

University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2017

Drug-Target Interaction Networks Prediction Using Short-linear Motifs

Wenxiao Xu
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Xu, Wenxiao, "Drug-Target Interaction Networks Prediction Using Short-linear Motifs" (2017). *Electronic Theses and Dissertations*. 6029.
<https://scholar.uwindsor.ca/etd/6029>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Drug-Target Interaction Networks Prediction Using Short-linear Motifs

By

Wenxiao Xu

A Thesis

Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2017

©2017 Wenxiao Xu

Drug-Target Interaction Networks Prediction Using Short-linear Motifs

by

Wenxiao Xu

APPROVED BY:

D. Yang
Department of Mathematics and Statistics

M. Kargar
School of Computer Science

L. Rueda, co-Advisor
School of Computer Science

A. Ngom, Advisor
School of Computer Science

May 4, 2017

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Drug-target interaction (DTI) prediction is a fundamental step in drug discovery and genomic research, and contributes to medical treatment. Various computational methods have been developed to find potential DTIs. Machine learning (ML) has been currently used for new DTIs identification from existing DTI networks. There are mainly two ML-based approaches for DTI network prediction: similarity-based methods and feature-based methods. In this thesis, we propose a feature-based approach, and firstly use short-linear motifs (SLiMs) as descriptors of protein. Additionally, chemical substructure fingerprints are used as features of drug. Moreover, another challenge in this field is the lack of negative data for the training set because most data which can be found in public databases is interaction samples. Many researchers regard unknown drug-target pairs as non-interaction, which is incorrect, and may cause serious consequences. To solve this problem, we introduce a strategy to select reliable negative samples according to the features of positive data. We use the same benchmark datasets as previous research in order to compare with them. After trying three classifiers k nearest neighbours (k -NN), Random Forest (RF) and Support Vector Machine (SVM), we find that the results of k -NN are satisfied but not as excellent as RF and SVM. Compared with existing approaches using the same datasets to solve the same problem, our method performs the best under most circumstance.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Dr. Alioune Ngom and Dr. Luis Rueda for their constant guidance and encouragement during my whole Master's period at the University of Windsor. Without their valuable help, this thesis would not have been possible.

I would also like to express my appreciation to my thesis committee members Dr. Mehdi Kargar and Dr. Dilian Yang. Thank you all for your valuable guidance and suggestions to this thesis.

Last but not least, I want to express my gratitude to my parents and my friends who give me consistent help over the past two years.

TABLE OF CONTENTS

| | |
|---|-------------|
| DECLARATION OF ORIGINALITY | III |
| ABSTRACT | IV |
| ACKNOWLEDGEMENTS | V |
| LIST OF TABLES | VIII |
| LIST OF FIGURES | IX |
| 1 Introduction | 1 |
| 1.1 Drug-target Interaction Network | 1 |
| 1.2 Research Motivation | 2 |
| 1.3 Problem Restatement | 4 |
| 2 Review of the Literature | 5 |
| 2.1 Similarity-based Method | 5 |
| 2.2 Feature-based Method | 6 |
| 2.2.1 Structural and Physicochemical Properties | 7 |
| 2.2.2 PSSM Based Evolutionary Information | 10 |
| 2.2.3 Class Imbalance-aware Ensemble Learning | 12 |
| 2.3 Summary | 15 |
| 3 Material and Methodology | 16 |
| 3.1 Gold Standard Dataset | 16 |
| 3.2 Protein Feature Representation | 18 |
| 3.2.1 Short-linear Motif | 18 |
| 3.2.2 <i>I</i> -Score Approach | 21 |
| 3.2.3 Sliding Window Score Approach | 24 |
| 3.3 Drug Feature Representation | 25 |
| 3.4 Negative Samples Selection | 28 |
| 3.5 Classification and Validation | 30 |
| 3.5.1 <i>k</i> -NN | 30 |
| 3.5.2 Random Forest | 31 |
| 3.5.3 Support Vector Machine | 32 |
| 3.5.4 mRMR Feature Selection | 32 |
| 3.5.5 Performance Evaluation | 33 |
| 4 Results and Discussion | 35 |
| 4.1 Results | 35 |
| 4.2 Comparison | 36 |
| 5 Conclusion | 45 |

| | |
|---------------|----|
| REFERENCES | 46 |
| VITA AUCTORIS | 52 |

LIST OF TABLES

| | | |
|----|---|----|
| 1 | The values of AUC among NN, BLM, PKM for different datasets . . . | 6 |
| 2 | Performances comparison between 5-fold and independent validation . | 9 |
| 3 | Performances comparison between random and balanced sampling . . | 13 |
| 4 | The number of motifs extracted for each dataset | 21 |
| 5 | Position-specific Probability Matrix of Ion Channel No. 86 SLiM . . . | 22 |
| 6 | Counting sites matrix | 23 |
| 7 | Drug Feature Matrix | 28 |
| 8 | Results of Nuclear Receptor dataset | 37 |
| 9 | Results of GPCR dataset | 38 |
| 10 | Results of Ion Channel dataset | 39 |
| 11 | Results of Enzyme dataset | 40 |
| 12 | Results of mRMR feature selection applied on SVM with <i>I</i> -score and 881 fingerprints | 43 |
| 13 | The comparison of AUC among existing methods using benchmark datasets | 44 |

LIST OF FIGURES

| | | |
|----|---|----|
| 1 | A drug-target interaction network graph [9] | 2 |
| 2 | Amino acids attributes and division | 8 |
| 3 | ROC curves of Bigram-PSSM model using random and balanced datasets for benchmark datasets | 12 |
| 4 | Flowchart of proposed model | 17 |
| 5 | The format of benchmark dataset | 17 |
| 6 | A short-linear motif expression | 19 |
| 7 | Flowchart of using MEME to extract SLiMs | 19 |
| 8 | FASTA format | 20 |
| 9 | (a) Motif 7, (b) Motif 8, (c) Motif 9 | 23 |
| 10 | Example of counting sites | 23 |
| 11 | An example to show how to find real sites in sliding window score method ($\lambda = 0.6$) | 26 |
| 12 | The first 25 chemical substructure fingerprints in PubChem | 27 |
| 13 | An example of SMILES format searching process | 27 |
| 14 | Command in R and its output to identify drug descriptors | 28 |
| 15 | Comparison among different SLiMs scoring methods, classifiers and different types of fingerprints for each dataset | 41 |
| 16 | Comparison among different datasets using RF with 881 fingerprints . | 42 |
| 17 | Comparison of AUC between with and without mRMR | 43 |

CHAPTER 1

Introduction

Drug research is necessary for treatment of disease. At the molecular level, proteins are the main targets for drugs. In this case, identification of drug-target interaction networks is a fundamental step of genomic drug discovery, drug design and pharmacology, which is also the problem needed to be addressed in this thesis. Because biochemical experiments in wet lab (aka in vitro methods) spend too much time, expense and human resources, computational methods (aka in silico methods) have been developed to predict drug-target interaction on a large scale nowadays.

1.1 Drug-target Interaction Network

Drug can be defined as a compound (molecular entity) that interacts with one or more molecular targets and effects a change in biological state [18]. Target proteins refer to bio-molecules (functional modules) of living organisms [3] formed by amino acid sequences. There are mainly four types of target proteins commonly known in humans, which are Nuclear Receptors, G-protein Coupled Receptors (GPCRs), Ion Channels, and Enzymes. Functions of target proteins can be affected by interacting with compounds [24], which means that drug compounds can enhance or inhibit functions carried out by target proteins. Such a pair of drug and its target protein is regarded as a drug-target interaction, where each drug or protein can be represented by a node, while an interaction can be shown as a link. In this case, a large number of drug-target pairs constitute drug-target interaction networks. The data about drug-target interaction can be obtained from KEGG BRITE [20], DrugBank [38],

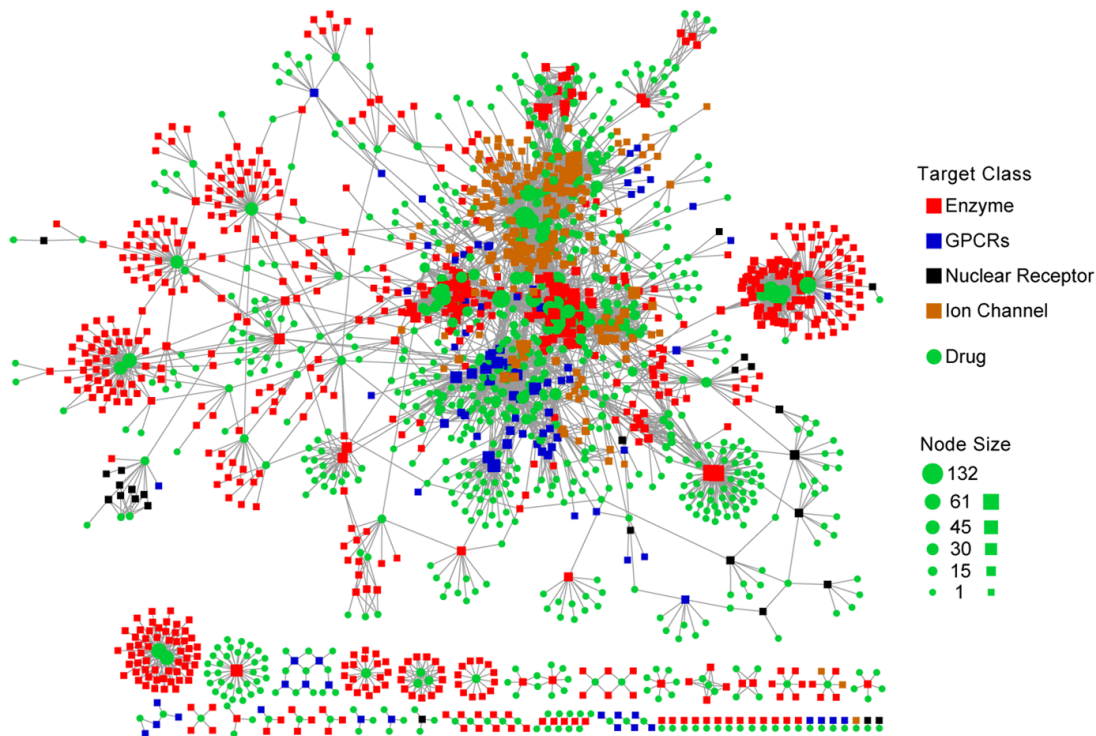


Fig. 1: A drug-target interaction network graph [9]

BRENDA [30], SuperTarget and Matador [16], ChEMBL [35] and GLIDA [25].

Fig. 1 shows an example of drug-target interaction network for four benchmark datasets: Enzymes (red), Ion Channels (orange), GPCRs (blue), and Nuclear Receptors (black). Circles represent for drugs, while rectangles are targets. An edge between a pair of drug-target means a known interaction. Since the number of known interactions is quite small, our purpose is to discover new drug-target interactions based on known information to enlarge the network.

1.2 Research Motivation

The amount of known drug-target interactions is limited, so more interactions need to be detected in order to support drug research. Many researchers focus on *in silico* drug-target interaction prediction because of labor-intensive, time-consuming and costly experimental process. As the assistance of *in vitro* experiments, computational

methods provide much useful information to narrow down experimental subjects. There are mainly two aspects in terms of *in silico* methods: docking simulation and machine learning. The main idea of docking simulation is to simulate the molecular recognition process between drug and its targets on the computer, which requires 3-dimensional structures for both of them. Characteristics of binding behaviours can be observed clearly during simulation. However, there is a severe limitation that 3-dimensional structures of most membrane proteins cannot be found in public databases. Thus, facing to an enormous amount of target proteins, machine learning is an appropriate choice which is used in this thesis. Moreover, approaches based on machine learning can also be categorized into two aspects: similarity-based and feature-based. More details about previous research in this field will be explained in Chapter 2 Review of the Literature.

Feature-based is the method we use in this thesis. In previous research, n -gram has been used as the feature of protein, which is a simple chain consisting of n amino acids with no meaning. However, short-linear motifs (SLiMs) are different from n -gram although they are both composed of amino acids. SLiMs are involved in recognition and targeting activities, which may contain the information relevant to binding with compounds. The concept of SLiMs has been used to predict protein-protein interaction, and gained a satisfied result, but no one has tried it in the field of drug-target interaction prediction. Thus, we firstly use SLiMs as features of proteins in this thesis.

Another problem is that only interaction data which normally called positive data can be obtained in public databases. Although some drugs and proteins are not shown as interacted, it does not mean they are non-interacted. It is possible that the interaction relationship has not been discovered, or their interactions are relatively weak, so they are not shown as positive. Many researchers regard all the unknown data as negative samples (non-interactive data), which is incorrect, and may cause serious consequence. In this case, negative samples selection is a crucial problem. Based on the strategy proposed by [33], we propose a new approach considering the degree of drugs and proteins.

1.3 Problem Restatement

In conclusion, the number of known DTIs data is quite a few, so we need to find target proteins for more drug compounds to enlarge DTI network. The problem can be defined as follows. Given a new pair of drug and protein, we aim to predict whether they are interacted or not, so that potential DTIs can be found. This research is very valuable which contributes to medical treatment, such as new drug discovery or drug side effects study. To solve this problem, the main idea of ML-based method is to build models based on known DTIs data by different classification algorithms, and use them to do prediction. Among ML-based methods, we use feature-based approach, and firstly use SLiMs as the feature of protein.

CHAPTER 2

Review of the Literature

This chapter reviews the previous research and publication on prediction of drug-target interaction networks using machine learning (ML) method, which can be mainly categorized into two aspects: similarity-based [12][14] and feature-based [6][24][13]. They will be introduced in details below. Beside of these, there are some graph-based approaches, such as network diffusion [9], and random walk [8].

