Tenenbaum JB, Silva Vd, Langford JC: **A global geometric framework for nonlinear dimensionality reduction.** *Science* 2000, **290**(5500):2319-2323.
85. Roweis ST, Saul LK: **Nonlinear dimensionality reduction by locally linear embedding.** *Science* 2000, **290**(5500):2323-2326.
86. Belkin M, Niyogi P: **Laplacian eigenmaps and spectral techniques for embedding and clustering.** In: *NIPS: 2001*. 585-591.
87. Laarhoven TV, Nabuurs SB, Marchiori E: **Gaussian interaction profile kernels for predicting drug-target interaction.** *Bioinformatics* 2011, **27**(21):3036-3043.
88. Raymond R, Kashima H: **Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs.** In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*. Edited by Balcázar JL, Bonchi F, Gionis A, Sebag M. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010: 131-147.
89. Rifkin R, Klautau A: **In defense of one-vs-all classification.** *Journal of machine learning research* 2004, **5**(Jan):101-141.
90. Mei J-P, Kwoh C-K, Yang P, Li X-L, Zheng J: **Drug-target interaction prediction by learning from local information and neighbors.** *Bioinformatics* 2013, **29**(2):238-245.

91. Laarhoven TV, Marchiori E: **Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile.** *PLOS ONE* 2013, **8**(6):e66952.
92. Cheng F, Liu C, Jiang J, Lu W, Li W, Liu G, Zhou W, Huang J, Tang Y: **Prediction of drug-target interactions and drug repositioning via network-based inference.** *PLOS Computational Biology* 2012, **8**(5):e1002503.
93. Wang W, Yang S, Li J: **Drug target predictions based on heterogeneous graph inference.** In: *Pacific Symposium on Biocomputing: 2013*. World Scientific: 53-64.
94. Vanunu O, Magger O, Ruppin E, Shlomi T, Sharan R: **Associating genes and protein complexes with disease via network propagation.** *PLOS Computational Biology* 2010, **6**(1):e1000641.
95. Chen X, Liu M-X, Yan G-Y: **Drug-target interaction prediction by random walk on the heterogeneous network.** *Molecular BioSystems* 2012, **8**(7):1970-1978.
96. Fakhraei S, Huang B, Raschid L, Getoor L: **Network-based drug-target interaction prediction with probabilistic soft logic.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2014, **11**(5):775-787.
97. Ba-alawi W, Soufan O, Essack M, Kalnis P, Bajic VB: **DASPfind: New efficient method to predict drug-target interactions.** *Journal of Cheminformatics* 2016, **8**(1):15.
98. Gönen M: **Predicting drug-target interactions from chemical and genomic kernels using Bayesian matrix factorization.** *Bioinformatics* 2012, **28**(18):2304-2310.

99. Cobanoglu MC, Liu C, Hu F, Oltvai ZN, Bahar I: **Predicting drug–target interactions using probabilistic matrix factorization.** *Journal of Chemical Information and Modeling* 2013, **53**(12):3399-3409.
100. Zheng X, Ding H, Mamitsuka H, Zhu S: **Collaborative matrix factorization with multiple similarities for predicting drug-target interactions.** In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Chicago, Illinois, USA.* 2487670: ACM 2013: 1025-1033.
101. Liu Y, Wu M, Miao C, Zhao P, Li X-L: **Neighborhood regularized logistic matrix factorization for drug-target interaction prediction.** *PLOS Computational Biology* 2016, **12**(2):e1004760.
102. Johnson CC: **Logistic matrix factorization for implicit feedback data.** *Advances in Neural Information Processing Systems* 2014, **27**.
103. Klabunde T: **Chemogenomic approaches to drug discovery: Similar receptors bind similar ligands.** *British Journal of Pharmacology* 2007, **152**(1):5-7.
104. Hao M, Bryant SH, Wang Y: **Predicting drug-target interactions by dual-network integrated logistic matrix factorization.** *Scientific Reports* 2017, **7**:40376.
105. He Z, Zhang J, Shi X-H, Hu L-L, Kong X, Cai Y-D, Chou K-C: **Predicting drug-target interaction networks based on functional groups and biological features.** *PLOS ONE* 2010, **5**(3):e9603.
106. Peng H, Long F, Ding C: **Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy.** *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 2005, **27**(8):1226-1238.

