# LEARNING REPRESENTATIONS FOR DISCOVERING PATTERNS IN NETWORKS

HU PENGWEI

PhD

The Hong Kong Polytechnic University

2019

The Hong Kong Polytechnic University
Department of Computing

# Learning Representations for Discovering Patterns in Networks

Hu Pengwei

A thesis submitted in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy

October 2018

## CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

HU Pengwei

# ABSTRACT

A network is made up of a set of objects and their links. It can be represented as a graph, with vertices representing objects and edges representing links between objects. An algorithm capable of learning useful representation in the network can have many applications in many disciplines. For example, such a technique can be used to learn node representation in drug-target interaction networks for link prediction or to extract a discriminative graph representation for social network analysis. An appropriate representation of a network can make it easier to extract valuable patterns when performing such tasks as link classification or node clustering in a network. Suppose we want to learn a representation that makes density estimation easier. The distribution of more independence is easier to model. The most common methods are to use feature selection in conjunction with machine learning. However, these approaches these methods have the disadvantage that they do not consider all available heterogeneous information. For example, those approaches to learning representations in heterogeneous networks are expected to be optimized against high dimensionality and multimodality. Hence, there is a need for the development of algorithms that can learn representations retaining heterogeneous information carried by the network. Except for the integrity of the heterogeneous information, patterns are required to have specific explainable property in some studies. Prevalent approaches to learning network representations tend to pay more attention to network topology. So, we also need the representation learning algorithms that maintain interpretability on content features characterizing the nodes. In this thesis, we attempt to address this challenging issue by proposing effective approaches that are essential to a reliable framework for learning network representations for pattern discovery.

To transform data into a learnable form, we propose a multi-scale method to transform the raw data into the multi-scale representation, this preliminary step is to fully transform the data information. Then, we propose multi-scale feature deep representations inferring interactions (MFDR) to classify links in a network. MFDR use Auto-encoder as building blocks of deep networks to map high-dimensional features into low-dimensional space. As for learning representations from network, we concentrate on two categories that are integrated representation learning and interpretable representation learning. For integrated representation learning, we propose deep multiple networks fusion (DMNF), which is a novel graph clustering approach by learning latent representation from multi-networks. To perform the task, DMNF first constructs a network representing the degree of interrelationship between pairwise vertices by utilizing a fusion method. Given the fused network data, DMNF attempts to learn the latent network representation by making use of a deep neural network model. We also propose a new algorithm to predict unknown links from the fused representation through deep network fusion (DFNet). Given heterogeneous networks, DFNet implements a network completion method improves network confidence. For interpretable representation learning, we present GraphSE to learning significant subgraphs in graphs so that these subgraphs can be used for the link prediction task.. In particular application, given the attributed graphs, we can find a set of subgraphs that can be explained and can be used to predict whether a node can be linked to a specific target. In the clustering tasks mentioned above, few of latent network representation can be summarized. To address this challenge, we propose a novel latent representation model for community identification and summarization, which is named as LFCIS. To perform the task, LFCIS formulates an objective function that evaluating the overall clustering quality by taking into the consideration both edge topology and node features in the network. At last, we try to take a small step forward to solve the unbalanced link prediction problem. We adopt a support vector data description to learn the one-class data representation for summarizing small samples.

The approaches to data transformation, and the models for learning network representations presented in this thesis have been used in various real applications. In particular, we have applied them to drug-target interaction prediction, drug and side-effect (SE) link prediction and social network clustering. The experimental results show that the learned representations can improve the performance of traditional algorithms and outperform state-of-the-art approaches.

## Publications arising from the thesis

### Journal papers

[1] Yu-An Huang**, Peng-Wei Hu,** Keith C.C. Chan and Zhu-Hong You, "Graph convolution for predicting associations between miRNA and drug resistance using raw data" Submitted to Bioinformatics.

[2] **Peng-Wei Hu** and Keith C.C. Chan, "Machine Learning in Biomedicine: A Review of Computational Methods" Submitted to Current Medicinal Chemistry Journal. (Accepted)

[3] Tiantian He, Keith C.C. Chan and **Peng-Wei Hu**$^{*}$, "Learning Latent Factors for Community Identification and Summarization" IEEE Access Journal (2018).

[4] **Peng-Wei Hu**, Yu-An Huang, Keith C.C. Chan "Learning Multimodal Networks from Heterogeneous Data for Prediction of lncRNA–miRNA Interactions" Submitted to IEEE IEEE/ACM Transactions on Computational Biology and Bioinformatics (Minor revision)

[5] Hu, Lun, Xiaohui Yuan, **Peng-Wei Hu**, and Keith C.C. Chan. "Efficiently predicting large-scale protein-protein interactions using MapReduce." Computational Biology and Chemistry 69: 202-206 (2017).

[6] **Peng-Wei Hu**, Keith C.C. Chan, Yanxing Hu, "Predicting drug-target interactions based on small positive samples," Current Protein & Peptide Science, vol. 19, no. 5, 2018, pp. 479-487.

[7] You, Zhu-Hong, Keith C.C. Chan, and **Peng-Wei Hu**, "Predicting Protein-Protein Interactions from Primary Protein Sequences Using a Novel Multi-Scale Local Feature Representation Scheme and the Random Forest" PLOS One 10.5 (2015): e0125811.

**Conference papers**

[1] **Peng-wei Hu**, Shaochun Li, Shuhang Gu, Lun Hu, Keith C.C. Chan. "Inductive Matrix Completion for Network Fusion." Submitted to International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2019)

[2] **Pengwei Hu**, Eryu Xia, Shochun Li, Xin Du, Changsheng Ma, Jianzeng Dong, Keith C.C Chan, "Network-based Prediction of Major Adverse Cardiac Events in Acute Coronary Syndromes from Imbalanced EMR Data" Submitted to 17th World Congress of Medical and Health Informatics (Medinfo 2019)

[3] **Peng-Wei Hu**, Zhu-Hong You, Tiantian He, Shaochun Li, Shuhang Gu, and Keith C.C. Chan. "Learning Latent Patterns in Molecular Data for Explainable Drug Side Effects Prediction." IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2018 (Accepted)

[4] **Peng-Wei Hu**, Zhaomeng Niu, Tiantian He and Keith C.C. Chan, "Learning Deep Representations in Integrated Large Network for Social Community Identification." IEEE International Conference on Artificial Intelligence and Knowledge Engineering 2018

[5] **Peng-Wei Hu**, Yu-An Huang, Keith CC Chan, and Zhu-Hong You. "Discovering an Integrated Network in Heterogeneous Data for Predicting lncRNA-miRNA Interactions." In International Conference on Intelligent Computing, pp. 539-545. Springer, Cham, 2018.

[6] **Peng-Wei Hu**, Keith C.C. Chan, Lun Hu and Henry Leung "Discovering second-order sub-structure associations in drug molecules for side-effect prediction." In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 2250-2253. IEEE, 2017.

[7] **Peng-Wei Hu**, Keith C.C. Chan, Tiantian He and Henry Leung "Deep Fusion of Multiple Networks for Learning Latent Social Communities." In Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on, pp.

765-771. IEEE, 2017.

[8] **Peng-Wei Hu**, Keith C.C. Chan, and Tiantian He. "Deep Graph Clustering in Social Network." In Proceedings of the 26th International Conference on World Wide Web Companion, pp. 1425-1426. WWW 2017.

[9] Lun Hu, Xiaohui Yuan, **Peng-Wei Hu**, Keith C. C. Chan, “Parallel Identification of Variable-length Patterns for Large-scale Prediction of Protein-protein Interactions Using MapReduce Using MapReduce” Asia Pacific Bioinformatics Conference. APBC 2017.

[10] **Peng-Wei Hu**, Keith C.C. Chan and Zhu-Hong You, "Large-Scale Prediction of Drug-Target Interactions from Deep Representations" International Joint Conference on Neural Networks. pp. 1236-1243. IJCNN 2016.

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Professor Keith C. C. Chan. All the time, His unique insight into things and rich experience in solving difficult problems have helped me a lot. In spirit, his serious attitude towards work and spirit of exploration in research have set me the best example. Without his considerate guidance and consistent inspiration, this thesis would not have been completed.

I would like to express my thankfulness to Prof. Zhuhong You, who always inspires me by sharing his significant experience in academic and life. Also, I would like to thank all my friends and colleges: Prof. Henry Leung, Dr. Lun Hu, Dr. Tiantian He, Dr. Xin Luo, Dr. Wei Du, Dr. Peiyuan Zhou, Dr. Tingting Wang, Dr. Hao Chen, Mr. Ran Bi, Mr. Hui Su, Mr. Songran Liu, Mr. Weimin Luo, Mr. Yu-an Huang Mr. Yanxing Hu and Miss. Luyao Tan for the accompany and illuminating discussions in the last five years.

I would like to express my special thanks to my parents for their support and sacrifice for my education. Your love has motivated me to engage in scientific research. To them, I dedicate this thesis.

# Table of Contents

## List of Figures

## List of Tables

# 1. INTRODUCTION

Machine learning has proven to be a useful tool for resolving artificial intelligence and data mining issues over the years. When we use machine learning, our goal is to develop algorithms that can learn rules from raw data and to make accurate predictions about input data through models. In the real world, machine learning has successfully solved many problems. For example, it has been used in computer vision recognition, natural language understanding, a recommendation system and other fields. There are a lot of differences in the application of these fields, but a common issue with machine learning, regardless of the area of application, is that its performance depends on the quality of data representation. That is to say, learning representation is a critical step in promoting classification, clustering, and application tasks [1-2]. With more and more large-scale data available, the key issue of various machine learning applications has gradually shifted from optimization prediction to optimization representation. To meet this challenge, researchers adopted data preprocessing, feature selection and data fusion methods when deploying algorithms, to provide the best data representation for machine learning models. The study of this issue has always been highly valued, but with the development of significant data technology, the more and more data is becoming more and more challenging. In particular, existing representation algorithms have long sought to be able to extract and organize information that's easy to distinguish from data. Two data characteristics mainly cause these challenges: the scale of the data is getting larger and the value representation is hiding deeper and deeper; more and more sources of data have become available, and the data representation has become increasingly diverse. How to transform raw data into a representation that machine learning tools can handle is the first step in representation learning. These raw data indicate that they are usually recorded manually, require a significant amount of expert resources input, and are not well extended to the relevant fields. Some classical algorithms are useful in some monotonous applications, but they may not be able to cope with specific tasks. For example, in many areas such as artificial intelligence, bioinformatics, and recommendation systems, classical models always ignore some interesting domain knowledge [3-5]. In other words, learning the patterns of data and discovering the knowledge of existing values from the data, to increase the cognitive ability of the model, is a critical step in the deployment of the machine learning tool. Refer to real-world problems faced by the human, the limitations of machine learning is not set up their own cognitive systems from various prior knowledge, which makes progress in specific areas. A strong representation engineering has to be able to unlock the potential explanatory factors in the rough data. With this technology, machine learning can extend its applicability and usability. For example, in the process of automatic drug screening, such as the link between drug and protein prediction task, it is an essential requirement to represent the sequence data of proteins into data that can be directly calculated by machine learning tools. Again, the recommendation system needs first to explore representative data and use it to discover interesting communities. How to use multi-domain data to construct user maps and find interesting clusters from them has attracted a lot of attention recently. In this thesis, we attempt to start with the feature representation, then develop into an integrated representation learning method to combine heterogeneous data, finally try to explore the interpretative representation and propose interpretable graph representation solutions.

The rest of this section is organized as the following. In Section 1.1, the challenges existing in the state-of-the-art learning representations that may motivate us to propose more effective computational methods are illustrated. In Section 1.2, what kinds of problems need to be solved in learning representations are introduced. In Section 1.3, the algorithms that may address the challenges are introduced. We introduced the application prospects of the representation learning, especially in network link prediction and node clustering. Section 1.4, we give the organization of the thesis.

## 1.1. Motivation

As mentioned previously, many real-world applications require the first to extract computable data representation from different data sources. Therefore, the method of transforming data from a specific domain into a computable representation has attracted full attention. There are many ways to solve this problem. For example, in the techniques of biological sequences representation, the graphical representation for DNA sequences and normalized feature vectors [6-7] can well transform the biological sequence data into machine learning identifiable data. By feature representation, some algorithms may use attribute clustering and high order pattern combination to train the machine learning model. For example, attribute clustering [8] and high-order patterns discovering [9] can collect important new representation combinations based on the attributes related to features. Moreover, cognitive models should be able to extract comprehensive representations from heterogeneous data. For example, the potential broad representation [10-12] was found for the nodes in the graph and was used to detect the associations in the network, calculate the integrated representation of multiple modal data and predict the links. Although some existing algorithms have been used in specific applications, we find the following challenges in data representation, which may prompt us to develop more efficient algorithms.

First of all, the raw data of many objects contain treasure, such as biological sequences containing multiple functional annotations and location information, and the chemical 3D structure includes multi-dimensional information so that unwrought representations may lose some meaningful information. The new scheme should able to capture multi-scale local information by varying the representations of the object.

Second, though different techniques are used, most classical algorithms process raw data directly when performing prediction tasks, and these rough data may lead to the model being limited to the optimal solution of data in a particular region. For example, there are some algorithms predict drug and protein links in the network by using their expressions directly. This kind of solution may lead to losing a lot of meaningful representations in the training process.

Third, the current method of data representation focuses on homogeneous relations, which is to measure the relationship between entities with one kind of information. However, each node in the network usually has some heterogeneous expressions. By using heterogeneous expression, we can find more patterns in the network. How to explore the comprehensive representation of heterogeneous information becomes the factor that attracts us to study representation learning.

Fourth, though some algorithms can use the heterogeneous information to represent the node in the network, their solutions may determine that certain types of patterns may not be truly revealed. For example, [13] could represent drugs and predict interactions by integrating various drug features. This is a straight set of heterogeneous properties rather than a set of isomorphic representation. Moreover, the strength of the relationship between two nodes can only be calculated by the direct collection of heterogeneous properties when constructing graph representation. This way may also reduce the quality of the detected subgraph.

Fifth, more and more attention has been paid to the interpretability of results as well as to the accuracy of real-world applications. In other words, existing algorithms are difficult to implement if the relationship between the results and the newly discovered data representation cannot be explained, for example, the relationship between a molecular graph and drug reaction. In network-based tasks, it may be preferable to transform the data into the graph representation and look for interpretable subgraphs.

## 1.2. Problem statement of learning data representation

How to transform the raw data into a more suitable representation is the premise of machine learning modeling. A sophisticated representation learning framework consists of multiple parts that can solve different problems and combinations of components that can cover more complex issues. In such ways, we can find more patterns in the network from learned representations and further refine the link prediction and node clustering models. We break learning representation down into many different problems and give solutions. The basic problem is to learn feature representation first, which is defined as data forms that machine learning tools can read directly. Based on the need for a deeper understanding of the feature representation, the deep representation is a new expression with higher intrinsic relation to original feature representation after learning from deep neural network. The feature representation obtained by a single standard tends to ignore the information in other states, but multiple networks are established to identify the fused network that can be used as a comprehensive view. It should be pointed out that general deep learning cannot retain the interpretability of the patterns found, so many scenarios cannot use this algorithm. In this case, we will use high-order patterns and graph representation to try to solve the problem of interpretability. All the methods proposed in this thesis are intended to use different algorithms to optimize the data in a more reasonable representation.

To start with the illustration of the proposed method, we first introduce the following problems that will be tackled in this thesis:

*Feature Representation:* Let $S$ be the set of samples containing $|S|$ samples and each sample can be made up of $|F|$ different representation. That is to say, for each sample, its feature representation can be denoted as $s_i = \{fr_{i1}, fr_{i2}, fr_{i3} \ldots fr_{i|F|}\}$.

*Network Representation:* Let $G = \{V, E\}$ represent a network, where $V = \{u_1, u_2, \ldots, u_n\}$ is a set of $n$ vertices representing all the nodes in the network, and the $E \ = \{e_{ij}\}$ is the edge set containing the edges between pairwise vertices, and their values represent how similar these vertices are. To achieve a similarity matrix of modes, we assess nodes similarity by use of similarity calculation method based on the feature representation. That is to say, edge weights of the network are constructed by a $n \times n$ similarity matrix $L$ and $L(i, j)$ representing the similarity between nodes.

*Fused Network Representation:*

Traditional approaches do not allow multiple networks to be considered although the rapid development of techniques results in a growing diversity of network data. The multiple domain representations of the targets in the networks are usually ignored. Given *m* kinds of feature representations, we can construct *m* graphs$\{G_1, G_2, \ldots, G_m\}$ , the fused graph representation can be obtained by a network-based fusion scheme.

To interpret representation, we need to tackle the following problem that has not been addressed previously:

*Explainable Representation:* To obtain an interpretable representation of the entities, we should identify the relationship between each entity and each desirable representation. It allows highly related representations to each entity to be discovered and these representations can be seen as the interpretable attributes used to characterize each label in each sample of the entity.

## 1.3. Contributions

Given the challenges also the motivations mentioned, we propose to perform the task of learning representations using a series of solutions. In algorithm development, feature representation is one of the most critical components that significantly affect the performance of the computational model. To be able to use machine learning methods to predict links in raw format data, one of the most important challenges is how to adequately represent a sequential data by a fixed length feature vector in which the important information content of samples is fully encoded. A novel Multi-scale Local Descriptor (MLD) feature representation scheme is proposed to extract features from raw data. In the real-world applications, this scheme can capture multi-scale local information by varying the length of protein-sequence segments. Based on the MLD, an ensemble learning method, the Random Forest (RF) method, is used as a classifier. The MLD feature representation scheme facilitates the mining of link network from multi-scale continuous amino acid segments, making it easier to capture multiple overlapping continuous binding patterns within a protein sequence.

Then, we propose MFDR, which is a promising algorithm for predicting link in the network. MFDR use Auto encoder as building blocks of deep networks to reconstruct two kinds of feature representations to low-dimensional space. In a real-world application, we adopt large-scale drug chemical structures and target protein sequences to machine learning model predict if certain human protein link to a specific compound. Our approach is the first one that applies Stacked Auto encoder to represent large-scale drug-target interactive features for prediction.

And, we propose DMNF, which is a novel deep-model-based approach to learn latent structural representation from multi-domain data. DMNF can discover an aggregated deep representation, by taking into consideration the multiple networks, which represent heterogeneous information carried by the network data. To perform the task, DMNF first constructs a network representing the total degree of interrelationship between nodes. Here, we use a fusion method to compute such degree by taking into consideration various information embedded in the network data, e.g., node connection, and different kinds of properties. Given the fused network data, DMNF attempts to learn the latent network representation making use of a deep neural network model. Such learned representation can reveal the node cluster, e.g., social communities in the social network.

We also propose DFNet, which is an effective algorithm to introduce network fusion and matrix decomposition to identify the links in the network. Given complex heterogeneous networks, DFNet implements a network completion approach to network confidence increasing. Matrix factorization is used to complement similar networks and the integrated representation of multiple such networks is learned. This is a new model of a comprehensive view of learning that involves multiple networks, and most existing methods rely on one. The proposed model captures deep representations of a fused network that is able to generate deep relations contain all related domain information.

Also, we developed a GraphSE algorithm for interpretable representations discovering. Given a raw dataset, the GraphSE algorithm can learn interesting patterns among multi-labels, among features, and between multiple features and the labels. GraphSE performs its tasks by first computing an association measure to determine the significance of co-occurrence of each data and each specific label. Based on it, an attributed graph can be constructed for each task by defining a measure of attribute similarity based on a low-rank approximation scheme. Given the attributed graphs, we can discover in them a set of subgraphs that can be explainable and can be used to predict if a drug may lead to a certain SE using a Bayesian approach. Extensive experiments using real-world drug side-effect reports show that GraphSE can be potentially very useful. In the node clustering task, few of above latent network representation can be used to summarize the closely related nodes. To address this challenge, we propose a novel latent representation model for community identification and summarization, which is named as LFCIS. To perform the task, LFCIS firstly formulates an objective function that evaluating the overall clustering quality by taking into consideration both edge topology and node features in the network. In the objective function, LFCIS also adopts an effective component that ensures those vertices sharing with both similar local structures and features to be located in the same clusters.

Finally, scalable feature representation gives us the opportunity to express data more fully, and sufficient information also gives us the opportunity to predict links in unbalanced network data, using support vector data descriptions. Known links usually make up a small percentage of the population, and in many fields, especially in bioinformatics, negative samples carry a large number of potential positive samples. Existing approaches can be further improved to better prediction. To this end, we propose a new method to transform raw data into multi-scale representation and build a hyperplane model based on the known link sample. One primary task of our method is to discover association patterns between interacting drugs and proteins from the chemical structure and the protein sequence.

### 1.4. Thesis organization

To illustrate how we address the challenges mentioned, we organize the rest of the thesis as the following.

In Section 2, we present an overview of the previous works that are related to representation learning. These related works are categorized based on the link classification and node clustering in the networks.

In Section 3, how to transform original sequence information into multi-scale representation is introduced. This step can sufficiently transform sequence data, extract useful information, and provide sufficient computable representations for machine learning models.

In Section 4, how stacked auto-encoder transfers feature representations in the deep representations as a constrained optimization problem and how to solve the link classification and node clustering problem by using useful algorithms are presented. We propose DFNet to implements a network completion approach for drug-target link prediction. DMNF also be introduced to integrate multiple networks. It's network fusion algorithm with deep representation technique which we applied in the social network clustering. Besides, the experiments that may test the efficiency and effectiveness of proposed solutions and other baselines are presented.

In Section 5, we present the background under which we propose the algorithm GraphSE at first. Then, how they formulate the discovering of interpretable representations as a pattern mining problem, and the experiments which may test the efficiency and effectiveness of proposed algorithms and the compared baselines are introduced. Last, we present the algorithm LFCIS, which is an algorithm for identifying interesting sub-graphs making use of local information on topology and associated attribute values. Our approach may be sufficient for discovering communities in the network, and it's able to identify communities and summarize their features simultaneously. The details of the proposed algorithm and how to test the effectiveness of the proposed algorithm and other baselines, using the experiments related to the real application, i.e., social network community identification and summarization, are presented.

In Section 6, we also solve the data imbalance problem in the link prediction task. We present the background under which we propose the ODT to learn the one-class data representation at first. Then, we try to take a support vector data description with mutual information to discover the one-class data representation for summarizing small samples. Then we can take link prediction by modeling the one-class data representation. The details of the proposed algorithm and how to test the effectiveness of the proposed algorithm and other baselines, using the experiments related to the real application, i.e., drug and protein link prediction in a biological network, are presented.

At last, in Section 7, we summarize the significances of the thesis and propose future works.

# 2. OVERVIEW OF LITERATURE

Poor data quality has long been regarded as one fundamental threat to compromise the performance of standard classifiers, which would also lead to performance degradation for most graph clustering approaches. To learn the data representations, several algorithms have been proposed. Although there are many existing algorithms, these algorithms have different emphasis on link classification and node clustering respectively. So, the techniques used, and the scenarios used are different. In this section, the state-of-the-art related to link classification and node clustering are introduced respectively.

## 2.1 Traditional feature learning algorithms

The application of representation learning to reconstruct features is a critical step in optimizing prediction model. For machine learning models, the first step is usually to learn the transformation of data so that useful information can be extracted more easily when building learning functions. Feature selection is a straightforward way for handling high-dimensional data by removing partial data. Most existing feature selection methods can select feature in a pre-processing phase to convert original data into a lower-dimensional form. Then they convey calculable data scale to machine learning models. The focus of feature selection for machine learning is to select the significant subset of variables from the ultrahigh dimensional feature input which can efficiently describe the input data while reducing effects from noise, redundant or irrelevant variables. Recent years have witnessed some extend feature selection methods using optimum structure, sampling or geometric model to solve substantial high-dimensional challenges [14-15]. Take PCA (Principal Component Analysis) as an example, it is a classical linear algorithm for dimensionality reduction, which produces new attributes as linear combinations of the original variables. As a new component, there is usually keep orthogonal to each other and extract the largest variation. Some PCA algorithms released to address data scale concerns and be computed very efficiently using randomized algorithms to randomly projects the original large matrix down to lower dimensions [25]. Further, such problems will have chance result easily and get arbitrarily close to the optimal solution by using approximation algorithms. For some detail applications, the approximation will not work at all for precise target [16]. Approximation effectively plays dimensionality reduction like [179] adapted an approximate linear programming approach for finding L1 regularization. Group incremental approach to updating approximations under rough set also find new feature subset fast [180]. Another alternative way of handling high-dimensional biological data is sparse representation. The fundamental idea is to transform redundant data to a new less representation which obtained representation is the simplest possible [181]. Data can be approximated by a sparse linear combination of basis vectors [1]. Such kind of approaches has been widely used in the field of link classification and node clustering. Almost all tasks of sweep away large-scale redundant or irrelevant biomedical features can rely on variant feature selection [182-183].

## 2.2 Learning deep representations for link prediction

Learning representations for link prediction has been a longstanding open problem in classification. There are a lot of algorithms that can learn the representations of the data besides the shallow models mentioned above. One of the most common scenarios for link prediction is biochemical interaction analysis. The link prediction algorithm can be used in applications including drug-target interaction prediction, protein-protein interaction prediction, and drug-drug interaction prediction. Previously, the leading research area in link prediction is similarity-based approaches that use similarities to represent original features. Such similarity-based methods derived from self-similarities exploring. In [62], they used a similar score to describe a genomic space and pharmaceutical space. And then, they proposed a kernel regression approach to predict links in the drug-target interaction network. In [72], they used the drug and target similarities as the support SVM kernels and classified link twice and merge the experimental results to provide link predictions. KBMF2K firstly map the drug and target spaces to low-dimensional spaces similarity for link prediction [73]. In [74], they developed a network-consistency-based prediction method to predict links in drug-target interaction network, which rely on drug similarity network and the target similarity network integration. Some above previous works enjoyed very high prediction accuracy.

However, the similarity-based direction has an underlying problem that support data is not a direct biological expression. The similarity represents another dimension of original properties that may make experiment only achieve a high rate of errors regarding millions of candidates. Even more interesting is feature-based methods have been attempts to use a classifier to infer links adopt different encoding schemes impose different descriptors on the original features.

Deep learning is the broad term for algorithm has multiple layers aims to learn of feature representations in each layer that can be used to represent large-scale given data [1]. Therefore, a deep learning framework for unsupervised feature representation is attractive. The deep learning to data analysis emphasizes high volume and scalable models that promising chances of research into the automated extraction of complex data representations at high levels of abstraction [66]. It has a hierarchical architecture that develops several layers of units for feature extraction and transformation, then the supervised or unsupervised model of feature representations in each layer where higher-level features are defined regarding lower-level features [85]. As the first and most popular deep learning root, deep neural networks provide the noteworthy solutions to many data representation problems. Deep Learning built multi-layer architecture neural networks and trained with the greedy layer-wise unsupervised pre-training algorithms. According to [85], deep learning will keep valuable information after executing the training process. Deep neural network is applying the greedy layer-wise unsupervised pre-training mechanism that can reconstruct the original raw data set. We can learn valuable features with neural network instead of traditional features filtering method. Then, we can use classifier and obtain higher accuracy with better generalization from the learned features. This function makes deep representation like a good recipe for link prediction because of it able to extract complex representations from high volume unsupervised data. Some work has been used to predict the link, such as drug-protein [184] and drug-drug [185]. In [184], Wen et al. used deep belief network (DBN) to predict drug-target interactions. However, due to each link has two types of feature representation, these deep learning based approaches are not handled link prediction well.

## 2.3 Learning graph representations for node clustering

The trained features typically do not represent connectivity patterns of edges and relationships between the nodes. Such drawbacks lead to representations that are not particularly useful features for machine learning tools. Fortunately, above data representation algorithms such as deep representation also can be applied to graphs. It's a vivid way to represent data and output as graphs. Learning graph representation would typically be motivated to find recurrent substructures and identify them that best discriminate between the different classes. In the case of the molecular network, the number of nodes in the graph perspective may be massive. Such nodes are often corresponding to atoms, and the links are considered as bonds between the atoms. The similarity-based method can facilitate links prediction through the use of heterogeneous graph representations. For example, the random walk has been used on such a heterogeneous network [119]. They simulated a random walker's transition to uncover the association between drugs and targets. A recent work [173] that used network-based Laplacian regularized least square synergistic for drug combination prediction, and it was alleged that several types of information such as known synergistic drug combinations, unlabeled combinations, drug-target interactions, and drug chemical structures were integrated. We can achieve chemical compound classification by such techniques have been used to find the set of graph representations occurring in at least some given minimum support threshold of the given graphs [187-188]. A subgraph is also referred to as important means of identifying markedly interacting nodes from networks [8, 24].

Most algorithms related to graph analytics concerns the identification of communities. To identify the communities in the network data, there have been some so-called graph clustering algorithms proposed. Most of them can detect communities based on, not surprisingly some pre-defined measures on edge structure. One of the most widely used measures is modularity [153], which is defined as a function of the differences in density within communities and a null-graph in which nodes are randomly connected. Based on this measure, two approaches [154] [155] are proposed to detect communities making use of modularity maximization. Besides these algorithms, there are also other approaches proposed to detect graph communities, utilizing other topological measures.

For example, a clique-percolation based method is proposed in [156]. In [96], a clustering method called affinity propagation (AP) is proposed to detect clusters by maximizing the similarities of edge structure between candidates of cluster centers and other vertices. In [157], spectral clustering (SC) is proposed to detect communities in graphs by making use of normalized cut [158] which may reveal similar edge structures of vertices within the same communities. To reveal more meaningful communities in the network, there are some approaches proposed by taking into consideration both edge structure and attributes that may characterize the vertices. In [159] and [160], SA-Cluster and inc-Cluster are proposed to discover network community by making use a neighborhood random walk model in which the transition probability between each pair of vertices is evaluated by taking into the consideration edge density and attribute similarity. Though effective in discovering communities in network data, most of them cannot identify the representations that are able to characterize the discovered communities. To discover such representations, there are also some approaches that are able to discover graph communities by grouping vertices with similar attribute values. For example, there are some attempts making use of k-means algorithm [163] to group nodes with a higher similarity of attributes into the same clusters. In [164], an algorithm called MAC, which is based on a probabilistic generative model is proposed to discover graph clusters in which vertices are labeled with Boolean attribute values. In [165], a graph summarization algorithm called k-SNAP is proposed to detect graph clusters by grouping nodes into the same cluster according to a similarity measure of the attribute values. Though such algorithms may reveal the representations that may characterize the graph communities, they are not effective in discovering meaningful community structures as these methods ignore the edge structure of the network data.

# 3. FEATURE EXTRACTION AND PRE-PROCESSING

Representational learning is a new opportunity for machine learning, a technology that is often placed at the beginning of the entire model. One of the difficulties of learning is that it is hard to establish a clear goal. In link classification and node clustering tasks, the goal is obvious, that is, to minimize the error prediction number of training data sets. In the case of representing learning, we cannot define the goal, because we focus on traditional machine learning. The general case is to learn a predictor after using the new data representation, and an appropriate data representation can eliminate potential changes and improve the performance. At present, several methods have been used to find the optimal representation of data features. Although different algorithms can be used, these algorithms cannot be applied to all fields, nor can special attributes, especially sequence data, be considered. Since the algorithm cannot directly read the sequence information, the feasible method is to transform the sequence into a computable vector including the real number or the discrete vector set. In this section, the state-of-the-art related to extracting multi-scale feature representation in protein sequence are introduced.

## 3.1 Overview

Protein-protein interaction networks (PPI) are typical large-scale networks. Link prediction in such PPI networks is a significant problem in both machine learning and bioinformatics. Protein-protein interactions (PPIs) play a crucial role in various biological processes and functions in living cells, including metabolic cycles, DNA transcription and replication, and signaling cascades [17]. Instead of acting individually to perform functions, proteins do so by interacting with other proteins in cellular environments. The study of how proteins interact plays a critical role in investigating the molecular mechanisms behind biological processes. Moreover, it may also be able to discover unknown functions of proteins based on the known functions of those interacting with them. Recently, people have recognized the biological significance of the interacting protein, and therefore many people have begun to try to develop techniques capable of effectively predicting protein interactions.

Until now, how to extract meaningful features used to represent proteins is still challenging. The experimental methods are costly and time-consuming. Therefore, current PPI links obtained with experimental methods covers only a small fraction of the complete PPI networks [18]. Also, large-scale experimental methods usually suffer from high rates of both false positive and false negative predictions [19]. Hence, it is of great practical significance to develop the reliable computational methods to facilitate identification of PPI [20, 3].

Recently, a couple of methods which derive information directly from the amino acid sequence are of particular interest [6, 7, 3, 29, 35, 38–39]. Many researchers have engaged in the development of a sequence-based method for discovering new PPI [38, 42], and the experimental results showed that the information of amino acid sequences alone is sufficient to predict PPI [3,29,43]. Among them, one of the excellent works is an SVM-based method developed by Shen et al. [3]. In the study, the 20 amino acids are clustered into seven classes according to their dipoles and volumes of the side chains, and then the conjoint triad method abstracts the features of protein links based on the classification of amino acids. When applied to predict human PPI, this method yields a high prediction accuracy of 83.9%. Because the conjoint triad method cannot take neighboring effect into account, and the links usually occur in the discontinuous amino acids segments in the sequence, on the other work Guo et al. developed a method based on SVM and autocovariance to extract the interactions information in the discontinuous amino acids segments in the sequence [44]. Their method yielded a prediction accuracy of 86.55% when applied to predict Saccharomyces cerevisiae PPI. In some previous works, they also obtained good prediction performance by using autocorrelation descriptors and correlation coefficient, respectively [35, 45].

Also, some computational techniques have been proposed to provide either complementary information or supporting evidence to experimental methods [22–25]. Existing methods generally utilize different protein properties or origin, such as protein structure information [26, 27], protein domains, gene neighborhood, phylogenetic profiles, gene expression, to infer PPI interactions [17, 28-31]. However, these methods cannot be implemented if such pre-knowledge about the proteins is not available [33, 34]. Regarding the extraction of patterns from protein sequences, most of the existing algorithms intend to discover k-mers, each of which is an amino acid sequence segment with length k. These k-mers are then used to compose a feature vector for each protein sequence. Different examples of making use of k-mers to predict PPIs can be found in different kinds of literature. However, in recent years, with rapid development of high-throughput genomic technologies, the vast amount of PPI data makes it difficult to perform an efficient process of extracting variable-length k-mer patterns. Therefore, besides the effectiveness from the viewpoint of accuracy, it is becoming increasingly urgent that the factor of efficiency is another issue that should be taken into account when predicting large-scale PPIs.

In this study, a novel feature representation method for extraction of the protein sequence is proposed. We hypothesize that the contiguous amino acids segments with different segment lengths play an essential role in determining the links between proteins. In other words, the proposed protein representation method gives adequate consideration to mine the link information from multi-scale continuous amino acid segments at the same time. Thus, it can sufficiently capture multiple overlapping continuous binding patterns within a protein sequence.

To sum up, in this chapter we propose a novel multi-scale local descriptor (MLD) protein feature representation method. And, we combine this sequence-based approach with random forest (RF) model for the prediction of protein-protein interaction. To evaluate the performance, the proposed method is applied to Saccharomyces cerevisiae PPI dataset. The experiment results show that our approach achieved 94.72% prediction accuracy with 94.34% sensitivity at the precision of 98.91%. The prediction model is also assessed using the independent dataset of the Helicobacter pylori PPI and yielded 88.30% prediction accuracy, which further demonstrates the effectiveness of our method.

### 3.2 Method

In this section, we describe the proposed MLD-RF approach for predicting protein links from protein sequences. Our method to predict the PPI depends on two steps: (1) Represent protein sequences as a vector by using the proposed MLD feature representation; (2) RF predictor is used to perform protein interactions prediction tasks. In algorithm development, feature extraction is one of the most important components that significantly affect the performance of the computational model. To successfully predict PPI from protein sequences using machine learning, one of the most important computational challenges is how to effectively represent a protein sequence by a fixed length feature vector in which the important information content of proteins is fully encoded. Although researchers have proposed various sequence-based methods to predict new PPI, one flaw of them is that the interactions information cannot be drawn from multi-scale continuous amino acids segments with different segment lengths at the same time. To overcome this shortcoming, in this study we propose a novel MLD sequence representation approach to transform the protein sequences into feature vectors by using a binary coding scheme. A multi-scale decomposition technique is used to divide protein sequence into multiple sequence segments of varying length to describe overlapping local regions. Here, the continuous sequence segments are composed of residues which are local in the polypeptide sequence.