2.1 Similarity-based Method

The main idea of similarity-based methods is based on drug and protein respective similarity matrixes to make a prediction. The element of i -th row and j -th column in drug similarity matrix is the similarity of drug i and drug j . In the same way, protein similarity matrix stores the similarity score among proteins. [12] introduces several models of similarity-based method, such as nearest neighbour (NN) [5], bipartite local models (BLM) [5][4][23], pairwise kernel method (PKM) [19], and etc.

Concerning NN method, they create a binary vector for each drug to present whether a target protein interacts with this drug or not, which is called drug interaction profile. Similarly, target protein interaction profile can also be generated. Given a new drug, its most similar known drug can be obtained according to drug similarity matrix, and called as nearest neighbours (NN). The interaction profile of a new drug can be computed by multiplying each value of its NN interaction profile by their similarity score. Interaction profile of a new target is generated in the same way. The average of these two results is the final score. NN is a very efficient model

Table 1: The values of AUC among NN, BLM, PKM for different datasets

| Models | Enzyme | Ion Channel | GPCR | Nuclear Receptor |
|--------|--------|-------------|-------|------------------|
| NN | 0.898 | 0.889 | 0.852 | 0.820 |
| BLM | 0.928 | 0.918 | 0.884 | 0.694 |
| PKM | 0.966 | 0.967 | 0.937 | 0.856 |

spending less time.

BLM extends the method of local models. Given an unknown pair of drug and target. In order to predict whether they are interacted, they first focus on this drug, and give every known target a label as +1 or -1 based on interaction data, which aims to divide them into two classes. Then train an SVM classifier according to protein similarity matrix which is regarded as attribute vectors of targets to predict the label of this unknown protein. The unknown drug can also be predicted in a similar way. This method costs a large computation because it needs to train two classifiers for every unknown pair of drug and protein prediction.

PKM is an SVM-based method. Based on the similarities between two proteins and two drugs, they obtain a similarity score of these two drug-target pairs by similarities multiplication. In this case, a similarity matrix of drug-target pairs can be created, and named as kernel matrix. They regard interacted and non-interacted data as two classes, and kernel matrix as the input to train an SVM classifier. Compare with the other two methods mentioned above, the result of PKM performs the best. Table 1 shows AUC values of these three models for different datasets.

2.2 Feature-based Method

The main idea of feature-based method is to find different descriptors for both proteins and drugs, then generate feature matrix which is represented by feature vectors for all interaction and non-interaction data to do prediction. Different papers use different drug-target representations.

2.2.1 Structural and Physicochemical Properties

[6] proposes a feature-based method to predict drug-target interaction, and uses MACCS fingerprints as features of drug, while use protein sequential representation as features of protein.

New Idea

This paper extends structure-activity relationship (SAR) methodology, and uses drug topological structures and protein sequential representation.

Datasets

This paper uses the benchmark datasets which are proposed by [39] in 2008. These datasets are used in much research, and they are divided by four types of target proteins, which are nuclear receptor, G-protein coupled receptors (GPCR), ion channel, and enzyme. It shows interactions between drug and its target protein. The number of proteins in these four datasets is 26, 95, 204, 664 respectively, and the number of drugs is 54, 223, 210 and 445 respectively. Moreover, there are 90, 635, 1476 and 2916 interactions between a pair of drug-target.

Datasets only show interaction pairs which are also known as positive data, so authors need to select negative data (non-interactive samples) in the first step. They regard all the unknown pairs as negative data, which means that all the possible drug-target combinations except positive samples are negative ones. In this case, the data is unbalanced, and may cause bias. To solve this problem, Negative samples are randomly selected as one to two times of positive ones. Authors generate ten different negative samples in the same way.

Methodology

MACCS substructure fingerprints are used as drug descriptors, which use a dictionary of MDL keys consisting of 166 features. Each feature is a functional molecular fragment. If a fragment can be found in this drug, then set the value of this position

as 1. If the feature is not included in this drug, then set it as 0. In this case, a 166-dimensional vector can be generated for each drug.

Amino acid attributes and the division of the amino acids into three groups for each attribute.

| | Group 1 | Group 2 | Group 3 |
|---------------------------------|-----------------------------------|---|---------------------------------------|
| Hydrophobicity | Polar R, K, E, D, Q, N | Neutral G, A, S, T, P, H, Y | Hydrophobicity C, L, V, I, M, F, W |
| Normalized van der Waals volume | 0-2.78 G, A, S, T, P, D | 2.95-4.0 N, V, E, Q, I, L | 4.03-8.08 M, H, K, F, R, Y, W |
| Polarity | 4.9-6.2 L, I, F, W, C, M, V, Y | 8.0-9.2 P, A, T, G, S | 10.4-13.0 H, Q, R, K, N, E, D |
| Polarizability | 0-1.08 G, A, S, D, T | 0.128-0.186 C, P, N, V, E, Q, I, L | 0.219-0.409 K, M, H, F, R, Y, W |
| Charge | Positive K, R | Neutral A, N, C, Q, G, H, I, L, M, F, P, S, T, W, Y, V | Negative D, E |
| Secondary structure | Helix E, A, L, M, Q, K, R, H | Strand V, I, Y, C, W, F, T | Coil G, N, P, S, D |
| Solvent accessibility | Buried A, L, F, C, G, I, V, W | Exposed R, K, Q, E, N, D | Intermediate M, S, P, T, H, Y |

Fig. 2: Amino acids attributes and division

To present proteins, the authors use four types of descriptors, which are amino acids, composition (C), transition (T) and distribution (D). Amino acid chains need to be obtained first for each protein. Regarding amino-acid property consisting of 20 kinds of amino acids, it shows the number of each kind of amino acid in a protein sequence. Moreover, according to seven different attributions such as hydrophobicity, polarizability, polarity, and etc, amino acids can be divided into 3 groups for each attribute, which are shown as Fig. 2. The value of C, T and D can be calculated based on Fig. 2. First, they assign every amino acid in protein sequence an index from 1. Then according to hydrophobicity division, they give each amino acid a number of '1', '2', or '3', and compute the number of amino acids marked as '1', '2', or '3' separately denoted by n_1 , n_2 and n_3 respectively. In term of descriptor C, there are 3 features including a composition of group 1, 2 and 3. They can be calculated as $n_p \times 100 / (n_1 + n_2 + n_3)$, where p means the p^{th} group. There are still 3 features of descriptor T, which are the transition from group '1' to group '2' (or '2' to '1'), from group '1' to group '3' (or '3' to '1'), and from group '2' to group '3' (or '3' to '2'). After score the number of each transition, they denote them as n_i , n_{ii} and n_{iii} . In this case, the value of each feature in T can be computed as $(n_j / (n_i + n_{ii} + n_{iii})) \times 100$ where j means one transition. Concerning the last type D, assume that there are N_a amino acids for group 1, they can find out the index of the first amino acid in this group,

Table 2: Performances comparison between 5-fold and independent validation

| Datasets | 5-fold cross validation | | | | Independent validation set | | | |
|------------------|-------------------------|-------------|----------|-------|----------------------------|-------------|----------|-------|
| | sensitivity | specificity | accuracy | MCC | sensitivity | specificity | accuracy | MCC |
| Enzyme | 90.10 | 90.64 | 90.31 | 79.77 | 88.23 | 88.39 | 88.28 | 78.27 |
| Ion Channel | 89.38 | 88.20 | 88.91 | 77.41 | 88.27 | 87.34 | 87.64 | 76.32 |
| GPCR | 82.54 | 85.49 | 84.68 | 67.10 | 80.59 | 84.74 | 82.97 | 65.34 |
| Nuclear Receptor | 82.35 | 84.72 | 83.74 | 64.55 | 81.82 | 81.48 | 81.63 | 63.09 |

and also the first 25%, 50%, 75% and 100% amino acids in group 1, then mark their indexes in protein sequence as n_{a_1} , n_{a_2} , n_{a_3} , n_{a_4} and n_{a_5} . There are 5 features for each group in D, and can be computed as $n_{a_i}/N \times 100$, where $i = 1, 2, 3, 4, 5$ respectively, and N is the length of protein sequence. Other two groups can do the same operation, so 15 features can be obtained totally in terms of D. Repeat the operation above for all seven functional structures. With the addition of 20 amino acid descriptors, there are 167 features totally for protein.

After obtaining the features of drug and protein, they combine them together according to the interaction and non-interaction data. Then they use Support vector machine (SVM) classifier with 5-fold cross validation and independent validation to do the prediction, and use grid search to determine the best parameters of SVM.

Results and Discussion

Table 2 shows the results for four datasets, which contains the values of sensitivity, specificity, accuracy and Matthews correlation coefficient (MCC). The authors draw a receiver operating curve (ROC) using the value of sensitivity (true positives) and specificity (false positives) for each dataset. The area under ROC is called AUC, which is an important measure to validate the performance. The AUC values of Enzyme, Ion channel, GPCR, and Nuclear receptor are 94.86%, 94.28%, 89.02% and 88.22% respectively.

2.2.2 PSSM Based Evolutionary Information

[24] proposes a feature-based method, and extract evolutionary from position specific scoring matrix (PSSM).

New Idea

This paper firstly uses bi-gram as features of protein, and proposes Bigram-PSSM model to predict drug-target interaction. Besides, authors also use BRS-nonint [43] algorithm to do negative samples selection. This algorithm is used to select negative samples in protein-protein interaction area, and this paper firstly uses it in drug-target interaction.

Datasets

This paper uses the same dataset as [6], and also regards all unknown combinations as negative data. Authors use two ways selecting negative samples to avoid bias, random and balanced. Random method is to select data from unknown samples randomly until reach the similar size as positive samples, while balanced method is evolved from BRS-nonint algorithm which considers balanced degree distribution of drugs and proteins in both positive and negative datasets.

Methodology

Authors use 881 molecular substructure fingerprints defined by PubChem database as descriptors of drugs. After SMILES format of each drug obtained, they use *rcdk* package in R to find the fingerprints in this drug, and generate an 881-dimensional binary vector corresponding to 881 fingerprints. If the corresponding substructure can be found in drug molecule, then set it as 1. Otherwise, set it as 0.

For representing target proteins, authors propose a method called Bigram-PSSM, which regards the probabilities of two amino acids extracted from position specific scoring matrix (PSSM) as features of proteins. Every protein sequence has a PSSM where each column represents an amino acid. There are 20 kinds of amino acids

totally, so there are 20 columns in PSSM. Each row means one position in this protein sequence, so the number of rows in PSSM is the same as the length of the protein sequence. The value in i^{th} row and j^{th} column is the probability of j^{th} amino acid occurring in the i^{th} position. If there are N amino acids in a protein sequence, the value of bi-gram (a pattern of two amino acids) for this protein can be calculated followed by Equation (1).

$$B_{m,n} = \sum_{i=1}^{N-1} P_{m,i} \times P_{n,i+1} (1 \leq m \leq 20, 1 \leq n \leq 20) \quad (1)$$

where $P_{i,j}$ means the value of i^{th} column and j^{th} row in PSSM. There are 400 kinds of bi-gram combination, so each protein can be described as a 400-dimensional vector shown as below:

$$B = (B_{1,1}, B_{1,2}, \dots, B_{1,20}, B_{2,1}, B_{2,2}, \dots, B_{2,10}, \dots, B_{20,1}, \dots, B_{20,10}) \quad (2)$$

where $B_{i,j}$ denotes the value of regarding the i^{th} and j^{th} amino acids as a pattern.

According to the interaction and non-interaction data generated using random and balanced methods, the authors create a 1281-dimensional vector for each record, which is combined 881-dimensional drug vector and 400-dimensional protein vector together. Then they use Support Vector Machine (SVM) classifier to do the classification.

Results and Discussion

Fig. 3 shows ROC curves of Bigram-PSSM model using random and balanced methods for each dataset, while Table 3 summarizes the statistics measures of Bigram model. It is obviously that the performance of random sampling is better than balanced one, and also indicates that negative samples selection affects the performance of model. The authors state that the results are satisfied, and better than other existing research in terms of AUC.