107. Yu H, Chen J, Xu X, Li Y, Zhao H, Fang Y, Li X, Zhou W, Wang W, Wang Y: **A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data.** *PLOS ONE* 2012, **7**(5):e37608.
108. Xiao X, Min J-L, Wang P, Chou K-C: **iGPCR-Drug: A web server for predicting interaction between gpcrs and drugs in cellular networking.** *PLOS ONE* 2013, **8**(8):e72234.
109. Perlman L, Gottlieb A, Atias N, Ruppin E, Sharan R: **Combining drug and gene similarity measures for drug-target elucidation.** *Journal of Computational Biology* 2011, **18**(2):133-145.
110. Yamanishi Y, Pauwels E, Saigo H, Stoven Vr: **Extracting sets of chemical substructures and protein domains governing drug-target interactions.** *Journal of Chemical Information and Modeling* 2011, **51**(5):1183-1194.
111. Finn RD, Bateman A, Clements J, Coghill P, Eberhardt RY, Eddy SR, Heger A, Hetherington K, Holm L, Mistry J *et al*: **Pfam: The protein families database.** *Nucleic Acids Research* 2014, **42**(D1):D222-D230.
112. Tabei Y, Yamanishi Y: **Scalable prediction of compound-protein interactions using minwise hashing.** *BMC Systems Biology* 2013, **7**(Suppl 6):S3.
113. Broder AZ, Charikar M, Frieze AM, Mitzenmacher M: **Min-wise independent permutations.** *Journal of Computer and System Sciences* 2000, **60**(3):630-659.
114. Wang Y, Zeng J: **Predicting drug-target interactions using restricted Boltzmann machines.** *Bioinformatics* 2013, **29**(13):i126-i134.
115. Wang C, Liu J, Luo F, Tan Y, Deng Z, Hu Q-N: **Pairwise input neural network for target-ligand interaction prediction.** In: *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM): 2-5 Nov. 2014* 2014. 67-70.

116. Tian K, Shao M, Wang Y, Guan J, Zhou S: **Boosting compound-protein interaction prediction by deep learning.** *Methods* 2016, **110**:64-72.
117. Wan F, Zeng J: **Deep learning with feature embedding for compound-protein interaction prediction.** *bioRxiv* 2016.
118. Hu P-W, Chan KCC, You Z-H: **Large-scale prediction of drug-target interactions from deep representations.** In: *2016 International Joint Conference on Neural Networks (IJCNN): 24-29 July 2016* 2016. 1236-1243.
119. Wen M, Zhang Z, Niu S, Sha H, Yang R, Yun Y, Lu H: **Deep-learning-based drug-target interaction prediction.** *Journal of Proteome Research* 2017, **16**(4):1401-1409.
120. Wassermann AM, Geppert H, Bajorath J: **Ligand prediction for orphan targets using support vector machines and various target-ligand kernels is dominated by nearest neighbor effects.** *Journal of Chemical Information and Modeling* 2009, **49**(10):2155-2167.
121. Ezzat A, Zhao P, Wu M, Li X-L, Kwoh C-K: **Drug-target interaction prediction with graph regularized matrix factorization.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2016, **14**(3):646-656.
122. Cai D, He X, Han J, Huang TS: **Graph regularized nonnegative matrix factorization for data representation.** *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 2011, **33**(8):1548-1560.
123. Gu Q, Zhou J, Ding C: **Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs.** In: *Proceedings of the 2010 SIAM International Conference on Data Mining.* Society for Industrial and Applied Mathematics; 2010: 199-210.