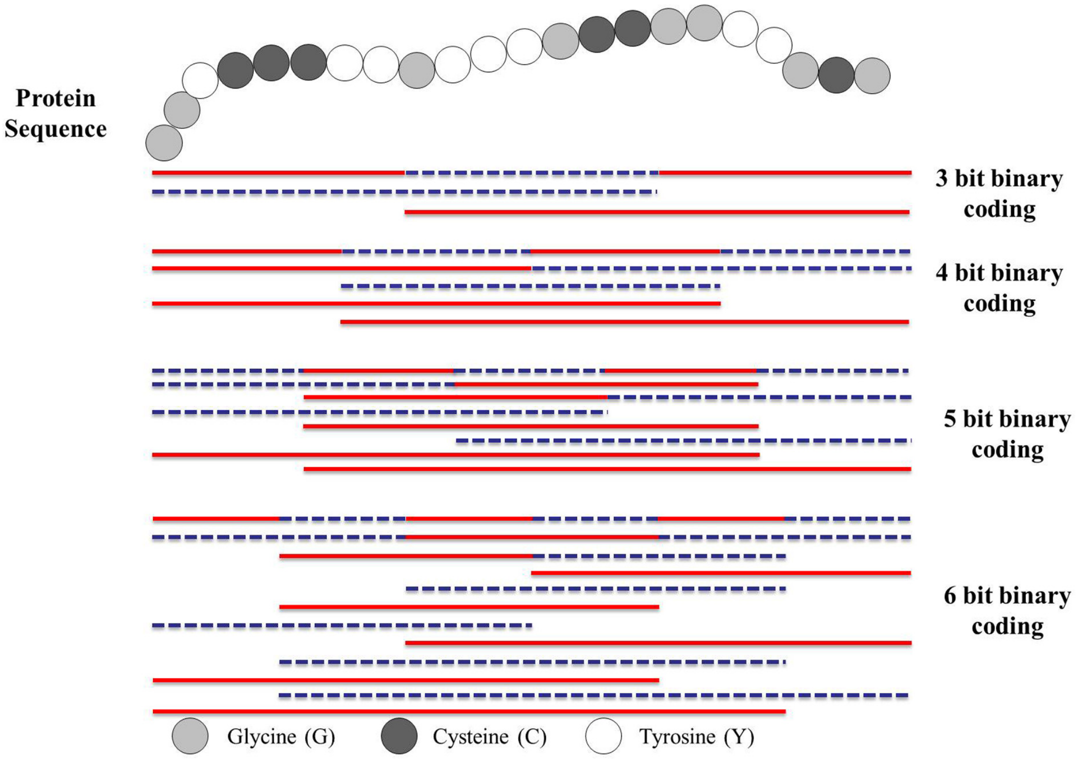


Fig. 1. The Schematic diagram for constructing multi-scale local descriptor regions for a hypothetical protein sequence using 3–6 bit binary form.

To extract the interaction information, we first divide the entire protein sequence into some equal length segments. Then a new binary coding scheme is adopted to construct a set of contiguous regions by the above partition. For example, consider a protein sequence “GGYCCCYYGYYYGCCGGYYGCG” containing 22 residues. To represent the sequence by a feature vector, let us first divide each protein sequence into multiple regions. Refer to Fig 1, the protein sequence is divided into four equal length segments (denoted by $S_1$, $S_2$, $S_3$ and $S_4$). Then it is encoded as a sequence of 1's and 0's of 4-bit binary form. In binary, these combinations are written as 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111. The number of states of a group of bits can be found by the expression $2^n$, where $n$ is the number of bits. It should be noticed that here 0 or 1 denote one of the four equal length region $S_1$—$S_4$ is excluded or included in constructing the continuous regions respectively. For example, 0011 denotes a continuous region constructed by $S_3$ and $S_4$ (the final 50% of the sequence). Similarly, 0111 represents a continuous region constructed by $S_2$, $S_3$ and $S_4$ (the final 75% of the sequence). These regions are illustrated in Fig 1. It should be noticed that the proposed feature representation method can be simply and conveniently edited at multiple scales, which offers a promising new way for addressing aforementioned difficulties in a simple, unified, and theoretically sound way to represent protein sequence. For a given number of bits, each protein sequence may take on only a finite number of contiguous regions. This limits the resolution of the sequence. If more bits are used for each protein sequence, then a higher degree of resolution is obtained. For example, if the protein sequence is encoded by 5-bit binary form, each protein sequence may take on 30 ($2^5$–2) different regions. Higher bit encoding requires more storage for data and requires more computing resource to process. In this study, only the continuous regions are used and the discontinuous regions are discarded.

For each continuous region, three types of descriptors, composition (C), transition (T) and distribution (D), are used to represent its characteristics. C is the number of amino acids of a particular property (e.g., hydrophobicity) divided by the total number of amino

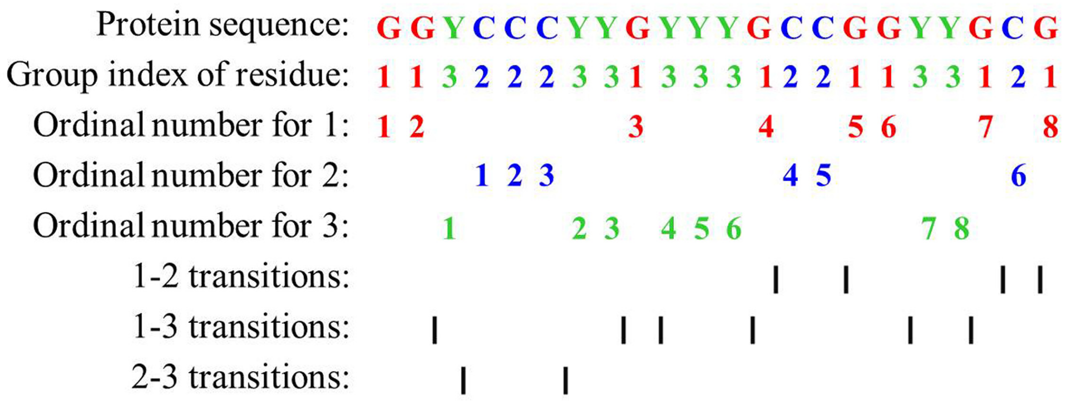

Protein sequence: G G Y C C C Y Y G Y Y Y G C C G G Y Y G C G
Group index of residue: 1 1 3 2 2 2 3 3 1 3 3 3 1 2 2 1 1 3 3 1 2 1
Ordinal number for 1: 1 2 3 4 5 6 7 8
Ordinal number for 2: 1 2 3 4 5 6
Ordinal number for 3: 1 2 3 4 5 6 7 8
1-2 transitions:
1-3 transitions:
2-3 transitions:

Fig. 2. Sequence of a hypothetic protein indicating the construction of composition, transition and distribution descriptors of a protein region.

acids in a local region.

T represents the percentage frequency of one characteristic amino acid after another. D measures the chain length within which the first, 25%, 50%, 75%, and 100% of the amino acids of a particular property are located, respectively [50].

The three descriptors can be calculated in the following ways. Firstly, to reduce the complexity inherent in the representation of the 20 standard amino acids, we firstly clustered them into seven groups based on the dipoles and volumes of the side chains. Amino acids within the same groups likely involve synonymous mutations because of their similar characteristics [3]. The amino ac49ids belonging to each group are shown in Table 1. Then, every amino acid in each protein sequence is replaced by the index depending on its grouping. For example, protein sequence “GGYCCCYYGYYYGCCGGYYGCG” is replaced by 1132223313331221133121 based on this classification of amino acids. There are eight ‘1’, six ‘2’ and eight ‘3’ in this protein sequence. The composition for these three symbols is 8/ (8+6+8) ×100% = 36.36%, 6/ (8+6+8) ×100% = 27.27% and 8/ (8+6+8) ×100% = 36.36%, respectively. There are 4 transitions from ‘1’ to ‘2’ or from ‘2’ to ‘1’ in this sequence, and the percentage frequency of these transitions is (4/21) ×100% = 19%. The transitions from ‘1’ to ‘3’ or from ‘3’ to ‘1’ in this sequence can similarly be calculated as (6/21) ×100% = 28.57%. The transitions from ‘2’ to ‘3’ or from ‘3’ to ‘2’ in this sequence can also similarly be calculated as (2/21) ×100% = 9.52%.

| Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 | Group 7 |
|---|---|---|---|---|---|---|
| A,G,V | C | M,S,T,Y | F,I,L,P | H,N,Q,W | K,R | D,E |

Table 1. Division of amino acids into seven groups based on the dipoles and volumes of the side chains.

For distribution D, there are 8 residues encoded as “1” in the example of Fig 2, the positions for the first residue ‘1’, the 2nd residue ‘1’ (25% × 8 = 2), the 4th ‘1’ residue (50% × 8 = 4), the 6th ‘1’ (75% × 8 = 6) and the 8th residue ‘1’ (100% × 8) in the encoded sequence are 1, 2, 13, 17, 22 respectively, so the D descriptors for ‘1’ are: (1/22) ×100% = 4.55%, (2/22) ×100% = 9.09%, (13/22) ×100% = 59.09%, (17/22) ×100% = 77.27%, (22/22)×100% = 100%, respectively. Similarly, the D descriptor for ‘2’ and ‘3’ is (18.18%, 18.18%, 27.27%, 63.64%, 95.45%) and (13.64%, 31.82%, 45.45%, 54.55%, 86.36%), respectively. For each continuous local region, the three descriptors (C, T and D) are calculated and concatenated, and a total of 63 descriptors are generated: 7 for C, 21 ((7×6)/2) for T and 35 (7×5) for D. Then, all descriptors from 9 regions (4 bit) are concatenated and a total 567-dimensional vector has been built to represent each protein sequence. Finally, the PPI pair is characterized by concatenating the two vector spaces of two individual proteins. Thus, an 1134-dimensional vector has been constructed to represent each protein pair and used as a feature vector for input into a prediction model.

Random Forest (RF) model is an ensemble classification algorithm that employs a collection of decision trees to reduce the output variance of individual trees and thus improves the stability and accuracy of classification. RF model is currently one of the most frequently employed machine learning techniques. RF takes advantage of two powerful machine-learning techniques: (1) the selection of training examples for each tree; (2) the random feature selection to split the data set. The first is performed by employing a bootstrap sample from original data (often referred to as bagging). Bagging works by sampling n samples with replacement from the original n samples, duplicating some examples and excluding some. The process results in two disjoint bags, one containing about 63.2% of examples of the training data and one bag containing the rest which is usually denoted as out-of-bag (OOB) examples. In general, a random forest is constructed using the in bag examples and the OOB examples are used to estimate its prediction performance. The second feature selection procedure works by sampling a small subset of features at each node in each classification tree. More specifically, at each node of a tree, RF randomly selected a constant number of features and the one with the maximum decrease in Gini index is chosen for the split when growing the tree.

The RF model construction is composed of two parts, the ensemble creation and the tree generation. Specifically, the model construction requires a set of examples S = ((($x_1$, $x_2$... $x_n$), $y$)...), where each example is described by a set of features X and a class label $y$; the number of trees to be constructed $T_n$; and the number of features to examine at each split Fn. In the ensemble creation step, $t$ = 1, 2..., $T_n$ trees are constructed from the in bag samples drawn with replacement from S. The tree construction algorithm starts by selecting $F_n$ random features which can reduce the Gini index most if split upon. If no feature is found that reduce the error, a leaf is created predicting the most probable class from those examples reaching the node. Otherwise, the data is partitioned into two subsets: those for which the feature is positive and those for which it is negative. The partitions are subsequently used to recursively build new trees, with edges from the previous node. The recursion continues until there is no more informative feature, the node is pure or the total number of examples at the current node is minor 2. In the experiment, we used the open source machine learning toolkit Weka to conduct this study.

## 3.3 Experiments with PPI datasets

### 3.3.1 Protein sequence and protein interaction dataset

To evaluate the performance of the proposed approach, we have used eight different PPI data sets in our experiment, two of which are S.cerevisiae, two are H. pylori, one is C.elegans, one is E.coli, one is H.sapiens, and one is M.musculus. The PPI dataset which was derived by Guo *et al.* [44], are used to build the first prediction model. The dataset was downloaded from S.cerevisiae core subset of the database of interacting proteins (DIP) [46]. After the protein links that contain a protein with fewer than 50 residues or have more than 40 percent sequence identity were removed, the remaining 5594 protein links formed the golden standard positive dataset (GSP). The construction of a negative PPI dataset is very important for training and evaluating prediction model. However, it is difficult to generate such a dataset because we have limited information about proteins that are non-interactive. Here, the negative dataset is generated by firstly selecting non-interacting link uniformly at random from the set of all proteins links that are not known to interact. Then the protein links with the same subcellular localization information are excluded. Finally, the golden standard negative dataset (GSN) consisted of 5594 protein links whose subcellular localization is different. By combining the above GSP and GSN datasets, the complete dataset contains 11188 protein links, where half are from the positive dataset and half from the negative dataset. Note that here we have used the same PPI dataset as used in Guo *et al.* [44].

However, some researchers argue that restricting negative examples to protein links localized in different cellular compartments is not appropriate for evaluating classifier accuracy [47, 48]. The use of such negative dataset for building a model can result primarily in predictions of protein co-localization [49]. The fact that interacting protein links have to be in the same place does not mean that all proteins in the same compartment will be interacting with each other. Therefore, we constructed the second PPI dataset by using positive samples from first PPI dataset, and following a simpler selection scheme—choosing negative examples uniformly at random—to construct the negative dataset. The second PPI dataset also consists of 11188 protein links, where half are from the positive dataset and half from the negative dataset.

The third PPI dataset is composed of 2916 Helicobacter pylori protein links (1458 interacting pair and 1458 non-interacting links) as described by Martin et al. [50]. Other five species-specific PPI dataset including C.elegans, E.coli, H.sapiens, M.musculus, and H.pylori are employed in our experiment to verify the effectiveness of the proposed method.

### 3.3.2 Evaluation measures

To measure the performance of the proposed method, we adopt five-fold cross-validation and a couple of validation measures in this study. These criteria are as follows: (1) the overall prediction accuracy (ACC) is the percentage of correctly identified interacting and non-interacting protein links and given by: $CC = \frac{TP+TN}{TP+FP+TN+FN}$ , the sensitivity (SN) is the percentage of correctly identified interacting protein links and given by:$SN = \frac{TP}{TP+FN}$ , the specificity (Spec) is the percentage of correctly identified non-interacting protein pairs and given by: $Spec = \frac{TN}{TN+FP}$, the positive predictive value (PPV) is the positive prediction value and given by: $PPV = \frac{TP}{TP+FP}$, the negative predictive value (NPV) is the negative prediction value and given by: $NPV = \frac{TN}{TN+FN}$, the F-score is a weighted average of the PPV and sensitivity, where an F-score reaches its best value at 1 and worst score at 0; The definitions are given as follows:$F_{score} = 2 \times \frac{SN \times PPV}{SN+PPV}$, the Matthew's correlation coefficient (MCC) is more stringent measure of prediction accuracy accounts for both under and over-predictions. Its definitions are given by: $MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN) \times (TN+FP) \times (TP+FP) \times (TN+FN)}}$, where true positive (TP) is the number of true PPIs that are predicted correctly; false negative (FN) is the number of true PPIs that are predicted to be non-interacting pairs; false positive (FP) is the number of true non-interacting pairs that are predicted to be PPIs, and true negative (TN) is the number of true non-interacting pairs that are predicted correctly.

| M | ACC(%) | SN(%) | Spec | PPV(%) | NPV(%) | F1 (%) | MCC(%) |
|---|---|---|---|---|---|---|---|
| 5 | 94.72±0.35 | 94.45±0.55 | 95.72±1.51 | 98.80±0.43 | 82.17±1.27 | 96.58±0.25 | 85.89±0.70 |
| 10 | 94.63±0.37 | 94.42±0.54 | 95.40±1.29 | 98.71 ±0.38 | 82.04±1.32 | 96.52±0.26 | 85.65±0.76 |
| 15 | 94.62±0.21 | 94.44±0.53 | 95.28±1.58 | 98.69±0.45 | 82.06±1.19 | 96.51±0.15 | 85.60±0.41 |
| 20 | 94.60±0.36 | 94.35±0.54 | 95.56±1.68 | 98.76±0.48 | 81,88±1.23 | 96.50±0.25 | 85.61 ±0.76 |
| 25 | 94.69+0.25 | 94.44±0.44 | 95.66±1.34 | 98.79±0.39 | 82.11 ±0.99 | 96.56±0.18 | 85.82±0.47 |
| 30 | 94.63±0.37 | 94.42±0.54 | 95.39±1.43 | 98.71 ±0.41 | 82.04±1.25 | 96.52±0.26 | 85.65±0.78 |

Table 2. The prediction performance for six testing datasets with various number of feature subsets M, where the tree size N is set to 60.

### 3.3.3 Experimental setting

In this chapter, the proposed sequence-based PPI predictor is implemented using MATLAB platform. All the simulations are carried out on a computer with 3.1 GHz 2-core CPU, 6 GB memory and Windows operating system. To achieve good experimental results, the corresponding parameters for random forest are firstly optimized. For RF model the parameters to be ascertained are the number of feature subset *M*, and the ensemble size *N*. The average prediction results for six testing datasets are listed in Table 2 by setting *M* to 5, 10, 15, 20, 25 and 30, respectively. It can be found that the performance under different conditions varies slightly, and none of the parameters take obvious advantage over the other ones. So there is no consistent relationship between the classification accuracy and feature subset *M*. In this study, the value of *M* is set to 10 in all experiments, which requires the relatively less computational cost. The average results with different ensemble sizes are shown in Fig 3. It can be found from Fig 3 that

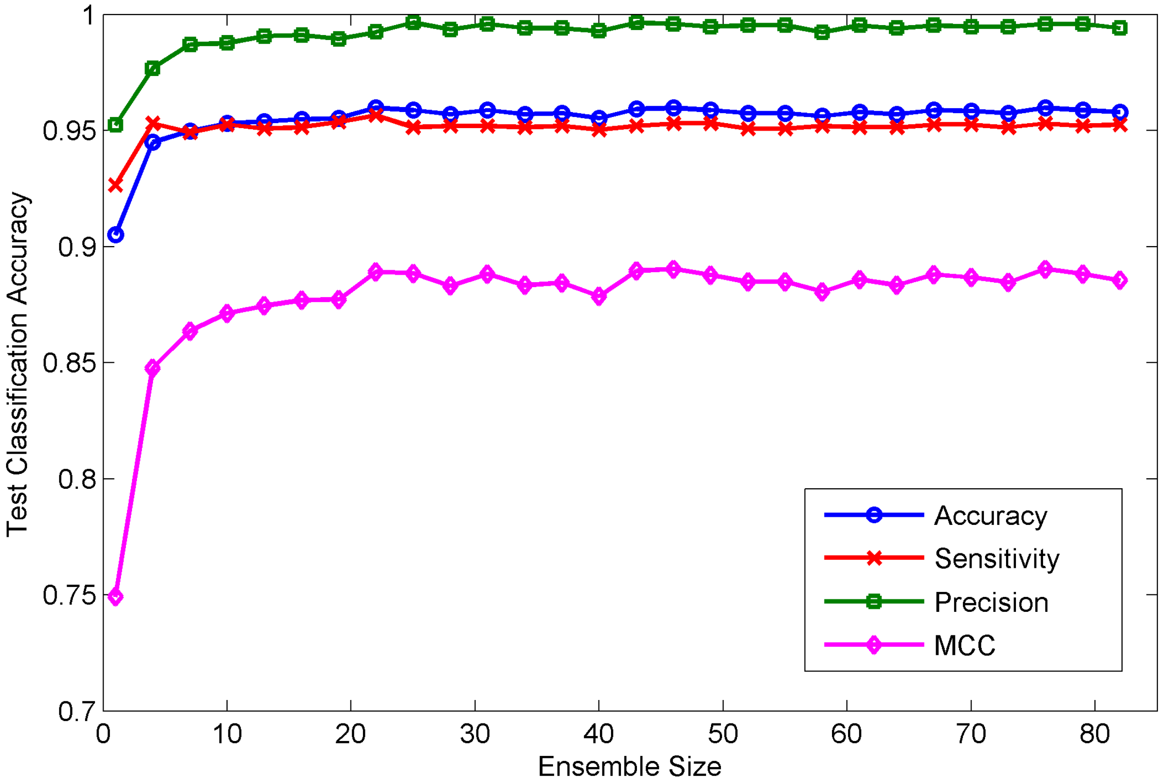


Fig. 3. The prediction performance for classification accuracy, sensitivity, precision and MCC of different tree size, where the number of feature subsets M is set to 10 and the unpruned decision tree is employed as the base classifier.

RF predictor performs well when only a few of base classifiers are employed. All the evaluation measures including average prediction accuracy, sensitivity, specificity, PPV, NPV and MCC keep improving with the ensemble size increase. However, the improvement becomes negligible when the ensemble size is larger than 10. From the above analyses, we can conclude that RF model is not sensitive to the choice of parameters. So for the H.pylori dataset, the parameters of the RF model do not need to be optimized again, assuming that they are set the same values as those adopted on the S.cerevisiae dataset.

### 3.3.4 Prediction performance of the proposed model

We evaluated the performance of the proposed model using the first PPIs dataset as investigated in Guo *et al.* [44]. In this experiment, we will guarantee the validity of the results and the predictive nature of the new data, the dataset is randomly partitioned into training and independent testing sets via five-fold cross-validation. Each of the five subsets acts as an independent holdout testing dataset for the model trained with the rest of four subsets. Thus, five models are generated for the five sets of data. The advantages of cross validation are that the impact of data dependency is minimized, and the reliability of the results can be improved.

| Model | Features | Classifier | SN(%) | PPV(%) | ACC(%) | MCC(%) |
|---|---|---|---|---|---|---|
| Our method | MLD | RF | 94.34±0.49 | 98.91±0.33 | 94.72±0.43 | 85.99±0.89 |
| Guos' work | ACC | SVM | 89.93±3.68 | 88.87±6.16 | 89.33±2.67 | N/A |
| | AC | SVM | 87.30±4.68 | 87.82±4.33 | 87.36±1.38 | N/A |
| Zhous' work | LD | SVM | 87.37±0.22 | 89.50±0.60 | 88.56±0.33 | 77.15±0.68 |
| Yangs' work | Cod1 | KNN | 75.81±1.20 | 74.75±1.23 | 75.08±1.13 | N/A |
| | Cod2 | KNN | 76.77±0.69 | 82.17±1.35 | 80.04±1.06 | N/A |
| | Cod3 | KNN | 78.14±0.90 | 81.86±0.99 | 80.41±0.47 | N/A |
| | Cod4 | KNN | 81.03±1.74 | 90.24±1.34 | 86.15±1.17 | N/A |

Table 3. Comparison of the prediction performance by the proposed method and some state-of-the-art works on the yeast dataset.

The prediction performance of RF predictor with MLD representation of protein sequence across five runs is shown in Table 3. It can be observed from Table 3 that high prediction accuracy of 94.72% is obtained for the proposed model. To better investigate the prediction ability of our model, we also calculated the values of Sensitivity, Positive Predictive Value, and MCC. From Table 3, we can see that our model gives good prediction performance with an average sensitivity value of 94.34%, PPV value of 98.91%, accuracy value of 94.72%, and MCC value of 85.99%. Further, it can also be seen in the Table 3 that the standard deviation of sensitivity, PPV, accuracy, and MCC are as low as 0.0049, 0.0033, 0.0043, and 0.0089, respectively. For the first PPI dataset, we define negative examples exploiting the fact that proteins from different cellular locations are unlikely to interact [50]. However, it was shown that this approach, when used to train PPI prediction methods, leads to a bias in the estimation of prediction accuracy since the additional constraints related to localization make the prediction task easier [47]. Another typical choice is to select non-interacting pairs uniformly at random from the set of all proteins links that are not known to interact. Therefore, in our experiments, we also use the second PPI dataset to verify the effectiveness of the proposed method. Table 3 illustrates the comparison of the prediction performance using two kinds of negative sample selection methods on the yeast dataset. As shown in the table, the performance of first PPI dataset (selecting negative examples using cellular localization information) is slightly better than that of the second PPI dataset (randomly selected negative examples without cellular localization information). We can explain the higher accuracy for the first PPI dataset by the fact that the constraint on localization

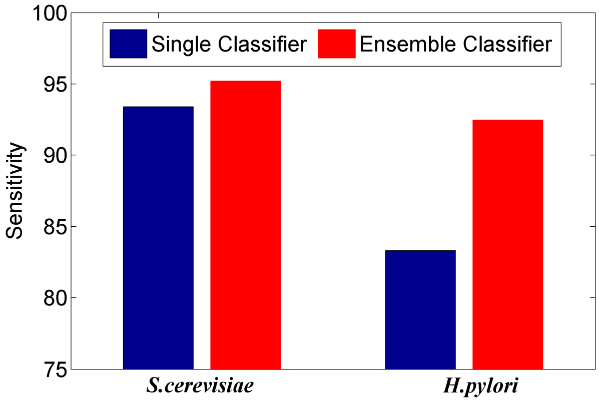


Fig. 4. Comparison for the Sensitivity value of the ensemble classifier versus single classifiers on the dataset of S.cerevisiae and H.pylori.

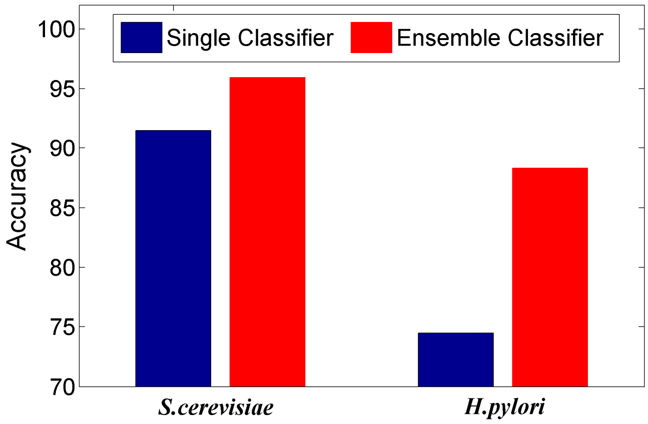


Fig. 5. Comparison for the Accuracy value of the ensemble classifier versus single classifiers on the dataset of S.cerevisiae and H.pylori.

restricts the negative examples to a sub-space of feature space, making the learning problem easier than when there is no constraint.

We further compared our method with Guo *et al.*[44], Zhou *et al.*[52] and Yang *et al.*[53], where the SVM, SVM and KNN is performed with the Auto Covariance (or Auto Cross Covariance), Local Descriptor, and Local Descriptor with four kinds of the coding scheme as the input feature vectors, respectively. From Table 3, we can see that the performance of all of these methods with different machine learning model and sequence-based feature representation are lower than ours, which indicates the advantages of our method. To sum up, we can readily conclude that the proposed approach generally outperforms the previous model with higher discrimination power for predicting PPIs based the information on protein sequences. Therefore, we can see clearly that our model is a much more appropriate method for predicting new protein

| Methods | Sensitivity | PPV | Accuracy | MCC |
|---|---|---|---|---|
| Phylogenetic bootstrap | 69.80% | 80.20% | 75.80% | N/A |
| HKNN | 86.00% | 84.00% | 84.00% | N/A |
| Signature products | 79.90% | 85.70% | 83.40% | N/A |
| Ensemble of HKNN | 86.70% | 85.00% | 86.60% | N/A |
| Boosting | 80.37% | 81.69% | 79.52% | 70.64% |
| Proposed method | 92.47% | 85.99% | 88.30% | 79.19% |

Table 4. Performance comparison of different methods on the H.pylori dataset.

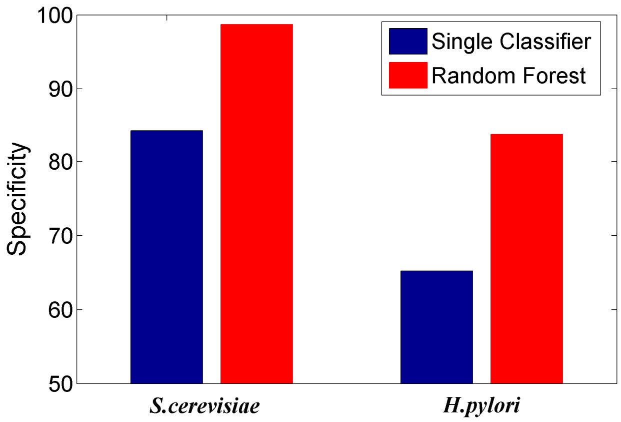


Fig. 6. Comparison for the Specificity value of the ensemble classifier versus single classifiers on the dataset of S.cerevisiae and H.pylori.

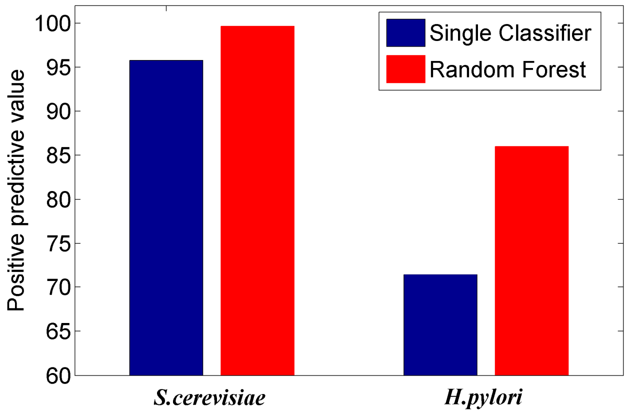


Fig. 7. Comparison for the Predictive PositiveV value of the ensemble classifier versus single classifiers on the dataset of S.cerevisiae and H.pylori.

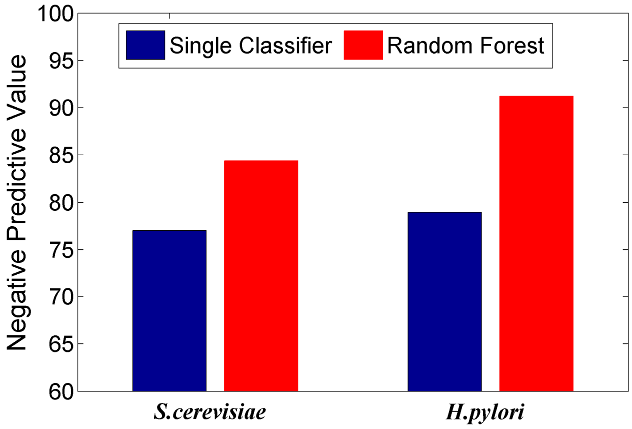


Fig. 8. Comparison for the Negative Positive Value of the ensemble classifier versus single classifiers on the dataset of S.cerevisiae and H.pylori.

interactions compared with the other methods. Consequently, it makes us more convinced that the proposed method can be beneficial in assisting the biologist to assist in the design and validation of experimental studies and for the prediction of interaction

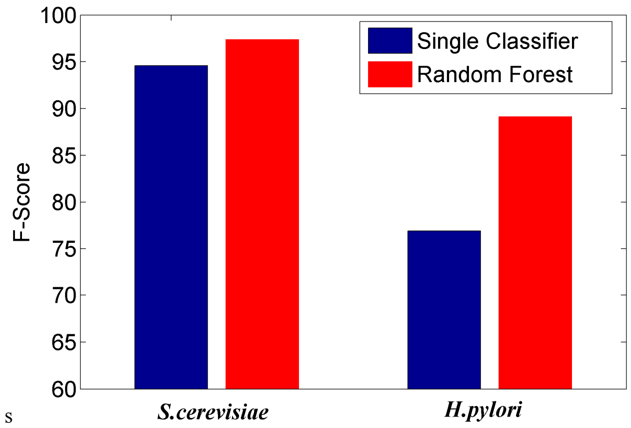


Fig. 9. Comparison for the F-Score value of the ensemble classifier versus single classifiers on the dataset of S.cerevisiae and H.pylori.

partners.

We here investigated whether the ensemble of classifiers can significantly improve the performance of PPI prediction compared against the individual classifier in the ensemble. Figs 4–10 plots the sensitivity, accuracy, specificity, PPV, NPV, F-Score, and MCC values for the component classifiers decision tree and the ensemble classifier random forest. The results in Figs 4–10 demonstrate that the ensemble classifier dominates the component classifiers. The PPV value obtained by the ensemble classifier is nearly 4.3% higher than the component classifier on the S.cerevisiae dataset. In addition, the sensitivity is improved from 92.66% to 95.15% while the accuracy is improved from 90.52% to 95.80%. Further, on the H.pylori dataset, it can also be seen from the Figs 2–8 that the ensemble classifier dominates the single classifier. We concluded that the ensemble classifier is much more accurate than the single classifier that makes them up.

To highlight the advantage of our model, it is also tested by Helicobacter pylori dataset. The H. pylo Figsri dataset is composed of 2,916 protein links (1,458 interacting pair and 1,458 non-interacting pairs) as described by Martin et al. [50]. This dataset gives a comparison of the proposed method with other previous works including phylogenetic bootstrap [56], signature products[50], HKNN[57], ensemble of HKNN[58] and boosting. The methods of phylogenetic bootstrap, signature products and HKNN are based on individual classifier system to infer PPI, while the methods of HKNN and

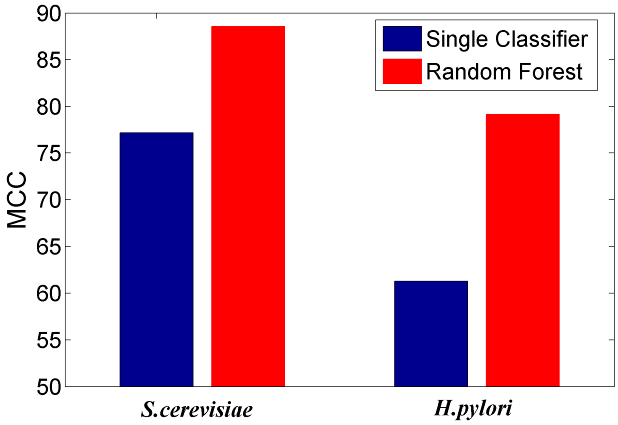


Fig. 10. Comparison for the Matthews Correlation Coefficient value of the ensemble classifier versus single classifiers on the dataset of S.cerevisiae and H.pylori.

boosting belong to ensemble-based classifiers. The average prediction performances of ten-fold cross-validation over six different methods are shown in Table 4. From Table 4, we can see that the average prediction performance, i.e. sensitivity, PPV, accuracy and MCC achieved by proposed predictor, are 92.47%, 85.99%, 88.30% and 79.19%, respectively. It demonstrates that our method outperforms all other individual classifier-based methods and the ensemble classifier systems (i.e. ensemble of HKNN and Boosting). All these results show that the proposed method not only achieves accurate performance but also substantially improves positive predictive value in the prediction of PPI.

### 3.4 Summary

In this chapter, we develop an efficient representation technique for extracting protein sequence information. For the link prediction task, we combined the proposed Multi-Scale Local Descriptor (MLD) feature representation with the RF model and predicted PPI based on the protein sequence. The MLD representation takes into account the factors that PPI usually occurs in continuous segments with varying lengths in the protein sequence. In our study, protein sequences are characterized by a number of regions using MLD representation, which is capable of capturing multiple overlapping continuous binding patterns within a protein sequence. The experimental results show that the proposed representation method can extract feature representation effectively on multiple PPI data sets. Combined with RF classifier, it performs better than all previous methods and can be used as a useful supplementary tool for traditional experimental methods.

# 4. DEEP REPRESENTATION

As mentioned above, the new feature engineering successfully makes the original feature expressed as thoroughly as possible through the multi-scale transformation algorithm. Comprehensive feature representation can improve the performance of the machine learning model to some extent, while the high-dimensional feature will cause overfitting of the traditional model. Bengio et al. [1] believe that a good data representation is not merely a distinction between "signal" and "noise", but that the most useful factors are retained, while the minimum valuable information is discarded. A good data representation can even reconstruct many internal factors that cannot be represented in the original data. In other words, when the data is presented, it maps the object to a representation of an estimate that separates the effect of the latent factors from the nuisance parameters and allows reconstruction of the observations from the representation. To find better data representation, several algorithms are proposed. Though effective to some extent, most of these algorithms are limited by the inability to find latent internal factors and the comprehensive representation of multiple views. Different from the traditional, deep representational learning has been proposed and successfully used in many applications. Feature extraction and transformation is performed by using cascades of multiple nonlinear processing units. Each successive layer uses the output of the previous layer as input. Learning corresponds to multiple representations of different levels of abstraction, which form a hierarchy. The stacked auto-encoder confirms that the greedy hierarchical training strategy is largely conducive to optimization by initializing weights in regions close to good local minima, generating internal distribution representation, which is a high-level abstraction of the input and leads to better generalization. In addition, the stacked auto-encoder is a good architecture for the multi-layer neural network to deal with the problems of model overfitting and gradient diffusion. The performance of link classification and nodes clustering in the network also depends on taking into account multi-view features. In other words, the node in each network can be described from multiple perspectives. However, the existing machine learning algorithm is difficult to optimize the model with multi view features. In this thesis, we innovatively combined deep network and network fusion into a new algorithm. In this section, we introduce the latest techniques related to the discovery of deep representation and deep fusion representation in link classification and nodes clustering.