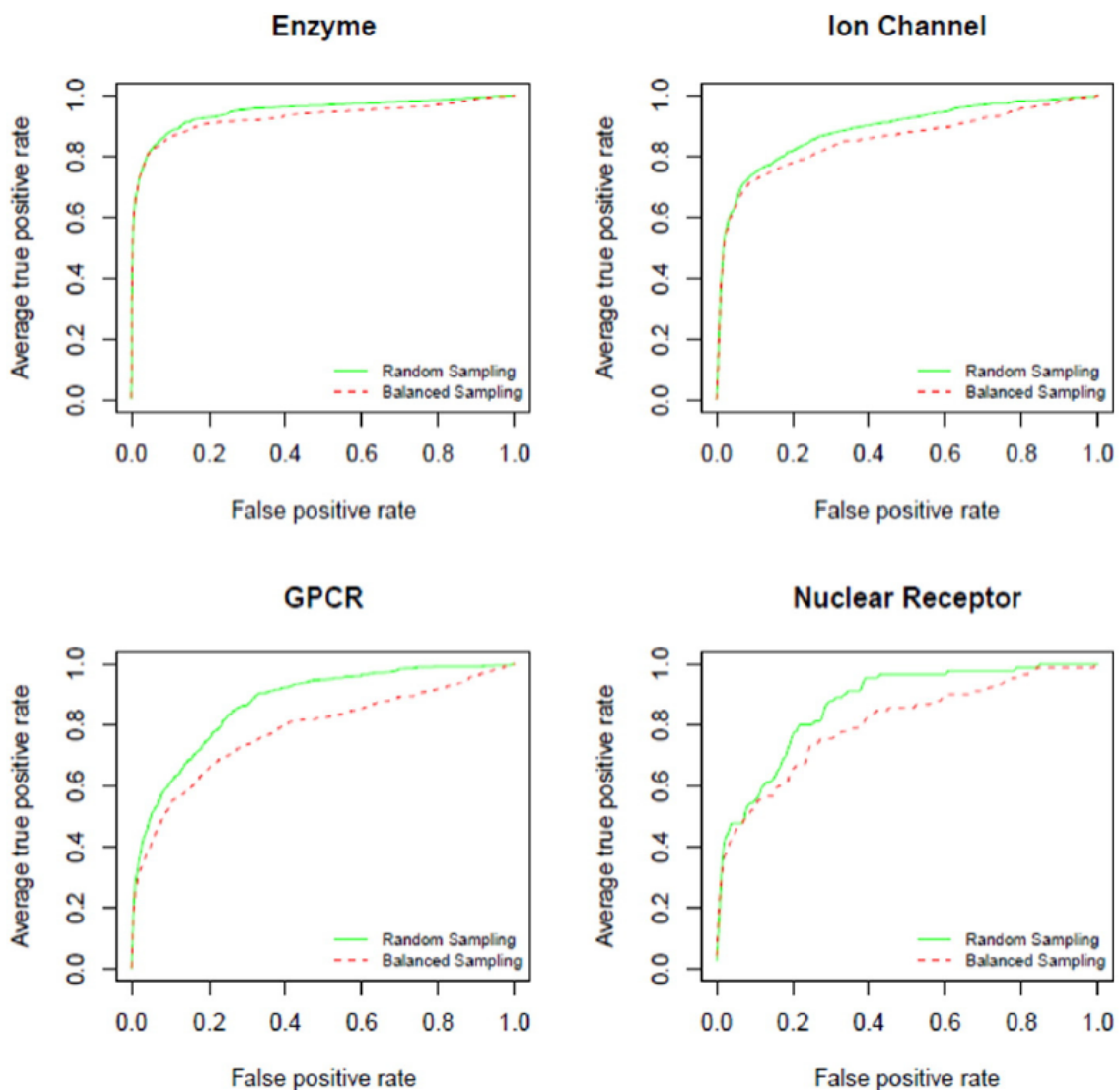


Fig. 3: ROC curves of Bigram-PSSM model using random and balanced datasets for benchmark datasets

2.2.3 Class Imbalance-aware Ensemble Learning

[13] is also an feature-based method. Besides of data representation chosen, authors find that classes imbalance may degrade prediction performance. Thus, this paper improve the performance by balancing classes.

New Idea

Class imbalance in a challenge in drug-target interaction prediction, and this problem is divided into two aspects: between class and within-class imbalance. Between-class

Table 3: Performances comparison between random and balanced sampling

| Datasets | Sampling | AUC | Sensitivity | Specificity | Precision |
|------------------|----------|------|-------------|-------------|-----------|
| Enzyme | Random | 94.8 | 60.9 | 99.4 | 48.9 |
| | Balanced | 92.8 | 57.6 | 99.6 | 59.1 |
| Ion Channel | Random | 88.9 | 49.5 | 98.2 | 49 |
| | Balanced | 85.5 | 47.3 | 98.3 | 49.8 |
| GPCR | Random | 87.2 | 30.9 | 98.6 | 42.8 |
| | Balanced | 78 | 3.8 | 99.9 | 47.2 |
| Nuclear Receptor | Random | 86.9 | 33.3 | 98.4 | 61.4 |
| | Balanced | 80.3 | 2.22 | 99.7 | 6.67 |

imbalance means that the number of positive samples is much less than negative ones, while within-class imbalance means that the numbers of different types of drug-target interaction are imbalanced. Some previous research ignores this problem. The authors state that performance will be better after solving this problem.

Data

Data used in this research is collected from DrugBank public database. It consists of 5877 drugs, 3348 target proteins, and 12674 interaction. Similar as the benchmark dataset used by [6] [24], there is no non-interactive pair in this dataset. They regard all other pairs which not occur in the positive dataset as negative samples. Additionally, this dataset mixes all the types together, rather than divided by different types of proteins like the gold standard dataset.

Data Representation

In terms of drug descriptor, they use *Rcpi* package to generate the 193-dimensional vector for each drug. To present target proteins, they regard amino acids, dipeptide, autocorrelation, quasi-sequence-order, amphiphilic pseudo-amino acid, and informa-

tion of composition, transition, and distribution as features of proteins. Authors try to combine all the protein properties extracted from [17][42][36] together to generate a list of 1290 attributes. Then normalize the value of these features.

Methodology

Authors propose a new classification method. Similar as Random Forest, they train T decision trees as predictors, and average all the scores generated by these decision trees to get the final score. positive dataset is denoted by P , while negative set is N . In this case, P_i means a subset of positive instances, and N_i is a subset of negative instances. To build one decision tree, they randomly select a feature subset names as F_i , and use it to extract some positive samples P_i . To avoid bias caused by within-class imbalance, the authors oversample P_i . They use k -means++ method clustering P_i data into K clusters, and each cluster represents a type of interaction. If there are 4 types, K will equal to 4. The sizes of these K clusters may be different, and the largest size is named as *maxClusterSize*. Then reselect the all these K clusters from P , until the size is the same as *maxClusterSize* for each cluster. P_i is replaced by this new set of positive instances. Next, in order to avoid bias caused by between-class imbalance, they randomly extract the same number as P_i from negative set N , so that $|P_i| = |N_i|$. They also modify negative samples in N_i by F_i which is the subset of features. After that, the samples selected in N_i is removed from the original set, which means each negative sample can only be used once. According to the same step, they build T decision trees.

Results and Discussion

Compare the value of AUC with Decision Tree (0.760), SVM (0.804), Random Forest (0.855), k -NN (0.814), this proposed method (0.900) performs the best. The AUC values of other classifiers are 0.760, RF is the second best one because RF can also deal with imbalanced classes. Thus, it is verified that predicted performance can be improved by deal with unbalanced data effectively. Authors also use this method predict some new drug and new protein interaction. Authors state that this method

is reliable to do drug-target interaction prediction.

2.3 Summary

Most of the results shown in this three literature are satisfied except [24]. As we can see in Table 3, the results are not satisfied, where the values of sensitivity and precision are extremely low. It indicates that TP rate are very low, and it is a terrible performance, not as satisfied as the authors claimed.

[24] and [13] both state that imbalanced classes have had an influence on final results, which should be considered in this thesis. Besides, all this three literature regards all the possible combinations which are not shown as interaction as negative data for no reason, which is incorrect, and may cause serious consequence. Such data can only be called as unknown samples because we cannot verify whether they are interacted or not. Reliable negative data selection is study-worthy.

After reviewing the literature, it can be found that what we need to do is not only seeking better and more meaningful descriptors to represent drug and protein, but also considering fully trying to decrease the bad influence from everywhere, such as imbalanced classes, unreliable negative data, and etc.

CHAPTER 3

Material and Methodology

In this chapter, we propose a new feature-based method to predict drug-target interaction. To put it simply, the main idea of feature-based approach is to find descriptors of protein and drug respectively, and give them values for both positive and negative samples as their features to do classification. The datasets and method will be introduced in detail in this chapter.

Fig. 4 is a flowchart showing every step of the model we proposed. After datasets obtained, the first step is to determine the representations of protein and drug. In this thesis, we firstly use short-linear motifs (SLiMs) as features of protein, chemical substructure fingerprints as features of drug. Then give each feature a value for every record. Specifically, SLiMs can be scored according to position-specific probability matrix (PSPM) in two different approaches, while chemical substructure fingerprints can be converted into a binary vector for each drug. Negative samples selection is also considered after that. As a result, a drug-target features matrix can be generated as an input data to do classification.

3.1 Gold Standard Dataset

To be comparable with previous research, we use the same gold standard dataset which was released by [39] in 2008. The data in this benchmark dataset is taken from KEGG BRITE, DrugBank, BRENDA and SuperTarget databases, and classified into four sets according to different types of proteins, which are Nuclear Receptor, GPCR, Ion Channel, and Enzyme.

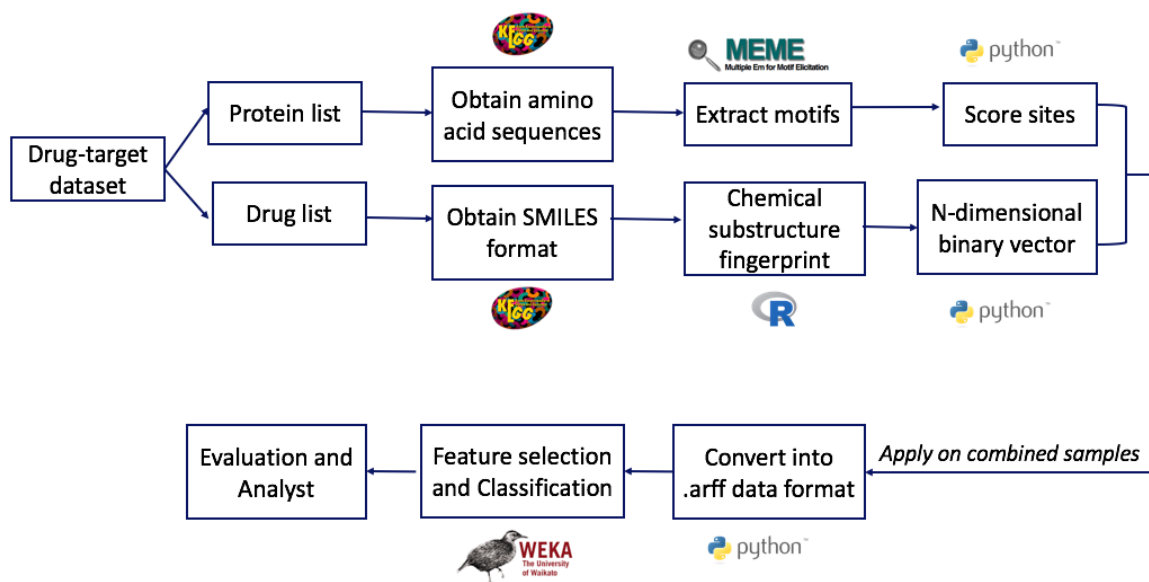


Fig. 4: Flowchart of proposed model

Fig. 5 shows the format of the dataset. The first column represents protein codes which can be used as entries to obtain amino-acid sequences from KEGG database, while the second column indicates drug codes which can be used to gain molecular formulas of compounds also from KEGG, and each row means that this pair of drug-target is interacted. The numbers of proteins in Nuclear Receptor, GPCR, Ion Channel, and Enzyme datasets are 26, 95, 204 and 664 respectively, while the numbers of drugs are 54, 223, 210 and 445 respectively. Moreover, there are 90, 635, 1476 and 2926 known interactions in each dataset separately.

| | |
|-----------|--------|
| hsa:10056 | D00021 |
| hsa:1017 | D02880 |
| hsa:1018 | D02880 |
| hsa:10188 | D01441 |
| hsa:1019 | D02880 |
| hsa:1020 | D02880 |
| hsa:1021 | D02880 |
| hsa:1022 | D02880 |
| hsa:1024 | D02880 |
| hsa:1025 | D02880 |
| hsa:10269 | D00279 |
| hsa:10279 | D00043 |
| hsa:10279 | D00160 |
| hsa:10295 | D00039 |
| hsa:10295 | D00065 |

Fig. 5: The format of benchmark dataset

As we can see, gold standard dataset only shows positive samples. However, in order to train a model for classification, negative samples are also needed in parallel with positive ones as two classes in a training set. In this case, reliable negative samples selection is a necessary step. It is worth noting that we cannot regard all the unknown samples as negative ones because some of them may have potential interactive relationships which have not been detected yet, while some may have weak interactions which do not mean entirely non-interacted although the relationship is not strong enough. Our purpose is to exclude distractors, and select true negative samples from unknown data. Then combine positive and negative ones together as a training set.

3.2 Protein Feature Representation

3.2.1 Short-linear Motif

Concept Introduction

In genetics, a motif is a nucleotide or amino-acid sequence pattern that is widespread and may have a biological significance [37]. In terms of protein, a motif should be a meaningful sequence of amino acid pattern extracted from a set of protein sequences. Short-linear sequence motifs (SLiMs) or minimotifs in protein sequences are short patterns of 3 to 10 amino acids that have been found to be interesting [1]. SLiMs are involved in recognition and targeting activities, which may contain the information relevant to binding with compounds. Thus, we use SLiMs as features of protein.

Fig. 6 shows an expression of Ion Channel No. 86 SLiM. Each letter represents for an amino acid, and the length of this SLiM is 10, which means there are 10 positions available for amino acids to place in this pattern. Another type of expression is $[VIA][AS]R[FL][ST]PYEW[YH]$. $[VIA]$ means amino acid V , I and A are all possible in the first place, but the probability may be different. The height of letters shown in Fig. 6 correspond to the probability of amino acids. In other words, the higher a letter is, the more probability this amino acid occurs. Each possible pattern



Fig. 6: A short-linear motif expression

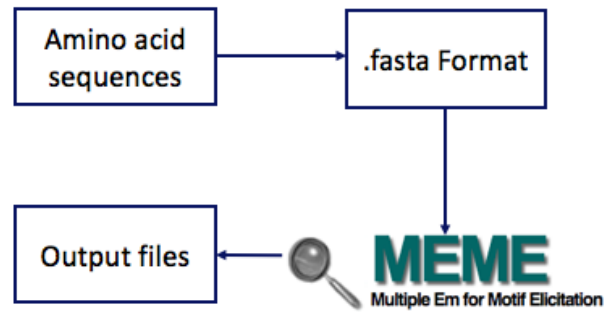


Fig. 7: Flowchart of using MEME to extract SLiMs

is called a site of this SLiM. For example, *VARFSPYEWY* and *IARFSPYEWY* are two sites, but they belong to the same SLiM.