124. Belkin M, Niyogi P, Sindhvani V: **Manifold regularization: A geometric framework for learning from labeled and unlabeled examples.** *The Journal of Machine Learning Research* 2006, **7**:2399-2434.
125. Sarwar B, Karypis G, Konstan J, Riedl J: **Application of dimensionality reduction in recommender system -- A case study.** In.: DTIC Document; 2000.
126. Langville AN, Meyer CD, Albright R, Cox J, Duling D: **Algorithms, initializations, and convergence for the nonnegative matrix factorization.** *arXiv preprint arXiv:14077299* 2014.
127. Davis J, Goadrich M: **The relationship between precision-recall and ROC curves.** In: *Proceedings of the 23rd International Conference on Machine Learning; Pittsburgh, Pennsylvania, USA*. 1143874: ACM 2006: 233-240.
128. Pahikkala T, Airola A, Pietilä S, Shakyawar S, Szwajda A, Tang J, Aittokallio T: **Toward more realistic drug–target interaction predictions.** *Briefings in Bioinformatics* 2014, **16**(2):325-337.
129. Gönen M, Alpaydın E: **Multiple kernel learning algorithms.** *Journal of Machine Learning Research* 2011, **12**:2211-2268.
130. He H, Garcia EA: **Learning from imbalanced data.** *IEEE Transactions on Knowledge and Data Engineering* 2009, **21**(9):1263-1284.
131. Weiss GM: **Mining with rarity: A unifying framework.** *SIGKDD Explorations Newsletter* 2004, **6**(1):7-19.
132. Holte RC, Acker L, Porter BW: **Concept Learning and the Problem of Small Disjuncts.** In: *IJCAI: 1989*. Citeseer: 813-818.
133. Ezzat A, Wu M, Li X-L, Kwoh C-K: **Drug-target interaction prediction via class imbalance-aware ensemble learning.** *BMC Bioinformatics* 2016, **17**(19):267-276.
134. Breiman L: **Random forests.** *Machine Learning* 2001, **45**(1):5-32.

135. Zhou Z-H: **Ensemble methods: Foundations and algorithms**: CRC press; 2012.
136. Sun S: **A survey of multi-view machine learning**. *Neural Computing and Applications* 2013, **23**(7):2031-2038.
137. Zhao J, Xie X, Xu X, Sun S: **Multi-view learning overview: Recent progress and new challenges**. *Information Fusion* 2017, **38**:43-54.
138. Arthur D, Vassilvitskii S: **K-means++: The advantages of careful seeding**. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms; New Orleans, Louisiana*. 1283494: Society for Industrial and Applied Mathematics 2007: 1027-1035.
139. Meir R, Rätsch G: **An Introduction to boosting and leveraging**. In: *Advanced Lectures on Machine Learning: Machine Learning Summer School 2002 Canberra, Australia, February 11–22, 2002 Revised Lectures*. Edited by Mendelson S, Smola AJ. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003: 118-183.
140. Japkowicz N, Stephen S: **The class imbalance problem: A systematic study**. *Intelligent data analysis* 2002, **6**(5):429-449.
141. Drummond C, Holte RC: **C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling**. In: *Workshop on learning from imbalanced datasets II: 2003*. Citeseer: 1-8.
142. Batista GEAPA, Prati RC, Monard MC: **A study of the behavior of several methods for balancing machine learning training data**. *SIGKDD Explor Newsl* 2004, **6**(1):20-29.
143. Fawcett T: **An introduction to ROC analysis**. *Pattern Recognition Letters* 2006, **27**(8):861-874.