## 4.1 Predicting large-scale drug–target interactions from integrated deep representations

As described above, drug-targeted interaction (DTIs) is a typical network link and predicting the link in the network and improving accuracy is a challenge. Traditional similarity-based approaches have taken hold, and they use the drug and target similarity matrix to infer the potential drug-target links. But these techniques do not handle biochemical data directly. While recent feature-based methods reveal simple patterns of physicochemical properties, efficient method to study large interactive features and precisely predict interactions is still missing. Deep learning has been found to be an appropriate tool for converting high-dimensional features to low-dimensional representations. These deep representations generated from drug-protein pair can serve as training examples for the interaction predictor. In this section, we propose a promising approach called multi-scale features deep representations (MFDR) inferring interactions. We extract the large-scale chemical structure and protein sequence descriptors to machine learning model predict if certain human target protein can interact with a specific drug. MFDR use Auto-Encoders as building blocks of deep network to reconstructing drug and protein features to low-dimensional new representations. Then, we make use of support vector machine to infer the potential drug-target interaction from deep representations. The experiment result shows that a deep neural network with Stacked Auto-Encoders exactly output interactive representations for the DTIs prediction task. MFDR is able to predict large-scale drug-target interactions with high accuracy and achieves results better than other feature-based approaches.

### 4.1.1 Overview

Drug discovery is a comprehensive study of diverse objects and provides detailed descriptions of the biological activity, genomic features and chemical structure to the disease treatment. Lead compound interacting with human protein is one of the critical procedures responsible for driving

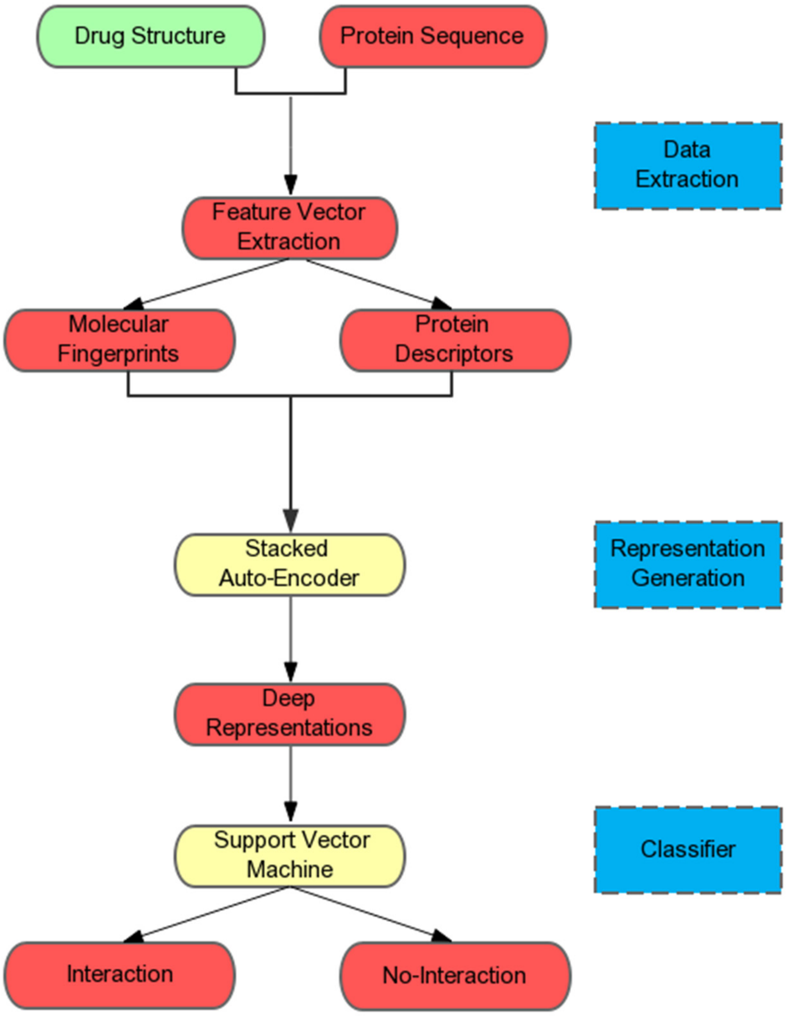


Fig. 11. The procedure of MFDR

critical biological actions within the human body cell. The mainly treatment processes within our body are carried out by adjusting proteins status that physically interacts to form the counter effect of the disease. Discovery of such drug-target interactions that take place within a human body can suggest new drug target protein and aid the design of new compounds by providing rational drug targets [59]. As the biggest drug database, PubChem collected more than 35 million compounds of which 7000 compounds are containing the target protein information. During the drug discovery processing, medicine production line often generates results different from the original goal. Such effects may be raised with hidden factors and biological domains in drug target selection and lead compound screening. Instability and no specificity of drug-target interactions have to be addressed appropriately before sending them to clinical phase. The complex procedure of confirming lead compound ranges from target identification to lead compound optimization is a long-term work. Even many optimal approaches have been proposed to tackle the problem of drug-target interaction prediction, and a new drug discovery still needs cost 6-10 years. Therefore, the identification of potential drug-target interactions is a challenging issue at the start stage of the drug development process [60].

To solve such problems, some interdisciplinary scientists introduce several computational approaches to cope with such issues. Previous attempts are divided broadly into the similarity-based approach and feature vector-based techniques [61]. Similarity-based methods are developed to discover potential DTIs through the similarity matrices of drug and protein. Some early works to discover drug-target interactions have been proposed based on their compound and protein sequence similarity [62]-[63]. Feature vector-based methods are regarded as more advanced strategies that face drug and protein features straightforward. They can uncover the description of the hidden knowledge concerning significant features and then generate rules to reproduce experts' decision process. These methods provide meaningful solutions for discovering interest patterns such as single molecule sub-structure influence, but they are not able to precisely reflect the molecule substructure and protein subspace interactions. It's also tricky for current techniques to is real high- sion tein ptor xcept parallel scheme [64]. Fortunatel it is y believed pr can olve deep ng mechanism [65], [1]. Co sed representation has bee large volumes of

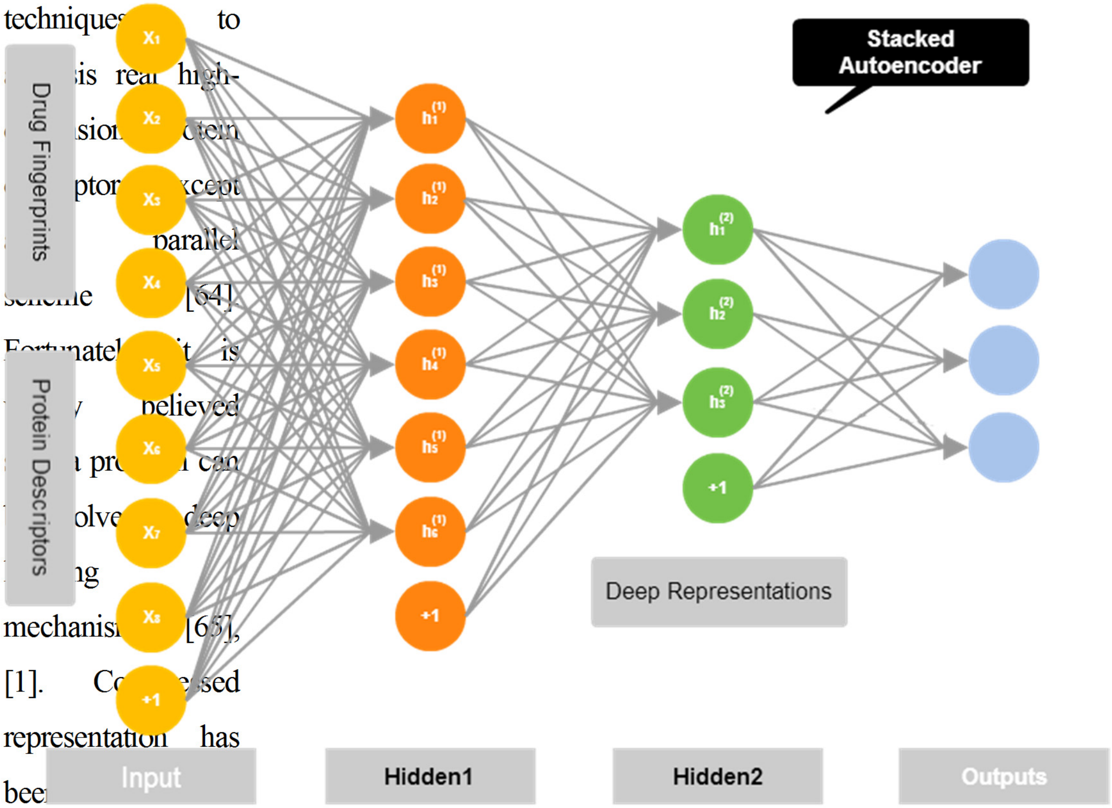


Fig. 12. A Stacked Auto-Encoder composed by two visible layers and two hidden layers

protein attributes and possibly uncovers significant hidden relationships exist in the protein [67]-[68].

From the perspective of the drug design, the previous drug-target prediction based on the data model is an excellent place to start. Previously, the leading research area in drug-target interaction is similarity-based approaches that use drug and target similarities. The conventional techniques for predicting drug-target interactions are favorite in screening potential drug candidates for further drug action verification. Such similarity-based methods derived from drug-drug and target-target similarities exploring. Drug similarity was tested from the molecules of drugs by using SIMCOMP [69]. Target similarity was computed from the protein sequence by using Smith-Waterman score [70]. In [61], they used the Smith-Waterman score to describe a genomic space and utilized SIMCOMP score to describe pharmaceutical space. And then, they proposed a kernel regression approach called bipartite graph learning to predict drug-target interactions. In [71], they used the drug and target similarities as the support SVM kernels and classified interaction twice and merge the experimental results to provide drug target predictions. KBMF2K firstly map the drug and target spaces to low-dimensional spaces similarity for drug–target interaction prediction [72]. In [73], they developed a network-consistency-based prediction method (NetCBP) to predict drug-target interactions, which rely on drug similarity network and the target similarity network integration. Some above previous works enjoyed very high prediction accuracy. However, the similarity-based direction has an underlying problem that support data is not a direct biological expression. The similarity represents another dimension of original drug and target properties that may make experiment only achieve a high rate of errors regarding millions of candidates. Even more interesting is feature-based methods have been attempts to use a classifier to infer drug-target interactions adopt different encoding schemes impose different descriptors on the protein sequences and compounds. A recent study [74] first try to address drug structures and protein sequence as a structure-activity relationship. They use SVM as a classifier to predict DTIs can be regarded as a significant direction even if a large-scale calculation is time-consuming. Bigram-PSSM takes advantage of PAAC descriptors for more accurate prediction [75]. Other studies of more recent interest include heterogeneous integration [76] or rare domain knowledge acquisition [77]. Even so, such methods cannot be required for high dimensional descriptor analysis. To overcome these drawbacks, we propose a novel model to extract large-scale drug-target descriptors and classify output information after a deep representation phase.

In this study, we develop a new deep learning-based method for the prediction of drug-protein interactions from protein sequence descriptors and molecule fingerprints with Support Vector Machine aiming at improving the efficiency and effectiveness of the classification accuracy. Firstly, we introduce a multi-scale local descriptor approach for discovering realistic large amino acid sequences descriptors and use chemical fingerprints to represent the chemical space. Secondly, to enhance the accuracy and transfer the large descriptors to deep representations, the extracted features of the input layer would be automatically learned by an unsupervised Stacked Auto-encoder for output a reconstruct lower dimensional layer. Finally, we focus on use classifier to judge whether one drug interacts with one target. Our method constitutes a significant advance because it logically considers coupled representations of protein and molecules that remain unobserved in any interaction. We adopt a popular data standard to test the proposed method which includes G protein-coupled receptor, enzyme, ion channel, and nuclear receptor dataset [61]. MFDR has been tested with Gold standard data sets that can be a beneficial approach to predict the DTIs. The necessary steps of MFDR, Fig 11 shows a procedure of the proposed method according to our definition.

### 4.1.2 MFDR in details

Our method undergoes two main computational steps: the compound-protein interaction representations discovery step briefly describe how to extract more meaningful protein sequence attributes. At the representation step, we introduce Stacked Auto-encoder to obtain new representations instead of original high-dimensional protein and compound features. New representations are the statistically significant solution to the sparse and large data set. Then, in the classification step, we use these discovered new space sets as input for assigning them as DTIs associated information.

#### **4.1.2.1** Feature representations of protein sequence and chemical structure

Feature extraction usually influences the quality of training data when we analyze large-scale biological data. Some elaborate protein extraction methods have revealed the valuable representations and also have had many remarkable discoveries [78-81]. To fully extract the interaction related features, we adopt an advanced multi-scale protein sequence representation method to extract feature vectors from sequences by using a binary coding scheme [81]. Typically, an original polypeptide sequence should contain multiple continuous sequence segments which are composed of residues. For collect specific feature vector of the protein sequence, we will take multi-scale descriptors to calculate and concatenate each continuous local region by introduced decomposition technique. Based on the actual situation, this approach is able to transform the protein sequences into multi-scale feature vectors which can span several length levels. Molecular fingerprints are descriptions of drug chemical sub-structures originally introduced to assist in chemical database searching [82]. Our chemical fingerprints set are generated by the PubChem System to encode the 3D structure of a molecule for our computing method. These fingerprints are used by PubChem for similarity neighboring and similarity searching to idealize 3D chemical structure. A fingerprint is an ordered list of binary (1/0) 881 bits in length. Fingerprints property is "CACTVS_SUBGRAPHKEYS" in PubChem and Base64 encoded to provide a textual representation of the binary data.

Each drug is represented by a chemical feature vector $D^{(chem)} = (d_1,\dots, d_q)^T$, where each element encodes for the presence or absence of each substructure by 1 or 0 and q is the number of fingerprints. Each target protein is represented by a sequence feature vector $T^{(protein)} = (t_1,\dots, t_p)^T$ , where each element encodes for the value of each descriptor range from 0 to 1 and p is the number of descriptors.

#### **4.1.2.2** Deep representations inferring interactions

After the multi-scale feature of drug and protein collected, known drug-target interactions will be represented by many factors includes the properties of drug compounds and the properties of the target protein. Some biochemical effect may be contained in these properties that including structure shape, amino acid composition, hydrophobicity, van der Waals force, Hydrogen bond, Water effects, Metal-ligand interactions and so on. In our method, each drug-target interaction sample will be represented by more than one thousand dimensional vectors. Each drug-target interaction features is made up of chemical substructures and multi-scale protein representations. That is to say, each drug-target interaction can be represented as $DT = (dt_1, dt_1, dt_1, \dots dt_s)$, where $dt_x$ is the $x_{th}$ drug-target interaction feature combined with $d_q$ and $t_p$. As with the original large-scale interactions, however, there are sparsity and imbalance issues if we have to deal with new representations directly. Dimensionality reduction has proven to be a useful method deal with large-scale data. The only problem is dimensionality reduction usually loss some important information of input data. Drug discovery is directly influencing the human body, so any information about the medicine should keep as more as possible. Fortunately, deep learning will keep valuable information after executing the training process. According to [83-84], deep learning built multi-layer architecture neural networks and trained with the greedy layer-wise unsupervised pre-training algorithms. DNN is about applying the greedy layer-wise unsupervised pre-training mechanism that can reconstruct the original raw data set. We can learn valuable features with deep representation instead of traditional features filtering method. Then, we can use a classifier and obtain higher accuracy with better generalization from the learned features. Also, the risk of fall in a local minimum rather than a global minimum problem in traditional training method has been solved by a deep network that greedily trained up hidden layer with Auto-encoder at a time. Because of the feature type of our drug data are real numbers and sparse distribution, we choose to stack Sparse Auto-Encoder for building a deep architecture of the neural network model.

Stacked Auto-Encoder is a stacked architecture network that applies Auto-Encoder in each layer [85]. In a neural network, each “neuron" in one layer is a computational unit that could be regarded as input vector $X = (x_1;\ x_2, \dots, x_n)$ (and a+1 intercept term), and outputs $h_{W,b}(x) = f\ (W^T x) = f\ (\sum_{i=1}^{3} W_i\, x_i + b)$ where a nonlinear function $f$ :

$\Re \rightarrow \Re$. is activation function. The connections among different neurons in the network can be taken as *a* weight matrix *W*. In usual cases of a neural network, sigmoid function is normally using $f(z) = \frac{1}{1+exp(-z)}$. A conventional Auto-Encoder would endeavor to learn a function $h_{W,b}(x) \approx$ x which means it is discovering an approximation to the identity function, to output an approximate outcome x̂. The identity function seems a typically small function trying to learn but by placing constraints on the network. For deep represent DTIs information, we can make discovery useful structure of drug-target interactions data from limiting hidden units. Take our multi-scale features as examples, $DT = (dt_1, dt_1, dt_1, \ldots dt_s)$ is defined as input vector $X$. Suppose the original feature representations are collected from a 1448-dimensional feature space, i.e. x ∈ $\Re^{1448}$ which means there are 1448 visible input units. If we set that there are 600 hidden units in the hidden layer1, according to the requirement $h_{W,b}(x) \approx$ x, the next layer need to learn a compressed representation of the input. This also means that the hidden layer will start to reconstruct the 1448-dimensional input x by a given vector of hidden unit activations $\mathrm{a}^{(2)} \in \Re^{600}$. Access to the multi-scale DTIs data could then be transformed into deep representations through the reconstruction process, instead of the high-dimensional and noise visible units. There is an interesting structure hide in the input data like two features have a relationship. Otherwise, this reconstructive function wouldn't work if the inputs features were completely random, i.e., each $x_i$ is independent of the other features. The overall cost function of non-sparse Auto-Encoder can be defined as:

$$J(W,b)$$

$$= \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W,b;x^{(i)},x^{(i)}\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2$$

$$= \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}||h_{W,b}(x^{(i)}) - x^{(i)}||^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(W_{ji}^{(l)})^2 \quad (1)$$

The first term in $J(W,b)$ is an average sum-of-squares error term, where $m$ is the training samples number. The second term is a regularization to prevent over-fitting, where λ be supposed to control the relative importance of the two terms. Normally, Auto-Encoder is aiming to minimize Equation (1) for that output $h_{W,b}(x) \approx x^{(i)}$ can approximate the raw data $x^{(i)}$ as much as possible. Further, largely hidden units still could be used to discover valuable information if we impose a sparsity constraint on the hidden unit [85]. Sparse Auto-Encoder tries to keep the output mean value of hidden layer to 0 which means most neurons are considered to be inactive. The overall cost function of Sparse Auto-Encoder can be defined as:

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} KL(\rho||\hat{\rho}_j) \tag{2}$$

Where

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})] \tag{3}$$

$$KL(\rho||\hat{\rho}_j) = \rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \tag{4}$$

where $\rho$ is a sparsity parameter, $s_2$ is the number of the hidden neurons and $\beta$ controls the weight of the sparsity penalty term. Eq. (3) is average activation of hidden unit $j$ and we need to enforce the constraint $\hat{\rho}_j = \rho$ . Eq. (4) is the Kullback-Leibler divergence between a Bernoulli random variable with mean $\rho$ and a Bernoulli random variable with mean $\hat{\rho}_j$ .

Specifically, we should stack Sparse Auto-Encoders layer by layer to a whole deep network. A typical two hidden layers Stacked Auto-Encoder structure diagram is shown in Fig 12 describes the main procedure of the proposed model. As in Fig 12, the input layer is a visible layer that takes the original data set. In every hidden layer, the neurons receive data from the previous layer, and then they compute the received data through an Auto-Encoder. At the end of each hidden layer, the neurons output the computed new features to the next hidden layer or visible layer. After a deep network processing, the original data set will represent by deeper feature spaces layer by layer. Therefore, Stacked

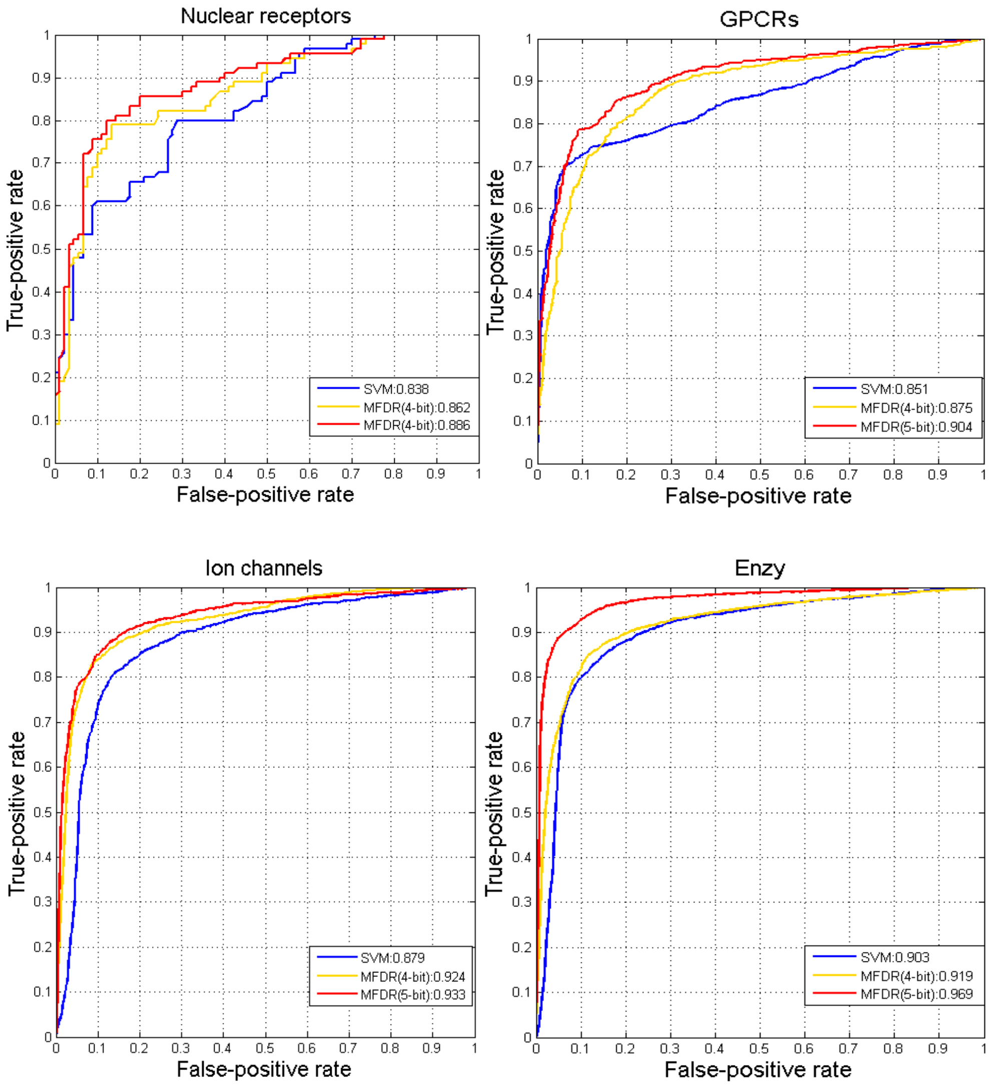


Fig. 13. ROC curves of four different drug-target interaction predictions

Sparse Auto-Encoders can learn enriched representations from the large-scale original data sets.

#### **4.1.2.3** Prediction model

After we bring in Stacked Auto-Encoder as an unsupervised learning model to get new representations, one useful classifier will be used to predict whether a given drug-target interaction is positive or not according to the gold standard dataset. SVM (Support Vector Machine) will be used as the classifier to build our predicting model. SVM is a popular classification algorithm initially developed by Vapnik et al. and it has been proved extremely effective in chemical and biological classifications [86-87]. The necessary procedure of utilizing an SVM model for DTIs prediction can be described briefly as follows. Firstly, we use the open source package from LIBSVM [87] to implement SVM. Then, SVM maps the inputted drug and target original representations space X into a high dimensional feature space F with a linear algorithm due to the linear relations exist in training data. After that, the SVM model will find out an optimized linear division within the feature F. According to our test and previous experiences, Radial Basis Functions (RBF) kernel is the best kernel selection of the traditional kernel, especially it's appropriate for the high-dimensional data sets and has better boundary response.

### 4.1.3 Results

#### **4.1.3.1** Data Preparation

In this study, the data which are used to predict DTIs come from [61]. There is a drug-target interactions gold standard dataset formed by four types of DTIs, which includes enzymes, ion channels, GPCRs, and nuclear receptors. After interactions collection, the final number of positive drug-target interactions in the gold standard dataset are 2926, 1476, 635 and 90, respectively. Each category is further organized in drugs are 445, 210 223and 54, respectively and the protein numbers are 664, 204, 95, 26 with four categories. Table 6 shows the statistics of the used dataset.

Table 5. Stacked auto-encoder parameters

| Parameter | Value |
|---|---|
| Neurons in hidden layer 1 | **600** |
| Neurons in hidden layer 2 | **200** |
| Beta (weight of sparsity penalty term) | **5** |
| Sparsity (desired average activation of the hidden units) | **0.05** |

Table 6. Drug-target data statistic

| Type | | Ion channel | Enzyme | GPCR | Nuclear receptor |
|---|---|---|---|---|---|
| Drugs | | 210 | 445 | 223 | 54 |
| 881 bits | | | | | |
| Target proteins | | 204 | 664 | 95 | 26 |
| 567 Descriptors | 1449 Descriptors | | | | |
| Positive Drug–target Interactions | | 1476 | 2926 | 635 | 90 |

Suppose that we have a set of n drugs with biological profiles of m target proteins. To encode drugs features, chemical structures of drug compounds are extracted from PubChem database which uses a fingerprint corresponding to t 881 chemical substructures. Each drug was represented by an 881-dimensional feature vector $D^{(chem)} = (d_1, \ldots, d_{881})^T$, where each element encodes for the presence or absence of each substructure by 1 or 0, respectively. To encode protein features, the protein sequence is extracted from the multi-scale local descriptor feature representation scheme. Regarding test larger descriptors influence, all descriptors calculated in 4-bit and 5-bit are concatenated. For 4-bit binary form, each sequence was represented by a 567 dimensional vector $T^{(protein)} = (t_1, \ldots, t_{567})^T$. For 5-bit binary form, a total 1449 dimensional vector $T^{(protein)} = (t_1, \ldots, t_{1449})^T$ has been built to represent the protein sequence. Each element encodes for the value of each descriptor range from 0 to 1.

On present understanding, known negative DTIs samples are generally much larger than the positive DTIs samples. Because the size of non-interactions is not comparable with the size of positive interactions, some works may take a high true negative result by the significant larger negative samples. With the goal of solving the above problems, previous feature-based approaches have randomly selected negative samples from the non-interactions until the ratio hitting the one-to-one scale. We considered all real positive drug–target interactions and randomly selected the same negative sample as many as the positive samples like [74]-[75]. In summary, the original features number can be extracted from MFDR of 4-bit is 1448 that comprise 881 chemical substructures and 567 protein descriptors. And so on, each of the drug-target interaction has 2330 features which extracted from MFDR of 5-bit.

**4.1.3.2** Performance Evaluation

Since our method like evaluates deep representations based on their profiles with protein and molecules, which in turn refer to specific sub-actions, we use a 2-layer Stacked Sparse Auto-encoder model to rebuild the drug and target features. The first hidden layer of our model is composed of 600 hidden units while the second hidden layer is composed of 200 hidden units. Which mean, there are 200 deep representations come out from thousands of original features. Table.5 shows the parameter configuration of the Stacked Sparse Auto-Encoder model. We performed the fivefold cross-validation to split gold-standard data into five subsets of equal size. Each subset was then taken in turn as a test set, and we performed the training on the remaining four sets. We used the grid search to select the best regularization parameter C and the kernel parameter γ for the radial basis function (RBF) based on the overall accuracy. In this study, the performance of MFDR was mainly evaluated by using ROC. The ROC (receiver operating characteristic curve) demonstrates the true-positive rate and false-positive rate of the experimental result, where true-positives are the number of correctly predicted drug-target interactions while

Table 7. Comparison for the Auroc of the MFDR versus others on the four dataset

| Data set | Feature-based | | | Similarity-based | | |
|---|---|---|---|---|---|---|
| | MFDR | Cao, D.S. et al. | Bigram-PSSM | Bipartite Graph Learning | KBMF2K | NetCBP |
| Nuclear receptors | **0.886** | 0.882 | 0.869 | 0.692 | 0.824 | 0.839 |
| GPCRs | **0.904** | 0.890 | 0.872 | 0.811 | 0.857 | 0.823 |
| Ion | 0.933 | 0.942 | 0.889 | 0.692 | 0.799 | 0.803 |
| Enzymes | **0.969** | 0.948 | 0.948 | 0.821 | 0.832 | 0.825 |

the false-positives are the number of not correctly predicted drug-target interactions. AUC refers to the area under ROC curve which is an important measure which can be used for evaluating the classification accuracy.

Given the larger number of protein sequence descriptors can reflecting structures more real [79], we followed two different regions outlined by 4-bit and 5-bit extraction. The resulting AUROC scores of MFDR(4-bit) for enzymes, ion channels, GPCRs, and nuclear receptors are 0.919, 0.924, 0.875 and 0.862 respectively. The resulting AUROC scores of classical SVM without deep representation are 0.903, 0.879, 0.851 and 0.838. As Fig 13 shows, our prediction accuracy is higher than the classical SVM that proved the multi-scale feature deep representations can improve the DTIs prediction while reducing high dimensional features. We also give the AUROC scores of MFDR(5-bit) are 0.969, 0.933, 0.904 and 0.886 respectively. It proved that MFDR has the chance to improve performance under a much larger range of descriptors even project to same low-dimensional representations. In addition, we believe MFDR should get significantly better performance regarding higher bit binary coding.

To evaluate the performance of the method in comparison to previous work, we considered three important studies in this similarity-based area and two feature-based

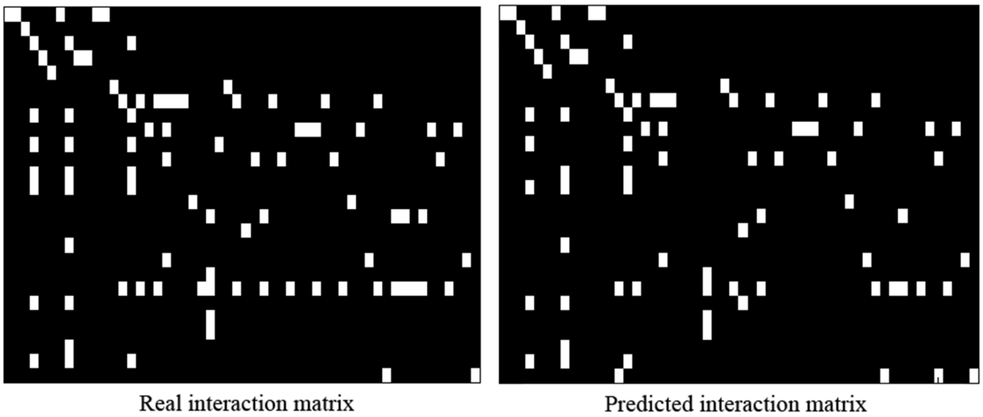


Fig. 14. Interaction matrix of nuclear receptor. Real interaction matrix is known drug-target interactions of nuclear receptor, predicted interaction matrix is generated by MFDR. White pixels represent the positive interactions, whereas Black pixels represent the negative interactions.

studies. We compared the best AUROC scores of the MFDR with these approaches including KBMF2K [72], NetCBP [73], Bipartite Graph Learning [61], Bigram-PSSM [75] and the proposed method by Cao, D.S. et al. [74]. Table 7 shows the AUROC scores of MFDR and others divided by four interaction types. As the results look like those shown above, the prediction accuracy of the MFDR is superior in comparison with most methods. We go further compared the predicted interactions of the nuclear receptor to real interactions as Fig 14 shown. As there is a high coincidence of the bright pixels to the predicted matrix and the real one, we can claim the deep representations successfully kept values from original large descriptors.

### 4.1.4 Summary

In this work, a new prediction model was developed for inferring drug-target interactions. We adopt the multi-scale optimization theory to extract the drug and protein details from limited biological information. Deep representation approach also introduced in our method for retaining the realistic biological properties and reducing the high-dimensional features. This is the first time that deep learning was used to predict drug-target interactions. The key aspects of the DTIs prediction model reflect a feasible way of mapping the large-scale drug-target descriptors to lower-dimensional representations rationally. The proposed Stacked Auto-encoder model can generate representations of the multi-scale data layer by layer. In the last step, our model successfully reconstructs the representative features from the stacked hidden layers and builds an SVM as the final classifier. We gathered several kinds of DTIs datasets that we used to train the deep representation model. The experimental result shows that MFDR is able to handle the large-scale pharmacological data effectively and improve the performances of the drug-target interaction prediction model. Our work can provide some important mechanisms of drug-target interaction to make the drug discovery navigation simpler. Also, deep learning successfully transfers high dimensional data to a relatively lower dimensional coupled description which make our model more sensitively reflect real actions. It has been proved in large-scale drug-target interactions and it should have the ability to solve other large biological data problem in the future.

## 4.2 Deep graph fusion for social network clustering

### 4.2.1 Overview

Given the fact that heterogeneous social network datasets can be collected, learning fused representations from multiple real-world networks is increasingly becoming an urgent pursuit of the nodes clustering model [88-89]. Though various types of data modalities, they are used to characterize the same entities, which in other words are vertices, or nodes, in the network data. For example, social networks typically use vertices to represent users and edges to represent social links between users, and contents that may characterize users. A protein-protein interaction (PPI) network can also be described with vertices representing proteins, edges representing interactions between proteins, and GO terms that are used to represent the biological meanings of those proteins [90-91]. The identification of meaningful sub-groups, in which data entities are cohesively interrelated, may significantly enhance the learning of knowledge hidden in these heterogeneous networks.

To discover such meaningful groups in social network data, which are also named as communities, or clusters, some approaches so-called graph clustering algorithms, have been proposed. Most traditional clustering algorithms only consider a single network [92-93]. Such methods, e.g., spectral clustering (SC) [94] and affinity propagation (AP) [95] mainly take into the consideration topological information of the network data while performing the task of clustering. To overcome the mentioned issue, several methods are proposed to detect network clusters by learning latent structures from different networks simultaneously. For examples, two model-based approaches, CESNA [96] and relational topic model (RTM) [97] may discover network clusters by taking into the consideration both node links and content information. In [98], an evolutionary community detection algorithm (ECDA) is proposed to detect graph clusters by grouping vertices into the same cluster according to the structural and attributed networks. In [99], an algorithm for mining interesting subgraphs in the attributed graph (MISAGA) is proposed to make use of statistical measure to solve subgraphs discovering as a constrained optimization problem. Though these mentioned algorithms are used to some extent, they either consider the information from the single network, e.g., topological or attribute similarity network or do not consider the common representations of network data which involve information from heterogeneous sources.