Multiple EM for Motif Elucidation (MEME)

Many tools can be used to extract SLiMs, such as Minimotif Miner(MnM) [29], SLiM-Search [11], SLiMFinder [10] and Multiple EM for Motif Elucidation (MEME) [2]. In this thesis, we use MEME as the tool to generate SLiMs. MEME use unsupervised and Expectation Maximization (EM) algorithm to optimize statistical parameters in order to get motifs from a set of protein, DNA or RNA sequences. It can be installed locally on Linux, OS X and Cygwin systems, and also provides web server online.

Fig. 7 shows that how to use MEME to extract SLiMs. First, amino acid sequences need to be obtained from public databases according to protein identifiers shown in the first column in benchmark dataset. Then, generate a FASTA file which can be read by MEME, and the format is shown as Fig. 8. This is a text-based format including protein names followed by their peptide sequences. Each letter represents an amino acid. Besides, DNA or RNA names with their nucleotide sequences are also acceptable, but we only consider protein sequences in this thesis. “>” is a symbol followed by a protein identifier indicating that previous protein sequence is over, and this is a new one. Its peptide sequence is placed on a new line. After FASTA file generated, it is regarded as an input file to extract SLiMs according to the command shell below.

```

>hsa:10008
METTINGTETWYESLHAVLKALNATLHSNLLCRPGPLGDPNQTEERRASLPGRDDNSYMYILFVMFLFAVTVGSLILGYTRS R KVDKRS
PYHVYIKNRVSMI
>hsa:10060
MSLSFCGNNISSYNINDGVLQNSCFVDALNLVPHVFLFITFPILFIGWGSQSSKVQIHHNTWLHFPGHNLRWILTFALLFVHVCEIAEG
IVSDSRRESRHLHLMFPAVMGFVATTTISIVYYHNIETSNFPKLLALFLYVWMAFITKTIKLVKYCQSGLDISNLRFCITGMMVILNGLL
MAVEINVIRVRRYVFFMNPQKVKPPEDLQDLGVRFLQPFVNLKSKATYWMNTLIISAHKKPIDLKAIGKLP IAMRAVTNYVCLKDAYEE
QKKKVADHPNRTPSIWLAMYRAFGRPILLSSTFRYLADLLGFAGPLCISGIVQRVNETQNGTNTTIGISETLSSKEFLENAYVLAVLLFL
ALILQRTFLQASYYYTIETGINLRGALLAMIYNKILRLSTSNLSMGEMTLGQINNLVAIETNQLMWFLFLCPNLWAMPVQIIMGVILLYN
LLGSSALVGAIVILLAPIQYFIATKLAEAQKSTLDYSTERLKKTNEILKGIKLLKLYAWEHIFCKSVETRMKELSSLKTFFALYTSLSI
FMNAAIPIAAVLATFVTHAYASGNLKPAAEFASLSLFHILVPLFLLSTVVRFVAVKAIISVQKLNFLSDEIGDSDWRTGESSLPFES
CKKHTGVQPKTINRKPGRYHLDSEYEQSTRRLRPAETEDIAIKVTNGYFVSWGGLATLSNIDIRIPTGQLTMIVGVQVCGKSSLLAILG
EMQTLLEGKVHWSNVNESEPSFEATRSRNRYSVAYAAQKPWLLNATVEENITFGSPFNKQRYKAVTDACSLQPDIDLLPFGDQTEIGERGI
NLSGGQRQRICVARALYQNTNIVFLDDPFSAIDHLSDHLMQEGILKFLQDDKRTLVLVTHKLQYLTHADWIIAMKDGSVLREGTLKDIQ
TKDVELYEHWKTLMNRQDQELEKDEADQTTLERKTLRRAMYSREAKAQMEDEDEEEEEDEDDNMSTVMRLRTPKMPWKCWRYLTSGG
FFLLILMIFSKLLKHSVIVAIDYWLATWTSEYSINNTGKADQTYVAGFSILCGAGIFLCLVTSLTVEWMGLTAAKNLHNNLKNKILGP
IRFFDTPPLGLILNRFSAIDNIHQHPPTLES LRSTLLCLSAIGMISYATPVFLVALLPLGVAFYFIQKYFRVASKDLQELDDSTQLP
LLCHFSETAEGLTIRAFRHETRFKQRMLELTDNINIAYLFLSAANRWLEVRTDYLGACIVLTASIASISGSSNSGLVGLLTYALTITN
YLNWVVRNLADLEVQMGAVKKNVNSFLTMESENYEGTMDPSQVPEHWPQEGEIKIHDLCVRYENNLKPVLVKHVKAYIKPGQKVGICGRTGS
GKSSLSLAFFRMVDFDGGKIVIDGIDISKLPLHLRSLRSIILQDPILFSGSIRFNLDPCECKTDDRLEWALEIAQLKNMVKSLPGGLDA
VVTEGGENFSVGRQLFLCLARAFVRKSSILIMDEATASIDMATENILQKVVMTAFADRTVVVIAHRVSSIMDAGLVLFVSEGLVVECDTV
PNLLAHKNGLFFSTLVMTNK
>hsa:10369
MGLFDRGVQMLLTTVGAAFAAFSLMTIAVGTDYWLYSRGVCKTKSVSENETSKKNEEVMTHSGLWRTCCLEGNFKGLCKQIDHFPEDADYE
ADTAEYFLRAVRASSIFPILSVILLFMGGLCIAASEFYKTRHNILSAGIFFVSAGLSNIIGIIVYISANAGDPSKSDSKKNSYSYGSWF
YFGALSFIIAEMVGLAVHMFIDRHKQLRATARATDYLQASAITRIPSYRYRQRRSRSSSRSTEPSHSRDASVPVGIKGFNTLPSTEISM
YTLSDRPLKAAATPTATYNSDRDNSFLQVHNCIQKENKDSLSHNTANRRTPP

```

Fig. 8: FASTA format

```
meme inputfile.fasta -oc outputfile -mod anr -nmotifs 50 -minw 3 -maxw 10
```

Regarding this command, “inputfile.fasta” is the FASTA file generated as input, while “outputfile” is the folder name to store results, and it can be set as any name we want. “-minw” is the minimum size of motifs, while “-maxw” represents the maximum size. As introduced previously, SLiMs mean the motifs whose size are between 3 to 10, so we set “-minw” as 3, while “-maxw” as 10. “-nmotif” in this command means the number of motifs needs to be extracted, and it is set as 50 in this instance. Table 4 shows this value for each dataset, where $N()$ means “the number of”. The original intention of SLiMs amount setting is the same number as proteins. However, 26 SLiMs for Nuclear Receptor dataset is too small to get a satisfying performance, so it is enlarged to 100. Also, in terms of Enzyme dataset, it spends approximately 340 hours to generate 50 SLiMs. To let the research proceed smoothly, 50 SLiMs is set tentatively in this thesis.

Table 4: The number of motifs extracted for each dataset

| Datasets | N (target proteins) | N (SLiMs) |
|-------------------|---------------------|-----------|
| Enzyme | 662 | 50 |
| Ion Channels | 204 | 204 |
| GPCR | 95 | 95 |
| Nuclear Receptors | 26 | 100 |

Position-specific Probability Matrix

From MEME output file, we can obtain position-specific probability matrix (PSPM) for every SLiM. Table 5 is the PSPM corresponding to the same SLiMs as Fig. 6. Each column represents for one amino acid. In this case, there are 20 columns in PSPM because proteins consist of 20 kinds of amino acids totally. Moreover, each row represents a position in this SLiM. As we know, the length of No.86 SLiM is 10, so this PSPM includes 10 rows. The value in this matrix means the probability of corresponding amino acid occurring in particular position. As we can see from the first row, amino acid *A*, *I* and *V* are all possible occurring in the first position of this pattern, and probabilities are 0.1, 0.3 and 0.5 respectively, which tallies with the information extracted from Fig. 6.

Based on PSPM, we can assign every motif in each protein a score using two approaches: *I*-score proposed by [28], and sliding window method [22], so that a protein feature matrix can be built as the features of proteins. These two scoring methods aim to solve protein-protein interaction problem, so after the considering concrete problem in this thesis, *I*-score and sliding window approaches are restated.

3.2.2 *I*-Score Approach

In *I*-score approach, we regard the SLiMs extracted by MEME as the features of protein.

Table 5: Position-specific Probability Matrix of Ion Channel No. 86 SLiM

| A | C | D | E | F | G | H | I | K | L | M | N | P | Q | R | S | T | V | W | Y |
|-----|---|---|---|-----|---|-----|-----|---|-----|---|---|---|---|---|-----|-----|-----|---|-----|
| 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.5 | 0 | 0 |
| 0.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.4 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0.7 | 0 | 0 | 0 | 0 | 0.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.7 | 0.2 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.5 |

Step 1: Counting Sites

As explained before, sites are the instances of a SLiM. In this step, we count the number of sites for each SLiM in every protein sequences. For example, Fig. 9 shows three SLiMs found from a set of proteins, and Fig. 10 is a protein sequence named Q13838. According to three known SLiMs, we can point out all the sites for each SLiM, which is the number of this motif. As shown in Fig. 9, we find out all the sites of motif 7, and mark them in red. In the same way, Patterns in purple are sites of motif 8, while green short chains are sites of motif 10. In this case, the score of motif 7 for protein Q13838 is set as 3, motif 8 is set as 2, and the number of motif 10 is 3. Assume that there are 10 SLiMs extracted, a matrix like Table 6 can be generated. Each row means a protein, while every column represents a SLiM.

Step 2: Scoring Sites using *I*-Formula

Rueda et al. [28] proposed *I*-formula in order to calculate *I*-score as the attributes of predict obligate and non-obligate protein interaction complexes, and get a satisfying

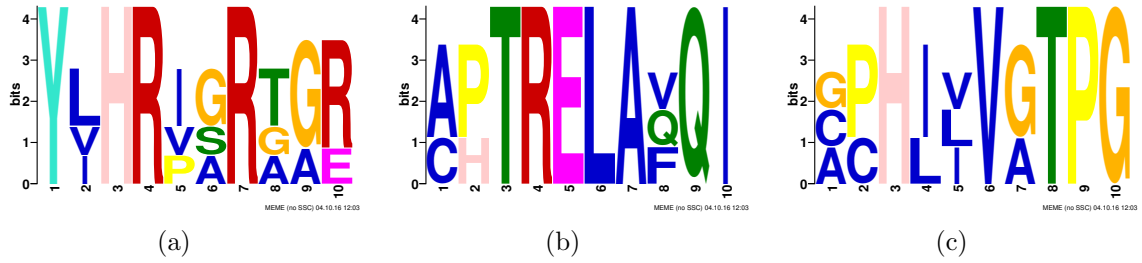


Fig. 9: (a) Motif 7, (b) Motif 8, (c) Motif 9

>Q13838

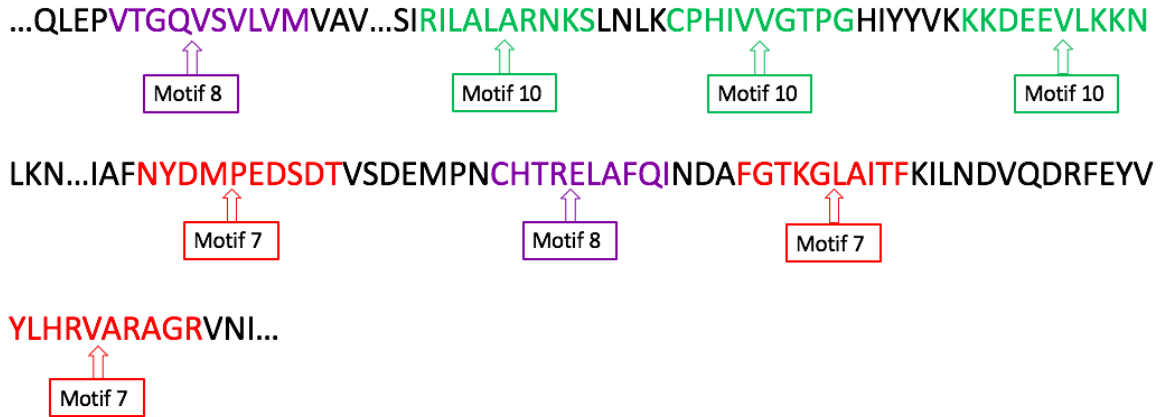


Fig. 10: Example of counting sites

result where the accuracy is more than 99%. Assume that given a protein sequence X of length L and a SLiM m of length l with n sites, the I -score of SLiM m for protein X is calculated according to Equation (1). In this formula, a_i means the i^{th} site of SLiM m , while a_{ij} is the j^{th} amino acid in site a_i . In this case, $P(a_{ij})$ is the probability of amino acid a_{ij} occurring on the j^{th} position of site a_i , which can be obtained from PSPM. Different length of SLiMs will affect the value, which may lead

Table 6: Counting sites matrix

| | SLiM 1 | ... | SLiM 7 | SLiM 8 | SLiM9 | SLiM10 |
|--------|--------|-----|--------|--------|-------|--------|
| Q13838 | 0 | 0 | 3 | 2 | 0 | 3 |
| ... | | | | | | |

to an unfair comparison. It means that a higher value may be caused by a larger size of SLiM since it is calculated by sum of values about probabilities of amino acids in each site. The more amino acids contained in a site, the larger score it may be. However, every SLiM is an equal attribute without any priority. To make it fair, it should be divided by l which is the length of a SLiM.

$$I(m|X) = -\frac{1}{l} \times \sum_{j=1}^l P(a_{ij}) \times \log(P(a_{ij})) \quad (1)$$

Since $0 \leq P(a_{ij}) \leq 1$, $\log(P(a_{ij}))$ is negative, the higher probability a SLiM is, the lower score we will get. To make it more meaningful, a negative sign is used in front of the equation. Besides, $\log(P(a_{ij})) = 0$ when $P(a_{ij}) = 1$, which is meaningless. To solve this problem, a regulation of $P(a_{ij})$ threshold is defined as follow:

$$\log(P(a_{ij})) = \begin{cases} \log(1 - \varepsilon) & \text{if } P(a_{ij}) > 1 - \varepsilon \\ \log(P(a_{ij})) & \text{otherwise} \end{cases} \quad (2)$$

where $\varepsilon > 0$, and ε is a sufficiently small value, and set as 0.01 in our experiment.