144. Lim E, Pon A, Djoumbou Y, Knox C, Shrivastava S, Guo AC, Neveu V, Wishart DS: **T3DB: A comprehensively annotated database of common toxins and their targets.** *Nucleic Acids Research* 2010, **38**(suppl_1):D781-D786.
145. Ezzat A, Wu M, Li X-L, Kwoh C-K: **Drug-target interaction prediction using ensemble learning and dimensionality reduction.** *Methods* 2017.
146. Kung SY: **Kernel methods and machine learning:** Cambridge University Press; 2014.
147. Bermingham ML, Pong-Wong R, Spiliopoulou A, Hayward C, Rudan I, Campbell H, Wright AF, Wilson JF, Agakov F, Navarro P *et al*: **Application of high-dimensional feature selection: evaluation for genomic prediction in man.** *Scientific Reports* 2015, **5**:10312.
148. de Jong S: **SIMPLS: An alternative approach to partial least squares regression.** *Chemometrics and Intelligent Laboratory Systems* 1993, **18**(3):251-263.
149. Cristianini N, Shawe-Taylor J: **Support vector machines and other kernel-based learning methods.** In.: Cambridge University Press, New York; 2000.
150. Tabei Y, Pauwels E, Stoven V, Takemoto K, Yamanishi Y: **Identification of chemogenomic features from drug–target interaction networks using interpretable classifiers.** *Bioinformatics* 2012, **28**(18):i487-i494.
151. Finn RD, Coghill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL, Potter SC, Punta M, Qureshi M, Sangrador-Vegas A *et al*: **The Pfam protein families database: Towards a more sustainable future.** *Nucleic Acids Research* 2016, **44**(D1):D279-D285.
152. Erdas O, Buyukbingol E, Alpaslan FN, Adejare A: **Modeling and predicting binding affinity of phencyclidine-like compounds using machine learning methods.** *Journal of Chemometrics* 2010, **24**(1):1-13.

153. Weiss GM, Hirsh H: **A quantitative study of small disjuncts**. In: *AAAI/IAAI*: 2000. 665-670.
154. Weiss GM: **The Impact of Small Disjuncts on Classifier Learning**. In: *Data Mining: Special Issue in Annals of Information Systems*. Edited by Stahlbock R, Crone SF, Lessmann S. Boston, MA: Springer US; 2010: 193-226.
155. Krawczyk B: **Learning from imbalanced data: open challenges and future directions**. *Progress in Artificial Intelligence* 2016, **5**(4):221-232.
156. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP: **SMOTE: synthetic minority over-sampling technique**. *Journal of artificial intelligence research* 2002, **16**:321-357.
157. He H, Ma Y: **Imbalanced learning: foundations, algorithms, and applications**: John Wiley & Sons; 2013.
158. Nascimento ACA, Prudêncio RBC, Costa IG: **A multiple kernel learning algorithm for drug-target interaction prediction**. *BMC Bioinformatics* 2016, **17**(1):46.
159. Nanni L, Lumini A, Brahnam S: **A set of descriptors for identifying the protein–drug interaction in cellular networking**. *Journal of Theoretical Biology* 2014, **359**(0):120-128.
160. Lin K, Simossis VA, Taylor WR, Heringa J: **A simple and fast secondary structure prediction method using hidden neural networks**. *Bioinformatics* 2005, **21**(2):152-159.
161. Wang W, Yang S, Zhang X, Li J: **Drug repositioning by integrating target information through a heterogeneous network model**. *Bioinformatics* 2014, **30**(20):2923-2930.
162. Li Y, Wu F-X, Ngom A: **A review on machine learning principles for multi-view biological data integration**. *Briefings in Bioinformatics* 2016:bbw113.