Besides existing in the network data, the feature of multiple modalities can also be found in various data types. Therefore, identifying an effective way to describe the common characteristics across different data modalities has drawn much attention in recent years. Among those presented works, there have been several ones that try to identify such common characteristics by taking into the consideration fusing data from multiple domains, such as remote sensing with multi-spectral and panchromatic images [100], target tracking with sensors data [101], biological applications with omics data [102], heterogeneous social networks detecting [103], and some applications with visual and temporal data [104]. These data from multiple sources are inherently correlated, and sometimes provide complementary information to each other. Therefore, data fusion has been paid much attention to, which mainly aims to generate most similar representation between entities under the existing domains. Many such fusion algorithms are proposed to attempt to fuse multiple data more effectively and efficiently. These techniques were drawn from a wide range of areas including data association, statistical estimation, and decision fusion. Data association-based methods, including the nearest-neighbor (NN) algorithm, the joint probabilistic data association (JPDA) method [105], the multiple hypothesis tracking approach (MHT) [106], Distributed JPDA and MHT, aim to establish the set of observations. For example, JPDA is a joint approach for tracking multi-target and the association probabilities are computed using all the observations. The goal of statistical estimation techniques is tracking the state of the target under measurements changing. They usually take probability theory to estimate a vector state from a vector measurement. Typical implementation models of estimation methods include maximum likelihood and Kalman filter [107-108]. Decision fusion discovers representations as sources and combines them to obtain a more accurate decision. Bayesian's methods are typically adopted techniques in this kind of approaches [109].

Besides, there are also some other fusion methods proposed for fusing multi-domain networked data. These approaches adopt heterogeneous descriptions of the same data entities, generate similarity networks representing the relationship between data entities, and then fuse them into one single network. These fusion methods for network data are relatively robust to noise and bias so that weak similarities are eliminated but the strong ones are preserved. In [88], similarity network fusion (SNF) is applied to fuse diverse types of genome-wide data and identify cancer subtypes. In [90], another algorithm called PCIA, which concatenates both topological and attribute network for clustering. Moreover, deep networks also have been successfully applied to feature learning from multiple modalities [110]. Though effective to some extent, these algorithms are not very well suited for the task mentioned in this thesis, i.e., discovering interesting groups, which are latent representations of multi-domain network data. Despite some fusion methods have been proposed, the effectiveness of them, especially when they are using in the task of network community detection, is not satisfying. Since equalized edge values usually result in a poor clustering result of the fused network, the algorithm also needs overcome the clustering difficulty if new edge values approach to the same after the fusion step.

To address the above challenges, in this section, we develop a new deep fusion method named Deep Multiple Networks Fusion (DMNF), to perform the task of learning latent communities in network data. We propose to treat the finding eigenvectors of a fused network as deep representing the process. More precisely, the deep representation can be used as a tool for discovering hidden relations under the fused network. Based on this idea, we first introduce a test statistical approach to eliminate the irrelevant attribute values. Second, to fuse the topology-based and content-based networks to a comprehensive network, we perform a nonlinear method that iteratively updates every network for making them more similar. Finally, we feed the fused network into a sparse stacked Auto-Encoder, in which we seek the best non-linear network representations that can approximate the input network. In Fig. 15, the procedure of the proposed method is shown. From the best of our knowledge, this is the first attempt that makes use of deep learning to deal with network fusion. The traditional fusion methods, including the

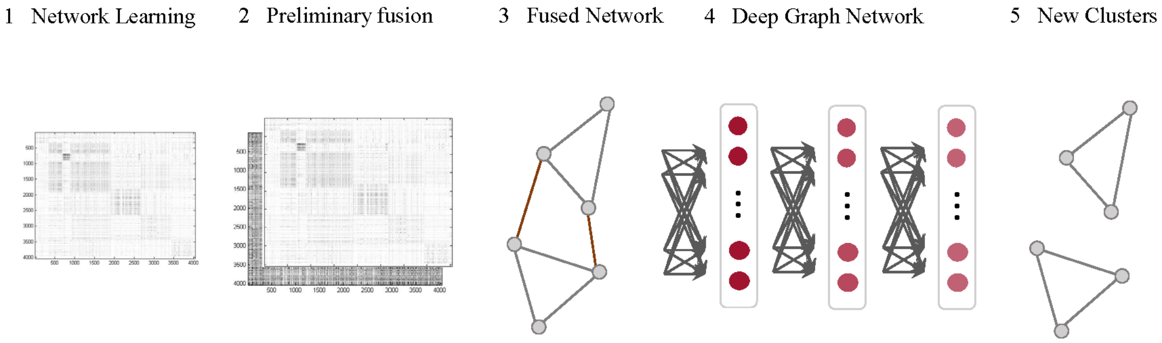


Fig. 15. The procedure of DMNF. (1) Example users similarity matrix of content expression and topology expression for the same users of social network. (2) Preliminary fused matrix that constructed by matrix fusion step (3) Users are represented by nodes and fused similarities are represented by edges. (4) Deep autoencoder eliminate edges by generating deep graph representations. (5) The final clustered network.

multimodal deep learning method, directly combine the features of various modes into a larger set of features. Even if they are fused in the deep network, it is still hidden in the hidden layer to combine multiple sets of features. Different from such methods, network fusion firstly builds the corresponding approximation network for the data of each mode and transforms the feature expression into network representation. The expression form of n*m is transformed into the network expression matrix of n*n. Next, each edge in the network is searched for its comprehensive representation based on multiple network edges.To evaluate the performance of the proposed method, we have performed experiments using real social network data. Experimental results show that DMNF outperforms most algorithms, which either or not are fusion-based. The communities detected by DMNF are better matching with the ground truth ones. Overall, the contributions of this section can be summarized as the following.

- We present a new model to learn comprehensive view which involves multiple networks, while most of the existing approaches do this relying on the one kind of network.
- The proposed model captures deep representations of the fused network that is able to generate deep relations contain both topology and content relations.
- The proposed model may avoid noise and ensure that the attribute values it considers for mining of interesting subgraphs are the relevant and interesting ones.

### 4.2.2 Method

Our method undergoes three main computational steps: 1) the network learning, which is used to extract two networks, representing the interrelationship concerning topology and content, between pairwise vertices. 2) the network fusion, which may generate a similarity network between vertices, based on the two original ones obtained in the first step. 3) the deep representation of the network, which introduces the stacked Auto-encoder for obtaining new representations instead of learned fusion network. New representations are statistically significant solutions to the sparse and large network. Then, in the network clustering step, we use these discovered new space sets as input for assigning network vertices into communities.

#### **4.2.2.1** Network Learning

For each vertex, different attribute values may have different contributions to the learning of network data. Interesting attribute value pair also influences the network learning. For example, an education degree could be a good attribute for grouping users with similar institutions in a social network. Hence, we like to make use of node attribute values relevance to eliminate the irrelevant content information while generating new nodes similarity within relevant internal attributes for the network learning task. To discover all such kinds of associated pairwise attribute values, we bring in a residual analysis approach from [111] to make a reasonable statistics to prove whether there is a relevance of $attribute\ value_p$ and $attribute\ value_j$ . Let $cor_{pj}$ be the number of connecting nodes that have $attribute\ value_p$and $attribute\ value_j$, as shown in Table 8. In this table, it shows the number of connections connecting pairwise vertices characterized by each pair of attribute values in the social network. Given this table, we may further define:

$$exp_{pj} = \frac{cor_{p+}cor_{+j}}{T} \tag{5}$$

as the expected frequency that $attribute\ value_p$and $attribute\ value_j$ are connected in the network, where

$$cor_{p+} = \sum_{k=1}^{n} cor_{pk} \tag{6}$$

$$cor_{+j} = \sum_{k=1}^{m} cor_{kj} \tag{7}$$

$$T = \sum_{m,n} cor_{mn} \tag{8}$$

We bring in an approach from [111] to make a reasonable statistical test to prove whether there is a relevance of two attribute values:

$$R_{pj} = \frac{z_{pj}}{\sqrt{(1-\frac{cor_{p+}}{T})(1-\frac{cor_{+j}}{T})}} \tag{9}$$

where $(1-\frac{cor_{p+}}{T})(1-\frac{cor_{+j}}{T})$ is defined as the adjusted term of $z_{pj}$.

$$z_{pj} = \frac{cor_{pj} - exp_{pj}}{\sqrt{exp_{pj}}} \tag{10}$$

In previous works, $R_{pj}$ has been shown to follow a standard normal distribution [111]. If the value of $R_{pj}$ is larger than 1.96, it would be considered there is a correlation between $attribute\ value_p$ and $attribute\ value_j$, at a 95% confidence level. We use these to evaluate whether there is a strong association between each pair of attribute values. After determining the association between attribute values, we then move to the similarity problem base on such associations. Let $G = \{V, E\}$ represent a network, where $V = \{u_1, u_2, \ldots, u_n\}$ is a set of $n$ vertices representing all the users in the network, and the $E = \{e_{ij}\}$ is the edge set containing the edges between pairwise vertices, and their values represent how similar these vertices are. To achieve a content-based similarity matrix of users, we assess nodes similarity by use of Jaccard similarity. That is to say, edge weights of the content network are constructed by a $n \times n$ similarity matrix $L$ and $L(i, j)$ representing the similarity between nodes.

Table 8. Observed Atttibute Values Co-occurrence

| | **ar$_1$** | **ar$_2$** | **…** | **ar$_j$** | **…** | **ar$_s$** |
|---|---|---|---|---|---|---|
| **ar$_1$** | cor$_{11}$ | cor$_{12}$ | … | cor$_{1j}$ | … | cor$_{1s}$ |
| **ar$_2$** | cor$_{21}$ | cor$_{22}$ | … | cor$_{1j}$ | … | cor$_{2s}$ |
| **…** | … | … | … | … | … | … |
| **ar$_p$** | cor$_{p1}$ | cor$_{p2}$ | … | cor$_{pj}$ | … | cor$_{ps}$ |
| **…** | … | … | … | … | … | … |
| **ar$_r$** | cor$_{r1}$ | cor$_{r2}$ | … | cor$_{rj}$ | … | cor$_{rs}$ |

For the topological network, we observe that link matrix is very sparse. As cosine similarity is very efficient for evaluating the interrelationship between sparse vectors, we propose to use a link-based similarity measure to evaluate the topological similarity between pairwise vertices in the social network. The measure is motivated by the previous work developed for the genome-wide data analysis [88]. Let $E_t$ denotes the set of $\mathrm{n}_E$ edges representing all the links in a network. If there is an edge $\mathrm{e}_{ij} \in E_t$, it means that the two vertices $u_i$ and $u_j$ are connected through the link of $e_{ij}$. That is to say, edges are represented by an $n \times n$ adjacent matrix $M$ with $M(i,j)$ indicating the existence of a link between pairwise nodes. Cosine distance has been proven to be useful in calculating the distance for binary variables. Hence, the link-based similarity matrix $L(i,j)$ is the topological weight of the edge between user $i$ and user $j$ which is computed as:

$$L(i,j) = \exp\left(-\frac{\rho^2(u_i,u_j)}{\mu\varepsilon_{i,j}}\right) \tag{11}$$

where

$$\rho(u_i,u) = 1 - \frac{\sum_i u_i u_j}{\sqrt{\sum_i u_i^2}\sqrt{\sum_j u_j^2}} = 1 - \frac{\langle u_i,u_j\rangle}{\|u_i\|\|u_j\|} \tag{12}$$

and

$$\varepsilon_{i,j} = \frac{\overline{\rho(u_i,U_i)}+\overline{\rho(u_j,U_j)}+\rho(u_i,u_j)}{3} \tag{13}$$

The $\rho(u_i,u_j)$ denotes the cosine distance between nodes $u_i$ and $u_j$, $\varepsilon_{i,j}$ is a scale adjusting function, $\mu$ is a hyperparameter and $\overline{\rho(u,U)}$ is the mean value of the cosine distances between a node and each of its neighbors.

#### **4.2.2.2** Network Fusion

Following the criteria defined for the desired content similarity network and link similarity network, we now formulate a fusion problem aiming at combining the content network and topological network. To achieve a fused network from heterogeneous ones, we use a normalized weight matrix $K = D^{-1}L$ as the full kernel on the vertex set $V$ [88]. $D$ is a diagonal matrix that entries$D(i,i) = \sum_j L(i,j)$, and $\sum_j K(i,j) = 1$. To arrive a better normalization, our approach looks at making this degree of the self-similarities on the diagonal entries of $L$, and keep $\sum_j K(i,j) = 1$:

$$K(i,j) = \frac{L(i,j)}{2\sum_{k \neq i} L(i,k)} \tag{14}$$

subject to the constraints: $i \neq j$ ,otherwise $K(i,j) = \frac{1}{2}$. Let denote $N_i$ as a set of $u_i$'s neighbors including $u_i$ in $G$. We then measure local affinity by using K nearest neighbors (KNN) as follows:

$$Q(i,j) = \frac{L(i,j)}{\sum_{k \in N_i} L(i,k)} \tag{15}$$

subject to the constraints: $j \in N_i$ ,otherwise $Q(i,j) = 0$

Let $K_{t=0}^{(v)}$ represent the initial $v$ status network, $Q_{t=0}^{(v)}$ represents the kernel matrix, and m represents how many networks we should consider. In this work, the $m = 2$ as we have a link-based network and content-based network. The fusion step is iteratively updating network corresponding to each of the data types:

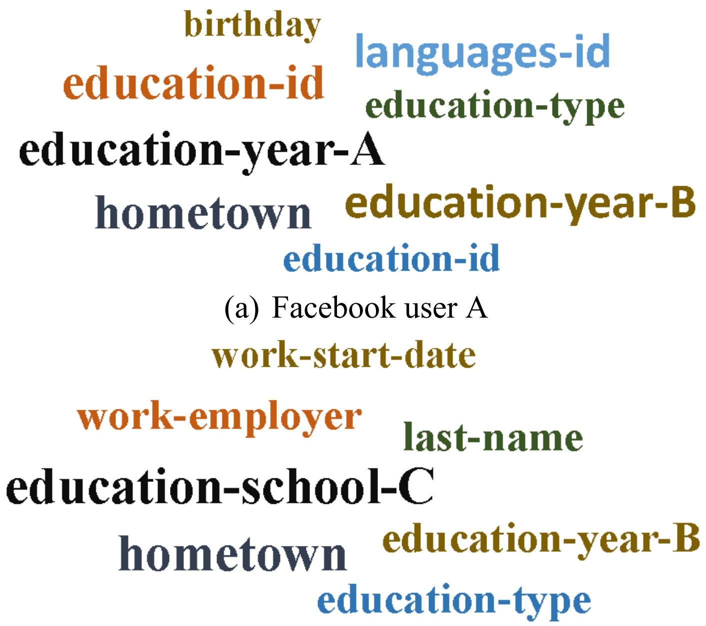
$$K^{(v)} = Q^{(v)} \times \left(\frac{\sum_{k \neq v} K^{(k)}}{m-1}\right) \times (Q^{(v)})^T, v = 1,2,3, \dots, m \tag{16}$$

We perform (14) on $K^{(v)}$ after each update for a normalization. We would run $v$ parallel interchanging diffusion processes by updates the status network each time in the above step. At last, we achieved the final status network:

$$K^{(v)} = \frac{\sum_{v=1}^{m} K^{v}}{m} \tag{17}$$

**4.2.2.3** Deep Graph

Let $G_f = (V_f,\ E_f)$ be the new fused graph with its similarity matrix S, the vertex set $V_f = \{v_{f1}, \dots, v_{fn}\}$ representing all the nodes. Two vertices have a fused similarity $s_{ij}$ between the two nodes, and $s_{ij}$ can be obtained by the fused method in the previous section. To infer new representation of the normalized fused network, we deploy stacked auto encoder [85, 112] to learn a non-linear embedding of the fused network since it has a similar optimizing path with spectral clustering, as well as it can also eliminate edges with lower degrees of similarity. Besides, according to work in [112], the computational



(a) Facebook user A

(b) Facebook user B

Fig. 16. Attributes of two Facebook users A and B in Facebook Dataset.

complexity of Auto-Encoder is lower than original spectral clustering in the perspective of capturing useful low-dimensional representation. We use the $n$ columns of normalized $S$ as $n$ nodes to Auto-Encoder. We know that optimization target of Auto-Encoder is to minimize the approximate error between reconstructed output $h_{W,b}(x) \approx x^{(i)}$ and inputted $D^{-1}S$. A complete deep neural network is stacked up by a layer of sparse auto-encoders. Finally, we introduce K-means to identify clusters of nodes from the fused deep represented network.

Table 9. Experimental performance of different approaches in social network data

| Methods | | NMI | |
|---|---|---|---|
| | | Caltech | Facebook |
| Fusion | DMNF | **0.337** | **0.610** |
| | CESNA | 0.221 | 0.58 |
| | RTM | 0.277 | 0.592 |
| | MISAGA | 0.310 | 0.573 |
| | ECDA | 0.156 | 0.353 |
| Non-fusion | SC | 0.305 | 0.569 |
| | AP | 0.279 | 0.580 |

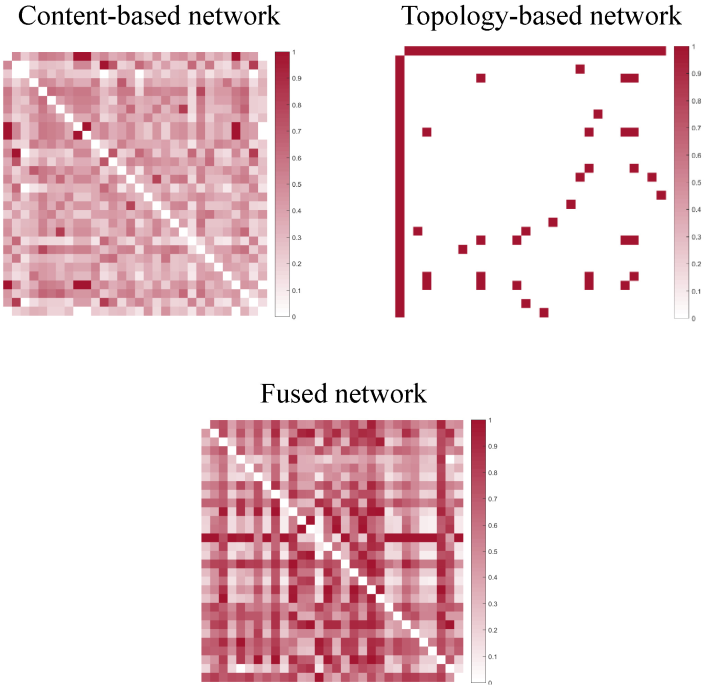
### 4.2.3 Experiment

In this section, we experimentally evaluate the proposed deep fusion algorithm with the application of social network clustering, using two sets of real social network data. The used datasets include Caltech and Ego-Facebook.

#### **4.2.3.1** Experimental Datasets

We perform experiments on two data collections, including Caltech [115] and Ego-Facebook [167]. Though both of these two datasets are related to social networks, the number of users, the social ties, and the attribute values that are used to characterize the social users are quite different from one to the other. The characteristics of these two datasets can be seen as the following.

Caltech: This is a social network dataset which is collected based on the relationship between social network users at California Institute of Technology. The truth community affiliation for each user is obtained by identifying the dormitory affiliation. The Caltech



Content-based network

Topology-based network

Fused network

Fig. 17. Sample of Fused Network obtained from topological and content network

dataset consists of a network of 769 vertices representing 769 social network users. It has 16656 edges connecting these users. There are 53 attribute values associated with the vertex that are used to represent the user profiles. In this social network dataset, there are 35 ground truth communities.

Ego-Facebook: The Ego-Facebook dataset is much larger than the Caltech dataset, as there are 4039 vertices and 88234 edges, representing the social network users, and social ties between them, respectively. Besides, 1283 attribute values are used to characterize users. In this social network dataset, there are 193 ground truth communities identified in the previous work [167]. Instead, we selected two nodes and showed their attribute values in Fig. 16. From Fig. 16, we noted these two users have some attributes in common which influence their similarity calculation.

#### **4.2.3.2** Comparison Methods and Performance Evaluation

To test the performance of DMNF, we selected several methods as compared baselines. Based on their features, they can be categorized as fusion-ones and non-fusion-ones. The details of these used baselines are described as the following.

Non-fusion

Spectral Clustering (SC): Given the assumption that vertices in the same cluster may share similar edge structure, spectral clustering is proposed to group those vertices with similar edge structures by minimizing the objective value of normalized cut of a given network.

AP: This method can find graph clusters by minimizing the distance between vertices and cluster centers in the network. These two no-fusion approaches are proposed to identify subgraphs by taking into consideration the information from the single network, e.g., the one containing the connections between vertices.

Fusion

CESNA: This is a method for learning latent representations of networks that are based on a generative process. By taking into the consideration both network structure and content information of each vertex, CESNA can find the cluster affiliation for each vertex by maximizing the posterior probabilities representing how possible each pair of vertices are connected.

RTM: RTM is a topic-modeling-based method which considers both graph topology and attributes values when detecting clusters in the network data.

ECDA: This is an effective method for social community detection. ECDA makes use of an evolutionary optimization approach to maximize the edge density within each cluster while minimizing that between clusters, when it performs the task of community detection in social networks.

MISAGA: MISAGA is an algorithm that is able to learn latent network representations by taking into consideration both edge structure and attribute information. The latent structures can be identified by solving a constrained optimization problem. These three fusion approaches are all proposed to identify interesting subgraphs considering both network topology and content information.

To quantitatively evaluate the performance of the proposed model and other basslines, we use the normalized mutual information (NMI) as the metrics. NMI is a very prevalent measure for evaluating the performance of clustering algorithms. It evaluates the average matching rate between identified clusters and ground truth ones. Assuming that $\mathrm{T} = \{\mathrm{T}_f\}$ $(1 \le \mathrm{f} \le \mathrm{k})$ is the expected result, NMI is defined as:

$$NMI = \frac{\sum_{f_1=1}^{k} \sum_{f_2=1}^{k} n_{C_{f_1}, T_{f_2}} \log(\frac{n_v n_{f_1, T_{f_2}}}{n_{C_f} n_{T_{f_2}}})}{\sqrt{(\sum_{f_1=1}^{k} n_{C_f} log \frac{n_{C_f}}{n_v})(\sum_{f_1=1}^{k} n_{T_f} log \frac{n_{T_f}}{n_v})}} \tag{18}$$

where $n_{C_f}$ is the number of users in $C_f$ , $n_{T_f}$ is the number of users in $T_f$ , and $n_{C_{f_1},T_{f_2}}$ is the number of users found in both $C_{f_1}$ and $T_{f_2}$. As for the NMI measure, their values are larger if the clustering result $\{C_f\}$ matches better with the expected result T.

#### **4.2.3.3** Experimental Results in Social network data

The experimental results of different algorithms on the two social network datasets are shown in Table 9. As it is shown in the table, DMNF outperforms all the other baselines in both two datasets. For the Caltech dataset, DMNF outperforms MISAGA, which ranks the second best, with 10%, when NMI evaluates them. In the Facebook dataset, DMNF is better than second-best (RTM) by 3%. Given these obtained results, it is said that DMNF is a practical approach to learning latent representations of network data. Besides evaluating the clusters detected by DMNF using the NMI measure, we also considered finding whether there is some difference between the fused and non-fused networks. To show this difference, we draw the matrix representations of a topology-based network, content-based network, and fused network in Fig. 17. The depth of the color represents

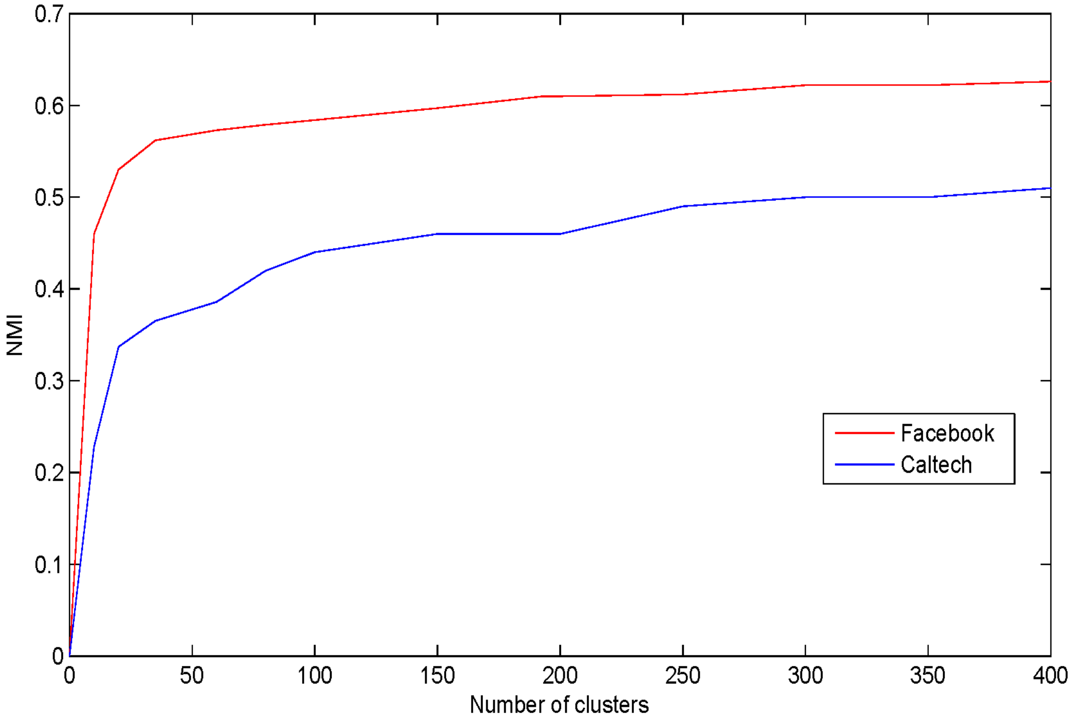


Fig. 18. Clustering Results on Facebook and Caltech Dataset

the edge values of the fused network. As it shows in the figure, DMNF can use much more information that represents the interrelationship between vertices for learning the latent representations. This might be the reason why DMNF may outperform other baselines in our experiments.

To identify the effect that is brought by the setting of cluster numbers, we used DMNF to detect clusters in the two mentioned social network datasets, using different numbers of clusters as input, and evaluate the discovered clusters using NMI. The variations of NMI against different settings of cluster numbers are shown in Fig. 18. As this figure shows, NMI can be maximized, when the number of clusters is set to approximate to that of ground truth communities. Based on the results obtained, it is seen that DMNF is a very promising approach for detecting latent communities in social network data.

### 4.2.4 Summary

In this chapter, a fusion-based approach, DMNF is proposed to detect latent communities in the social network data. The proposed method addressed the following key challenges in network fusion. One is how to fuse heterogeneous domains of social network data as a single network. The other is how to reduce the computational complexity caused by the high dimensionality of the data vectors representing the characteristics of the vertices in the social network data. Utilizing a novel network fusion technique, DMNF can find a network which preserves information from different domains. Then, DMNF finds the deep representations of the fused network by making use of stacked Auto-Encoder. Given the deep fused representations of each vertex in the network, DMNF finds the latent communities by performing k-means clustering which can minimize the reconstruction error for the original fused network. By replacing the step of finding k largest eigenvalues of the normalized fused network, DMNF makes use of the deep neural network to find low-dimensional representation vectors into which both local and global properties of the vertices in the network are embedded. Utilizing these vectors, DMNF can outperform most prevalent approaches when they perform the task of learning latent representations, which in other words cluster in the social network data.

DMNF has been tested with two sets of real-world social network data. The performance shows that DMNF may identify the social communities with higher accuracy, compared with other baselines.

## 4.3 Discovering an Integrated Network in Heterogeneous Data for Predicting Drug-Target Interactions

Many computational approaches have been developed to predict drug-target interactions (DTIs) perform their tasks based them on their similarity network. However, such methods do not allow multiple networks to be considered through the rapid development of techniques results in a growing diversity of biological network data. The numerous domain representations of the DTIs in the networks are usually ignored. Therefore, a more in-depth understanding of latent knowledge representing the DTIs network can be learned by combining the insights obtained from multiple, diverse networks. The comprehensive predicting of DTIs is highly desired for one to gain deep insights into both fundamental drug discovery processes and the system biology.

### 4.3.1 Overview

Some DTIs related information like sequence, structure, side-effects, and function of proteins have been collected to public databases. For example, there are hundreds of thousands of human proteins that are recorded in the UniProtKB database [21]. On the other hand, there are around thousands known drug compounds are deposited in Drug Bank [36]. Other databases such as Super Target and Matador [37], Comparative Toxicogenomics Database [40] and the SIDER database [51] have been designed as resources for drug and protein functions. These emerging public databases allow access to lists of many known proteins and drugs. Therefore, the existed a huge number of unexplored compounds and human proteins make it impossible to evaluate drug-target interactions effectively by biological experiment. Finely-set small-scale experiments are not only very expensive but also inefficient to identify numerous interactomes despite their high accuracy. Normal drug discovery processing may generate products different from the original treatment. Instability and no specificity of drug-target interactions have to be addressed appropriately during the screening and clinical phase. To reduce the huge time and financial cost of experimental approaches, many computational models have been built to elucidate interesting drug-target relationships of most promising candidates for further experimental validation. Various methods cared about drug similarity and drug-target nature representations respectively [54].

A kind of popular solutions is feature vector-based methods that face drug and protein features straight-forward [67]. They can uncover the description of the hidden knowledge regarding meaningful representations and then generate rules to repeat experts' decision process. These methods provide biological representations for learning interest patterns such as compound subset and protein subspace [84]. But, current feature-based methods are difficult to handle incomplete knowledge. For example, some protein expression like protein-protein interactome mapping hasn't been fully discovered [113].

Network-based solutions are developed to identify biological interaction by including the similarity matrices of related entities. Lots of computational approaches have been proposed to discover DTIs based on their compound and protein sequence similarity [116-120]. An attractive alternative approach is to integrate various descriptions of drug-target from multiple sources in a statistical learning framework. The comprehensive predicting of drug-target interactions (DTIs) is highly desired for one to gain multi-insights into both fundamental drug and protein function, yet the desire to further extract useful knowledge from these data leads to the problem of multiple similarity network fusion. Network approaches prove to be highly effective in addressing this problem [119]. However, their performance deteriorates significantly on two main stumbling blocks. One is incomplete putative related networks, and the other is each domain having multiple network representations.

There have also been several works of network fusion from multiple domains [88, 90]. These approaches adopt multiple data types of the same sample to generate similarity networks and then fuse them into one single network. Such fusion method is robust to noise and collection bias so that weak similarities are eliminated, and strong similarities are preserved. In [88], similarity network fusion is applied to fuse diverse types of genome-wide data and identify cancer subtypes. In [90], another algorithm concatenates both topological and attribute network for PPI complex clustering.

In this study, we develop a new network-based method called DFNet (integrated network representation through deep multiple network fusion) that motivated by the success of network fusion approaches to the problem of constructing networks into one network. DFNet predicts DTIs from multiple network's deep representations with inductive network completion aiming at training a robust model to rely on known interactions. Firstly, we introduce the non-negative matrix factorization (NMF) to complement the unreliable similarity network. Several drug and protein functions about nature features have been used to construct a similarity network as well. Secondly, to solve the difficult problem of multiple networks understanding, we perform a nonlinear method that iteratively updates every network for making them more similar. Finally, we feed the fused network into a sparse stacked Auto-Encoder, in which we seek the best non-linear network representations that can approximate the input network. To the best of our knowledge, this is the first attempt that makes use of deep learning to deal with the biological network fusion. To evaluate the performance of the proposed method, we have performed experiments using real social network data. Experimental results show that DFNet is better able to identify DTIs more accurately when compared with the state-of-the-art network fusion algorithm due to its considering of deep fused information.

### 4.3.2 DFNet in details

Suppose that we are given several drug similarity networks and protein similarity networks constructed from multiple domain information and we want to predict unknown interactions between the compounds and target proteins on a genome-wide scale. The proposed method consists of three steps: (i) the network fusion step, which introduces a fusion method fuses the similarity network obtained from multiple drug or protein information. (ii) the deep representation step, which introduces the stacked Auto-encoder for obtaining new representations instead of learned fusion network. (iii) New representations are statistically significant solutions to the sparse and large network. Then, we infer the unknown drug–target interactions based on the projection distance of their new representations.

#### **4.3.2.1** Constructing High Reliable Drug and Protein Networks

Let $G = \{V, E\}$ represent a network with a two-element tuple, where $V = \{v_i\}$ $(1 \leq i \leq n_v)$is a set of $n_v$vertices representing all the nodes in a network, and the edges $E = \{e_{ij}\}$ are weighted by how similar the nodes are. That is to say, edge weights of drug or protein network can be represented by an $n \times n$ similarity matrix $M$ with $M(i,j)$ indicating the similarity between nodes. For some drug and protein related similarity network, we observed that the matrix is very sparse, and the information that we obtain is incomplete as protein related diseases are not fully discovered. Since matrix factorization is efficient to complement matrix, we propose to use NMF [122] for similarity network completion. The target of NMF is to find an approximate factorization $V \approx WH$. The overall cost function of NMF can be defined using Kullback-Leibler divergence to measure the distance between two non-negative matrices [123]. We use matrix $M$ to denote one of the original similarity matrices. This matrix first approximately factorized into an $n\ x\ r$ matrix $W$ and an $r\ x\ n$ matrix $H$, and then we consider following formulations as optimization problems:

$$\min D_{KL}(V||WH) =$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\left(v_{ij}\ln\frac{v_{ij}}{[WH]_{ij}} - v_{ij} + [WH]_{ij}\right) \tag{19}$$

subject to the constraints $W, H \geq O$.

Where

$$W_{ik} \leftarrow W_{ik}\frac{\sum_{j=1}^{J} H_{kj}v_{ij}/[WH]_{ij}}{\sum_{j=1}^{J} H_{kj}} \tag{20}$$

and

$$H_{kj} \leftarrow H_{kj}\frac{\sum_{i=1}^{I} W_{ik}v_{ij}/[WH]_{ij}}{\sum_{i=1}^{I} W_{ik}} \tag{21}$$

The purpose of this step is to insert known values for these entries, then perform NMF, producing $W$ and $H$. Then, we can compute $WH$ as our estimate of original similarity network, and now have estimates for the missing information

#### **4.3.2.2** Network fusion and matrix completion for predicting drug–target associations

Above the similarity network from multiple sources are inherently correlated, and sometimes provide complementary information to each other. As a result, data fusion has been paid much attention to, which mainly aims to generate most similar representation between entities under the existing domains. Following the criteria defined for the desirable drug and protein similarity networks, we now formulate a fusion problem by combining several networks. This procedure is following in section 3.2. We first consider stack Sparse Auto-Encoders layer by layer to form a whole deep neural network. And then we apply inductive matrix completion [121] [124-125] to predict DTIs.