Step 3: Averaging I -Score

The final score of a SLiM should also be divided by the number of its sites which is obtained in Step 1. Therefore, I formula is modified as below:

$$\hat{I}(m|X) = -\frac{1}{n} \times \sum_{i=1}^n \left(\frac{1}{l} \times \sum_{j=1}^l P(a_{ij}) \times \log(P(a_{ij})) \right) \quad (3)$$

Thus, every SLiM for each protein sequence will get an I -score.

3.2.3 Sliding Window Score Approach

This method is proposed by [22], and found a new way to define sites rather than using the SLiMs extracted by MEME. Assume that given a protein sequence X of

length L , and the length of a potential site s is l . Suppose that there is a window which can show l amino acids, and place the window from the beginning of the protein sequence. The pattern shown in this window consisting of the first l amino acids is regarded as a potential site. It can be scored by the formula (4), where $P(s_i)$ means the probability of i^{th} amino acid in site s occurring in the i^{th} position of a SLiM. The value of $P(s_i)$ can be obtained from PSPM in MEME output file. Then move the window to the next position, and score the pattern selected by the window in the same way. Repeat the same operation until the window slides to the end of the protein sequence X .

$$P(s|X) = \frac{1}{l} \times \sum_{i=1}^l P(s_i) \quad (4)$$

Next, we define a threshold λ . If $P(s|X)$ is larger than λ , site s is considered as a real site, and marked as a , otherwise, it is not a site. In this thesis, we set value of λ as 0.4, 0.45, 0.5 and 0.6 to see which performance is better. Fig. 11 shows that given a pattern of SLiM and its PSPM, how to find real sites using this method. Letters in red represent the pattern shown in the window.

The purpose of this method is to calculate the value of SLiMs for every protein, so after the score of each real site gotten, we need to add them together and divide it by the number of real sites to get an average. Suppose that there are n real sites of SLiM m in protein X , and a_i means the i^{th} real site. According to Equation (5), we can get the score of SLiM m .

$$P(m|X) = \frac{1}{n} \times \sum_{i=1}^n P(a_i|X) \quad (5)$$

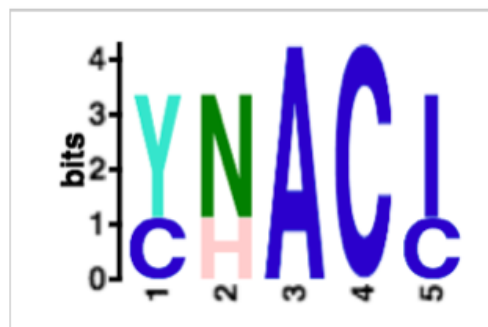
3.3 Drug Feature Representation

In this thesis, we use chemical substructure fingerprints [32][31][41] as drug feature descriptors. PubChem database defines 881 fingerprints, while Klekota and Roth [21] defines 4860 bits. Fig. 12 is a portion of fingerprints in PubChem, where the first

| Position | A | C | D | E | F | G | H | I | K | L | M | N | P | Q | R | S | T | V | W | Y |
|----------|---|-----|---|---|---|---|-----|-----|---|---|---|-----|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.6 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0.3 | 0 | 0 | 0 | 0 | 0.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0.3 | 0 | 0 | 0 | 0 | 0 | 0.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

>Q13838

...AYNACIACENB...



Step 1: AYNACIACENB



$$P(a|X) = \frac{1}{5} \times (0 + 0 + 0 + 0 + 0.3) < 0.6$$

NOT a site

Step 2: AYNACIACENB ...



$$P(a|X) = \frac{1}{5} \times (0.6 + 0.6 + 1 + 1 + 0.6) > 0.6$$

One site

Step 7: AYNACIACENB



$$P(a|X) = \frac{1}{5} \times (0 + 0 + 0 + 0 + 0)$$

NOT a site

Fig. 11: An example to show how to find real sites in sliding window score method ($\lambda = 0.6$)

column is position number starting from 0 and end with 880, and the second column is chemical structure. Klekota and Roth defines molecular fingerprints in a similar way, but more particularly. In the following part, we focus on PubChem to explain the principle, and Klekota and Roth is the same.

It is obvious that drug chemical formulas need to be gained first if we want to identify descriptors of drugs. Simplified molecular-input line-entry system (SMILES) is the chemical formula expression we used in this thesis. SMILES format can be searched on KEGG public database according to the drug codes in the dataset. Fig. 13 shows this process.

According to 881 drug fingerprints defined by PubChem, each drug can be encoded with 881 binary bits. If the substructure can be found in this compound, then set the value of this fingerprint as 1, otherwise, set it as 0. Many tools have been developed

| <u>Bit Position</u> | <u>Bit Substructure</u> |
|---------------------|-------------------------|
| 0 | >= 4 H |
| 1 | >= 8 H |
| 2 | >= 16 H |
| 3 | >= 32 H |
| 4 | >= 1 Li |
| 5 | >= 2 Li |
| 6 | >= 1 B |
| 7 | >= 2 B |
| 8 | >= 4 B |
| 9 | >= 2 C |
| 10 | >= 4 C |
| 11 | >= 8 C |
| 12 | >= 16 C |
| 13 | >= 32 C |
| 14 | >= 1 N |
| 15 | >= 2 N |
| 16 | >= 4 N |
| 17 | >= 8 N |
| 18 | >= 1 O |
| 19 | >= 2 O |
| 20 | >= 4 O |
| 21 | >= 8 O |
| 22 | >= 16 O |
| 23 | >= 1 F |
| 24 | >= 2 F |
| 25 | >= 4 F |

Fig. 12: The first 25 chemical substructure fingerprints in PubChem

to find molecular fingerprints in drugs. *rdck* package in *R* software is one of them. This package is a JAVA framework for chemoinformatics, and can be imported into *R* as the interface to CDK Libraries. It is developed to assess different kinds of chemical compound descriptors. It can output the corresponding position numbers whose substructures can be found in this drug based on SMILES format input. Fig. 14 is the *R* command and its output. We assign the SMILES expression to variable *smiles*, and set the type of database as '*pubchem*'. In this case, *R* prints the positions

Name in our dataset: D00021



Chemical name: L-phenylalanine



SMILES: C1=CC=C(C=C1)CC(C(=O)O)N

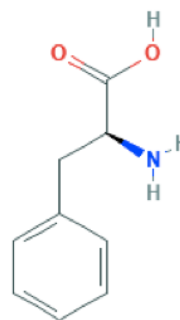




Fig. 13: An example of SMILES format searching process

whose fingerprints are satisfied.

Command in 

```
> smiles <- 'C[C@]12CC[C@H]3[C@H]([C@@H]1CC[C@H]2O)[C@@H]
(CCC4=C3C=CC(=C4)O)CCCCCCCCS(=O)CCCC(C(F)(F)F)(F)F'
> mols <- parse.smiles(smiles)
> fp <- get.fingerprint(mols[[1]], type='pubchem')
```

 Output:

```
name =
length = 881
folded = FALSE
source = CDK
bits on = 10 11 12 13 19 20 21 144 145 179 180 186 187 193 194 284 285 287 309 333 334 335 336
340 342 345 347 353 367 375 381 407 421 441 444 453 536 572 580 583 618 638 680 681 685 689 690
691 693 697 698 699 700 702 705 709 710 711 712 713 777 778 798 799 819 820 840 861
```

Fig. 14: Command in R and its output to identify drug descriptors

According to the output, we can create a matrix like Table 7, where each column represents a substructure position number, so there are 881 columns regarding PubChem. As we can see, the fingerprints on position 10, 11, 12, 13, 19 and 20 can be found in this drug, then set the value on these positions as 1.

Table 7: Drug Feature Matrix

| | 0 | ... | 10 | 11 | 12 | 13 | 14 | 15 | ... | 19 | 20 | 21 | ... | 880 |
|-------|---|-----|----|----|----|----|----|----|-----|----|----|----|-----|-----|
| Drug1 | 0 | | 1 | 1 | 1 | 1 | 0 | 0 | | 1 | 1 | 0 | | 0 |
| ... | | | | | | | | | | | | | | |

3.4 Negative Samples Selection

As mentioned before, reliable negative samples selection is a significant problem need to be solved before classification. Based on two strategies proposed by [33] and considering the specific problem in this thesis, we introduce a solution as follows.

Since interaction pairs between drugs and their target proteins are known, the pairs which not shown as interacted are considered as unknown samples. For each drug, we have known the features of its target proteins, and based on these features, the most different proteins can be fingered out as the non-interacted proteins in terms of this drug. To make it clear, suppose that there are m drugs and n

proteins in a dataset, and marked as drug $D = \{d_1, d_2, d_3, \dots, d_m\}$, target proteins $T = \{t_1, t_2, t_3, \dots, t_n\}$. If d_1 interacts with t_1, t_2 and t_3 , which belongs to set PT (positive target), then t_4, t_5, \dots, t_n are regarded as unknown state for d_1 , and such set is named as UT (unknown targets). The next step is to find out the proteins which have the largest differences between d_1 . Assume that there are k SLiMs as attributes for each protein, the protein properties $X = \{x_1, x_2, x_3, \dots, x_k\}$ where each element represents one attribute and consist of values for each protein of this feature. It means that $x_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}\}$ where x_{in} means the value of the i^{th} feature for the n^{th} protein.

To define the weight for each attribute can follow Equation (6).

$$W_i = \frac{mean(x_i)^2 / var(x_i)}{\sum_{j=1}^k (mean(x_j)^2 / var(x_j))} \quad (6)$$

where $mean(x_i)$ is the average of x_i :

$$mean(x_i) = \frac{1}{n} \times \sum_{j=1}^n x_{ij} \quad (7)$$

and $var(x_i)$ is its variance:

$$var(x_i) = \sum_{j=1}^n (x_{ij} - mean(x_i))^2 \quad (8)$$

Then the deviation of i^{th} protein t_i is calculate as follows.

$$\xi(t_i) = \sum_{j=1}^k (W_j \times \left| \frac{x_{ji} - mean(x_j)}{mean(x_j)} \right|) \quad (9)$$

The higher $\xi(t_i)$ is, the more deviation protein t_i has, and the further away from mean value. However, it can not illustrate that the most different point is in this position. Moreover, it should be compared with positive data according to Equation (10), where UT is a set storing unknown targets, while PT is a set of positive targets which means the proteins interacted with this drug.

$$P(t_i \in UT) = \left| \xi(t_i) - \text{mean} \left(\sum_{j \in PT} \xi(t_j) \right) \right| \quad (10)$$

After that, we sort the proteins based on the value of $P(t_i)$ from high to low, and selection preference is from high to low. To make it much fairer, we choose the same number of non-interacted proteins as target ones for every drug to maintain the same degree in positive and negative data. Next, the same operation can be done for each drug to find out the most likely non-interacted proteins for this drug, and vice versa. In terms of every protein, its targeted drugs can also be extracted, and using the same formulas based on the features of drugs. In the same way, the most different ones compared to the attributes of interacted drugs will be chosen, and the same amount of drugs can still be selected as interacted ones for this protein. For the rest of proteins, the same operations are repeated. After another negative set obtained, we combine all the negative pairs fingered out during these two screenings together. In this case, the number of negative data is twice larger than positive ones, which may cause bias. To avoid this situation, we randomly select the same number as positive ones from the combined set to be negative samples, and combine positive and negative pairs together as the dataset need to be used for classification.

3.5 Classification and Validation

We have tried several classifiers in this research, such as k Nearest Neighbours (k -NN), Random Forest (RF), and Support Vector Machine (SVM). Waikato Environment for Knowledge Analysis (Weka) is the tool used to do the classification and validation in this research. It integrates many classifiers including k -NN, RF and SVM, and also outputs results of evaluation.

3.5.1 k -NN

k Nearest Neighbours (k -NN) is one of the most fundamental and simple classification methods [27]. Assume that there are two classes, positive and negative. Given a test

sample, find out its k nearest known-label neighbours. If there are more samples belong to positive class among this k nearest neighbours, then regard this test sample as positive; Otherwise, predict it as negative. In this thesis, we use 1-NN method, which means that a test sample is predicted as the same class as its nearest training sample.

There is a general formula for distance calculation called Minkowski distance which is defined as Equation(11). d means dimension, which is the number of features.

$$d(x, y) = \sqrt[p]{\sum_{i=1}^d |x_i - y_i|^p} \quad (11)$$

When $p = 1$, Equation(11) is converted into Equation(12) which is called Manhattan distance.

$$d(x, y) = \sum_{i=1}^d |x_i - y_i| \quad (12)$$

When $p = 2$, Euclidean distance can be obtained as Equation (13). In this thesis, we use Euclidean distance to find the nearest neighbours for each sample.

$$d(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad (13)$$

3.5.2 Random Forest

Generally speaking, Random Forest (RF) combines the definition of bagging and decision tree together. It has an advantage in accuracy improvement, and does not need to reduce the dimensionality. As the name implies, ‘forest’ means many decision trees need to be built. Every time before building a decision tree, the training set is constructed by randomly sampling from the original dataset with replacement, until N samples chosen if N samples are needed. Samples unselected are regarded as a test set. Such sampling method is named as bootstrap sample. Decision tree may cause overfitting for the training set. To solve this problem, if there are M features totally, when a decision needs to be done, we can randomly choose m features of them where

$m \ll M$. Then choose one best feature from these m features to make a decision. We can use the same method to build many decision trees as predictors, and branches of these trees will not be pruned. Given a test sample, if it is classified as positive class by more decision trees, this sample is predicted as positive class. Otherwise, it is regarded as negative. This is like a voting system which depends on majority votes.