163. Zhang R: **An ensemble learning approach for improving drug–target interactions prediction.** In: *Proceedings of the 4th International Conference on Computer Engineering and Networks*. Edited by Wong WE: Springer International Publishing; 2015: 433-442.
164. Yuan Q, Gao J, Wu D, Zhang S, Mamitsuka H, Zhu S: **DrugE-Rank: Improving drug–target interaction prediction of new candidate drugs or targets by ensemble learning to rank.** *Bioinformatics* 2016, **32**(12):i18-i27.
165. Burges CJC: **From RankNet to LambdaRank to LambdaMART: An overview.** *Microsoft Research* 2010, **11**(23-581):81.
166. Zhou Z-H, Wu J, Tang W: **Ensembling neural networks: Many could be better than all.** *Artificial Intelligence* 2002, **137**(1):239-263.
167. Zhang J, Zhu M, Chen P, Wang B: **DrugRPE: Random projection ensemble approach to drug-target interaction prediction.** *Neurocomputing* 2017, **228**:256-262.
168. Breiman L: **Randomizing outputs to increase prediction accuracy.** *Machine Learning* 2000, **40**(3):229-242.
169. Gawehn E, Hiss JA, Schneider G: **Deep Learning in Drug Discovery.** *Molecular Informatics* 2016, **35**(1):3-14.
170. Ekins S: **The next era: Deep learning in pharmaceutical research.** *Pharm Res* 2016, **33**(11):2594-2603.
171. Miotto R, Wang F, Wang S, Jiang X, Dudley JT: **Deep learning for healthcare: review, opportunities and challenges.** *Briefings in Bioinformatics* 2017:bbx044-bbx044.
172. Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY: **Multimodal deep learning.** In: *Proceedings of the 28th international conference on machine learning (ICML-11): 2011*. 689-696.

173. Zhang Y, Ai Q, Chen X, Croft WB: **Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources**. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management; Singapore, Singapore*. 3132892: ACM 2017: 1449-1458.
174. Linghu B, Franzosa EA, Xia Y: **Construction of Functional Linkage Gene Networks by Data Integration**. In: *Data Mining for Systems Biology: Methods and Protocols*. Edited by Mamitsuka H, DeLisi C, Kanehisa M. Totowa, NJ: Humana Press; 2013: 215-232.
175. Cheng Z, Zhou S, Wang Y, Liu H, Guan J, Chen Y-PP: **Effectively identifying compound-protein interactions by learning from positive and unlabeled examples**. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2016, **PP**(99):1-1.
176. Lan W, Wang J, Li M, Liu J, Li Y, Wu F-X, Pan Y: **Predicting drug–target interaction using positive-unlabeled learning**. *Neurocomputing* 2016, **206**:50-57.
177. Yang J, Roy A, Zhang Y: **BioLiP: A semi-manually curated database for biologically relevant ligand–protein interactions**. *Nucleic Acids Research* 2013, **41**(D1):D1096-D1103.
178. Chen X, Liu M, Gilson MK: **BindingDB: A web-accessible molecular recognition database**. *Combinatorial Chemistry & High Throughput Screening* 2001, **4**(8):719-725.
179. Davis MI, Hunt JP, Herrgard S, Ciceri P, Wodicka LM, Pallares G, Hocker M, Treiber DK, Zarrinkar PP: **Comprehensive analysis of kinase inhibitor selectivity**. *Nature Biotechnology* 2011, **29**(11):1046-1051.
180. Metz JT, Johnson EF, Soni NB, Merta PJ, Kifle L, Hajduk PJ: **Navigating the kinome**. *Nature Chemical Biology* 2011, **7**(4):200-202.

181. Kim S, Thiessen PA, Bolton EE, Chen J, Fu G, Gindulyte A, Han L, He J, He S, Shoemaker BA *et al*: **PubChem Substance and Compound databases**. *Nucleic Acids Research* 2016, **44**(D1):D1202-D1213.
182. Tabei Y, Kishimoto A, Kotera M, Yamanishi Y: **Succinct interval-splitting tree for scalable similarity search of compound-protein pairs with property constraints**. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Chicago, Illinois, USA*. 2487637: ACM 2013: 176-184.
183. Zhang L, Tan J, Han D, Zhu H: **From machine learning to deep learning: progress in machine intelligence for rational drug discovery**. *Drug Discovery Today* 2017, **22**(11):1680-1685.
184. Chen X-W, Lin X: **Big Data Deep Learning: Challenges and Perspectives**. *IEEE Access* 2014, **2**:514-525.