Formally, let $X = \left[x_1, \ldots, x_{N_d}\right]^T$, $x_i \in R^{f_d}, i = 1, \ldots, N_d$ represent a fused network of the drugs, where each row *i* represents the corresponding feature vector of drug and $N_d$ stand for the numbers of drugs. That is to say, we can use $Y = \left[y_1, \ldots, y_{N_t}\right]^T$, $y_i \in R^{f_t}, i = 1, \ldots, N_t$ to denote the corresponding feature vector of target and $N_t$ stand for the numbers of targets. In particular, $X \in R^{N_d \times f_d}$ and $Y \in R^{N_t \times f_t}$ are generated from the final status matrix of the network fusion section. Let *A* be a drug–target interaction matrix, where each entry $A_{ij} = 1$ if drug *i* is known to interact with target *j*, and $A_{ij} = 0$ otherwise. To infer unknown drug–target interactions in *A,* we deploy a bilinear function to learn the projection matrix *P* between drug space and target space. Generally, the bilinear function can be defined as:

$$XPY^T \approx A \tag{22}$$

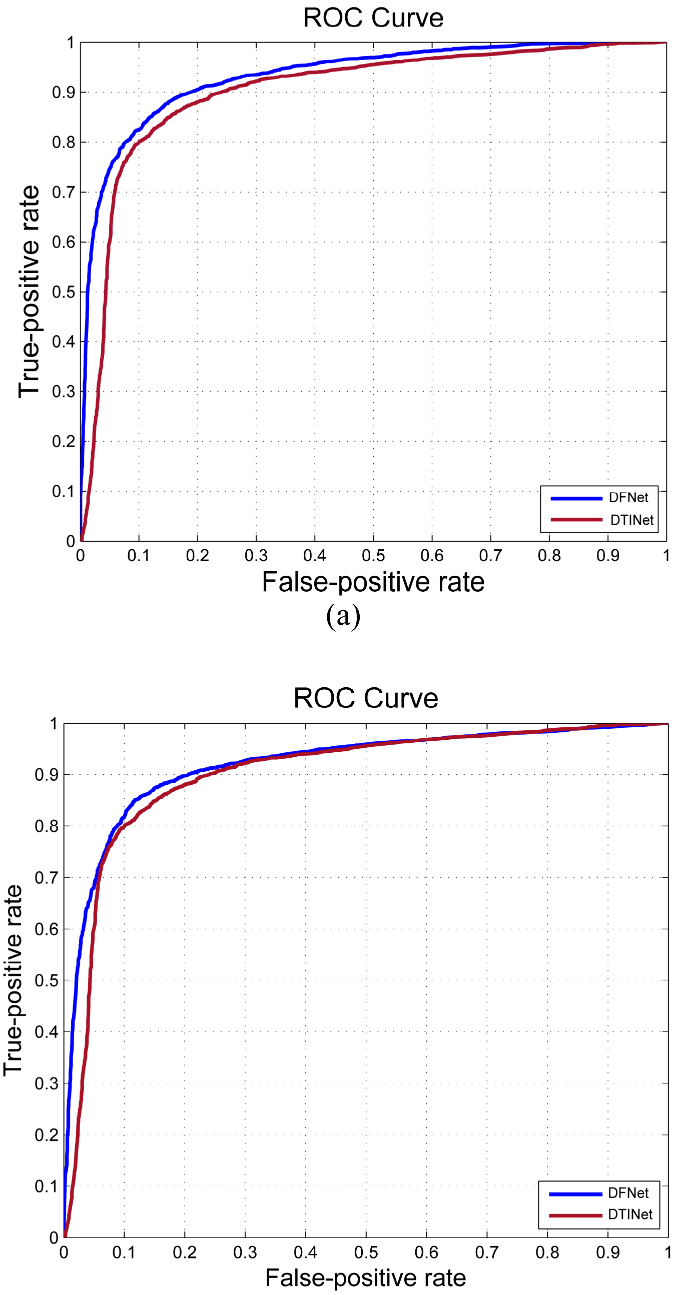
where $A \in R^{N_d \times N_t}$ denoted as the known drug–target interaction matrix and $P \in R^{f_d \times f_t}$ $R_{fd\_ft}$ is the projection matrix that we need to learn. We then measure the possibility of binding each pair of drug–target to determine whether drug *i* is more likely to interact with target *j*:

$$\mathrm{score}(i,j) = \mathrm{x}_i P \mathrm{y}_j^T \tag{23}$$

Obviously, the higher score means a greater chance of drug–target will interact with each other. We know that there's a significant correlation between the feature vectors of drugs or targets and they're geometrically close in space. So we have a chance to greatly reduce the number of effective parameters required in *P* to model drug-target interactions. Based on this idea, we apply a low-rank constraint on P by considering a low-rank decomposition of the form:

$$P = WH^T \tag{24}$$

where $W \in R^{f_d \times f_t}$ and $H \in R^{f_t \times f_t}$. We set up such function to learn a small number of latent factors. This kind of low-rank constraint not only alleviates the problem of



(a)

(b)

Fig. 19. Comparison of the ROC curves of DFNet and DTINet on two DTIs datasets. (a) The original dataset (b) the dataset removed a part of homologous proteins

overfitting, but also automatically separates the noise and clean data, and also has great benefits to the optimization process [125]. We assume that the interactions matrix is generated by applying feature vectors associated with its row as well as column entities to a low-rank matrix $P$. A standard relaxation of the low-rank constraint is to minimize the trace norm of the (24). That is to say, minimizing the trace-norm of (24) is equivalent to minimize:$\frac{1}{2}(\|W\|_F^2 - \|H\|_F^2)$. Therefore, factoring $P$ into $W$ and $H$ are obtained as solutions to the following optimization problem by alternating minimization:

$$\min_{W,H} \sum_{(i,j)} \left\|A_{ij} - x_i W H^T y_j^T\right\|_2^2 + \frac{\lambda}{2}(\|W\|_F^2 - \|H\|_F^2) \qquad (25)$$

### 4.3.3 Experiments

Table 10. Comparison for the Auroc, Auprc, Mcc And F-Measure Values of the DFNet versus DTINet on the two dataset

| Measures | DFNet | | DTINet | | MFDR | |
|---|---|---|---|---|---|---|
| | Original dataset | Removed dataset | Original dataset | Removed dataset | Original dataset | Removed dataset |
| AUROC | **0.929** | **0.905** | 0.908 | 0.88 | 0.898 | 0.851 |
| AUPRC | **0.946** | **0.932** | 0.925 | 0.902 | 0.905 | 0.882 |
| ACC | **0.902** | **0.881** | 0.855 | 0.840 | 0.823 | 0.817 |
| MCC | **0.8208** | **0.801** | 0.748 | 0.731 | 0.723 | 0.714 |
| F-MEASURE | **0.8917** | **0.866** | 0.844 | 0.829 | 0.822 | 0.806 |

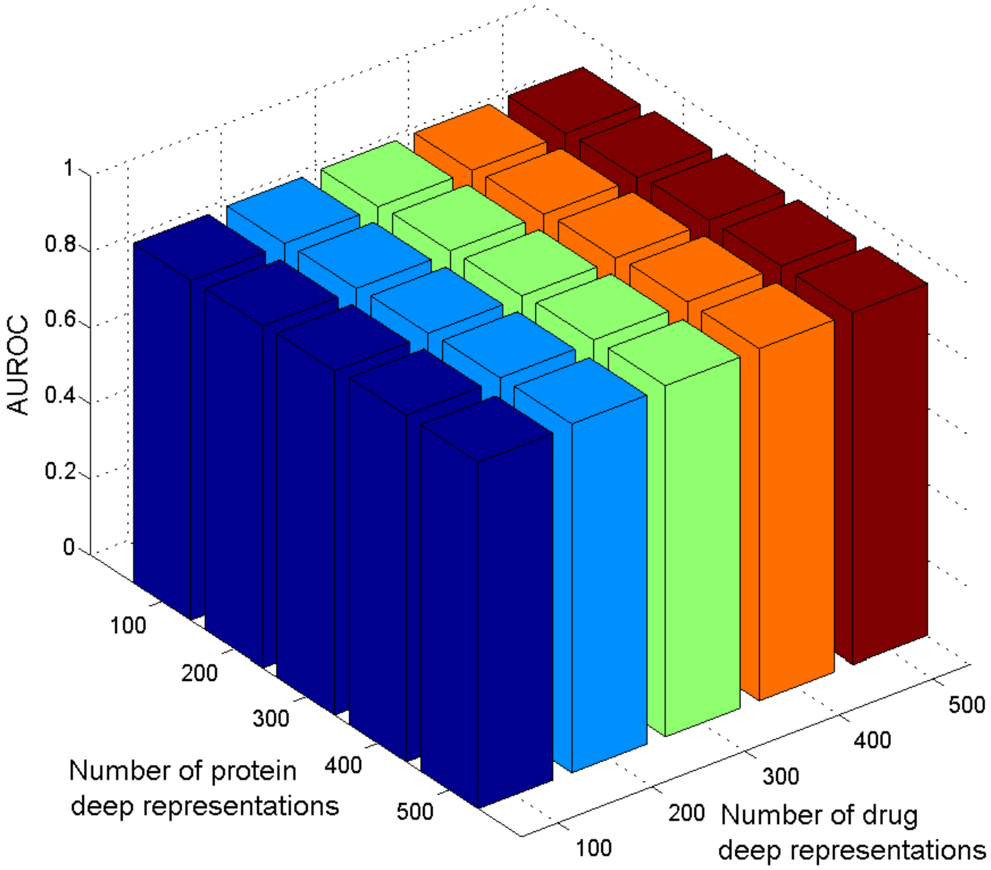


Fig.20. Area under ROC of DFNet on original dataset with different drug and protein representations

In this study, the drug and protein networks which we used to predict DTIs come from [121]. There are two different scale datasets: the original dataset and removed dataset. The fewer number one is reconstructed by removing the homologous proteins with high sequence identity scores from the larger one. In the removed dataset, only the DTIs of the original dataset which proteins conditioned the low sequence identity scores can be retained. There are 708 drugs and 1512 protein in both datasets and we observed the number of known DTIs in original dataset and removed dataset are 1923 and 1332 respectively. All the drugs were extracted from the DrugBank database, their related disease information was extracted from the Comparative Toxicogenomics Database and their side-effect profiles were collected from the SIDER database. All the proteins include the protein-protein interactions were extracted from the HPRD database [126] and the protein related disease associations were also extracted from Comparative

Toxicogenomics Database. The data which are used to represent 708 drugs include four types of expressions: drug-drug interactions, drug-disease associations, drug-side-effect associations and their chemical structures. That is to say, we would construct four 708×708 similarity network for drugs. The data which are used to represent 1512 proteins include three types of expressions: protein-protein interactions, protein-disease associations, and their sequences. That is to say, we would construct three 1512 ×1512 similarity network for proteins.

#### **4.3.3.1** Evaluation measures

Regarding imbalance prediction performance evaluation, it has argued for using ranking measures like AUC (area under the ROC curve) that the prediction performance can be safely unbiased [127]. We infer interactions and compare against the held-out interactions, measuring performance using the AUC for our evaluations. ROC curves are created by plotting the true positive rate versus the false positive rate at various thresholds. The results are shown as a ROC curve where TPR is plotted against FPR, calculated as follows TPR=TP/TP+FN , FPR=FP/FP+TN, where TP (true-positives) is the number of correctly predicted drug-target interactions while the FP (false-positives) is the number of not correctly predicted drug-target interactions. TN (true negative) is the number of drug-target interactions predicted not to be in a class that is not observed in that class, and FN (false negative) is the number of drug-target interactions predicted not to be in a class that is observed in that class. To further evaluate the performance of the proposed method, we also use the several more measure like following criteria: the overall prediction accuracy, recall, precision, Matthews's correlation coefficient (MCC) and F-measure were calculated.

#### **4.3.3.2** Compare with State-of-the-art Approach

To evaluate the performance of the proposed prediction method, we also applied the DTINet [28] to predict DTIs from multiple similarity networks. To the best of our knowledge, there is only one network fusion method for DTIs prediction so far. Regarding comparison theory, we examined the learned deep fusion representations in the same inductive matrix method with DTINet. DTINet's deep learning module was proposed based on MFDR. Therefore, we also introduced the MFDR model based on drug chemical molecular information and protein sequence expression, and made a comparison. This comparison can help us prove that the network fusion method is better than the single network method. Hence, the experimental result reflects the effect of the proposed method. Due to our DTIs dataset is a high-class imbalance, we used 10-fold cross validation, where each fold leaves out 10% of the positive and negative samples for testing. Since the discovered positive samples are too small that may lead to imbalance bias if randomly divide the dataset. We randomly sampled known interactions and negative pairs and divided both into each fold equally.

Table 10 reports the accuracies of different algorithms on both original dataset and removed dataset. The resulting Auroc for DFNet and DTINet on the original dataset are 0.929 and 0.908, respectively. The resulting Auroc for same approaches on the removed dataset is 0.905 and 0.88, respectively. We got a 2% increase in the ROC score. The scores of AUPRC, ACC, MCC, and F-MEASURE also be listed in Table 10. We further compared the performance of each method by the ROC curve. Figure 19 shows the ROC curves of the two algorithms on the original dataset and removed dataset. As expected, among all measures, DFNet achieves the higher score. This result shows that DFNet extracted more meaningful representations to drug and protein from the fused network and improved the prediction performance.

#### **4.3.3.3** Cross-Test for Parameter Setting

In this subsection, we conduct a series of cross-test experiments on the original dataset to evaluate the influence of neurons in output layers parameters in DFNet, which are the dimensions of deep representations of drugs and the dimensions of deep representations of proteins. We test the number of deep representations among {100 200, 300, 400, 500}, proteins also follow the above parameters. The results in Figure 4 show that the accuracy increases gradually as the value of two dimensions increases. And, the accuracy approaches to a stable value from 300 to 500. Figure 20 also indicates that the best parameters for dimensions of drug and dimensions of protein are 400 and 500, respectively.

### 4.3.4 Summary

In this chapter, a network-based fusion approach for DTIs prediction is proposed. The proposed method addressed the following key challenges in biological net-work incompletion and multiple network fusion. One is fusing the diversity of heterogeneous information embedded in the network data, the other is reducing the incompleteness brought by the vertex features in the heterogeneous network data not fully discovered, e.g. Side-effects usually found slowly even if the drug has been listed. DFNet introduces NMF, which first characterizes the higher reliable information of some individual network. And then, we are applying an interchanging diffusion algorithm to multiple networks. In addition, we use stacked Auto-Encoder compute deep representations for each node in the networks to approximate the fused network. This is because both auto-encoder and spectral method minimize the reconstruction error for the original normalized similarity matrix. Deep neural network subtly replaces the step of find k largest eigenvalues of the normalized graph similarity matrix in a spectral procedure. These low-dimensional representations encode both global and local topological properties for nodes in the networks and are readily incorporable for the downstream predictive models. Given the fused deep graph representations, we used an inductive matrix completion for predicting unknown DTIs. We have demonstrated that DFNet can display excellent ability in network integration for accurate DTIs inferring and achieve substantial improvement over the advanced approach. Moreover, experimental results on two real-world networks dataset demonstrate that DFNet able to achieves a good detecting performance.

# 5. INTERPRETABLE REPRESENTATION

Deep representation has been widely used in various fields to improve performance. In predicting outcome after reaching the accuracy of the acceptable, the users want to get the prediction results can be explained. But in some real-world applications, accuracy is not the only criterion. The features and patterns related to the results, as well as their influence on the decision-making, have become the new focus of researchers. Existing machine learning methods are also complicated to be accurate and interpretable as a common optimization goal. In general, the algorithm will be less accurate if it tries to guarantee the interpretability of the results. For example, what small molecules link to protein targets needs to be recognized in drug screening. In the recommendation system, it is also necessary to explain what factors and their combination will influence the strength of the recommendation. Therefore, we propose two interpretable frames for link classification and nodes clustering respectively. Both methods are based on graph analytics to provide an interpretable lens for existing models. By translating the original data into the representation of the graph, we use the interpretable subgraph as the basis for link prediction and nodes clustering. Flexible conversion of the input application can be made with no loss in accuracy and can get information on helpful nodes and edges. We achieve state-of-the-art results on drug and side-effects link prediction and social network users clustering using the two proposed algorithms.

## 5.1 Discovering Graph Representation for SE prediction

### 5.1.1 Overview

The accuracy of prediction can be satisfied in most link prediction tasks, but in some such as medical network and biological network, the interpretability of prediction is of vital importance. Drug side-effects (SEs) can range from mild irritations to serious health problems [128]. The link between the drug and its effects is lethal. According to statistics, millions of patients suffer from various adverse drug reactions every year and 0.1% die as a result. Unlike toxic reactions caused by excessive dosage or long-term medication, SEs might be caused by properties inherent in drug structures, or due to limited selectivity and broad effects of pharmacological actions. Pharmaceutical companies usually have to invest a lot of resources to screen a large number of candidate compounds to identify the most suitable ones to test and then manufacture. The cost for such drug screening can be tremendously reduced if an effective computational approach can be used to better predict SEs based on various properties of drug structures. For this reason, SE prediction for drugs has drawn much attention recently.

To effectively execute this mission, some computational methods have been proposed to discover SEs in drug-related data [129-133]. Empirical laboratory methods for predicting SEs is to investigate into molecular toxicology in the early stages of drug discovery [134]. To speed up the investigation, some machine learning algorithms have been used to attempt to uncover the relationship between SEs and drug molecules, but they cannot be used to identify which sub-component in a drug molecule that is responsible for causing adverse SEs [131]. A useful algorithm should be able to discover significant patterns in drug molecules to allow components that are responsible for generating the SEs to be identified, explicable and understood [133]. And to do that, several machine learning algorithms have been proposed to determine the possible association between drug substructures and drug SEs [134-135]. Unfortunately, the accuracy of prediction of SEs is not high enough. This is because most of those methods ignore the fact that drug reactions are typically found to be associated with groups, rather than individual, molecular substructures. And, a set of explainable patterns extracted from the original molecules to explain the prediction results is the urgent need of pharmaceutical companies. As active small molecules are usually hidden under bigger molecular compounds [128], a method that can take into consideration the sub-structures in a drug molecule is more desirable.

Given the features of previous approaches for predicting drug SEs, they may perform the task by mainly concerning two aspects, i.e., the recognition of global functions, or the extraction of inducements relation. By taking into the consideration of the former aspect, these bottom-up methods may group all drug substructures for discovering SEs, making use of existing techniques, like SVM and ensemble learning [136]. Though different types of information, such as the association between one drug substructure and SE [103, 135, 137-138] might be used, these methods, may not be able to reveal the interrelationship between substructure sets and corresponding SE. While, those methods concerning the latter aspect, are model-based and perform the task by optimizing the objective functions which may measure the discriminants between pairwise drug substructure and SE. For examples, two approaches proposed in [132-133] can predict SEs by fitting the model which may represent the molecular influence. Though these mentioned approaches might be useful to some extent, they do not take into the consideration chemical substructure set when performing the task of SEs prediction. Compared with them, we may discover molecular subgraphs that jointly identify inherent attribute relations and SEs cross relations. The performance of graph mining methods heavily depends on the choice of data representations. Since drug data only provides structured molecular data from which it is hard to learn chemical concepts, we need an appropriate representation learning method for drug data. Many works are also proposed for improving the optimization algorithm of low-rank approximation [139-140]. Therefore, we use a graph description for drug data to achieve good performance and explainable representation.

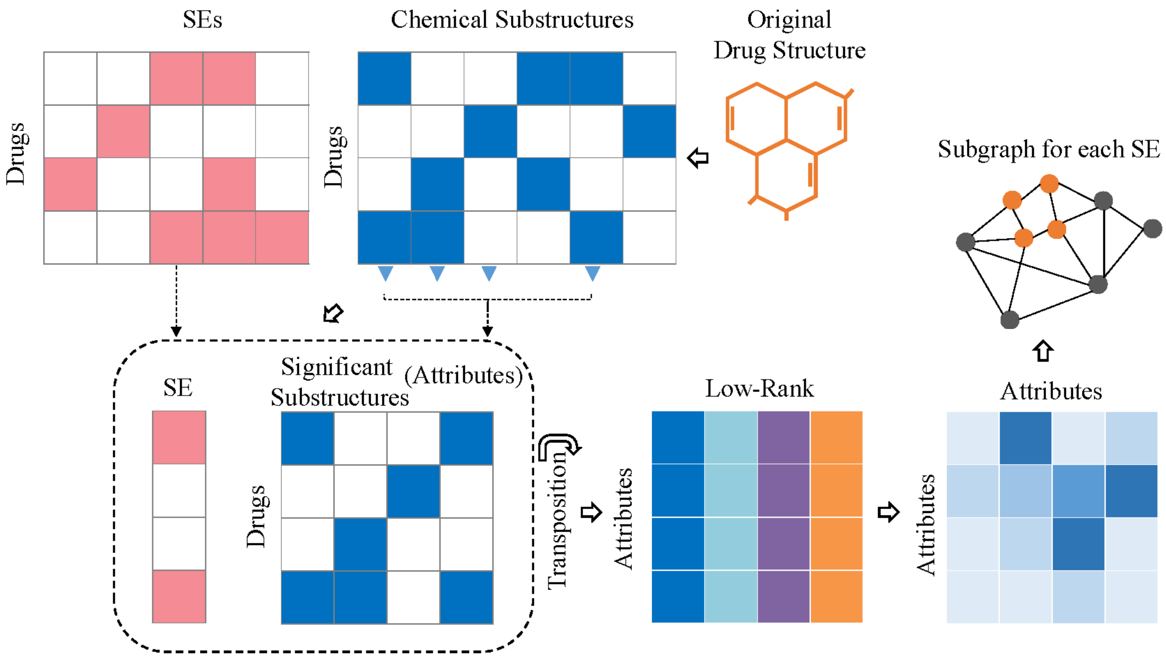


Fig. 21. Flow diagram of GraphSE.

In this section, we present a methodology, called GraphSE, to learning significant subgraphs in drug molecular graphs for each SE with low-rank approximation so that these subgraphs can be used for SE prediction. To do so, GraphSE first identifies significant drug substructures by computing a degree-of-correlation between drug substructures and each SE. It then attempts to find low-rank representation between drug substructures using an NMF. The use of NMF has the advantage that they can effectively deal with sparse noise or outliers when determining a low-rank representation of the significant substructures. Based on them, GraphSE constructs an attributed graph with nodes representing the substructures and edges representing the existence of molecular binding between substructures. From these attributed graphs, GraphSE attempts to look for latent patterns in the form of significant subgraphs that are represented as clusters of interrelated molecular substructures. Once these latent subgraphs are identified, then we make use of a Bayesian approach to predict SEs in the candidate drug molecules. In Figure 21 below, we summarize the subgraph identification process for SE prediction.

GraphSE has the following unique characteristics: (i) it can predict SEs by rep-resenting molecular structures as graphs so that significant substructures that interrelate with different SEs for each drug can be more easily identified and explained; (ii) Some SEs can be interrelated with each other but such interrelationship are usually not considered in previous work. In the case of GraphSE, it attempts to identify such interrelating SEs by considering co-occurring frequencies of different SEs. (iii) GraphSE can reduce noise and outliers by using an appropriate low-rank matrix to represent the significant substructures.

GraphSE has been tested real data sets and its performance shows that it is a promising method for predicting drug SEs both because the prediction can be explainable and more accurate. It has good potential to be used to improve the automated drug screening process and to prevent fatal drug SEs.

### 5.1.2 Pattern discovery

To start with the illustration of the proposed method, we first introduce the following definitions:

*Drug Representation:* Let $D$ be the set of drugs containing $|D|$ drugs samples and each drug sample can be made up of $|P|$ different chemical substructures. That is to say, for each example of the drug, its drug substructure can be represented as $d_i = \{cs_{i1}, cs_{i2}, cs_{i3} \ldots cs_{i|P|}\}$, where $cs_{ix}$ is equal to 1 if $d_i$ contains the xth chemical substructure, and vice versa.

*Drug Side Effect Representation:* For $|D|$ samples of drugs and $|S|$ possible side effects, we use $s_i = \{se_{i1}, se_{i2}, se_{i3} \ldots se_{i|S|}\}$ to represent whether a sample of the drug, $d_i$, has the side-effect $se_y$. It should be noted that each SE may be incurred by the drug substructure of a drug sample.

*Molecular Graph Representation:* For each SE, we use a graph, $G = \{V, E\}$ to represent the affinity between pairwise drug substructures, where $V = \{v_i\}\ (1 \leq i \leq |P|)$ is the vertex set representing all the drug substructures and the edge set $E = \{e_{ij}\}$ are weighted by how similar between a pair of substructures are. As a result, we have |S| molecular graphs in each of which the similarity between pairwise chemical substructures for a particular SE can be represented.

To predict SEs, we need to tackle two sub-problems that have not been addressed previously.

*SE Related Attributes Learning:* To obtain an accurate predictor of SE for the drugs, the first problem is we should identify the relationship between each SE, say $se_j$ and each drug substructure $cs_i$. It allows highly related drug substructures to each SE to be discovered and these substructures can be seen as the attributes used to characterize each SE in each sample of the drug.

*Explainable Subgraph Learning:* The second problem can be stated as follows. Given the attributes for each SE in all drug samples, the low-rank representation of all attributes in each drug sample can be obtained by a low-rank approximation scheme and an affinity graph $G$ which has been defined can be constructed by using the low-rank representations. For an edge, say $e_{ij}$ in $G$, it represents the similarity between drug substructures i and j, given a particular SE. Given $G$, we can identify some subgraphs representing the sets of drug substructures that are highly related to a particular SE, by utilizing an appropriate clustering method. Here, the problem of SE prediction is transferred to the learning of sub-graphs in *G*, which is the main contribution of this section and there are no similar approaches proposed before.

### **5.1.2.1** SE Related Attributes Learning with Mutual Information

In this section, we describe how to identify significant substructures for each SE. To identify such attributes for SEs, we attempt to identify the degree interrelationship between any pair of chemical substructure ($cs_i$) and SE ($se_j$). To determine such degree, we attempt to measure the difference between the observed frequency (o($cs_i$, $se_j$)) that each drug incurs each SE, and the expected frequency (e($cs_i$, $se_j$)) that each drug incurs each SE. According to Section If such difference is sufficiently large, we may determine that $cs_i$ may incur $se_j$, which in other words, those drugs containing $cs_i$ might incur the side-effect $se_j$. To determine whether or not $cs_i$ may incur $se_j$, we adopt the proposed statistical method from Section 3.2.2, it is defined as the following:

$$e(cs_i, se_j) = \frac{cs_{i+} * se_{+j}}{T} \tag{26}$$

where

$$cs_{i+} = \sum_{j=1}^{S} cs_{ij} \tag{27}$$

$$se_{+j} = \sum_{p=1}^{|S|} se_{pj} \tag{28}$$

$$T = \sum_{i,j} o(cs_i, se_j) \tag{29}$$

we make a reasonable statistic to prove whether there is a relevance of drug SE and chemical substructure

$$R(cs_i, se_j) = \frac{Z_{pj}}{\sqrt{\left(1-\frac{cs_{i+}}{T}\right)\left(1-\frac{se_{+j}}{T}\right)}} \tag{30}$$

where $\left(1 - \frac{cs_{i+}}{T}\right)\left(1 - \frac{se_{+j}}{T}\right)$ is used to adjust likelihood of $Z(cs_i, se_j)$.

$$Z(cs_i, se_j) = \frac{o(cs_i, se_j) - e(cs_i, se_j)}{\sqrt{e(cs_i, se_j)}} \tag{31}$$

In previous works, $R$($cs_i$, $se_j$) measure has been shown to approximately follow the *standard normal distribution* [141143]. Thus, whether or not the occurrence of $cs_i$ and $se_j$ can incur each other is at some confidence level if $R(cs_i, se_j)$ is larger than a predefined value. The correlation between pairwise drug substructures and SEs at different confidence levels can be identified by setting corresponding thresholds. By making use of $R$ measure, we may identify those significantly correlated links of drug substructures and SEs, while eliminating those that are insufficiently significant.

#### **5.1.2.2** Inferring High-Order Pattern Candidates

In this section, how *GraphSE* formulates the problem of SEs prediction as the detection of latent substructures in the matrix obtained by low-rank approximation, and how *GraphSE* solves the formulated problem, are illustrated.

Though significantly correlated links of drug substructure and SE can be identified by the $R$ measure proposed in Section 3, those attribute sets, which are contributing factors triggering SEs, cannot be identified immediately. To find these patterns, we assume that the drug attributes can be generated only from those significantly correlated links of drug substructure and SE. Given this assumption, whether a sample of drug contains a set of significant substructures that may lead to a particular SE, say $d_i$, can be represented as $a_i = \{scs_{i1}, scs_{i2}, scs_{i3} \ldots scs_{io}\}$, where $scs_{ix}$ is equal to 1 if $i$th drug contains the $x$th significant chemical substructure resulting into a particular SE, and vice versa. $O$ is the total number of drug substructures that may associate with a specific symptom of SE. Given an SE and $|D|$ drugs, an attribute expression matrix, $A$ which has the dimension of $|D|$ by $O$ can be formulated, in which each element, say $a_{ij}$, represents whether a significant drug substructure may lead to a particular side effect in drug $d_i$.

Given a set of $|D|$ drugs, a straightforward but efficient way to represent meaningful attributes is to check whether a significant chemical substructure is contained in those $|D|$ drugs. In other words, we used the matrix Y, which is the transpose of $A$, to investigate the occurrence of a drug substructure in drugs that may result in a particular SE. In this case, $Y$ can be seen as an attribute expression matrix. It is well known that structural converting, like constructing $Y$, based on the samples of drugs and SEs, may lead to inevitable structural loss and sparse noise. Thus, the reduction of such loss and noise may let one find better representations of the converted data. Inspired by NMF, we facilitate the robust affinity constructing of attributes. In this case, it would be considered there is a low-rank attribute matrix *X* by NMF. To achieve the molecular graph *G* for each SE, given a set of attributes $S = \{s_1, \dots, s_n\}$ with their new representation *X*, we may obtain an affinity matrix *M* in which $M_{ij} = \exp(-\|s_i - s_j\|^2/2\sigma^2)$, if $i \neq j$, and $M_{ii} = 1$. *M* can be used to represent the similarity between pairwise attributes.

**5.1.2.3** Attribute Graph Clustering

In this section, how *GraphSE* formulates the problem of SEs prediction as the detection of latent substructures in the matrix obtain by low-rank approximation, and how *GraphSE* solves the formulated problem, are illustrated.

For clustering attributes in *S*, *GraphSE* adopts an objective-function based method to detect clusters in *M*. The objective function can be used to evaluate the overall clustering quality and it is defined as:

$$\text{maximize } O = tr(C^T MC) - \alpha |CC^T|_F^2 - \beta |C|_F^2$$
$$\text{Subject to } C \geq 0 \tag{32}$$

where *C* is an n-by-k latent matrix in which each variable represents degree of cluster affiliation between a particular drug and each of k clusters, $|CC^T|_F^2$ is a penalization term which avoids the variables in *C* increasing infinitely within the clustering procedure, $|\mathrm{C}|_F^2$ is the regularization term for smoothing the variables in *C*, $|\ |_F$ represents the matrix Frobenius norm, and $\alpha$ and $\beta$ are parameters for controlling the effect of the penalization and regularization terms. The proposed method is able to obtain the optimal cluster arrangement for each attribute when (32) is optimized. Those found clusters can be seen optimal sets of molecular sub-graphs that may lead to a particular SE.

To optimize the objective function (32), we derive the following iterative updating rule for inferencing the variables in C. Let $\eta = [\eta_{ij}]$ be a n-by-k matrix in which elements represent the Lagrange multipliers for corresponding variables in C. The following Lagrange function can be formulated:

$$L(\eta, \mathrm{C}) = O - tr(\eta^T C) \tag{33}$$

Based on the KKT optimality conditions for constrained optimization, we have the following element-wise equation system:

$$\frac{\partial L}{\partial c_{ij}} = [2 * \mathrm{MC} - 4\alpha * \mathrm{CC}^T\mathrm{C} - 2\beta * \mathrm{C} - \eta]_{ij} = 0$$

$$\eta_{ij} \cdot c_{ij} = 0$$

$$\eta_{ij} \geq 0 \tag{34}$$

By solving the above system, we have the following iterative updating rule for variables in C:

$$c_{ij} \leftarrow c_{ij}\sqrt{\frac{\mathrm{MC}}{2\alpha*\mathrm{CC}^T\mathrm{C}+\beta*\mathrm{C}}} \tag{35}$$

By using the updating rule (35), (32) is able to be optimized in a finite number of iterations. Once the optimization process is done, the *GraphSE* obtains the optimal clustering arrangement for all the corresponding chemical attributes. By making use of *C*, those highly correlated chemical attributes can be found in the same cluster.

After finding the subgraphs in the attributed graph for each SE, *GraphSE* identifies a particular number of sets of chemical substructures that are potentially related to each SE. Hence, the subgraphs could be utilized in the construction of prediction rules. According to [131], Naïve Bayes (NB) is a possible approach to produce a final

Table 11. AUROC scores of different algorithms on two datasets

| **Methods** | **ROC** | |
|---|---|---|
| | **Liu's** | **Pauwels's** |
| *GraphSE* | **0.887** | **0.892** |
| *GraphSE*-RankClus | 0.871 | 0.872 |
| *GraphSE*-NCut | 0.869 | 0.870 |
| NB | 0.868 | 0.863 |
| SVM | 0.872 | 0.871 |
| OCCA | 0.845 | 0.856 |
| SCCA | 0.868 | 0.873 |
| LDA | 0.848 | 0.850 |

prediction. Given data on a set of $n$ molecular subgraphs for training $S = \{(SX_1, SE_1), (SX_2, SE_2), ..., (SX_s, SE_s)\}$, $SX_i = \{SG_{i1}, SG_{i2}, ..., SG_{in}\}$ is a vector of subgraphs where $n$ relies on the number of subgraphs and each binary vector whose element of the users encode for the presence or absence of a subgraphs using 1 and 0. $SE_i$ is the corresponding SE label and $s$ relies on the number of SEs. Here, we would like to grow a score for new subgraphs as follows:

$$score = P(Y)\prod_t P(SG_t|SE) \tag{36}$$

To predict SEs, the final score for *SE* can be defined as: $\sum_{i=1}^{n} score_n$.

### 5.1.3 Experiment

**5.1.3.1** Datasets and Features

GraphSE was evaluated using two sets of real publicly available data obtained from different sources. They include Pauwels's dataset [135] and Liu's dataset [131]. They have been widely used to test the effectiveness of SEs prediction.

The data used to test for 881 substructures in drugs were obtained from the PubChem Compound Database [144], DrugBank [21] and KEGG DRUG [145]. The data of drug SE come from SIDER [51] which collects SEs data from FDA Adverse Event Reporting System (FAERS). It contains information on marketed medicines and their recorded drug SEs reports. The information is extracted from public documents and hospital medical report. Each drug can be encoded as 881-dimensional binary finger-prints whose elements encode for the presence or absence of a chemical substructure by 1 or 0, respectively.

Pauwels's dataset builds a dataset containing 888 drugs and 1385 kinds of SE and 61102 drug-side-effect interactions in all. Liu's dataset builds a dataset containing 832 drugs and 1385 kinds of SE and 59205 drug-side-effect interactions in all. That is to say, each drug can be encoded as a 1385-dimensional binary vector whose elements encode for the presence or absence of a side-effect by 1 or 0, respectively.

| **Drug substructure** | **Peripheral Ischaemia** | **Rales** | **Mydriasis** |
|---|---|---|---|
| **>= 8 H** | -0.332 | -0.119 | 1.098 |
| **>= 16 H** | -0.132 | 0.080 | **1.417** |
| **>= 32 H** | 0.372 | 0.570 | **2.324** |
| **>= 4 C** | -0.325 | -0.111 | 1.110 |
| **>= 2 N** | -0.054 | -0.114 | **1.542** |
| **>= 8 O** | **1.815** | **2.484** | **2.677** |

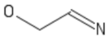
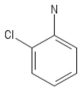
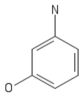
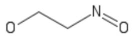
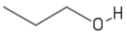
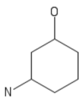
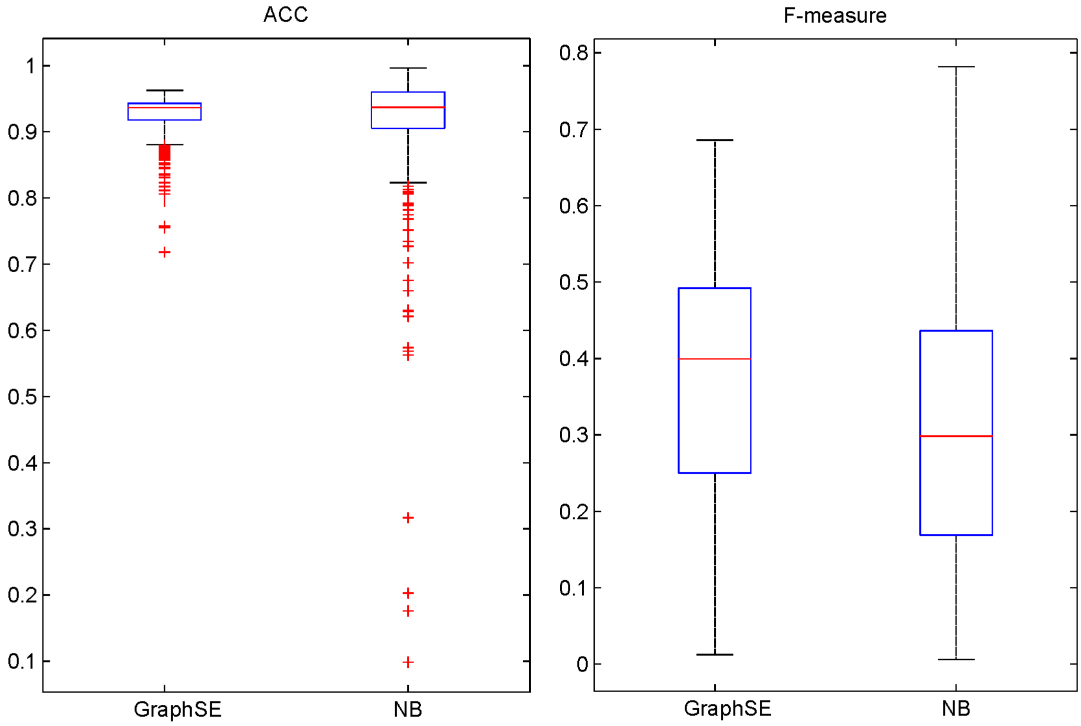
| | | | |
|---|---|---|---|
| **Subgraph 1** | **O-C-C=N** |  | **Side-effect: Cystitis** |
| | **Nc1c(Cl)cccc1** |  | |
| | **Oc1cc(N)ccc1** |  | |
| **Subgraph 2** | **O=N-C:C-O** |  | |
| | **C-C-C-O-[#1]** |  | |
| | **C-C-C-O-[#1]** |  | |

Table 12. Example of interpretable representation of drug substructure for side-effects.