3.5.3 Support Vector Machine

Support vector machine (SVM) is a supervised learning method. The aim of SVM is to separate the training data by a hyperplane with the largest margin. It uses the concept of kernel instead of mapping data into a higher dimensional space. There are mainly three types of kernel, which are linear, polynomial, and radial basis function (RBF). In this thesis, we use RBF as the kernel of SVM. There are two parameters which affect the performance of this classifier, γ and c . γ is the parameter of RBF kernel, which controls the shape of separating function. To make it simple, it defines how far the influence of a single training example reaches. c means cost, which is a parameter for the soft margin cost function, and it indicates the level of punishment. A lower c value allows higher error on the training set by finding larger margin, while a higher c value will select few points within soft margin, and may cause overfitting. We can improve the performance by modifying the values of parameter γ and c .

3.5.4 mRMR Feature Selection

mRMR is short for minimum-redundancy maximum-relevancy, which is a feature selection method proposed by [26]. The purpose of feature selection is to choose a subset of relevant features instead of using all the features to build a model for classification. Feature selection is not the way to improve the performance, but to make model more efficient. It will take less time, avoid curse of dimensionality, and make the model simpler and more general. mRMR feature selection method considers both high relevant between features and class and less redundancy among features.

Suppose that S represents feature subset, c is class label, f means feature, and

$I(a, b)$ is the information between a and b. Relevancy can be computed as Equation (14).

$$D(S, c) = \frac{1}{|S|} \sum_{f_i \in S} I(f_i; c) \quad (14)$$

Redundancy is calculated as follows.

$$R(S) = \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i; f_j) \quad (15)$$

We can maximize Φ in order to make relevancy maximum and redundancy minimum.

$$\Phi(D, R) = D - R \quad (16)$$

3.5.5 Performance Evaluation

In order to validate the performance of different classifiers, we compare predicted results with known classes. Cross-validation is a commonly used evaluation method. The original dataset can be divided into two sets, training set and testing set. Training set is used to build a model, while testing set is regarded as an unknown-label set to do the prediction using this model. Since the labels of two classes have been known, the results can be evaluated by some statistics measures, such as specificity, sensitivity and accuracy, which are defined as Equation (17)(18)(19).

In this thesis, 10-fold cross validation is used as the evaluation method. First, randomly divide the whole dataset into ten equal parts. Then select the first part as a test set, while the rest nine sets are combined as a training set which is used to build a model. After using this model to predict the test set, values of those measures can be computed. Next, select the second part as a test set, while other nine parts as a training set, and do the same operation to get another set of measured values. Repeat the same work until every part are selected as a test set, and other nine sets are used to build a model to do the prediction by the same classifier of the same parameters. In this case, ten sets of measured values are obtained, and we can get

an average for each as the final score for specificity, sensitivity and accuracy of this classifier.

$$\textit{specificity} = \frac{TN}{TN + FP} \quad (17)$$

$$\textit{sensitivity} = \frac{TP}{TP + FN} \quad (18)$$

$$\textit{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

Assume that there are two classes, positive and negative. TP is the number of True Positive samples, and True Positive means the samples are labeled as positive (interaction) and also predicted as positive data. FP is the number of False Positive, and False Positive means the samples are predicted as positive samples, but labeled as negative in the original dataset, so this kind of samples are not real positive samples, and called as False Positive. In the same way, TN means the number of True Negative samples, which are predicted as negative and indeed negative in fact. FN is the number of False Negative data, which are positive ones indeed, but classified as negative. Specificity is a measure that validates true negative rate, while sensitivity is to evaluate true positive rate, and accuracy is an overall assessment considering both positive and negative prediction.

AUC is another important measure which means the area under ROC curve. The x -axis of ROC curve is False Positive rate, while y -axis is True Positive rate. The best situation is that all the predicted positive samples are true, in this case, the value of AUC is 1.

CHAPTER 4

Results and Discussion

4.1 Results

In this chapter, we list all the results of four gold standard datasets generated using different databases, methods, parameters, and classifiers. In terms of drug features, we have tried two databases: PubChem and Klekota and Roth, consisting 881 and 4860 fingerprints respectively. For scoring SLiMs, we use two methods: *I*-score and sliding window score (SWS) method, and the value of SWS threshold λ is set as 0.4, 0.45, 0.5 and 0.6 separately. Besides, three classifiers are used to do the prediction Random Forest (RF), *k*-Nearest Neighbours (*k*-NN), and Support Vector Machine (SVM). We set *k* as 1 concerning *k*-NN. Additionally, after trying different values of parameter γ and *c*, we find that when $\gamma = 0.01$, $c = 100$, the performances of four datasets are relatively better. Thus, we set γ as 0.01 and *c* as 100 for SVM in this thesis. Moreover, there are four statistics measures used in this chapter, which are AUC, accuracy, sensitivity and specificity.

Table 8 shows the results of Nuclear Receptor dataset using *k*-NN, RF and SVM classifiers. Multi-rows of ‘881 fingerprints’ represent the results using 881 chemical substructure fingerprints defined by PubChem database, while rows of ‘4860 fingerprints’ are the measured values generated using 4860 fingerprints by Klekota and Roth database. The row of ‘*I*-score’ shows the performance using *I*-score method scoring SLiMs, while ‘SWS’ is short for sliding window score approach. Multi-rows of SWS method are the results of different thresholds. As we can see from Table 8, the performance of *k*-NN is relatively worse than RF and SVM in general. The value of λ

effect the results in a certain degree, but not too much. Concerning Nuclear Receptor dataset, The best AUC value is 0.8764 when regard 4680 fingerprints as drug features and use SWS method with λ equals to 0.6 scoring SLiMs by RF, and its accuracy is also the highest.

Table 9 shows the results of GPCR dataset, and the performance is much better than Nuclear Receptor. The values of AUC are mostly higher than 0.9, while in terms of RF, AUC values are all more than 0.96. It is evident that the performance of k -NN is satisfied, but still not as excellent as RF and SVM. Additionally, the results generated using 881 fingerprints are a little bit better than using 4860 fingerprints for all classifiers. Moreover, it is interesting to note, the result of I -score performs the best compared with SWS for k -NN, while this method has the worst performance in SVM. In terms of AUC, RF is better than SVM, and the highest value occurs when using PubChem database and SWS method with λ equals to 0.5, which AUC is 0.9756. However, considering accuracy, SVM has higher scores, and the best value is 0.938 with sensitivity is 0.928 sensitivity and 0.948 specificity.

As we can see from Table 10, the performance is satisfied in general. The values of AUC for all conditions are mostly more than 0.9. Additionally, the results of RF and SVM is still better than k -NN. Although AUC of RF is higher than SVM, the accuracy of SVM is better. When $\lambda = 0.45$ in SWS, the accuracy reaches the highest for both 881 and 4860 fingerprints, which are 0.929 and 0.934 separately. The difference between these two types of fingerprints is not much in Ion Channel dataset.

The performance of Enzyme is shown as Table 11. Generally, AUC by k -NN is around 0.95, while the accuracy is mostly higher than 0.94. In terms of RF, the accuracy is more than 0.97, and AUC is up to 0.99. For SVM, the values of accuracy are the same as AUC, which are mostly around 0.97.

4.2 Comparison

To compare the results intuitively for each dataset, we draw a series of bar charts shown as Fig.15, and accuracy is used as the measure. Each bar graph compares

Table 8: Results of Nuclear Receptor dataset

| | | | AUC | Accuracy | Sensitivity | Specificity | |
|--------------|----------------------|---------|------------------|----------|-------------|-------------|-------|
| <i>k</i> -NN | 881 Fingerprints | I-score | | 0.7259 | 0.722 | 0.733 | 0.711 |
| | | SWS | $\lambda = 0.4$ | 0.6718 | 0.644 | 0.733 | 0.556 |
| | | | $\lambda = 0.45$ | 0.7231 | 0.661 | 0.744 | 0.578 |
| | | | $\lambda = 0.5$ | 0.7118 | 0.683 | 0.700 | 0.667 |
| | | | $\lambda = 0.6$ | 0.7248 | 0.711 | 0.700 | 0.722 |
| | 4860 Fingerprints | I-score | | 0.7078 | 0.678 | 0.722 | 0.633 |
| | | SWS | $\lambda = 0.4$ | 0.695 | 0.633 | 0.667 | 0.600 |
| | | | $\lambda = 0.45$ | 0.7331 | 0.678 | 0.633 | 0.722 |
| | | | $\lambda = 0.5$ | 0.7353 | 0.667 | 0.600 | 0.733 |
| | | | $\lambda = 0.6$ | 0.7096 | 0.639 | 0.500 | 0.778 |
| RF | 881 Fingerprints | I-score | | 0.8319 | 0.767 | 0.789 | 0.744 |
| | | SWS | $\lambda = 0.4$ | 0.8459 | 0.789 | 0.833 | 0.744 |
| | | | $\lambda = 0.45$ | 0.8276 | 0.789 | 0.833 | 0.744 |
| | | | $\lambda = 0.5$ | 0.8426 | 0.800 | 0.789 | 0.811 |
| | | | $\lambda = 0.6$ | 0.8764 | 0.800 | 0.789 | 0.811 |
| | 4860 Fingerprints | I-score | | 0.8119 | 0.739 | 0.733 | 0.744 |
| | | SWS | $\lambda = 0.4$ | 0.8141 | 0.750 | 0.789 | 0.711 |
| | | | $\lambda = 0.45$ | 0.7729 | 0.717 | 0.767 | 0.667 |
| | | | $\lambda = 0.5$ | 0.7910 | 0.744 | 0.744 | 0.744 |
| | | | $\lambda = 0.6$ | 0.7516 | 0.711 | 0.689 | 0.733 |
| SVM | 881 Fingerprints | I-score | | 0.7278 | 0.728 | 0.722 | 0.733 |
| | | SWS | $\lambda = 0.4$ | 0.7000 | 0.700 | 0.756 | 0.644 |
| | | | $\lambda = 0.45$ | 0.7500 | 0.750 | 0.767 | 0.733 |
| | | | $\lambda = 0.5$ | 0.7500 | 0.750 | 0.744 | 0.756 |
| | | | $\lambda = 0.6$ | 0.7778 | 0.778 | 0.811 | 0.744 |
| | 4860 Fingerprints | I-score | | 0.7889 | 0.789 | 0.744 | 0.833 |
| | | SWS | $\lambda = 0.4$ | 0.7056 | 0.706 | 0.733 | 0.678 |
| | | | $\lambda = 0.45$ | 0.7389 | 0.739 | 0.767 | 0.711 |
| | | | $\lambda = 0.5$ | 0.7333 | 0.733 | 0.733 | 0.733 |
| | | | $\lambda = 0.6$ | 0.7222 | 0.722 | 0.756 | 0.689 |

Table 9: Results of GPCR dataset

| | | | AUC | Accuracy | Sensitivity | Specificity | |
|--------------|----------------------|---------|------------------|----------|-------------|-------------|-------|
| <i>k</i> -NN | 881 Fingerprints | I-score | | 0.9159 | 0.916 | 0.934 | 0.898 |
| | | SWS | $\lambda = 0.4$ | 0.9177 | 0.917 | 0.934 | 0.901 |
| | | | $\lambda = 0.45$ | 0.9217 | 0.921 | 0.928 | 0.915 |
| | | | $\lambda = 0.5$ | 0.9183 | 0.918 | 0.920 | 0.917 |
| | | | $\lambda = 0.6$ | 0.9191 | 0.916 | 0.915 | 0.917 |
| | 4860 Fingerprints | I-score | | 0.8874 | 0.887 | 0.902 | 0.871 |
| | | SWS | $\lambda = 0.4$ | 0.9048 | 0.903 | 0.918 | 0.888 |
| | | | $\lambda = 0.45$ | 0.9199 | 0.919 | 0.918 | 0.920 |
| | | | $\lambda = 0.5$ | 0.8906 | 0.890 | 0.896 | 0.883 |
| | | | $\lambda = 0.6$ | 0.9004 | 0.894 | 0.891 | 0.896 |
| RF | 881 Fingerprints | I-score | | 0.9699 | 0.919 | 0.937 | 0.901 |
| | | SWS | $\lambda = 0.4$ | 0.9703 | 0.920 | 0.946 | 0.893 |
| | | | $\lambda = 0.45$ | 0.9739 | 0.928 | 0.939 | 0.917 |
| | | | $\lambda = 0.5$ | 0.9756 | 0.934 | 0.942 | 0.926 |
| | | | $\lambda = 0.6$ | 0.9733 | 0.942 | 0.942 | 0.942 |
| | 4860 Fingerprints | I-score | | 0.9614 | 0.915 | 0.923 | 0.907 |
| | | SWS | $\lambda = 0.4$ | 0.9664 | 0.922 | 0.928 | 0.917 |
| | | | $\lambda = 0.45$ | 0.9689 | 0.920 | 0.928 | 0.912 |
| | | | $\lambda = 0.5$ | 0.9634 | 0.921 | 0.926 | 0.917 |
| | | | $\lambda = 0.6$ | 0.9612 | 0.924 | 0.918 | 0.929 |
| SVM | 881 Fingerprints | I-score | | 0.9205 | 0.920 | 0.920 | 0.921 |
| | | SWS | $\lambda = 0.4$ | 0.9236 | 0.924 | 0.926 | 0.921 |
| | | | $\lambda = 0.45$ | 0.9370 | 0.937 | 0.942 | 0.932 |
| | | | $\lambda = 0.5$ | 0.9354 | 0.935 | 0.931 | 0.940 |
| | | | $\lambda = 0.6$ | 0.9378 | 0.938 | 0.928 | 0.948 |
| | 4860 Fingerprints | I-score | | 0.9118 | 0.912 | 0.902 | 0.921 |
| | | SWS | $\lambda = 0.4$ | 0.9181 | 0.918 | 0.918 | 0.918 |
| | | | $\lambda = 0.45$ | 0.9213 | 0.921 | 0.913 | 0.929 |
| | | | $\lambda = 0.5$ | 0.9244 | 0.924 | 0.922 | 0.924 |
| | | | $\lambda = 0.6$ | 0.9134 | 0.913 | 0.902 | 0.924 |

Table 10: Results of Ion Channel dataset

| | | | AUC | Accuracy | Sensitivity | Specificity | |
|-----------------|----------------------|-----------------|------------------|----------|-------------|-------------|-------|
| <i>k</i> -NN | 881 Fingerprints | I-score | | 0.9209 | 0.897 | 0.883 | 0.911 |
| | | SWS | $\lambda = 0.4$ | 0.8940 | 0.893 | 0.888 | 0.898 |
| | | | $\lambda = 0.45$ | 0.9042 | 0.899 | 0.896 | 0.903 |
| | | | $\lambda = 0.5$ | 0.9146 | 0.899 | 0.887 | 0.911 |
| | | $\lambda = 0.6$ | 0.9185 | 0.902 | 0.909 | 0.894 | |
| | 4860 Fingerprints | I-score | | 0.9232 | 0.897 | 0.884 | 0.909 |
| | | SWS | $\lambda = 0.4$ | 0.8904 | 0.887 | 0.892 | 0.881 |
| | | | $\lambda = 0.45$ | 0.9069 | 0.901 | 0.896 | 0.907 |
| $\lambda = 0.5$ | | | 0.9205 | 0.904 | 0.888 | 0.921 | |
| | $\lambda = 0.6$ | 0.9175 | 0.903 | 0.910 | 0.896 | | |
| RF | 881 Fingerprints | I-score | | 0.9523 | 0.888 | 0.872 | 0.904 |
| | | SWS | $\lambda = 0.4$ | 0.9475 | 0.874 | 0.888 | 0.860 |
| | | | $\lambda = 0.45$ | 0.9572 | 0.892 | 0.881 | 0.902 |
| | | | $\lambda = 0.5$ | 0.9542 | 0.888 | 0.874 | 0.901 |
| | | $\lambda = 0.6$ | 0.9639 | 0.907 | 0.899 | 0.915 | |
| | 4860 Fingerprints | I-score | | 0.9586 | 0.900 | 0.871 | 0.928 |
| | | SWS | $\lambda = 0.4$ | 0.9417 | 0.865 | 0.844 | 0.886 |
| | | | $\lambda = 0.45$ | 0.9588 | 0.898 | 0.875 | 0.921 |
| $\lambda = 0.5$ | | | 0.9573 | 0.899 | 0.877 | 0.921 | |
| | $\lambda = 0.6$ | 0.9623 | 0.913 | 0.904 | 0.921 | | |
| SVM | 881 Fingerprints | I-score | | 0.8852 | 0.885 | 0.910 | 0.860 |
| | | SWS | $\lambda = 0.4$ | 0.9194 | 0.919 | 0.928 | 0.911 |
| | | | $\lambda = 0.45$ | 0.9289 | 0.929 | 0.931 | 0.927 |
| | | | $\lambda = 0.5$ | 0.9119 | 0.912 | 0.915 | 0.909 |
| | | $\lambda = 0.6$ | 0.9160 | 0.916 | 0.911 | 0.921 | |
| | 4860 Fingerprints | I-score | | 0.8916 | 0.892 | 0.904 | 0.879 |
| | | SWS | $\lambda = 0.4$ | 0.9123 | 0.912 | 0.913 | 0.911 |
| | | | $\lambda = 0.45$ | 0.9339 | 0.934 | 0.934 | 0.934 |
| $\lambda = 0.5$ | | | 0.9272 | 0.927 | 0.917 | 0.938 | |
| | $\lambda = 0.6$ | 0.9051 | 0.905 | 0.911 | 0.899 | | |

Table 11: Results of Enzyme dataset

| | | | AUC | Accuracy | Sensitivity | Specificity | |
|--------------|----------------------|---------|------------------|----------|-------------|-------------|-------|
| <i>k</i> -NN | 881 Fingerprints | I-score | | 0.9565 | 0.951 | 0.957 | 0.945 |
| | | SWS | $\lambda = 0.4$ | 0.9519 | 0.947 | 0.957 | 0.937 |
| | | | $\lambda = 0.45$ | 0.9665 | 0.955 | 0.956 | 0.953 |
| | | | $\lambda = 0.5$ | 0.9656 | 0.954 | 0.956 | 0.953 |
| | | | $\lambda = 0.6$ | 0.9587 | 0.949 | 0.955 | 0.942 |
| | 4860 Fingerprints | I-score | | 0.9460 | 0.943 | 0.950 | 0.936 |
| | | SWS | $\lambda = 0.4$ | 0.9355 | 0.933 | 0.954 | 0.912 |
| | | | $\lambda = 0.45$ | 0.9493 | 0.945 | 0.955 | 0.934 |
| | | | $\lambda = 0.5$ | 0.9571 | 0.949 | 0.958 | 0.941 |
| | | | $\lambda = 0.6$ | 0.9495 | 0.943 | 0.956 | 0.930 |
| RF | 881 Fingerprints | I-score | | 0.9914 | 0.977 | 0.974 | 0.980 |
| | | SWS | $\lambda = 0.4$ | 0.9913 | 0.972 | 0.965 | 0.979 |
| | | | $\lambda = 0.45$ | 0.9924 | 0.975 | 0.969 | 0.981 |
| | | | $\lambda = 0.5$ | 0.9901 | 0.975 | 0.967 | 0.982 |
| | | | $\lambda = 0.6$ | 0.9904 | 0.977 | 0.971 | 0.983 |
| | 4860 Fingerprints | I-score | | 0.9896 | 0.976 | 0.974 | 0.979 |
| | | SWS | $\lambda = 0.4$ | 0.9885 | 0.976 | 0.977 | 0.975 |
| | | | $\lambda = 0.45$ | 0.9900 | 0.976 | 0.972 | 0.974 |
| | | | $\lambda = 0.5$ | 0.9904 | 0.977 | 0.975 | 0.978 |
| | | | $\lambda = 0.6$ | 0.9884 | 0.975 | 0.975 | 0.976 |
| SVM | 881 Fingerprints | I-score | | 0.9719 | 0.972 | 0.963 | 0.980 |
| | | SWS | $\lambda = 0.4$ | 0.9742 | 0.974 | 0.969 | 0.980 |
| | | | $\lambda = 0.45$ | 0.9764 | 0.976 | 0.969 | 0.984 |
| | | | $\lambda = 0.5$ | 0.9754 | 0.975 | 0.967 | 0.984 |
| | | | $\lambda = 0.6$ | 0.9772 | 0.977 | 0.967 | 0.987 |
| | 4860 Fingerprints | I-score | | 0.9687 | 0.969 | 0.973 | 0.965 |
| | | SWS | $\lambda = 0.4$ | 0.9740 | 0.974 | 0.973 | 0.975 |
| | | | $\lambda = 0.45$ | 0.9719 | 0.972 | 0.972 | 0.972 |
| | | | $\lambda = 0.5$ | 0.9762 | 0.976 | 0.975 | 0.977 |
| | | | $\lambda = 0.6$ | 0.9750 | 0.975 | 0.974 | 0.976 |

different scoring SLiMs methods, classifiers and various types of fingerprints. The dark blue bars represent the accuracy using I -score method, yellow bars indicate the accuracy when $\lambda = 0.6$ in SWS method, grey bars means $\lambda = 0.5$, bars in orange are performances of $\lambda = 0.45$, and light blue ones are $\lambda = 0.4$. Generally speaking, different scoring methods have little influence on the final results. The performance of k -NN is quite satisfied, but relatively not as good as RF and SVM. Besides, the results of 881 fingerprints are a little better than 4860 ones. In this case, if we focus on the accuracy using 881 fingerprints, concerning RF, when the threshold is 0.6, the accuracies are the best under most circumstances. For SVM, $\lambda = 0.45$ is the best choice. However, the influence of different thresholds is not that much.

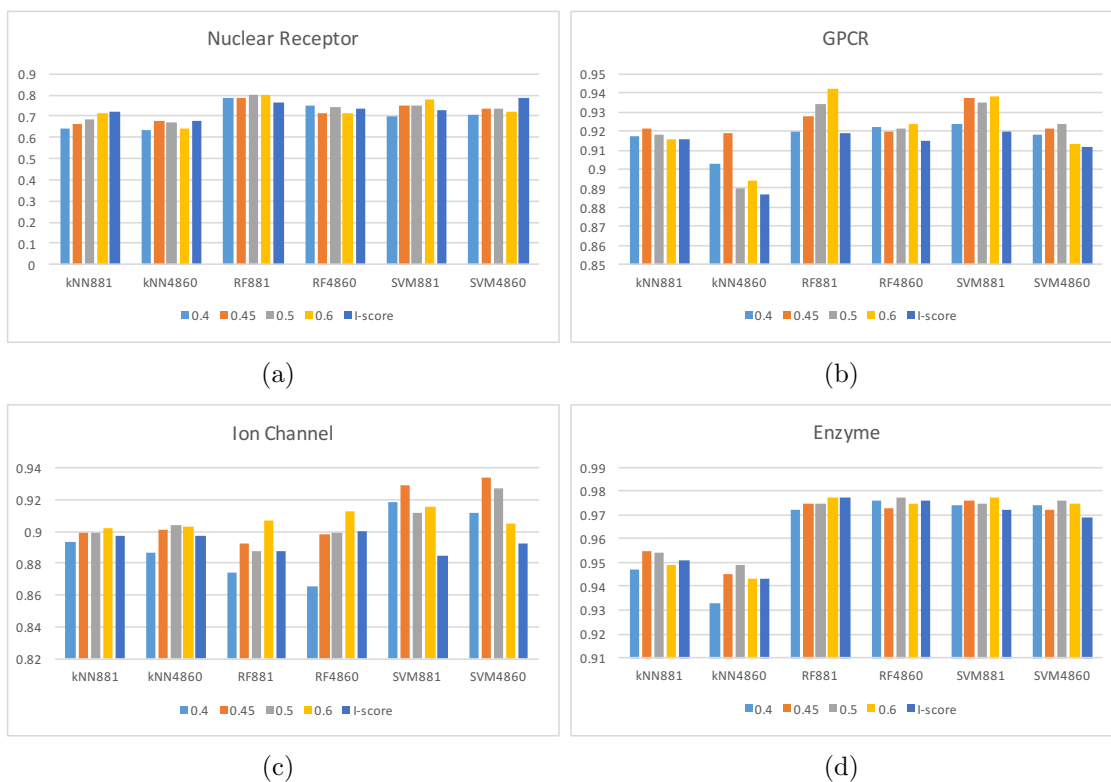


Fig. 15: Comparison among different SLiMs scoring methods, classifiers and different types of fingerprints for each dataset

We choose the performance by RF with 881 fingerprints as a condition comparing the results among different datasets. The performance of I -score is moderate, so we use I -score to do this comparison which is shown in Fig. 16. Obviously, the performance of Nuclear Receptor is relatively not as satisfied as others. This may caused

by less training samples. We believe that if enlarge the dataset, the performance will be better. The performance of Enzyme proves this hypothesis.

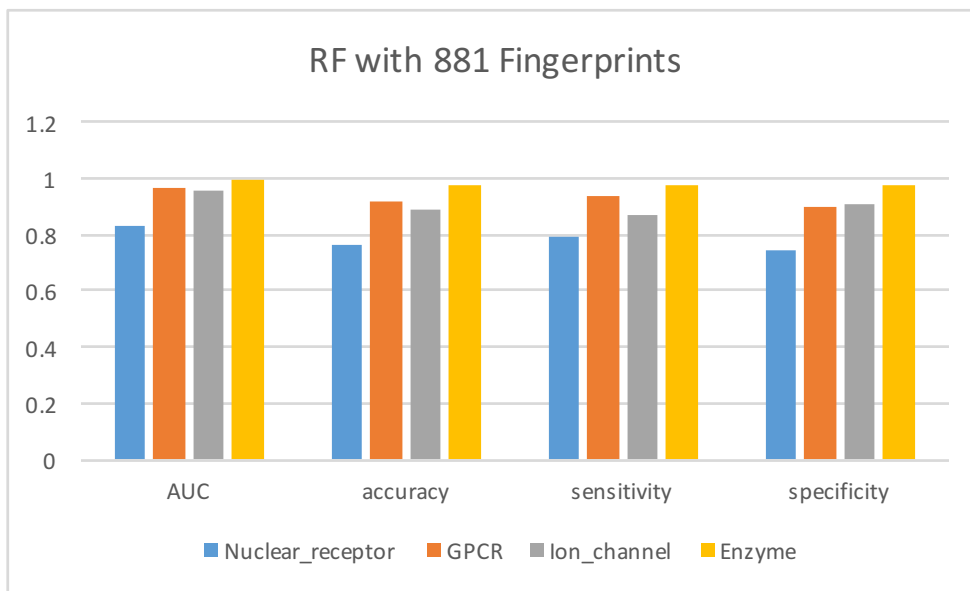


Fig. 16: Comparison among different datasets using RF with 881 fingerprints

Besides, we also want to compare the performance before mRMR feature selection and after it. We choose AUC as the measure to evaluate the performance, and use SVM with I -score and 881 fingerprints as the classifier. Table 12 shows the results of SVM using mRMR, while Fig. 17 compares AUC of SVM using all the features and feature subset selected by mRMR method. As we can see, AUC values become lower after mRMR. This is a normal situation because features support classifiers to do prediction. The results may be affected after reducing features. However, the aim of feature selection is not to improve the accuracy, but make models simpler, so that it may cost less time and avoid curse of dimensionality. Additionally, we can also find out the most important features using mRMR.