### **5.1.3.2** Evaluation and Comparisons of Different Methods

To evaluate the performance of the proposed prediction method, we applied well-known solutions of SEs prediction include SVM, OCCA, SCCA, and LDA with Gibbs Sampling in the experimental study. To prove the superiority of the subgraph discovering scheme, two state-of-the-art approaches for graph partitioning are also considered. One is NCut



(a) (b)

Fig. 22. (a) Boxplot for the ACC scores of side-effects obtained from GraphSE and NB; (b) Boxplot for the F-measure obtained from GraphSE and NB.

[146] and the other is RankClus [147]. In terms of biological theory, we examined the sets of drug substructures identified by GraphSE, and the SE related substructure sets are explainable. To measure the performance of the methods, we adopted 5-fold cross-validation. In this scenario, the results at the top K suspects might matter the most. Hence, we draw the receiver operating characteristic curve (ROC) for the experimental result.

Table 11 reports the accuracies of different algorithms on both Pauwels's dataset and Liu's dataset. As the table shows, GraphSE outperforms other baselines in both two data sets. The resulting Auroc for GraphSE, GraphSE-RankClus, GraphSE-NCut, Naïve Bayes, SVM, OCCA, SCCA and LDA on Pauwels's dataset are 0.892, 0.872, 0.866, 0.863, 0.871, 0.856, 0.873 and 0.85, respectively. The resulting Auroc on Liu's dataset are 0.887, 0.869, 0.870, 0.872, 0.845, 0.868 and 0.840, respectively. As it shows, among all comparisons, we achieve the highest Auroc. This means GraphSE is able to extract more meaningful information to drug SEs from molecular subgraphs and so that improve

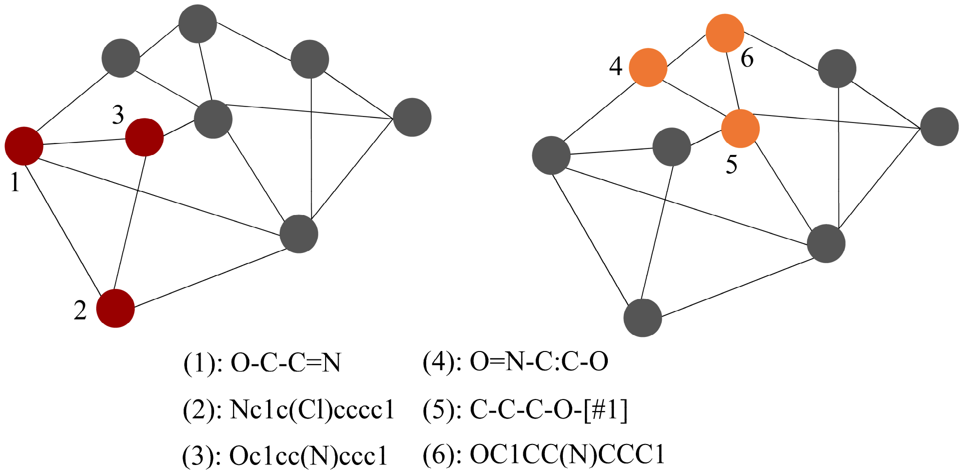


Fig. 23. Example of interesting subgraphs for cystitis, the red part represents subgraph 1 and the yellow part represents subgraph 2.

the prediction performance.

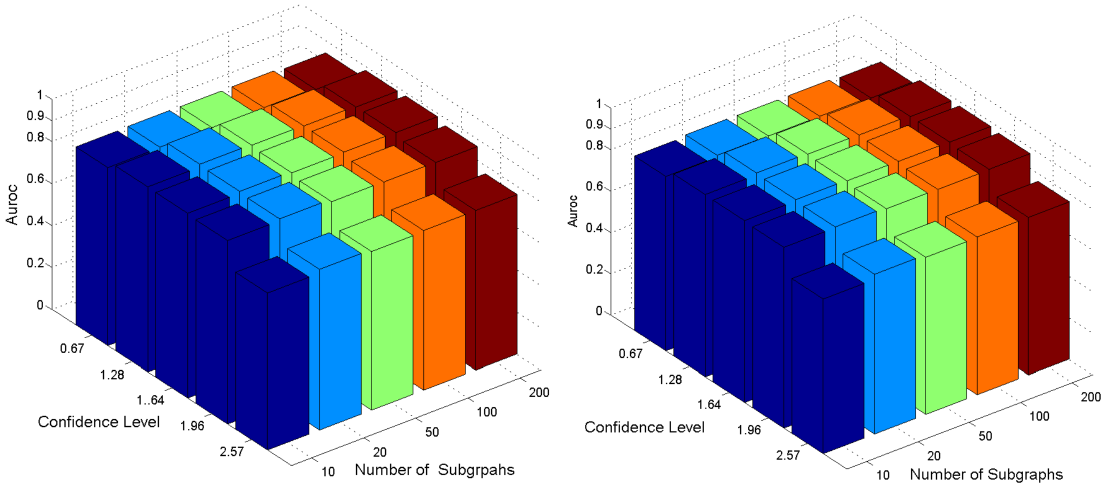
**5.1.3.3** Discovered Associations Study

To validate the interpretability of discovered patterns, we compared the proposed method and the base method Naïve Bayes, by using two more evaluation measures, including overall prediction accuracy (ACC) and f-measure. Since both methods use possibility scores of relationships between substructure and SEs, we would like to learn the prediction performance of predicted SEs for each drug. Figure 22 shows the distribution of the resulting ACC and F-measure for 888 drugs in GraphSE and NB. As can be seen, two boxplots show significant upward and stable tendency when the subgraphs introduced.

An example of the significant patterns between the drug substructures and SEs is given in Table 12. This table lists a series of significant substructure for some SEs. As we can see series of Table 12, the numbers of the bold font represent a strongly correlated chemical structure and SE. And, Figure 23 draws an example of two explained subgraphs include four substructure which has a strong correlation to cystitis the depth of the edge color represents the weight of the connection. As the figure shows, the cystitis is strongly related with two attribute sub-graphs: {O-C-C=N, Nc1c(Cl)cccc1, Oc1cc(N)ccc1} and {O=N-C:C-O, O=N-C:C-O, O=N-C:C-O}. And, these subgraphs that cause cystitis are listed in Figure 23 in the form of molecular structure diagrams. Above examples show the explainable ability for the SEs prediction of GraphSE. Compared with the previous research work, this model can infer SEs at the data level, as well as visualize which small compounds may influence the SEs during the process of automated compounds screening.

### **5.1.3.4** Cross-Test for Parameter Settings

In this subsection, we conduct a series of cross-test experiments on Pauwels's dataset to evaluate the influence of main parameters in GraphSE, including the confidence level to an adjusted residual between SEs and chemical substructures, and the number of subgraphs. We test confidence level using different thresholds, including 50%, 80%, 90%, 95%, and 99%, corresponding to adjusted residual as 0.674, 1.28, 1.645, 1.96, and 2.57, and test number of subgraphs K using 10, 20, 100, 150, 200. The testing results are shown in Figure 24. As the figure shows, the accuracy increases gradually as the threshold of adjusted residuals increase. And, the accuracy achieves to a steady performance when the threshold of adjusted residuals ranging from 1.28 to 1.96. Based on the results shown in Figure 24, we may find the best parameter setting for our model is 90% of confidence level, and 50 subgraphs, respectively.



(a) (b)

Fig. 24. (a) Area under ROC of GraphSE on Pauwels's dataset with different confidence level and number of subgraphs. (b) Area under ROC of GraphSE on Liu's dataset with different confidence level and number of subgraphs.

### 5.1.4 Summary

In this chapter, we propose a novel machine learning approach called GraphSE to predict the links between drug and SEs. It is a unique approach in the sense that it represents an attempt to make use of attributed subgraphs within drug substructures to predict potential SEs of different drugs. By taking into consideration significant relationships between molecular representations and SEs, the experimental results show that GraphSE is able to predict SEs accurately. Also, it can allow predictions to be explained easily so that the relationship between substructures and various SEs can be understood. Graphse is a promising interpretable algorithm that can be widely used in network link prediction. GraphSE also can be a promising intelligent tool for assisting in decision making related to the establishment of lead compounds at the beginning of drug design. It has good potential to improve the efficiency and effectiveness of automated drug screening.

## 5.2 Learning latent representations for clustering

In section 3.2, we have explored ways to find clusters in a social network using graph mining and fusion algorithm. Our approach improves the performance of social network clustering, but fusion is difficult to maintain the interpretability of results at the same time. The identification of communities and their representations becomes one of the most significant tasks in the analytics of different networks. To perform the task, several approaches have been proposed, taking into the consideration different categories of information carried by the network data, e.g., edge structure, node attributes, or those above. Such approaches may be practical for discovering communities in the network, but few of them can discover communities and summarize their representations simultaneously.

### 5.2.1 Overview

A Network can be modeled as a graph contains a set of vertices and edges, representing data entities, and inter-relationship between them, respectively. Different from random graphs, there is some particular latent structure in those real-world graphs. There are many types of hidden structures that have been looked into, e.g., triadic patterns in social networks [148-150]. Among these latent structures, network communities, which also named as clusters are the most typical ones. How to identify such communities and the representations that may characterize them has drawn much attention in recent years [151-152]. There are some approaches to discover communities effectively. Different from those model-based ones, such algorithms make use of different objective functions to measure the overall quality of discovered communities and the community membership of each vertex is obtained by optimizing the objective function. For example, MISAGA [99] is an approach to community detection in graphs, which can perform the task by maximizing the objective function measuring the overall edge density and attribute similarity in all the clusters. In [100], an evolutionary algorithm for community detection in social networks (ECDA) is proposed. ECDA can discover communities in network data by maximizing the intra-degrees of attribute similarity between connecting vertices in the same clusters. Inspired by probabilistic topic models [161], there are several topic-model based algorithms, including Relational Topic Model and iTopicModel [162], proposed to discover communities in relational data. The community membership is modeled as a posterior probability measuring the possibility that vertices in the same cluster are labeled with similar topics. As such topic model-based methods always require for a high computational effort, they are not efficient algorithms for discovering communities and summarizing their features in the network data [97]. Given the prevalent algorithms, we have the following findings that may motivate us to develop a novel approach. First, most algorithms are proposed to either discover communities or summarize community representations. There are almost no effective algorithms that are able to complete both two tasks simultaneously. Second, though there have been some approaches which can simultaneously detect graph community and summarize their representations, e.g., those topic-models based ones, their high computational requirement leads them to be infeasible for the analytics in large network data. To address the mentioned challenges, we propose a novel Latent Factor Model for Community Identification and Summarization (LFCIS). By modeling edge structure, attribute similarity between pairwise vertices, community features, and community membership as low-dimensional latent spaces, LFCIS formulates the tasks of the identification of community membership and representations as a single optimization problem which is related to the learning of the factor in the mentioned latent spaces. The corresponding latent spaces learned by LFCIS may reveal the community membership for each vertex, taking into consideration both edge structure and attribute similarity, and common representations of each community. For the test of the performance, LFCIS is used with both synthetic and real-world datasets of social network data. The experimental results are evaluated against known ground-truth data. It is found that LFCIS outperforms most of the state-of-the-art in both effectiveness and efficiency. Based on its performance, LFCIS is a very promising approach for community identification and community summarization.

### 5.2.2 Preliminaries

Given a set of network data containing $n$ vertices, $m$ node attributes, and |E| edges, it can be represented as a graph G = {V, E, Λ}, where V, E, and Λ represent the vertex, edge, and attribute set in the network data, respectively. For the vertex set, it is defined as V = $\{v_i \,|1\leq i\leq n\}$. The edge set, is defined as E = $\{e_{ij}=1|\ v_i$ and $v_j$ are connected}. And the attribute set is defined as Λ = $\{\Lambda_i \,|1\leq i\leq m\}$. LFCIS makes use of two matrices, **M** and **F**, to represent the edge structure and node attributes in G. **M** is an $n$-by-$n$ adjacency matrix each element of which, say $\mathbf{M}_{ij}$, equals to 1 if $v_i$ and $v_j$ are connected in G, and 0 if they are disconnected. **F** is an $m$-by-$n$ matrix each element of which say $\mathbf{F}_{ij}$, equals to 1 if vertex $v_j$ is associated with attribute $\Lambda_i$, and vice versa.

For notations, we use a subscript, e.g., $\mathbf{M}_i$, to represent the $i$th column of a given matrix, say **M**. We use $\mathbf{M}_{ij}$, to represent the entry of **M**, in $i$th row, $j$th column. $tr(\cdot)$ represents the matrix trace. $|\cdot|_F$ and $|\cdot|_1$ represent the matrix Frobenius norm, and $l_1$ norm, respectively. All these mentioned mathematical preliminaries and notations are used by LFCIS to model the problem of community identification and summarization.

### 5.2.3 LFCIS in details

In this section, how LFCIS models the community identification and summarization as an optimization problem, making use of different latent spaces, and how the factors in these latent spaces are fitted, are introduced in detail.

#### **5.2.3.1** Modeling community identification and summarization

As mentioned above, there are two sub-tasks, i.e., identifying latent communities, and summarizing their representations, that LFCIS has to complete through latent space modeling. For the identification of network communities, LFCIS attempts to assign those vertices sharing similar edge structure and node attributes into the same communities. To project each vertex in G from a high dimension into a lower one, LFCIS makes use of a *k*-by-*n* latent matrix, **S** to represent the latent edge structure for each vertex. Each column of **S**, say $\mathbf{S}_i$, represents the inter-relationship w.r.t. edge structure between a vertex, say $v_i$, and *k* latent structural components. Obviously, a larger value of an element in **S**, say $\mathbf{S}_{ij}$, means $v_j$ has a stronger relationship with *i*th latent component. Using another *k*-by-*n* matrix, **C** to represent community membership between each vertex and *k* communities, LFCIS makes use of the difference between the original adjacency matrix of a graph, G and the one that is jointly constructed by **S** and **C**, to measure the structural loss after using **S** and **C** to project the edge structures of *n* vertices into the *k*-dimensional latent spaces. It is apparent that a minimum of such loss leads to a better projection. And this quality function is defined as

$$\begin{gathered} \textit{Minimize} \\ O_1 = \left| \mathbf{M} - \mathbf{S}^T \mathbf{C} \right|_F^2 \end{gathered} \tag{37}$$

Besides considering the edge structure of the vertices within the same community, LFCIS also takes into the consideration attribute similarity between each pair of vertices. As the feature vectors for a pair of vertices $v_i$ and $v_j$, $\mathbf{F}_i$ and $\mathbf{F}_j$ are always with high dimensionality and are always different, LFCIS makes use of the following kernel function to measure the attribute similarity between a pair of vertices, $v_i$ and $v_j$ in G

$$\mathbf{X}_{ij} = \exp(-\frac{\left| \mathbf{F}_i - \mathbf{F}_j \right|^2}{2\delta^2}) \tag{38}$$

(38) is a standard Gaussian kernel which can be used to measure the overall similarity w.r.t. attributes associated with any pair of vertices in G. A higher value of that means there are more attributes commonly associated with both $v_i$ and $v_j$. Such pairwise vertices are considered to be more similar w.r.t. attributes. After obtaining the attribute similarity

between each pair of vertices, LFCIS makes use an *n*-by-*n* matrix, **X** to represent the attribute similarity between any pair of vertices in G. Similarly, LFCIS uses a *k*-by-*n* latent matrix, **B** to represent the latent attribute similarity between each vertex and *k* latent attribute components. For an element in **B**, say $\mathbf{B}_{ij}$, its value means the strength of attribute similarity between $v_j$ and *i*th latent component. Similar to (37), LFCIS makes use of the difference between **X** and the one jointly constructed by **B**, and **C**, to measure the loss of attribute similarity after projection. It is defined as

$$\begin{gathered} Minimize \\ O_2 = \left|X - B^T C\right|_F^2 \end{gathered} \tag{39}$$

As LFCIS aims to find *k* communities in each of which vertices are connecting more and sharing higher attribute similarity, it makes use of the following objective function to regulate the structure of latent spaces of **S** and **B**

$$\begin{gathered} Minimize \\ O_3 = \left|S - B\right|_F^2 \end{gathered} \tag{40}$$

By making use of (40), the latent spaces, **B** and **S** are regulated to share the similar structure so that LFCIS is able to assign those vertices a sharing higher similarity of edge structure and attributes when fitting. To summarize the features that are able to characterize the communities, LFCIS assumes that, the community representations are hidden in those *m* attributes in G, and the representations for one community are always different from those for others. Based on this assumption, LFCIS utilizes a *k*-by-*m* latent matrix, **A** to represent the inter-relationship between each of *m* attributes and *k* communities. It is apparent that a higher value of an entry in **A**, say $\mathbf{A}_{ij}$, means $\Lambda_j$ is more possible to become a representation characterizing community *i*. By making use of **C** as the latent matrix representing the community membership, LFCIS utilizing the following objective function to measure the overall difference between **F** and the one constructed by **A** and **C**

$$\begin{gathered} Minimize \\ O_4 = \left|F - A^T C\right|_F^2 \end{gathered} \tag{41}$$

It is apparent that when (41) is minimized, the corresponding latent spaces represented by **A** may best interpret the representations characterizing the *k* found communities. Having introduced objective functions that LFCIS uses to complete the sub-tasks of community identification and summarization, we may know that minimizing the following function means completing these tasks simultaneously

$$\begin{gathered}\mathit{Minimize}\\ O = \left|\mathrm{M} - \mathrm{S}^T\mathrm{C}\right|_F^2 + \left|\mathrm{X} - \mathrm{B}^T\mathrm{C}\right|_F^2 + \left|\mathrm{S} - \mathrm{B}\right|_F^2 + \left|\mathrm{F} - \mathrm{A}^T\mathrm{C}\right|_F^2\end{gathered} \tag{42}$$

Here we assume that the objectives O$_1$, O$_2$, and O$_4$ share the same latent space representing the community membership. Based on the introduction of each term in (42), we know that those vertices sharing a higher similarity of edge structure and attributes can be grouped, so that the community representations can be summarized based on the community membership, when (42) is minimized. By ignoring the terms which are irrelative to the model optimization, (42) is equivalent to

$$\begin{gathered}\mathit{Maximize}\\ O = tr(\mathrm{M}^T\mathrm{S}^T\mathrm{C} + \mathrm{X}^T\mathrm{B}^T\mathrm{C}) + tr(\mathrm{F}^T\mathrm{A}^T\mathrm{C}) + tr(\mathrm{S}^T\mathrm{B})\\ - \frac{1}{2}\left[\left|\mathrm{S}^T\mathrm{C}\right|_F^2 + \left|\mathrm{B}^T\mathrm{C}\right|_F^2 + \left|\mathrm{A}^T\mathrm{C}\right|_F^2 + \left|\mathrm{B}\right|_F^2 + \left|\mathrm{S}\right|_F^2\right]\end{gathered} \tag{43}$$

As all the entries in **M**, **X**, and **F** are non-negative, we propose the following objective function used by LFCIS to perform the tasks of community detection and summarization

$$\begin{gathered}\mathit{Maximize}\\ O = tr(\mathrm{M}^T\mathrm{S}^T\mathrm{C} + \mathrm{X}^T\mathrm{B}^T\mathrm{C}) + tr(\mathrm{F}^T\mathrm{A}^T\mathrm{C}) + tr(\mathrm{S}^T\mathrm{B})\\ - \frac{1}{2}\left[\left|\mathrm{S}^T\mathrm{C}\right|_F^2 + \left|\mathrm{B}^T\mathrm{C}\right|_F^2 + \left|\mathrm{A}^T\mathrm{C}\right|_F^2 + \Omega(\mathrm{A}, \mathrm{B}, \mathrm{S}, \mathrm{C})\right]\\ \Omega(\mathrm{A}, \mathrm{B}, \mathrm{S}, \mathrm{C}) = \left|\mathrm{A}\right|_F^2 + \alpha\left|\mathrm{A}\right|_1 + \left|\mathrm{B}\right|_F^2 + \left|\mathrm{S}\right|_F^2 + \left|\mathrm{C}\right|_F^2\\ \mathit{subject\ \ to}\ \mathrm{C},\ \mathrm{B},\ \mathrm{S},\ \mathrm{A} \geq 0\end{gathered} \tag{44}$$

where **Ω** contains the regularization terms preventing the factors in the latent spaces from overfitting. If (44) can be optimized, LFCIS is able to find *k* communities in each of which vertices are densely connected, share relatively high attribute similarity with each other, and the community representations can be obtained from *m* attributes in G.

#### **5.2.3.2** Latent factor Inference through optimization

To identify optimal latent spaces that can be used to represent the community structure and community representations, LFCIS has to optimize (44). Given the characteristics of (44), we find that it is convex for variables in **C**, **A**, **B**, and **S** respectively, when fixing all variables in other matrices. Given this, we may derive a series of iterative rules for inferring the optimal latent factors in **C**, **A**, **B**, and **S**.

- Inference of C

Let $\boldsymbol{\beta}_{ij}$ be the Lagrange multiplier for $\mathbf{C}_{ij}\geq 0$, the Lagrange function for variables in **C** is shown as the following

$$L(\mathbf{C},\boldsymbol{\beta}) = O - tr(\boldsymbol{\beta}^T\mathbf{C}) \tag{45}$$

Based on the KKT conditions for constrained optimization, we have the following element-wise equation system

$$\begin{gathered} \frac{\partial L(\mathrm{C},\beta)}{\partial \mathrm{C}_{ij}} \\ = [\mathrm{SM} + \mathrm{BX} + \mathrm{AF} - \mathrm{SS}^T\mathrm{C} - \mathrm{BB}^T\mathrm{C} - \mathrm{AA}^T\mathrm{C} - \mathrm{C} - \beta]_{ij} \\ \beta_{ij}\cdot \mathrm{C}_{ij} = 0 \\ \beta \geq 0 \end{gathered} \tag{46}$$

Solving the equation system in (46), we may derive the element-wise updating rule for inferring the latent factors in **C**

$$\mathbf{C}_{ij} \leftarrow \mathbf{C}_{ij}\cdot\frac{[\mathbf{SM}+\mathbf{BX}+\mathbf{AF}]_{ij}}{[\mathbf{SS}^T\mathbf{C}+\mathbf{BB}^T\mathbf{C}+\mathbf{AA}^T\mathbf{C}+\mathbf{C}]_{ij}} \tag{47}$$

- Inference of S

Let $\boldsymbol{\gamma}_{ij}$ be the Lagrange multiplier for $\mathbf{S}_{ij}\geq 0$, the Lagrange function for variables in **S** is shown as the following

$$L(\mathbf{S},\boldsymbol{\gamma}) = O - tr(\boldsymbol{\gamma}^T\mathbf{S}) \tag{48}$$

Based on the KKT conditions for constrained optimization, we have the following element-wise equation system

$$\frac{\partial L(\mathrm{S},\gamma)}{\partial \mathrm{S}_{ij}} = \left[\mathrm{CM} + \mathrm{B} - \mathrm{CC}^T\mathrm{S} - \mathrm{S} - \gamma\right]_{ij}$$
$$\gamma_{ij} \cdot \mathrm{S}_{ij} = 0 \tag{49}$$
$$\gamma \geq 0$$

Solving the equation system in (49), we may derive the element-wise updating rule for inferring the latent factors in **S**

$$\mathbf{S}_{ij} \leftarrow \mathbf{S}_{ij} \cdot \frac{[\mathbf{CM}+\mathbf{B}]_{ij}}{[\mathbf{CC}^T\mathbf{S}+\mathbf{S}]_{ij}} \tag{50}$$

- Inference of B

Let $\boldsymbol{\eta}_{ij}$ be the Lagrange multiplier for $\mathbf{B}_{ij}\geq 0$, the Lagrange function for variables in **B** is shown as the following

$$L(\mathbf{B},\boldsymbol{\eta}) = O - tr(\boldsymbol{\eta}^T\mathbf{B}) \tag{51}$$

Based on the KKT conditions for constrained optimization, we have the following element-wise equation system

$$\frac{\partial L(\mathrm{B},\eta)}{\partial \mathrm{B}_{ij}} = \left[\mathrm{CX} + \mathrm{S} - \mathrm{CC}^T\mathrm{B} - \mathrm{B} - \eta\right]_{ij}$$
$$\eta_{ij} \cdot \mathrm{B}_{ij} = 0 \tag{52}$$
$$\eta \geq 0$$

Solving the equation system in (52), we may derive the element-wise updating rule for inferring the latent factors in **B**

$$\mathbf{B}_{ij} \leftarrow \mathbf{B}_{ij} \cdot \frac{[\mathbf{CX}+\mathbf{S}]_{ij}}{[\mathbf{CC}^T\mathbf{B}+\mathbf{B}]_{ij}} \tag{53}$$

- Inference of A

Let $\boldsymbol{\mu}_{ij}$ be the Lagrange multiplier for $\mathbf{A}_{ij}\geq 0$. The Lagrange function for variables in **A** is shown as the following

$$L(\mathbf{A},\boldsymbol{\mu}) = O - tr(\boldsymbol{\mu}^T\mathbf{A}) \tag{54}$$

Based on the KKT conditions for constrained optimization, we have the following element-wise equation system

$$\frac{\partial L(\mathrm{A}, \mu)}{\partial \mathrm{A}_{ij}} = \left[\mathrm{CF}^T - \mathrm{CC}^T\mathrm{A} - \mathrm{A} - \mu\right]_{ij} + \alpha$$
$$\mu_{ij} \cdot \mathrm{A}_{ij} = 0 \qquad (55)$$
$$\mu \geq 0$$

Solving the equation system in (55), we may derive the element-wise updating rule for inferring the latent factors in **A**

$$\mathbf{A}_{ij} \leftarrow \mathbf{A}_{ij} \cdot \frac{\left[\mathbf{CF}^T\right]_{ij}}{\left[\mathbf{CC}^T\mathbf{A} + \mathbf{A}\right]_{ij} + \alpha} \qquad (56)$$

By iteratively updating latent factors in **C**, **S**, **B**, and **A**, respectively, while fixing the others, LFCIS is able to find the optimal factors that maximize (44).

#### **5.2.3.3** Convergence analysis

To prove the convergence of the algorithm, we may make use of one property of an auxiliary function that is also used in the proof of the Expectation-Maximization

To prove the convergence of the algorithm, we may make use of one property of an auxiliary function that is also used in the proof of the Expectation-Maximization algorithm [166]. The property of the auxiliary function is described as the following. If there exists an auxiliary function satisfying the conditions that $Q(x, x') \leq F(x)$ and $Q(x, x) = F(x)$, then $F$ is non-decreasing under the updating rule that

$$x^{t+1} = \arg\max_x Q(x, x^t) \qquad (57)$$

The equality $F(x^{t+1}) = F(x^t)$ holds only when $x$ is a local maximum of $Q(x, x')$. By iteratively updating $x$ according to (57), $F$ will converge to the local maximum $x_{max} = \mathrm{argmax}_x F(x)$. By defining an appropriate auxiliary function for $O$, we may show the convergence of (44).

First, we may prove the convergence of the updating rule (47) for the inference of **C**. Let $\mathbf{C}_{ij}$ be any element in **C**, $O_{Cij}$ be the partial of (44) that is related to $\mathbf{C}_{ij}$, $O_{Cij}(\mathbf{C}'_{ij})$ be the partial objective value of (44) that is related to $\mathbf{C}_{ij}$ when $\mathbf{C}_{ij}$ is equal to some value, say $\mathbf{C}'_{ij}$. Since the updating rule for **C** is element wise, it is sufficient to show $O_{Cij}$ is non-decreasing according to the updating rule (47). To prove this, we define the following auxiliary function for $O_{Cij}$:

$$Q(c,\mathbf{C}^t_{ij}) = O_{c_{ij}}(\mathbf{C}^t_{ij}) + O'_{c_{ij}}(c-\mathbf{C}^t_{ij}) \\ -\frac{\left[\mathbf{SS}^T\mathbf{C}+\mathbf{BB}^T\mathbf{C}+\mathbf{AA}^T\mathbf{C}+\mathbf{C}\right]_{ij}}{2\mathbf{C}^t_{ij}}(c-\mathbf{C}^t_{ij})^2 \tag{58}$$

where $O'_{Cij}$ is the first order partial derivative relevant to $\mathbf{C}_{ij}$. Although the auxiliary function is defined in (58), we need to prove it satisfies the aforementioned conditions. Apparently, $Q(c, c) = O_{Cij}(c)$. Hence, the left we need to prove is $Q(c, \mathbf{C}^t_{ij}) \leq O_{Cij}(c)$. To prove this, we compared $Q(c, \mathbf{C}^t_{ij})$ shown in (58) with the Taylor expansion of $O_{Cij}$ near to $\mathbf{C}^t_{ij}$

$$O_{c_{ij}}(c) = O_{c_{ij}}(\mathbf{C}^t_{ij}) + O'_{c_{ij}}(c-\mathbf{C}^t_{ij}) + \frac{1}{2}O''_{c_{ij}}(c-\mathbf{C}^t_{ij})^2 \tag{59}$$

where $O'_{cij}$ and $O''_{cij}$ are the first and second order partial derivatives relevant to $c_{ij}$. Note that

$$O'_{c_{ij}} = \frac{\partial O}{\partial \mathbf{C}_{ij}} = \begin{bmatrix}\mathbf{SM}+\mathbf{BX}+\mathbf{AF} \\ -\mathbf{SS}^T\mathbf{C}-\mathbf{BB}^T\mathbf{C}-\mathbf{AA}^T\mathbf{C}-\mathbf{C}\end{bmatrix}_{ij} \\ O''_{c_{ij}} = \frac{\partial^2 O}{\partial(\mathbf{C}_{ij})^2} = -\left[\mathbf{SS}^T+\mathbf{BB}^T+\mathbf{AA}^T\right]_{jj} - 1 \tag{60}$$

Using (60) to replace the relevant terms in (59), we can see that if $Q(c, c^t_{ij}) \leq O_{cij}(c)$, the following inequality must hold

$$-\frac{(\mathrm{SS}^T\mathrm{C}+\mathrm{BB}^T\mathrm{C}+\mathrm{AA}^T\mathrm{C}+\mathrm{C})_{ij}}{2\mathrm{C}^t_{ij}} \leq \frac{1}{2}O''_{c_{ij}} \\ = -\frac{1}{2}\left[\mathrm{SS}^T+\mathrm{BB}^T+\mathrm{AA}^T\right]_{ii} - \frac{1}{2} \tag{61}$$

Therefore, to show $Q(c, c^t_{ij}) \leq O_{cij}(c)$, it is equivalent to show

$$(\mathbf{SS}^T\mathbf{C}+\mathbf{BB}^T\mathbf{C}+\mathbf{AA}^T\mathbf{C}+\mathbf{C})_{ij} \\ \geq \mathbf{C}^t_{ij}\left[\mathbf{SS}^T+\mathbf{BB}^T+\mathbf{AA}^T\right]_{ii} + \mathbf{C}^t_{ij} \tag{62}$$

Since the elements in **C**, **D**, **B** and **S** are non-negative, we have

$$(\mathrm{SS}^T\mathrm{C}+\mathrm{BB}^T\mathrm{C}+\mathrm{AA}^T\mathrm{C}+\mathrm{C})_{ij} = \\ \sum_l(\mathrm{SS}^T)_{il}\mathrm{C}^t_{lj} + \sum_l(\mathrm{BB}^T)_{il}\mathrm{C}^t_{lj} + \sum_l(\mathrm{AA}^T)_{il}\mathrm{C}^t_{lj} \\ \geq (\mathrm{SS}^T)_{ii}\mathrm{C}^t_{ij} + (\mathrm{BB}^T)_{ii}\mathrm{C}^t_{ij} + (\mathrm{BB}^T)_{ii}\mathrm{C}^t_{ij} \tag{63}$$

To prove the convergence of the algorithm, we may make use of one property of an auxiliary function that is also used in the proof of the Expectation-Maximization

Up to here, $Q(c, c^{t}_{ij}) \leq O_{cij}(c)$ has been proved thus (54) is an auxiliary function for $O_{cij}$. Next, we will define the auxiliary functions regarding the updating rules for the inference of **S**, **B**, and **A**, which are shown in (50), (53), and (56). Similarly, let $O_{Sij}$, $O_{Bij}$, and $O_{Aij}$ be the partial of (41) relevant to $\mathbf{S}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{A}_{ij}$, $O_{Sij}(\mathbf{S}'_{ij})$, $O_{Bij}(\mathbf{B}'_{ij})$, and $O_{Aij}(\mathbf{A}'_{ij})$ be the partial objective values when $\mathbf{S}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{A}_{ij}$ equal to $\mathbf{S}'_{ij}$, $\mathbf{B}'_{ij}$ and $\mathbf{A}'_{ij}$, respectively. Since the updating rules for the inferring **S**, **B**, and **A** are also element wise, it is sufficient to show that $O_{Sij}$, $O_{Bij}$, and $O_{Aij}$ are non-decreasing according to the updating rules (50), (53), and (56). Let the following be the auxiliary functions regarding to $O_{Sij}$, $O_{Bij}$, and $O_{Aij}$:

$$\begin{gathered}
Q(s, \mathrm{S}^{t}_{ij}) = \\
O_{S_{ij}}(\mathrm{S}^{t}_{ij}) + O'_{S_{ij}}(d - \mathrm{S}^{t}_{ij}) - \frac{(\mathrm{CC}^{T}\mathrm{S} + \mathrm{S})_{ij}}{2\mathrm{S}^{t}_{ij}}(s - \mathrm{S}^{t}_{ij})^{2} \\
Q(b, \mathrm{B}^{t}_{ij}) = \\
O_{b_{ij}}(\mathrm{B}^{t}_{ij}) + O'_{b_{ij}}(b - \mathrm{B}^{t}_{ij}) - \frac{(\mathrm{CC}^{T}\mathrm{B} + \mathrm{B})_{ij}}{2\mathrm{B}^{t}_{ij}}(b - \mathrm{B}^{t}_{ij})^{2} \\
Q(a, \mathrm{A}^{t}_{ij}) = \\
O_{b_{ij}}(\mathrm{A}^{t}_{ij}) + O'_{b_{ij}}(a - \mathrm{B}^{t}_{ij}) - \frac{(\mathrm{CC}^{T}\mathrm{A} + \mathrm{A})_{ij} + \alpha}{2\mathrm{A}^{t}_{ij}}(a - \mathrm{A}^{t}_{ij})^{2}
\end{gathered} \tag{64}$$

Since the proof for the above functions to be auxiliary functions for $O_{Sij}$, $O_{Bij}$, and $O_{Aij}$ is similar to that for $O_{Cij}$, we don’t show the proof in detail due to the space limitation. Having obtained the auxiliary functions for $O_{Cij}$, $O_{Sij}$, $O_{Bij}$, and $O_{Aij}$, now we can show the convergence of (44) using the updating rules (47), (50), (53) and (56). Since (58) is an auxiliary for $O_{Cij}$, according to (59), we have

$$\begin{aligned}
\mathrm{C}^{t+1}_{ij} &= \arg\max_{c} Q(c, \mathrm{C}^{t}_{ij}) \\
&= \mathrm{C}^{t}_{ij} \cdot \frac{(\mathrm{SM} + \mathrm{BX} + \mathrm{AF})_{ij}}{(\mathrm{SS}^{T}\mathrm{C} + \mathrm{BB}^{T}\mathrm{C} + \mathrm{AA}^{T}\mathrm{C} + \mathrm{C})_{ij}}
\end{aligned} \tag{65}$$

The above result is same to the updating rule (47). Since (58) is an auxiliary function, $O_{Cij}$ is non-decreasing when $\mathbf{C}_{ij}$ is updated according to (65) or (47). This is equivalent to say that $O$ is non-decreasing when $\mathbf{C}_{ij}$ is updated according to (47) for $\mathbf{C}_{ij}$ is any element of **C**. Since (64) are auxiliary functions for $O_{Sij}$, $O_{Bij}$, and $O_{Aij}$, according to (57), we have

$$\begin{aligned}
\mathbf{S}_{ij}^{t+1} &= \arg\max_{s} Q(s,\mathbf{S}_{ij}^{t}) = \mathbf{S}_{ij}^{t} \cdot \frac{[\mathbf{CM}+\mathbf{B}]_{ij}}{[\mathbf{CC}^{T}\mathbf{S}+\mathbf{S}]_{ij}} \\
\mathbf{B}_{ij}^{t+1} &= \arg\max_{b} Q(b,\mathbf{B}_{ij}^{t}) = \mathbf{B}_{ij}^{t} \cdot \frac{[\mathbf{CX}+\mathbf{S}]_{ij}}{[\mathbf{CC}^{T}\mathbf{B}+\mathbf{B}]_{ij}} \\
\mathbf{A}_{ij}^{t+1} &= \arg\max_{a} Q(a,\mathbf{A}_{ij}^{t}) = \mathbf{A}_{ij}^{t} \cdot \frac{[\mathbf{CF}^{T}]_{ij}}{[\mathbf{CC}^{T}\mathbf{A}+\mathbf{A}]_{ij}+\alpha}
\end{aligned} \tag{66}$$

The above results are same to the updating rules (50), (53), and (56). Since (64) are auxiliary functions, $O_{Sij}$, $O_{Bij}$, and $O_{Aij}$ are non-decreasing when $\mathbf{S}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{A}_{ij}$ are updated according to (50), (53), and (56). This is equivalent to say that $O$ is non-decreasing when $\mathbf{S}_{ij}$, $\mathbf{B}_{ij}$ and $\mathbf{A}_{ij}$ are updated according to (50), (53), and (56), respectively. The above proof shows that $O$ is non-decreasing when **C**, **S**, **B** and **A** are iteratively updated according to (47), (50), (53) and (56). Thus, we have

$$\begin{aligned}
& O(\mathrm{C}^0, \mathrm{S}^0, \mathrm{B}^0, \mathrm{A}^0) \leq O(\mathrm{C}^1, \mathrm{S}^0, \mathrm{B}^0, \mathrm{A}^0) \\
& \leq O(\mathrm{C}^1, \mathrm{S}^1, \mathrm{B}^0, \mathrm{A}^0) \leq \cdots \leq O(\mathrm{C}^{opt}, \mathrm{S}^{opt}, \mathrm{B}^{opt}, \mathrm{A}^{opt})
\end{aligned} \tag{67}$$

where $O$ shows a non-decreasing trend in each iteration of updating and it may finally achieve the local optima.

### 5.2.3.4 The termination of optimization

As **C**, **S**, **B** and **A** are iteratively updated, the objective value converges to the local optima asymptotically. Simultaneously, the variation of the four matrices becomes less evident as the elements in each matrix are approximate to the magnitudes which lead the objective value to local optima. Thus, we may use the following stopping criterion to terminate the optimization process and LFCIS may obtain optimal latent factors in matrices **C**, **S**, **B** and **A** that lead $O$ to converge approximately

$$\left| \mathrm{C}^{i} - \mathrm{C}^{i-1} \right|_{F} < \tau \tag{68}$$

where $\mathbf{C}^{i}$ stands for the latent space representing the community membership after the $i$th iteration of updating, $\tau$ represents the predefined tolerance which the Frobenius norm of the difference of **C** between two iterations should satisfy. When $\tau$ is set to be a relatively small value, LFCIS may obtain a latent matrix **C** which is very approximate

```
Inference of latent factors in LFCIS
Input:              M, X, F, $\alpha$, $max\_iteration$, $\tau$, $k$

Output:             C, S, B, A
Randomly initialize C, S, B, A;

for count=1: $max\_iteration$
    Fixing S, B, A
     update C according to (11);
    Fixing C
     update S according to (14);
    update B according to (17);
    update A according to (20);
     if ($|\mathbf{C}^i$ - $\mathbf{C}^{i-1}|_F<\tau$)
        compute objective value according to (9);
        break;
     end if
end for

return C, S, B, A;
```

Fig. 25. Model fitting of LFCIS

to the optimal.

**5.2.3.5** Summary remarks

Having obtained the updating rules for **C**, **S**, **B** and **A** and the stopping criterion for the optimization process, now we may describe the details of LFCIS. Based on the aforementioned description, the proposed latent factor model can be summarized as the pseudo codes shown in Fig. 25. As it is seen in the figure, there are not many parameters that need to be input. After the parameters of maximum number of iteration *max_iteration*, tolerance for improvement $\tau$, penalty factor $\alpha$ and the dimensionality of latent spaces, $k$ are determined, LFCIS will iteratively update the matrices **C**, **S**, **B** and **A**, in which are the latent representations representing the community membership and features, till the variation of **C** between each two iterations is less than $\tau$ or the objective function converges to the local maxima. After the optimization process is terminated, LFCIS obtains the matrices for community membership and features. In this case, **C** and

**A** which contain the optimal or approximately optimal membership between each vertex and *k* communities and the communities representations generated based on the *m* attributes in G. Given **C**, LFCIS can directly identify the best community membership for each vertex in the attributed graph and in this regard, it is more efficient than other approaches.

### 5.2.4 Experiments and analysis

To evaluate the effectiveness of LFCIS, we performed some experiments using both synthetic and real-world datasets. In this section, we will detail the data sets we use, the criteria we used to evaluate the performance, and how we performed the experiments.

#### **5.2.4.1** Experimental set-up and performance metrics

- Data sets descriptions

We used both synthetic and real datasets with known ground truth for performance evaluations. We used synthetic data to test the effectiveness, efficiency of LFCIS and other compared baselines, and parameter sensitivity of LFCIS. We used real-world datasets to test the robustness of different algorithms. The details of datasets we used are described below.

There are five real-world datasets used in our experiments, including Caltech [115], Twitter [97], Ego-facebook [167], Googleplus-1, and Googleplus-2 [167]. Compare with the experiment of DMNF, this part of the experiment also introduced Twitter, Googleplus-1, and Googleplus-2 data sets

Twitter dataset is constructed based on a number of social circles extracted from twitter.com. For this dataset, there are 2511 vertices representing twitter users, 37154 edges representing the friendship between them, and 9067 attributes, representing social topics they concern, and the locations where the users post twits. There are 132 social circles that have been verified as ground truth communities.

Googleplus-1 is a set of online social network data which are collected from plus.google.com. There are 5630 vertices, 463537 edges, and 4229 attributes in the dataset. In this dataset, vertices, edges, and attributes represent googleplus users, friendships, and user profiles, respectively. There are 58 social circles that have been verified as ground truth communities in Googleplus-1.

Googleplus-2 is another set of social network data which are constructed based on the sub-networks from plus.google.com. There are 7856 vertices, 321268 edges, and 2024 attributes in the dataset. In this dataset, there are 91 social communities of ground truth that are able to be used for benchmarking the identified ones.

Syn1k is a set of synthetic data which is generated based on the rule that the probability of intra-community edges is higher than that of inter-community edges and that vertices in the same cluster are more related to each other than those that are not. For this dataset, we used 1000 vertices that are divided into 4 disjoint ground truth communities, 9900 edges and 50 attributes that are possibly associated with each vertex.

The above data sets are used to test the effectiveness of LFCIS and other algorithms. In addition, to test the scalability of LFCIS, we have generated several additional synthetic datasets ranging in size from 5,000 to 100,000 for our experiments.

- Evaluation metrics

For performance evaluation, we are considering different evaluation measures which are widely used for evaluating graph clustering algorithms. For measures used for validating graph clusters, we also used the Normalized Mutual Information (NMI) same with the measurement of DMNF, and the Average Accuracy (Acc) [168].

The NMI measures the overall accuracy of the matches between detected communities and those that are considered as “ground truth”. Contrary to the NMI, the Acc measure evaluates individually detected community. It is defined as

$$Acc = \sum_{c} \frac{|C_i|}{|C|} f\left(C_i, C^*\right) \tag{69}$$

where $|C|$ means the size of the detected communities, and $f(\cdot)$ stands for a mapping function between cluster $i$ and the ground truth. For our purpose, we define $f(\cdot)$ to be the maximum overlap between detected community *i* and a ground-truth community. Thus, *Acc* evaluates the best matching of each cluster. A higher value of *Acc*, therefore means that each detected community has a better match with the ground truth. The higher the *Acc* of all communities detected by an algorithm, therefore means that the algorithm is more effective.

- Baselines for comparison

To test the effectiveness of LFCIS, we selected a number of approaches as compared baselines. These algorithms include Affinity Propagation clustering (AP), Spectral clustering (SC), k-means clustering, Relational topic model (RTM), ECDA, and MISAGA. Selecting these algorithms as baselines are because they are either the latest algorithms or classical ones and have all been used effectively to detect network communities in various networks. Specifically, AP and SC may detect graph clusters that take different topological properties of network graph data. For our experiments, we used the SC that makes use of the normalized cut in graph clustering. K-means is able to detect graph communities by grouping together those vertices with similar attributes. Therefore, we used the information in Λ as the input that is used to compute the similarity between pairwise vertices for k-means. Algorithms like RTM, ECDA, and MISAGA are ones taking into consideration both graph topologies and attributes. RTM has been shown to be a very effective topic-model based approach to segment relational data. ECDA performs its tasks using an evolutionary graph clustering algorithm. MISAGA is a very effective algorithm which is proposed recently. It can perform the task of community detection in graphs by taking into the consideration edge structure and

| *Approach* | *NMI* | *Acc* |
|---|---|---|
| *AP* | 0.152 | 0.747 |
| *SC* | 0.232 | 0.528 |
| *k-means* | 0.691 | 0.835 |
| *RTM* | 0.797 | 0.797 |
| *ECDA* | 0.272 | 0.466 |
| *MISAGA* | 0.981 | 0.996 |
| *LFCIS* | **0.995** | **0.999** |

Table 13. NMI and Acc in Syn1k

attribute similarity between pairwise vertices.

For performance benchmarking, we used the source code or executables made available by the authors. All the experiments were conducted in the same environment which included a workstation with 4-core 3.4GHz CPU and 16GB RAM.

### **5.2.4.2** Experimental results using synthetic data

- The performance of community detection

For performance evaluation, we used a set of synthetic network data containing 1000 vertices to test the effectiveness of all different algorithms. There are four disjoint ground truth clusters in the synthetic dataset. As mentioned above, the synthetic data are generated by assuming that the probability of vertices within the same community to be connected with other vertices to be higher than that of the probability between communities. For our experiment, the data set Syn1k was generated by setting the probability of intra-community connections to be 0.05 and the probability of inter-community connections to be 0.01.

The performance of LFCIS and other algorithms on the synthetic dataset Syn1k concerning  NMI, and Acc is given in Table 13. As the table shows, LFCIS performs

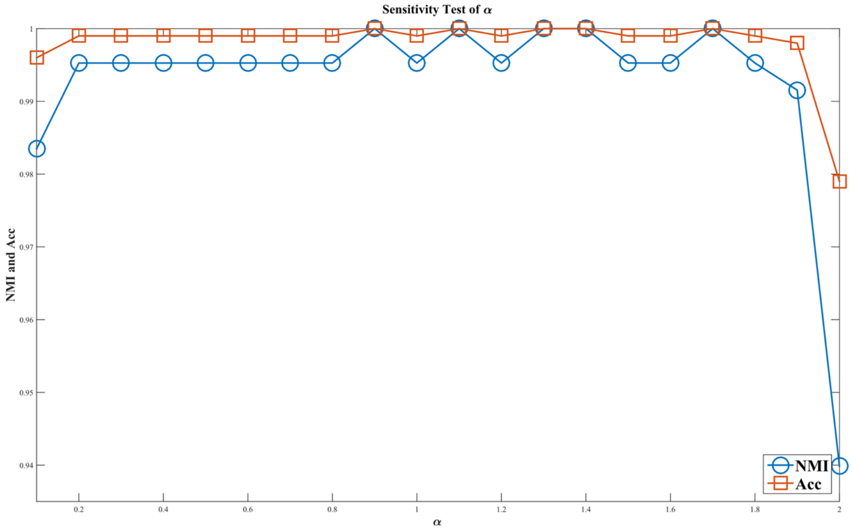


Fig. 26.  Sensitivity test of α.

better than other algorithms. No matter which of NMI, or Acc is considered, LFCIS may outperform all the compared baselines in dataset Syn1k. These experimental results show that LFCIS can be very effective with the discovering of communities in the synthetic attributed graph.

- Sensitivity test of α

As mentioned in the above section, there is only one parameter, α, which is used to control the sparsity of A in LFCIS that might take effect on the performance of the model. To investigate how the parameter α may take effect on the performance of LFCIS, we performed the sensitivity test using the dataset Syn1k. In our experiment, α was set to different values from 0.1 to 2, with an increment of 0.1, and LFCIS was used under these different settings to fit the latent factors for discovering communities. The performance was measured with NMI, and Acc and the results are shown in Fig. 26.

As it is shown in the figure, LFCIS may obtain a worse performance when α is set to be either near 0, or near 2. LFCIS may perform steadily when α is set to a value between 0.2 and 1.5. In our experiments, we set α to 0.5, when LFCIS performs the tasks of community identification and summarization in all the datasets. Using this setting may guide LFCIS to exclude those attributes with relatively lower possibility of being ones that may characterize the identified communities while preserving those that are more possible to be community representations.

### **5.2.4.3** Experimental results using real-world data

Community detection is significant to network analytics. To test the effectiveness of LFCIS and other compared baselines, we use them to perform the task of community detection in five sets of real-world network graph data, including Caltech, Twitter, Ego-facebook, Googleplus-1, and Googleplus-2. These five sets of real-world data are different from vertex size and the dimensionality of attributes that are used to characterize the vertices. All these datasets have known ground truth communities which have been verified in the previous works. For this reason, the performance of LFCIS and other baselines can be more objectively compared.

Table 14. Experimental results in real-world data

| ***Dataset*** / ***Approach*** | *Caltech* | | *Twitter* | | *Ego-facebook* | |
|---|---|---|---|---|---|---|
| | *NMI* | *Acc* | *NMI* | *Acc* | *NMI* | *Acc* |
| *AP* | 0.34 | 0.458 | 0.598 | 0.479 | 0.528 | 0.416 |
| *SC* | 0.338 | 0.335 | 0.493 | 0.305 | 0.52 | 0.447 |
| *k-means* | 0.176 | 0.268 | 0.298 | 0.237 | 0.385 | 0.276 |
| *RTM* | 0.11 | 0.146 | 0.028 | 0.099 | 0.227 | 0.167 |
| *ECDA* | 0.148 | 0.202 | 0.529 | 0.385 | 0.322 | 0.234 |
| *MISAGA* | 0.2 | 0.256 | 0.65 | **0.503** | 0.54 | 0.452 |
| *LFCIS* | **0.485** | **0.475** | **0.67** | 0.493 | **0.613** | **0.512** |
| *Improvement (%)* | **42.65** | **3.71** | **3.08** | -2.03 | **13.52** | **13.27** |

| ***Dataset*** / ***Approach*** | *Googleplus-1* | | *Googleplus-2* | |
|---|---|---|---|---|
| | *NMI* | *Acc* | *NMI* | *Acc* |
| *AP* | 0.412 | 0.525 | 0.355 | 0.273 |
| *SC* | 0.284 | 0.321 | 0.33 | 0.296 |
| *k-means* | 0.16 | 0.221 | 0.154 | 0.195 |
| *RTM* | 0.075 | 0.309 | 0.023 | 0.151 |
| *ECDA* | 0.443 | 0.468 | 0.333 | 0.255 |
| *MISAGA* | 0.546 | 0.735 | 0.399 | 0.363 |
| *LFCIS* | **0.578** | **0.741** | **0.457** | **0.473** |
| *Improvement (%)* | **5.86** | **0.82** | **14.54** | **30.3** |

The experimental results of NMI and Acc obtained with these datasets are summarized in Table 14. As the table shows, LFCIS performs more robustly, compared with other baselines. When NMI is considered, LFCIS is better than any other baselines in all the five datasets. LFCIS outperforms the second-best methods by 42.65%, 3.08%, 13.52%, 5.86%, and 14.54% in Caltech, Twitter, Ego-facebook, Googleplus-1, and Googleplus-2, respectively. When Acc is considered, LFCIS is better than any other baselines, except the case in Twitter dataset. In Caltech, Ego-facebook, Googleplus-1, and Googleplus-2, the improvement related to Acc, is 3.71%, 13.27%, 0.82%, and 30.3%, respectively, when LFCIS is compared with the second-best algorithms. Given the robust performance obtained by LFCIS in these real-world datasets, it is said that LFCIS is a very effective model for identifying latent communities in network data, while ensuring the community representations also to be identified.

### 5.2.5 Summary

In this section, a novel latent factor model for community detection and summarization, LFCIS is proposed. By taking into the consideration edge structure and attribute similarity between each pair of vertices in the network, LFCIS is able to find the optimal assignment of community membership for each vertex by making use of a convergent updating algorithm to fit the latent factors. Different from prevalent approaches that focus on either community identification or community summarization, LFCIS is able to summarize the representations of each identified community while performing the task of community identification in the network. Such representations are able to characterize both community itself and their members. Having been used with both synthetic and real-world network data, LFCIS outperforms most state-of-the-art approaches in experiments related to the test of effectiveness and efficiency. It is concluded that LFCIS is a very promising approach to identifying communities and summarizing their representations in the network data.

# 6. ONE CLASS REPRESENTATION

There is a particular case in the task of link prediction of the network, which is sample imbalance. An essential problem in the task of supervised link prediction is that there are very few samples of a particular type. In other words, the corresponding features representing one class of samples are also very few. Small samples also mean few patterns to be excavated, and machine learning does not work well in this case. For example, the exception instance in fault prognostic usually accounts for only one thousandth of the entire sample. It has become a huge challenge to reconstruct the one-class representation and improve the prediction using the representations of one-class samples. This means that just example objects of the target class can be used and that no information about the other class of outlier objects is present. In this section, we present a novel approach for such a purpose. It applies a one-class representation algorithm to discover association patterns between interacting targets from their original features.

## 6.1 Overview

Drug and target protein interaction represents the procedural nature of drug act on the human body, and thus a crucial step in drug discovery is the identification of small molecules that effectively modulate the functions of disease-related target proteins. Past discovery campaigns have indicated that the high failure rate of drug discovery can be mainly attributed to the improper DTIs (drug-target interactions) occurrence. Even though data manufacturing technologies are accelerating faster than Moore's Law, drug discovery is still a costly and inefficient process. The cost of discovering a new FDA approved drug has doubled every 9 years since 1950, with costs for each new drug estimated at $2.6 billion from a 2013 estimation [169-170]. And with the enormous investments, pharmaceutical communities lock into their greatest losses when a drug fails in the later stages of development and post-market. Since the Human Genome Project (HGP) presented in the 80’s late and early 90’s, it gives a future promise of increasing the number of the potential target protein and carries it out. Many DTIs related information like molecule structure, protein sequence, and protein functions have been collected to public databases. For example, there are hundreds of thousands human proteins are recorded in UniProtKB database [171]. On the other hand, there are only around thousands of known drug compounds are deposited in Drug Bank [21]. Other databases such as Super Target and Matador [172] and Therapeutic Target Database (TTD) [37] have been designed as resources for target functions. Therefore, the existed a huge number of unexplored compounds and human proteins make it impossible to evaluate drug-target interactions effectively by biological experiment. Normal drug discovery processing may generate products different from the original treatment. Instability and no specificity of DTIs have to be addressed appropriately during the screening and clinical phase. In addition, possible drug-target interactions also supporting other drug discovery work like drug combination prediction, adverse drug reaction prediction new biomarkers discovery [173]. To reduce the huge time and financial cost of experimental approaches, many computational models have been built to elucidate interesting drug-target relationships of most promising candidates for further experimental validation. Various methods are caring drug similarity and drug-target nature representations respectively [54]. Similarity-based methods are developed to identify biological interaction by including the similarity matrices of related entities. Multiple computational techniques have been proposed to discover DTIs based on their molecule and protein sequence similarity [174-175], [61, 119]. An attractive approach is to integrate various descriptions of drug-target from multiple sources in a machine learning framework. Feature vector-based methods are regarded as more advanced strategies that face drug and protein features straightforward. These methods provide biological representations for learning interest patterns such as compound subset and protein subspace. But, current feature-based methods are not able to train full DTIs because of the large computing cost. It's also difficult for current techniques to build a prediction model rely on reliable positive samples without the uncertain negative sample. Fortunately, it is widely believed such a problem can be resolved by one-classification techniques as follows:

- Over-represented representations observed in interacting drug-target links can be used to infer DTIs. Drug chemical information and sequence descriptor about nature features can be used to predict potential DTIs as well.
- Original large-scale candidate interactions have been extracted from DTIs matrix. A promising approach is to combine various representations and propose a one-class classification method for only keep reliable positive samples in our training model. And

all positive and negative samples can be used to test in this model.

In this chapter, we develop a new feature-based method called ODT (one class drug-target interaction prediction). ODT predicts drug-target interactions from protein sequence descriptors and molecule fingerprints with one-class classification aiming at training a robust training model to rely on known positive samples. Firstly, we introduce two descriptor approaches to discover protein sequences descriptors and use chemical fingerprints for representing the chemical space. Secondly, to solve the difficult problem of no certain negative samples, we introduce Support Vector Data Description (SVDD) to judge whether one drug interacts with one target by describing a boundary. Our method constitutes a significant advance because it logically just considers certain nature representations of positive known DTIs that remained as training data. We adopt all popular data standard to test the proposed method which includes G protein-coupled receptor, enzyme, ion channel, and nuclear receptor dataset [61]. ODT has been tested with Gold standard data sets that can be a beneficial approach to predict the DTIs.

### 6.2 ODT in details

The crucial issue of drug-target interactions problem is that reliable interactions are available for the only positive class, called the known interactions. In turn, most negative instances cannot be regarded as absolutely non-interactive samples for the negative concept. In experimental biology, it is hard to obtain certain interaction instances, and we have not enough resource to filter all the negative samples. Before, it is very challenging to label the uncertain samples when we train the machine learning model to predict interaction. When deciding which drug and protein to be grouped, previous approaches usually make use of both the positive and all negative interaction samples. Because too large-scale samples are trained, and some of them are wrongly labeled, general ways are difficult to build an efficient and reliable classifier. Consequently, a smart classification method is needed to generate a more accurate descriptive model based on representations of a certain positive sample.

A skillful one-class classification model called Support vector data description (SVDD) [176-178] is constructed with the aim of characterizing the representation of one-class examples, and then it is used to distinguish test examples which should be classified into the target category or not. A pretty subtle step of SVDD is making a constrained geometry with less number to describe the data by mapping them to a high-dimensional space. Here, the boundary provides a standard for classifying the outliers and targets. The optimal solution is one kernel find a spherically shaped boundary around the targets that have minimum volume containing all data. SVDD like to detect the radius and the center of the hypersphere by using the known data samples. The sketch map of SVDD in 2D graph is presented in Fig. 27. Given an instance set $X = [x_1, x_2, \ldots, x_i]^T \in R^{N \times M}$, where $N$ is the number of drug-target interaction instances, and $M$ is the number of their representations.

As SVDD considers all data points with the same importance, the trained model is very sensitive to noise points which greatly affect the acceptance accuracy of SVDD [33]. The quantity of noise point in drug-target representations is considerable compared with the general data which should be reduced before the hyperplane making. Information gain ratio is a typical feature selection technique, here we present it to process the original high-dimensional data first. It is a widely used entropy-based analysis technique that allows reducing the dimensionality of the features while preserving information on the classified influence. The basic idea of information gain ratio [34] is to determine a ratio of information gain to the intrinsic information. Information gain tells us which

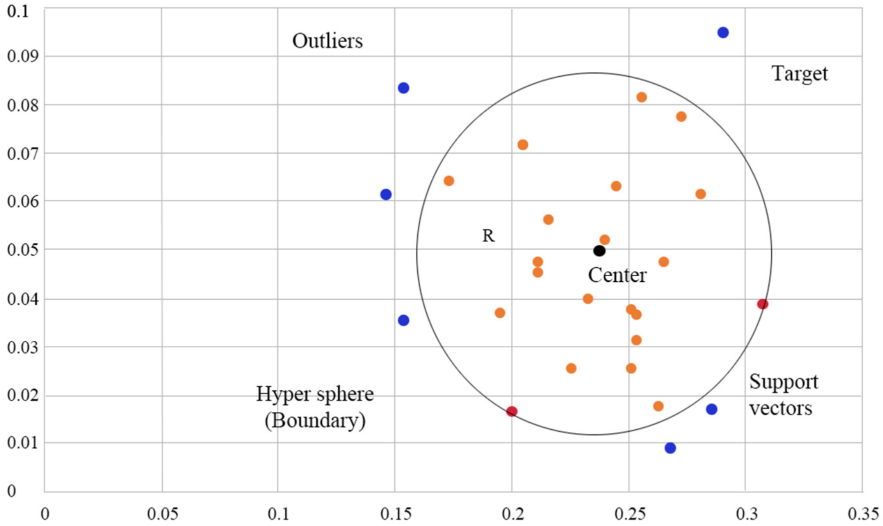


Fig. 27. Sketch map of SVDD in 2D

training attribute is most useful for deciding the instance's classification. All original high-dimensional drug-target patterns can be optimally transformed to an abridged feature space with lower dimensionality.

Usually, SVDD first maps new input space with a nonlinear transformation to the feature space via a mapping function $\varphi$ (.). Then, the general task is determining a sphere with a minimal volume containing all or most of the mapped data objects in the feature space. For a drug-target interaction data set containing N samples $\{S_i,\ i = 1, \ldots, N\}$, let hypersphere characterized by a center α and radius R. Hence, SVDD is then to describe the input samples by using a hypersphere with minimized radius for a minimum sphere volume that enclosed the target samples as many as possible. If some given objects have a large distance from the center, the built large sphere may decrease the model performance. Therefore, the far outliers should be penalized, we allow for some outliers outside the sphere by introducing slack variables $\xi_i$ associated with the deviation, and the goal is to describe the hypersphere with minimum radius R. The error function to minimize radius is an optimization problem as follows.

$$minL(R, a) = R^2 + C\sum_{i=1}^{N}\xi_i \tag{70}$$

subject to the constraints:

$$\|x_i - a\|^2 \leq R^2 + \xi_i \tag{71}$$

$$\xi_i \geq 0 \forall i.$$

where the parameter *C* controls the tradeoff between the size of the hypersphere and the miss errors. Eq. (71) can be incorporated into Eq. (70) by using Lagrange multipliers with the $\alpha_i \geq 0$ and $\beta_i \geq 0$. The corresponding Lagrange function becomes:

$$L(R, a, \alpha_i, \beta_i, \xi_i) =$$

$$R^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}(R^2 + \xi_i - \|x_i - a\|^2)\alpha_i - \sum_{i=1}^{N}\beta_i\xi_i \tag{72}$$

by using Lagrange multipliers with the $\alpha_i \geq 0$ and $\beta_i \geq 0$. The corresponding Lagrange function becomes:

The Lagrange function should be minimized with respect to $R$ ,$\xi_i$ , $a$, and maximized with respect to $\alpha_i$ and $\beta_i$ with the constraints. We should set the following limit conditions at the solution point:

$$\sum_{i=1}^{N} \alpha_i = 1 \tag{73}$$

$$a = \sum_{i=1}^{N} \alpha_i x_i \tag{74}$$

$$C = \alpha_i + \beta_i \tag{75}$$

where the $0 \leq \alpha_i \leq C$ from Eq. (75) because the $\alpha_i \geq 0$ and $\beta_i \geq 0$ . Then resubstituting Eqs. (73) - (75) into Eq. (72), the problem will turn into maximizing the following function $L$:

$$max \;\; L = \sum_{i=1}^{N} \alpha_i (x_i \cdot x_i) - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i, \alpha_j (x_i \cdot x_j) \tag{76}$$

subject to the constraints:

$$0 \leqslant \alpha_i \leqslant C, \; a = \sum_{i=1}^{N} \alpha_i x_i , \sum_{i=1}^{N} \alpha_i = 1 \tag{77}$$

To detect meaningful clusters in the graph data, there have been several so-called graph clustering algorithms proposed. These algorithms can be categorized based on the information of graph data they utilize

According to the location of the training instances, there are three types of training instances. If $x_i$ is mapped to the inside of the hypersphere, $\alpha_i$ becomes zero. And the corresponding data samples $\alpha_i$ are called support vectors. If the training instance $x_i$ is mapped on the hypersphere, $\alpha_i$ lies on between 0 and C. If the $x_i$ is mapped to the outside of the hyperspheres, $\alpha_i$ becomes C.

When an object $z$ be tested, it requires the calculation of the distance from the object $z$ to the center of the hypersphere. The distance to the center of the hypersphere is calculated by equation (78). The test object $z$ is accepted within the hypersphere when the distance is smaller than the radius $R$.

$$\| z - a \|^2 = (z \cdot z) - 2 \sum_{i=1}^{N} \alpha_i (z \cdot x_i) + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i, \alpha_j (x_i \cdot x_j) \leqslant R^2 \tag{78}$$

$R^2$ is the squared distance from the center of the sphere a to the boundary. If $\alpha_i = C$ , the support vectors fall outside the sphere are excluded. Therefore:

$$R^2 = (x_k \cdot x_k) - 2 \sum_{i=1}^{N} \alpha_i (x_i \cdot x_k) + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i, \alpha_j (x_i \cdot x_j) \tag{79}$$

Figure 28 shows a Samples of Gold-standard dataset by random three features in a 3D graph. Because the sphere is not always a tight description for the boundary of sample distribution, we can replace the inner product by a kernel function $K(x,\ y)\ =\ (\varphi(x)\ \cdot\ \varphi(y))$ that make more flexible. Some kernel functions have been used for the SVDD classifier. The Gaussian kernel is a nonlinear function that maps the sample data to a new feature space, and it has been proved to work well on the data descriptions in many previous cases. Hence, we adopt the Gaussian kernel function in this work. Then, we replace the inner product, and it will be obtained in the following form:

$$K\left(x_i, x_j\right) = exp\left(\frac{-\left\|x_i - x_j\right\|^2}{\sigma^2}\right),\ \sigma > 0 \tag{80}$$

where the variance parameter $\sigma$ is a width parameter that controls how tight the description is around the data.

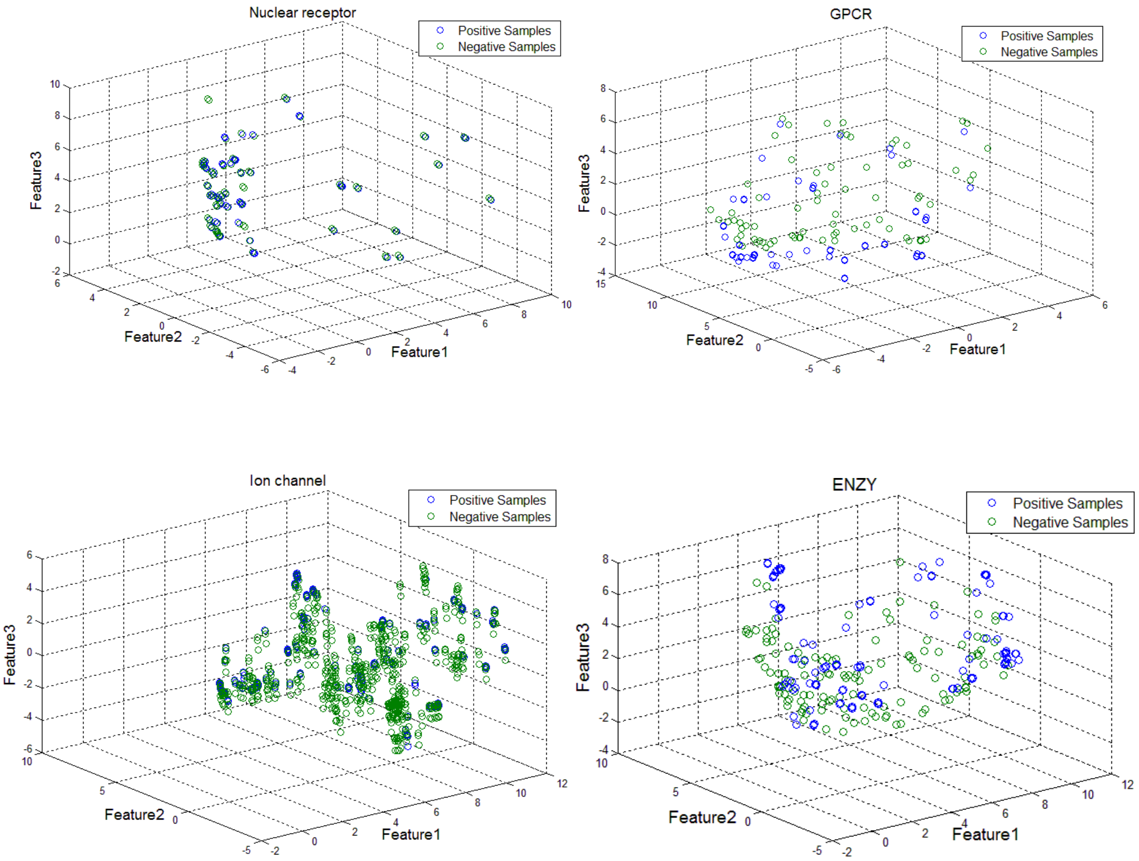


Fig. 28. Samples of Gold standard data set by random three features in 3D graph

## 6.3 Experiments

### 6.3.1 Performance Evaluation

For drug-target interactions, we used the DTIs dataset from Golden Standard Dataset same with the MFDR used. The ratio of positive-negative to enzymes, ion channels, GPCRs, and nuclear receptors is 99.984, 28.024, 32.362 and 14.6 respectively. Previous feature-based approaches have randomly selected negative samples from the non-interactions until the ratio hitting the one-to-one scale. And yet, we considered all drug-target interactions in our work. Due to our DTIs dataset is a high-class imbalance, we use 5-fold cross-validation, where each fold leaves out 20% of the positive and negative samples for testing. Since the true positive examples are too small, that may lead to no positive training samples if randomly divide the dataset. We separate positive and negative samples and divide both into each fold equally. Regarding extreme imbalance prediction performance evaluation, it has argued for using ranking measures like AUROC (area under the ROC curve) that the prediction performance can be safely unbiased. We infer interactions and compare against the held-out interactions, measuring performance using the AUC for our evaluations. ROC curves are created by plotting the true positive rate versus the false positive rate at various thresholds. We then computed precision-recall curves for each dataset. AUPR (area under the precision-recall curve) which takes into account both recall and precision, calculated as follows $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$. We also used the grid search to select the best parameter C (0.1-1) and sigma (0.1-2.5) for the Gaussian kernel based on the ROC scores. SVDD and evaluation methods were implemented using Dd_tools [178]. Since the high-dimensional of protein and drug descriptors, we collected 1448 features for each drug-target interaction sample outlined by protein profiles and chemical profiles. The resulting AUROC scores of ODT for nuclear receptors, GPCRs, Ion channels, and Enzymes are 0.766 obtained when C=0.5 and σ=1.3, 0.885 obtained when C=1 and σ=1.15, 0.847 obtained when C=0.2 and σ=2.15 and 0.850 obtained when C=0.85 and σ=1.15 respectively. The resulting AUPRC scores of ODT for nuclear receptors, GPCRs, Ion channels, and Enzymes are 0.3762, 0.2992, 0.3661 and 0.3263 respectively. As Figure 29 shows, our average prediction accuracy of a 5-fold dataset is good.