After comparing the performance among different sets and different methods, existing research comparison is also needed. We choose the AUC values generated by RF with 881 fingerprints and SWS method with $\lambda = 0.6$ as the result of our method. Table 13 lists the AUC values of some existing methods using the same gold standard dataset, which are Cao et al (2012) [6], Bigram-PSSM [24], Yamanishi et al. (2008) [39], Wang et al. (2010) [34], Yamanishi et al. (2010) [40], KBMF2K

Table 12: Results of mRMR feature selection applied on SVM with I -score and 881 fingerprints

| | Nuclear Receptor | GPCR | Ion Channel | Enzyme |
|-------------|------------------|--------|-------------|--------|
| AUC | 0.7222 | 0.9024 | 0.8032 | 0.8832 |
| Accuracy | 0.722 | 0.902 | 0.803 | 0.883 |
| Sensitivity | 0.756 | 0.915 | 0.781 | 0.972 |
| Specificity | 0.689 | 0.890 | 0.825 | 0.794 |

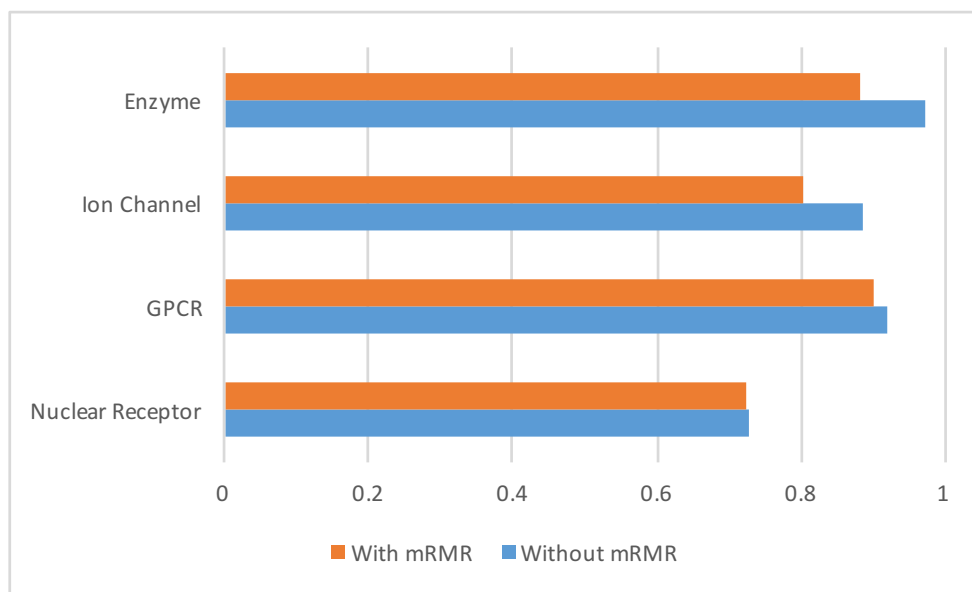


Fig. 17: Comparison of AUC between with and without mRMR

Table 13: The comparison of AUC among existing methods using benchmark datasets

| Algorithms | Enzyme | Ion Channel | GPCR | Nuclear Receptor |
|-------------------------|--------|-------------|--------|------------------|
| Proposed Method | 0.9904 | 0.9639 | 0.9733 | 0.8764 |
| Cao et al. (2012) | 0.9486 | 0.9428 | 0.8902 | 0.8822 |
| Bigram-PSSM | 0.948 | 0.889 | 0.872 | 0.869 |
| Yamanishi et al. (2008) | 0.904 | 0.851 | 0.899 | 0.835 |
| Wang et al. (2010) | 0.886 | 0.893 | 0.873 | 0.824 |
| Yamanishi et al. (2010) | 0.892 | 0.812 | 0.827 | 0.835 |
| KBMF2K | 0.832 | 0.799 | 0.857 | 0.824 |
| NetCBP | 0.8251 | 0.8034 | 0.8235 | 0.8394 |
| DBSI | 0.8075 | 0.8029 | 0.8022 | 0.7578 |

[15], NetCBP [7], and DBSI [9]. These existing methods are not only feature-based, but using the same dataset to solve the same problem, so they are comparable. The method proposed in this thesis outperforms the others regarding most datasets. In terms of Nuclear Receptor dataset, the AUC value of our method is the second best, which is also a satisfied result.

CHAPTER 5

Conclusion

We propose a new feature-based method to predict drug-target interaction which firstly introduces short-linear motifs (SLiMs) as protein features into this field. Different from n -gram, SLiMs have biological meanings, so that they can represent protein features better. There are two approaches to score SLiMs: I -score and sliding window score (SWS) which are calculated based on position-specific probability matrix (PSPM), in order to generate protein feature matrix. Concerning drug features, we select two kinds of chemical substructure fingerprints defined by PubChem and Klekota and Roth databases, then generate a drug feature binary matrix as the representation of drugs.

Another contribution of this research is to find a strategy to extract negative data from unknown samples. This point is often ignored by many previous studies, but necessary to be considered. We select negative samples by finding out drug-target pairs with the largest difference from known interacted samples, and also considering balanced degrees between positive and negative data to avoid bias.

After getting all the results, we find that the performance of RF and SVM is better than k -NN, and using the fingerprints defined by PubChem is the best choice. The influence of different values of threshold λ in SWS method is not much for the final results. Additionally, when $\lambda = 0.6$, the accuracy of RF classifier is relatively higher in most conditions. Besides, SVM with $\lambda = 0.45$ is also a good choice. Compared with the other existing study using the same dataset, our results are the best under most circumstance. It indicates that this method is efficient and reliable.

REFERENCES

- [1] Bailey, T. L., Bodén, M., Whittington, T., and Machanick, P. (2010). The value of position-specific priors in motif discovery using meme. *BMC bioinformatics*, 11(1):179.
- [2] Bailey, T. L. and Elkan, C. (1995). The value of prior knowledge in discovering motifs with meme. In *Ismb*, volume 3, pages 21–29.
- [3] Bharadwaja, A. (2014). Similarity based learning method for drug target interaction prediction.
- [4] Bleakley, K., Biau, G., and Vert, J.-P. (2007). Supervised reconstruction of biological networks with local models. *Bioinformatics*, 23(13):i57–i65.
- [5] Bleakley, K. and Yamanishi, Y. (2009). Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics*, 25(18):2397–2403.
- [6] Cao, D.-S., Liu, S., Xu, Q.-S., Lu, H.-M., Huang, J.-H., Hu, Q.-N., and Liang, Y.-Z. (2012). Large-scale prediction of drug–target interactions using protein sequences and drug topological structures. *Analytica chimica acta*, 752:1–10.
- [7] Chen, H. and Zhang, Z. (2013). A semi-supervised method for drug-target interaction prediction with consistency in networks. *PloS one*, 8(5):e62975.
- [8] Chen, X., Liu, M.-X., and Yan, G.-Y. (2012). Drug–target interaction prediction by random walk on the heterogeneous network. *Molecular BioSystems*, 8(7):1970–1978.

- [9] Cheng, F., Liu, C., Jiang, J., Lu, W., Li, W., Liu, G., Zhou, W., Huang, J., and Tang, Y. (2012). Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol*, 8(5):e1002503.
- [10] Davey, N. E., Haslam, N. J., Shields, D. C., and Edwards, R. J. (2010). Slimfinder: a web server to find novel, significantly over-represented, short protein motifs. *Nucleic acids research*, page gkq440.
- [11] Davey, N. E., Haslam, N. J., Shields, D. C., and Edwards, R. J. (2011). Slimsearch 2.0: biological context for short linear motifs in proteins. *Nucleic acids research*, 39(suppl 2):W56–W60.
- [12] Ding, H., Takigawa, I., Mamitsuka, H., and Zhu, S. (2014). Similarity-based machine learning methods for predicting drug–target interactions: a brief review. *Briefings in Bioinformatics*, 15(5):734–747.
- [13] Ezzat, A., Wu, M., Li, X.-L., and Kwoh, C.-K. (2016a). Drug-target interaction prediction via class imbalance-aware ensemble learning. *BMC Bioinformatics*, 17(19):267.
- [14] Ezzat, A., Zhao, P., Wu, M., Li, X., and Kwoh, C. K. (2016b). Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.
- [15] Gönen, M. (2012). Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, 28(18):2304–2310.
- [16] Günther, S., Kuhn, M., Dunkel, M., Campillos, M., Senger, C., Petsalaki, E., Ahmed, J., Urdiales, E. G., Gewiess, A., Jensen, L. J., et al. (2008). Supertarget and matador: resources for exploring drug-target relationships. *Nucleic acids research*, 36(suppl 1):D919–D922.
- [17] He, Z., Zhang, J., Shi, X.-H., Hu, L.-L., Kong, X., Cai, Y.-D., and Chou, K.-C.

- (2010). Predicting drug-target interaction networks based on functional groups and biological features. *PloS one*, 5(3):e9603.
- [18] Huang, R., Southall, N., Wang, Y., Yasgar, A., Shinn, P., Jadhav, A., Nguyen, D.-T., and Austin, C. P. (2011). The ncgc pharmaceutical collection: a comprehensive resource of clinically approved drugs enabling repurposing and chemical genomics. *Science translational medicine*, 3(80):80ps16–80ps16.
- [19] Jacob, L. and Vert, J.-P. (2008). Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156.
- [20] Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., Kawashima, S., Katayama, T., Araki, M., and Hirakawa, M. (2006). From genomics to chemical genomics: new developments in kegg. *Nucleic acids research*, 34(suppl 1):D354–D357.
- [21] Klekota, J. and Roth, F. P. (2008). Chemical substructures that enrich for biological activity. *Bioinformatics*, 24(21):2518–2525.
- [22] Li, Y. (2016). Prediction of high-throughput protein-protein interactions and calmodulin binding using short linear motifs.
- [23] Mordelet, F. and Vert, J.-P. (2008). Sirene: supervised inference of regulatory networks. *Bioinformatics*, 24(16):i76–i82.
- [24] Mousavian, Z., Khakabimamaghani, S., Kavousi, K., and Masoudi-Nejad, A. (2016). Drug-target interaction prediction from pssm based evolutionary information. *Journal of pharmacological and toxicological methods*, 78:42–51.
- [25] Okuno, Y., Tamon, A., Yabuuchi, H., Niijima, S., Minowa, Y., Tonomura, K., Kunimoto, R., and Feng, C. (2008). Glida: Gpcr-ligand database for chemical genomics drug discovery-database and tools update. *Nucleic acids research*, 36(suppl 1):D907–D912.

- [26] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238.
- [27] Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.
- [28] Rueda, L. and Pandit, M. (2014). A model based on minimotifs for classification of stable protein-protein complexes. In *Computational Intelligence in Bioinformatics and Computational Biology, 2014 IEEE Conference on*, pages 1–6. IEEE.
- [29] Schiller, M. R., Mi, T., Merlin, J. C., Deverasetty, S., Gryk, M. R., Bill, T. J., Brooks, A. W., Lee, L. Y., Rathnayake, V., Ross, C. A., et al. (2011). Minimotif miner 3.0: database expansion and significantly improved reduction of false-positive predictions from consensus sequences. *Nucleic acids research*, page gkr1189.
- [30] Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004). Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl 1):D431–D433.
- [31] Tabei, Y., Pauwels, E., Stoven, V., Takemoto, K., and Yamanishi, Y. (2012). Identification of chemogenomic features from drug–target interaction networks using interpretable classifiers. *Bioinformatics*, 28(18):i487–i494.
- [32] Tabei, Y. and Yamanishi, Y. (2013). Scalable prediction of compound-protein interactions using minwise hashing. *BMC systems biology*, 7(6):S3.
- [33] Wang, J. T., Liu, W., Tang, H., and Xie, H. (2014). Screening drug target proteins based on sequence information. *Journal of biomedical informatics*, 49:269–274.
- [34] Wang, Y.-C., Yang, Z.-X., Wang, Y., and Deng, N.-Y. (2010). Computationally probing drug-protein interactions via support vector machine. *Letters in Drug Design & Discovery*, 7(5):370–378.

- [35] Warr, W. A. (2009). Chembl. an interview with john overington, team leader, chemogenomics at the european bioinformatics institute outstation of the european molecular biology laboratory (embl-ebi). *Journal of computer-aided molecular design*, 23(4):195–198.
- [36] Wassermann, A. M., Geppert, H., and Bajorath, J. (2009). 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*, 49(10):2155–2167.
- [37] Wikipedia (2017). Sequence motif — wikipedia, the free encyclopedia. [Online; accessed 21-March-2017].
- [38] Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., and Hassanali, M. (2008). Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl 1):D901–D906.
- [39] Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., and Kanehisa, M. (2008). Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240.
- [40] Yamanishi, Y., Kotera, M., Kanehisa, M., and Goto, S. (2010). Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics*, 26(12):i246–i254.
- [41] Yamanishi, Y., Pauwels, E., Saigo, H., and Stoven, V. (2011). Extracting sets of chemical substructures and protein domains governing drug-target interactions. *Journal of chemical information and modeling*, 51(5):1183–1194.
- [42] Yu, H., Chen, J., Xu, X., Li, Y., Zhao, H., Fang, Y., Li, X., Zhou, W., Wang, W., and Wang, Y. (2012). A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data. *PloS one*, 7(5):e37608.
- [43] Yu, J., Guo, M., Needham, C. J., Huang, Y., Cai, L., and Westhead, D. R.

- (2010). Simple sequence-based kernels do not predict protein–protein interactions. *Bioinformatics*, 26(20):2610–2614.

VITA AUCTORIS

NAME: Wenxiao Xu

PLACE OF BIRTH: Tianjin, China

YEAR OF BIRTH: 1991

EDUCATION: Southwest University, B.Eng., Computer Science and
Technology, Chongqing, China, 2014

University of Windsor, M.Sc in Computer Science,
Windsor, Ontario, 2017