### 6.3.2 Performance Comparison

At present, many studies are subtly designed for classifying the unknown drug and protein interactions. However, all feature-based methods are restricting practicability in the unreliable and incomplete training data. And, most similarity-based methods are restricting biological availability in the unessential representations. Thus, a reliable and efficient computational way is highly required to extract accurate information and target label directly. To evaluate the performance of ODT, we tested it with several previous works for comparison. Since traditional classifiers are time-consuming and costly to identify the high-dimensional drug-protein representations consist in hundred thousand DTI samples by performing experimental tests, there is no similar feature-based prediction work can support whole large-scale DTIs prediction. Because of computation limitations, we examined the prediction performance for original SVM on Nuclear receptors and GPCRs. We still used the grid search to select the best parameter C (1-50)

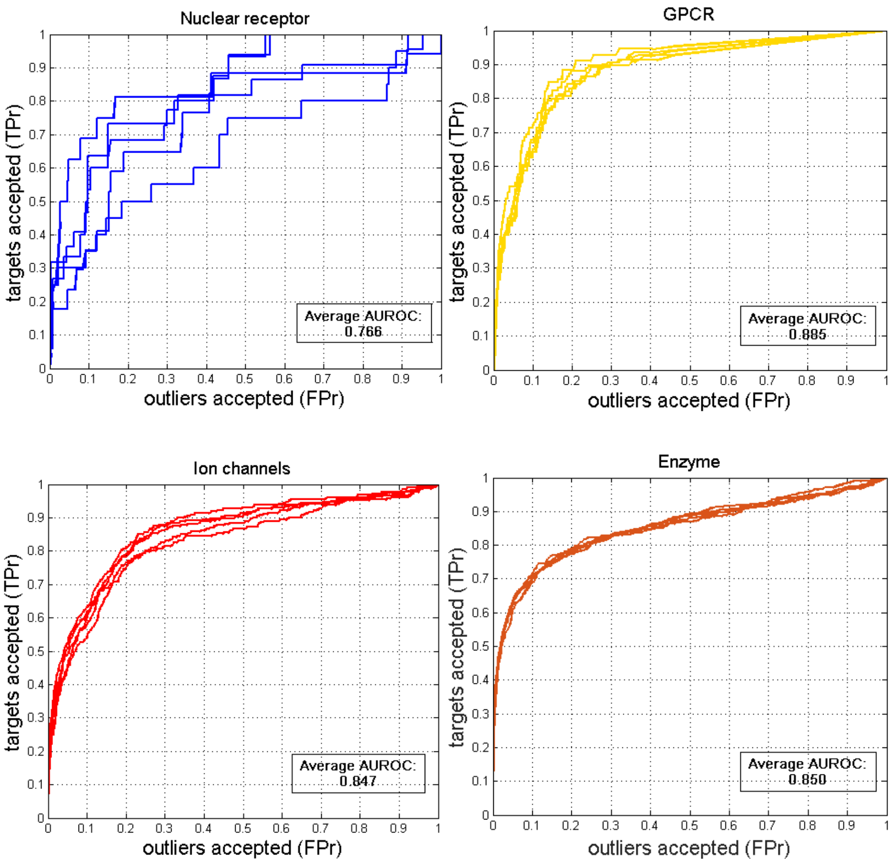


Fig. 29. 5-Fold ROC curves of Nuclear Receptor, GPCR, Ion channel, and Enzyme

and sigma (0.001-1) for the RBF kernel based on the ROC scores. The best AUROC of SVM for Nuclear receptors and GPCRs are 0.7917 and 0.641 respectively. It seems that the proposed performance of SVDD similar to SVM on Nuclear receptors dataset. However, SVDD outperforms SVM on GPCRs which demonstrates the proposed SVDD have better prediction power on the larger scale and more imbalanced DTIs.

We further considered five important studies in this similarity-based area which have been implemented in the whole DTIs. As AUC is the most commonly reported measure in our related publications and it allows us to compare against the published results of other methods on the same dataset. We compared the best AUROC scores of the ODT with these approaches including KBMF2K [73], NetCBP [74], Bipartite Graph Learning [62], DBSI [119] and PUDT [55]. Table 15 shows the AUROC scores of ODT and others divided by four interaction types. As the results look like that shown above, the prediction accuracy of the ODT is superior in comparison with most methods at GPCRs and Ion channels. In addition, instead of the common two class classifier, we made use of one class solution to create boundary of target class. We usually choose one class classifier for the imbalance data classification task. The minority class should be defined to target class. Using this SVDD, known positive interactions will be used to training model and it's a small-scale sample under large-scale DTIs. So it has ability to obtained good results under considerable training time even we can collect more DTI samples.

### 6.3.3 Case Study

Table 15. The average AUC of different methods on the four type dataset

| Method<br>Data set | ODT | PUDT | DBSI | Bipartite Graph Learning | KBMF2K | NetCBP |
|---|---|---|---|---|---|---|
| Nuclear receptors | 0.766± 0.024 | 0.885 | 0.758 | 0.692 | 0.824 | 0.839 |
| GPCRs | **0.885** ± 0.005 | 0.878 | 0.802 | 0.811 | 0.857 | 0.823 |
| Ion channels | **0.847**± 0.004 | 0.831 | 0.803 | 0.692 | 0.799 | 0.803 |
| Enzymes | 0.850± 0.002 | 0.884 | 0.808 | 0.821 | 0.832 | 0.825 |

Table 16. The newly confirmed drug-target interactions by public database with high scores

| Drug ID | Protein ID | Evidence |
|---|---|---|
| Levodopa | hsa:1814 | KEGG |
| Levodopa | hsa:1816 | KEGG |
| Duratool | hsa:3355 | Drugbank |
| Misoprostol | hsa:5031 | KEGG |
| Isoetharine | hsa:154 | KEGG |
| Metoprolol | hsa:1814 | Drugbank |
| Clozapine | hsa:154 | Drugbank |
| Dipivefrin | hsa:2099 | Drugbank |

After evaluating the effectiveness of the proposed model by using the 5-fold cross validation method, we here calculate the interaction possibility for all potential drug-target pairs in the datasets of GPCRs and Nuclear Receptors. The predicted drug-target pairs with top ranks in the drug's potential target lists are considered as highly potential drug-target interactions and further verified by four public databases (i.e. KEGG [145], Drugbank [21]). These databases have been supplemented by some newly detected drug-target interactions since the gold standard data explored in this study was collected in 2008. All the predicted possibilities for all potential drug-target interactions in GPCRs and Nuclear Receptors can be obtained in Table 16. As a result, 8 new drug-target interactions are finally confirmed. Note that the high-ranked interactions that are not reported yet may also exist in reality. Based on these results, we anticipate that the proposed model is feasible to predict new drug-target interactions.

### 6.4 Summary

In this chapter, we proposed a novel method to predict potential drug-target interaction based on their chemical structures and their target protein sequences. We adopted two representation methods for multiple responses to deal with two object data sources. And then, we introduced SVDD to solve the large high-dimensional data sets and unreliable negative samples. The general task of SVDD can be regarded as drawing a decision boundary for accepting most of the target and rejecting most of the outliers at the same time. As the known positive samples make up only a small portion of large-scale candidate interactions, the smart method is training target classification boundary obtained from calculable positive samples excluded from confused samples. To our knowledge, no previous feature-based work relies on all reliable positive samples in the training of drug-target interaction prediction. The originality of the proposed method lies in the get high-dimensional chemical features and biological features together in a unified input. At early stages, our approach could help screen the candidate molecules for the further development process. ODT can be used for link prediction in large networks. The generalization method is to filter features with mutual information first, and then to build the hypersphere based on one-class representation. The experimental result shows that ODT can handle the one-class drug-target interaction data effectively and provide good performance of drug-target interaction prediction. The proposed method depends highly on the scale of known positive samples, high-dimensional biological and chemical profiles also influence the model performance.

# 7. CONCLUSION AND FUTURE WORK

To discover representation in the networks, several algorithms have been proposed. Though different computational methodologies might be utilized, these algorithms can be categorized according to the specific properties that are considered, the techniques that are used, and the particular field into which they are applied. In this section, the state-of-the-art related to discovering clusters in graphs are introduced categorically.

## 7.1 Conclusion

In this thesis, we mainly address the challenges in learning appropriate and explainable network representations for pattern discovery. To learn representations that may consider different characteristics of the network data, we have proposed five different algorithms.

First, we propose MFDR, which is an effective model for learning multi-scale representations from the network. MFDR is the first attempt to utilize stacked autoencoder to represent large-scale bimodal features, i.e., network topology and node attributes. The network representations learned by MFDR can effectively perform the task of link prediction in biological network data. Second, we propose DMNF, which is an effective model for learning meaningful representations from networks carrying heterogeneous information. DMNF is able to construct a new network by fusing the heterogeneous information carried by the network and learn representations via a deep fusion method. The learned representations can effectively uncover the clusters hidden in the network. Third, we further propose a model, named as DFNet, which can learn representations from the fused network via matrix completion. Fourth, we propose GraphSE to learn interpretable network representations to construct a set of significant sub-networks which can be used for predicting the existence of links in the drug-side-effect network data. At last, we propose an effective clustering model called LFCIS. Different from previous ones, the low dimensional network representations learned by LFCIS can be used to summarize the group-wise node features that are hidden in the network data. Besides, we also attempt to address the problem of sample imbalance in link prediction by proposing ODT. ODT tackles the problem of unbalanced samples with support vector data descriptor. It can identify those one-class nodes in a densely connected hypersphere, as well as the attributes of nodes are selected by mutual information.

The proposed models have been used to analyze a wide range of real-world network data related to society and biology and have been compared to a number of prevalent approaches. The proposed methods are able to outperform most compared baselines in most real datasets. This indicates that the network representations learned by the proposed models may well capture the information carried by the network and they are effective in different tasks of network analytics

## 7.2 Future work

In the future, we attempt to improve representation learning in the network data from the following aspects. First, we will attempt to propose effective fuzzy-based models for learning representations from network data. Compared with traditional methods for learning representations in the network, fuzzy representation learning may reveal the significance of learned latent features via the degree of membership. As a result, those latent features that are used to dominantly describe the vertices can be easily utilized by a classifier and the accuracy of the discovered patterns in the network is expected to be higher. Second, we will attempt to investigate how the mutual effect between different domains of networks may affect the performance of a representation learner in multiple network data. Robust network representations are expected to be learned if such mutual effect can be quantified and considered in the learning process. Third, we will try TO adopt new deep learning frameworks, such as end-to-end techniques. In addition, deep SVDD will be an important goal for next study.

## REFERENCES

[1] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35, no. 8 (2013): 1798-1828.

[2] Mahadevan, Sridhar. "Learning representation and control in Markov decision processes: New frontiers." Foundations and Trends® in Machine Learning 1, no. 4 (2009): 403-565.

[3] Shen, Juwen, Jian Zhang, Xiaomin Luo, Weiliang Zhu, Kunqian Yu, Kaixian Chen, Yixue Li, and Hualiang Jiang. "Predicting protein–protein interactions based only on sequences information." Proceedings of the National Academy of Sciences 104, no. 11 (2007): 4337-4341.

[4] Xia, Rongkai, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. "Supervised hashing for image retrieval via image representation learning." In AAAI, vol. 1, p. 2. 2014.

[5] Choi, Jae Young, Dae Hoe Kim, Konstantinos N. Plataniotis, and Yong Man Ro. "Classifier ensemble generation and selection with multiple feature representations for classification applications in computer-aided detection and diagnosis on mammography." Expert Systems with Applications 46 (2016): 106-121.

[6] Yu, Hong-Jie, and De-Shuang Huang. "Graphical representation for DNA sequences via joint diagonalization of matrix pencil." IEEE Journal of Biomedical and Health Informatics 17, no. 3 (2013): 503-511.

[7] Yu, Hong-Jie, and De-Shuang Huang. "Normalized feature vectors: a novel alignment-free sequence comparison method based on the numbers of adjacent amino acids." IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 10, no. 2 (2013): 457-467.

[8] Au, Wai-Ho, Keith CC Chan, Andrew KC Wong, and Yang Wang. "Attribute clustering for grouping, selection, and classification of gene expression data." IEEE/ACM transactions on computational biology and bioinformatics 2, no. 2 (2005): 83-101.

[9] Wong, Andrew KC, Wai-Ho Au, and Keith CC Chan. "Discovering high-order patterns of gene expression levels." Journal of Computational Biology 15, no. 6 (2008): 625-637.

[10] Jacob, Yann, Ludovic Denoyer, and Patrick Gallinari. "Learning latent representations of nodes for classifying in heterogeneous social networks." In Proceedings of the 7th ACM international conference on Web search and data mining, pp. 373-382. ACM, 2014.

[11] Gui, Lin, Yu Zhou, Ruifeng Xu, Yulan He, and Qin Lu. "Learning representations from heterogeneous network for sentiment classification of product reviews." Knowledge-Based Systems 124 (2017): 34-45.

[12] Luo, Yunan, Xinbin Zhao, Jingtian Zhou, Jinglin Yang, Yanqing Zhang, Wenhua Kuang, Jian Peng, Ligong Chen, and Jianyang Zeng. "A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information." Nature communications 8, no. 1 (2017): 573.

[13] Zhang, Wen, Yanlin Chen, Feng Liu, Fei Luo, Gang Tian, and Xiaohong Li. "Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data." BMC bioinformatics 18, no. 1 (2017): 18.

[14] Wei, Hua-Liang, and Stephen A. Billings. "Feature subset selection and ranking for data dimensionality reduction." IEEE transactions on pattern analysis and machine intelligence 29, no. 1 (2007).

[15] Maji, Pradipta. "A rough hypercuboid approach for feature selection in approximation spaces." IEEE Transactions on Knowledge and Data Engineering 26, no. 1 (2014): 16-29.

[16] Saha, Barna, and Divesh Srivastava. "Data quality: The other face of big data." In Data Engineering (ICDE), 2014 IEEE 30th International Conference on, pp. 1294-1297. IEEE, 2014.

[17] You, Zhu-Hong, Ying-Ke Lei, Jie Gui, De-Shuang Huang, and Xiaobo Zhou. "Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data." Bioinformatics 26, no. 21 (2010): 2744-2751.

[18] Zhou, Fengfeng, Victor Olman, and Ying Xu. "Large-scale analyses of glycosylation in cellulases." Genomics, proteomics & bioinformatics 7, no. 4 (2009): 194-199.

[19] Luo, Xin, Zhuhong You, Mengchu Zhou, Shuai Li, Hareton Leung, Yunni Xia, and Qingsheng Zhu. "A highly efficient approach to protein interactome mapping based on collaborative filtering framework." Scientific reports 5 (2015): 7702.

[20] You, Zhu-Hong, Zheng Yin, Kyungsook Han, De-Shuang Huang, and Xiaobo Zhou. "A semi-supervised learning approach to predict synthetic genetic interactions by combining functional and topological properties of functional gene network." Bmc Bioinformatics 11, no. 1 (2010): 343.

[21] Law, Vivian, Craig Knox, Yannick Djoumbou, Tim Jewison, An Chi Guo, Yifeng Liu, Adam Maciejewski et al. "DrugBank 4.0: shedding new light on drug metabolism." Nucleic acids research 42, no. D1 (2013): D1091-D1097.

[22] Shoemaker, Benjamin A., and Anna R. Panchenko. "Deciphering protein–protein interactions. Part II. Computational methods to predict protein and domain interaction partners." PLoS computational biology 3, no. 4 (2007): e43.

[23] Zhao, Xing‐Ming, Xin Li, Luonan Chen, and Kazuyuki Aihara. "Protein classification with imbalanced data." Proteins: Structure, function, and bioinformatics 70, no. 4 (2008): 1125-1132.

[24] Lam, Winnie WM, and Keith CC Chan. "Discovering functional interdependence relationship in PPI networks for protein complex identification." IEEE transactions on biomedical engineering 59, no. 4 (2012): 899-908.

[25] You, Zhu-Hong, Ying-Ke Lei, Lin Zhu, Junfeng Xia, and Bing Wang. "Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis." In BMC bioinformatics, vol. 14, no. 8, p. S10. BioMed Central, 2013.

[26] Zhang, Qiangfeng Cliff, Donald Petrey, Raquel Norel, and Barry H. Honig. "Protein interface conservation across structure space." Proceedings of the National Academy of Sciences 107, no. 24 (2010): 10896-10901.

[27] Zhang, Qiangfeng Cliff, Donald Petrey, Lei Deng, Li Qiang, Yu Shi, Chan Aye Thu, Brygida Bisikirska et al. "Structure-based prediction of protein–protein interactions on a genome-wide scale." Nature 490, no. 7421 (2012): 556.

[28] Lei, Ying-Ke, Zhu-Hong You, Zhen Ji, Lin Zhu, and De-Shuang Huang. "Assessing and predicting protein interactions by combining manifold embedding with multiple information integration." In BMC bioinformatics, vol. 13, no. 7, p. S3. BioMed Central, 2012.

[29] Xia, Jun-Feng, Xing-Ming Zhao, and De-Shuang Huang. "Predicting protein–protein interactions from protein sequences using meta predictor." Amino Acids 39, no. 5 (2010): 1595-1599.

[30] Zhao, Xing‐Ming, Luonan Chen, and Kazuyuki Aihara. "A discriminative approach for identifying domain–domain interactions from protein–protein interactions." Proteins: Structure, Function, and Bioinformatics 78, no. 5 (2010): 1243-1253.

[31] Zhao, Xing-Ming, Yiu-Ming Cheung, and De-Shuang Huang. "A novel approach to extracting features from motif content and protein composition for protein sequence classification." Neural Networks 18, no. 8 (2005): 1019-1028.

[32] Qi, Yanjun, Judith Klein-Seetharaman, and Ziv Bar-Joseph. "Random forest similarity for protein-protein interaction prediction from multiple sources." In Biocomputing 2005, pp. 531-542. 2005.

[33] Pandini, Alessandro, Arianna Fornili, Franca Fraternali, and Jens Kleinjung. "Detection of allosteric signal transmission by information-theoretic analysis of protein dynamics." The FASEB journal 26, no. 2 (2012): 868-881.

[34] Autore, Flavia, Mark Pfuhl, Xueping Quan, Aisling Williams, Roland G. Roberts, Catherine M. Shanahan, and Franca Fraternali. "Large-scale modelling of the divergent spectrin repeats in nesprins: giant modular proteins." PloS one 8, no. 5 (2013): e63633.

[35] Xia, Jun-Feng, Kyungsook Han, and De-Shuang Huang. "Sequence-based prediction of protein-protein interactions by means of rotation forest and autocorrelation

descriptor." Protein and Peptide Letters 17, no. 1 (2010): 137-145.

[36] Günther, Stefan, Michael Kuhn, Mathias Dunkel, Monica Campillos, Christian Senger, Evangelia Petsalaki, Jessica Ahmed et al. "SuperTarget and Matador: resources for exploring drug-target relationships." Nucleic acids research 36, no. suppl_1 (2007): D919-D922.

[37] Chen, Xin, Zhi Liang Ji, and Yu Zong Chen. "TTD: therapeutic target database." Nucleic acids research 30, no. 1 (2002): 412-415.

[38] Zhang, Ya-Nan, Xiao-Yong Pan, Yan Huang, and Hong-Bin Shen. "Adaptive compressive learning for prediction of protein–protein interactions from primary sequence." Journal of theoretical biology 283, no. 1 (2011): 44-52.

[39] Pan, Xiao-Yong, Ya-Nan Zhang, and Hong-Bin Shen. "Large-Scale prediction of human protein− protein interactions from amino acid sequence based on latent topic features." Journal of Proteome Research 9, no. 10 (2010): 4992-5001.

[40] Davis, Allan Peter, Cynthia Grondin Murphy, Robin Johnson, Jean M. Lay, Kelley Lennon-Hopkins, Cynthia Saraceni-Richards, Daniela Sciaky et al. "The comparative toxicogenomics database: update 2013." Nucleic acids research 41, no. D1 (2012): D1104-D1114.

[41] Pitre, Sylvain, Mohsen Hooshyar, Andrew Schoenrock, Bahram Samanfar, Matthew Jessulat, James R. Green, Frank Dehne, and Ashkan Golshani. "Short co-occurring polypeptide regions can predict global protein interaction maps." Scientific reports 2 (2012): 239.

[42] Wang, Hua, Heng Huang, Chris Ding, and Feiping Nie. "Predicting protein–protein interactions from multimodal biological data sources via nonnegative matrix factorization." Journal of Computational Biology 20, no. 4 (2013): 344-358.

[43] Mei, Suyu, and Hao Zhu. "AdaBoost based multi-instance transfer learning for predicting proteome-wide interactions between Salmonella and human proteins." PloS one 9, no. 10 (2014): e110488.

[44] Guo, Yanzhi, Lezheng Yu, Zhining Wen, and Menglong Li. "Using support vector machine combined with auto covariance to predict protein–protein interactions

from protein sequences." Nucleic acids research 36, no. 9 (2008): 3025-3030.

[45] Chou, Kuo-Chen, and Yu-Dong Cai. "Predicting protein−protein interactions from sequences in a hybridization space." Journal of Proteome Research 5, no. 2 (2006): 316-322.

[46] Salwinski, Lukasz, Christopher S. Miller, Adam J. Smith, Frank K. Pettit, James U. Bowie, and David Eisenberg. "The database of interacting proteins: 2004 update." Nucleic acids research 32, no. suppl_1 (2004): D449-D451.

[47] Ben-Hur, Asa, and William Stafford Noble. "Choosing negative examples for the prediction of protein-protein interactions." In BMC bioinformatics, vol. 7, no. 1, p. S2. BioMed Central, 2006.

[48] Smialowski, Pawel, Philipp Pagel, Philip Wong, Barbara Brauner, Irmtraud Dunger, Gisela Fobo, Goar Frishman et al. "The Negatome database: a reference set of non-interacting protein pairs." Nucleic acids research 38, no. suppl_1 (2009): D540-D544.

[49] Veres, Daniel V., Dávid M. Gyurkó, Benedek Thaler, Kristóf Z. Szalay, Dávid Fazekas, Tamás Korcsmáros, and Peter Csermely. "ComPPI: a cellular compartment-specific database for protein–protein interaction network analysis." Nucleic acids research 43, no. D1 (2014): D485-D493.

[50] Martin, Shawn, Diana Roe, and Jean-Loup Faulon. "Predicting protein–protein interactions using signature products." Bioinformatics 21, no. 2 (2004): 218-226.

[51] Kuhn, Michael, Monica Campillos, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. "A side effect resource to capture phenotypic effects of drugs." Molecular systems biology 6, no. 1 (2010): 343.

[52] Zhou, Yu Zhen, Yun Gao, and Ying Ying Zheng. "Prediction of protein-protein interactions using local description of amino acid sequence." In Advances in Computer Science and Education Applications, pp. 254-262. Springer, Berlin, Heidelberg, 2011.

[53] Yang, Lei, Jun-Feng Xia, and Jie Gui. "Prediction of protein-protein interactions from protein sequence using local descriptors." Protein and Peptide Letters 17, no. 9 (2010): 1085-1090.

[54] Ding, Hao, Ichigaku Takigawa, Hiroshi Mamitsuka, and Shanfeng Zhu. "Similarity-based machine learning methods for predicting drug–target interactions: a brief review." Briefings in bioinformatics 15, no. 5 (2013): 734-747.

[55] Lan, Wei, Jianxin Wang, Min Li, Fang-Xiang Wu, and Yi Pan. "Predicting drug-target interaction based on sequence and structure information." IFAC-PapersOnLine 48, no. 28 (2015): 12-16..

[56] Bock, Joel R., and David A. Gough. "Whole-proteome interaction mining." Bioinformatics 19, no. 1 (2003): 125-134.

[57] Nanni, Loris. "Hyperplanes for predicting protein–protein interactions." Neurocomputing 69, no. 1-3 (2005): 257-263.

[58] Nanni, Loris, and Alessandra Lumini. "An ensemble of K-local hyperplanes for predicting protein–protein interactions." Bioinformatics 22, no. 10 (2006): 1207-1210.

[59] Nagamine N, Shirakawa T, Minato Y, et al. “Integrating statistical predictions and experimental verifications for enhancing protein-chemical interaction predictions in virtual screening, ” PLoS Comput Biol2009;5(6):e1000397

[60] Parsons, Ainslie B., et al. "Integration of chemical-genetic and genetic interaction data links bioactive compounds to cellular target pathways," Nature biotechnology 22.1 (2004): 62-69.

[61] Yamanishi, Yoshihiro, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces." Bioinformatics 24, no. 13 (2008): i232-i240.

[62] Xia, Zheng, Ling-Yun Wu, Xiaobo Zhou, and Stephen TC Wong. "Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces." In BMC systems biology, vol. 4, no. 2, p. S6. BioMed Central, 2010.

[63] Van Laarhoven, Twan, Sander B. Nabuurs, and Elena Marchiori. "Gaussian interaction profile kernels for predicting drug–target interaction." Bioinformatics 27, no. 21 (2011): 3036-3043.

[64] You, Zhu-Hong, Jian-Zhong Yu, Lin Zhu, Shuai Li, and Zhen-Kun Wen. "A MapReduce based parallel SVM for large-scale predicting protein–protein interactions."

Neurocomputing 145 (2014): 37-43.

[65] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313, no. 5786 (2006): 504-507.

[66] Wang, Lei, Zhu-Hong You, Xing Chen, Xin Yan, Gang Liu, and Wei Zhang. "Rfdt: A rotation forest-based predictor for predicting drug-target interactions using drug structure and protein sequence information." Current Protein and Peptide Science 19, no. 5 (2018): 445-454.

[67] Spencer, Matt, Jesse Eickholt, and Jianlin Cheng. "A Deep Learning Network Approach to ab initio Protein Secondary Structure Prediction," Computational Biology and Bioinformatics, IEEE/ACM Transactions on 12.1 (2015): 103-112.

[68] Nguyen, Son P., Yi Shang, and Dong Xu. "DL-PRO: A novel deep learning method for protein model quality assessment," In Neural Networks (IJCNN), 2014 International Joint Conference on, pp. 2071-2078. IEEE, 2014.

[69] Hattori, Masahiro, Yasushi Okuno, Susumu Goto, and Minoru Kanehisa. "Heuristics for chemical compound matching." Genome Informatics 14 (2003): 144-153..

[70] Smith, Temple F., and Michael S. Waterman. "Identification of common molecular subsequences," Journal of molecular biology 147, no. 1 (1981): 195-197.

[71] Yu, Hua, Jianxin Chen, Xue Xu, Yan Li, Huihui Zhao, Yupeng Fang, Xiuxiu Li, Wei Zhou, Wei Wang, and Yonghua Wang. "A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data." PloS one 7, no. 5 (2012): e37608.

[72] Gönen, Mehmet. "Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization," Bioinformatics 28, no. 18 (2012): 2304-2310.

[73] Chen, Hailin, and Zuping Zhang. "A semi-supervised method for drug-target interaction prediction with consistency in networks," PLoS One. (2013): e62975.

[74] Cao, Dong-Sheng, et al. "Large-scale prediction of drug–target interactions using protein sequences and drug topological structures," Analytica chimica acta 752 (2012): 1-10.

[75] Mousavian, Zaynab, et al. "Drug–target interaction prediction from PSSM based evolutionary information." Journal of pharmacological and toxicological methods 78 (2016): 42-51.

[76] Harvard Yamanishi, Yoshihiro, Masaaki Kotera, Minoru Kanehisa, and Susumu Goto. "Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework," Bioinformatics 26, no. 12 (2010): i246-i254.

[77] Luo, Weimin, and Keith CC Chan. "Discovering patterns in drug-protein interactions based on their fingerprints." In BMC bioinformatics, vol. 13, no. 9, p. S4. BioMed Central, 2012.

[78] Liu, Bin, Shanyi Wang, and Xiaolong Wang. "DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation." Scientific reports 5 (2015): 15479.

[79] Liu, Bin, Deyuan Zhang, Ruifeng Xu, Jinghao Xu, Xiaolong Wang, Qingcai Chen, Qiwen Dong, and Kuo-Chen Chou. "Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection," Bioinformatics 30, no. 4 (2014): 472-479.

[80] Zeng, Jiancang, Dapeng Li, Yunfeng Wu, Quan Zou, and Xiangrong Liu. "An empirical study of features fusion techniques for protein-protein interaction prediction." Current Bioinformatics 11, no. 1 (2016): 4-12.

[81] You, Zhu-Hong, Keith CC Chan, and Pengwei Hu. "Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest." PLoS One 10, no. 5 (2015): e0125811.

[82] Kauvar, Lawrence M., Deborah L. Higgins, Hugo O. Villar, J. Richard Sportsman, Åsa Engqvist-Goldstein, Robert Bukar, Karin E. Bauer, Hara Dilley, and David M. Rocke. "Predicting ligand binding to proteins by affinity fingerprinting." Chemistry & biology 2, no. 2 (1995): 107-118.

[83] Pengwei Hu, Chan, Keith CC, and Zhu-Hong You. "Large-scale prediction of drug-target interactions from deep representations." In Neural Networks (IJCNN), 2016

International Joint Conference on, pp. 1236-1243. IEEE, 2016.

[84] Bengio, Yoshua, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. "Greedy layer-wise training of deep networks." In Advances in neural information processing systems, pp. 153-160. 2007.

[85] Ng, Andrew. "Sparse autoencoder." CS294A Lecture notes 72 (2011).

[86] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks," Machine learning 20, no. 3 (1995): 273-297.

[87] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." ACM transactions on intelligent systems and technology (TIST) 2, no. 3 (2011): 27.

[88] Wang, Bo, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. "Similarity network fusion for aggregating data types on a genomic scale." Nature methods 11, no. 3 (2014): 333.

[89] Xia, Youshen, and Henry Leung. "Performance analysis of statistical optimal data fusion algorithms." Information Sciences 277 (2014): 808-824.

[90] Hu, Allen L., and Keith CC Chan. "Utilizing both topological and attribute information for protein complex identification in ppi networks." IEEE/ACM transactions on computational biology and bioinformatics 10, no. 3 (2013): 780-792.

[91] He, Tiantian, and Keith CC Chan. "Evolutionary graph clustering for protein complex identification." IEEE/ACM transactions on computational biology and bioinformatics (2016).

[92] Girvan, Michelle, and Mark EJ Newman. "Community structure in social and biological networks." Proceedings of the national academy of sciences 99, no. 12 (2002): 7821-7826.

[93] Wang, Qi, and Junyuan Xie. "A Two-Dimensional Genetic Algorithm for Identifying Overlapping Communities in Dynamic Networks." In Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on, pp. 565-569. IEEE, 2016.

[94] Von Luxburg, Ulrike. "A tutorial on spectral clustering." Statistics and computing 17, no. 4 (2007): 395-416.

[95] Frey, Brendan J., and Delbert Dueck. "Clustering by passing messages between data points." science 315, no. 5814 (2007): 972-976..

[96] Yang, Jaewon, Julian McAuley, and Jure Leskovec. "Community detection in networks with node attributes." In Data Mining (ICDM), 2013 IEEE 13th international conference on, pp. 1151-1156. IEEE, 2013.

[97] J. Chan and D. Blei, “Relational topic models for document networks,” in Proc. 12th Int. Conf. Artif. Intell. Stat., 2009, pp. 81–88.

[98] He, Tiantian, and Keith CC Chan. "Evolutionary community detection in social networks." In Evolutionary Computation (CEC), 2014 IEEE Congress on, pp. 1496-1503. IEEE, 2014.

[99] He, Tiantian, and Keith CC Chan. "MISAGA: An Algorithm for Mining Interesting Subgraphs in Attributed Graphs." IEEE transactions on cybernetics 48, no. 5 (2018): 1369-1382.

[100] Li, Zhenhua, and Henry Leung. "Fusion of multispectral and panchromatic images using a restoration-based method." IEEE transactions on geoscience and remote sensing 47, no. 5 (2009): 1482-1491.

[101] Zhu, Hao, Henry Leung, and Ka-Veng Yuen. "A joint data association, registration, and fusion approach for distributed tracking." Information Sciences 324 (2015): 186-196.

[102] Acar, Evrim, Rasmus Bro, and Age K. Smilde. "Data fusion in metabolomics using coupled matrix and tensor factorizations." Proceedings of the IEEE 103, no. 9 (2015): 1602-1620.

[103] Zhang, Yutao, Jie Tang, Zhilin Yang, Jian Pei, and Philip S. Yu. "COSNET: Connecting heterogeneous social networks with local and global consistency." In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1485-1494. ACM, 2015.

[104] Zhang, Lan, Henry Leung, and Keith CC Chan. " Information fusion based

smart home control system and its application." IEEE Transactions on Consumer Electronics 54, no. 3 (2008).

[105] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. "Sonar tracking of multiple targets using joint probabilistic data association. " IEEE J. Oceanic Eng., 8(3):173–184, 1983.

[106] Reid, Donald. "An algorithm for tracking multiple targets." IEEE transactions on Automatic Control 24, no. 6 (1979): 843-854.

[107] Huang, Dongliang, and Henry Leung. "Maximum likelihood state estimation of semi-Markovian switching system in non-Gaussian measurement noise." IEEE Transactions on Aerospace and Electronic Systems 46, no. 1 (2010).

[108] Deng, Zili, Peng Zhang, Wenjuan Qi, Jinfang Liu, and Yuan Gao. "Sequential covariance intersection fusion Kalman filter." Information Sciences 189 (2012): 293-309.

[109] Zhu, Hao, Henry Leung, and Zhongshi He. "A variational Bayesian approach to robust sensor fusion based on Student-t distribution." Information Sciences 221 (2013): 201-214.

[110] Ngiam, Jiquan, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. "Multimodal deep learning." In Proceedings of the 28th international conference on machine learning (ICML-11), pp. 689-696. 2011.

[111] Chan, Keith CC, Andrew KC Wong, and David KY Chiu. "Learning sequential patterns for probabilistic inductive prediction." IEEE transactions on systems, man, and cybernetics 24, no. 10 (1994): 1532-1547.

[112] Tian, Fei, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. "Learning Deep Representations for Graph Clustering." In AAAI, pp. 1293-1299. 2014.

[113] Luo, Xin, Zhong Ming, Zhuhong You, Shuai Li, Yunni Xia, and Hareton Leung. "Improving network topology-based protein interactome mapping via collaborative filtering." Knowledge-Based Systems 90 (2015): 23-32.

[114] Traud, Amanda L., Eric D. Kelsic, Peter J. Mucha, and Mason A. Porter. "Comparing community structure to characteristics in online collegiate social networks." SIAM review 53, no. 3 (2011): 526-543.

[115] Leskovec, Jure, and Julian J. Mcauley. "Learning to discover social circles in ego networks." In Advances in neural information processing systems, pp. 539-547. 2012.

[116] Chen, Xing, Chenggang Clarence Yan, Xiaotian Zhang, Xu Zhang, Feng Dai, Jian Yin, and Yongdong Zhang. "Drug–target interaction prediction: databases, web servers and computational models." Briefings in bioinformatics 17, no. 4 (2015): 696-712.

[117] Yamanishi, Yoshihiro, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces." Bioinformatics 24, no. 13 (2008): i232-i240.

[118] Chen, Xing, Ming-Xi Liu, and Gui-Ying Yan. "Drug–target interaction prediction by random walk on the heterogeneous network." Molecular BioSystems 8, no. 7 (2012): 1970-1978.

[119] Cheng, Feixiong, Chuang Liu, Jing Jiang, Weiqiang Lu, Weihua Li, Guixia Liu, Weixing Zhou, Jin Huang, and Yun Tang. "Prediction of drug-target interactions and drug repositioning via network-based inference." PLoS computational biology 8, no. 5 (2012): e1002503.

[120] Takarabe, Masataka, Masaaki Kotera, Yosuke Nishimura, Susumu Goto, and Yoshihiro Yamanishi. "Drug target prediction using adverse event report systems: a pharmacogenomic approach." Bioinformatics 28, no. 18 (2012): i611-i618.

[121] Luo, Yunan, Xinbin Zhao, Jingtian Zhou, Jinglin Yang, Yanqing Zhang, Wenhua Kuang, Jian Peng, Ligong Chen, and Jianyang Zeng. "A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information." Nature communications 8, no. 1 (2017): 573.

[122] Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." Nature 401, no. 6755 (1999): 788.

[123] Lee, Daniel D., and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." In Advances in neural information processing systems, pp. 556-562. 2001.

[124] Jain, Prateek, and Inderjit S. Dhillon. "Provable inductive matrix completion."

arXiv preprint arXiv:1306.0626 (2013).

[125] Natarajan, Nagarajan, and Inderjit S. Dhillon. "Inductive matrix completion for predicting gene–disease associations." Bioinformatics 30, no. 12 (2014): i60-i68.

[126] Keshava Prasad, T. S., Renu Goel, Kumaran Kandasamy, Shivakumar Keerthikumar, Sameer Kumar, Suresh Mathivanan, Deepthi Telikicherla et al. "Human protein reference database—2009 update." Nucleic acids research 37, no. suppl_1 (2008): D767-D772.

[127] Huang, Jin, and Charles X. Ling. "Using AUC and accuracy in evaluating learning algorithms." IEEE Transactions on knowledge and Data Engineering 17, no. 3 (2005): 299-310.

[128] Edwards, I. Ralph, and Jeffrey K. Aronson. "Adverse drug reactions: definitions, diagnosis, and management." The lancet 356, no. 9237 (2000): 1255-1259.

[129] Murphy, Robert F. "An active role for machine learning in drug development." Nature chemical biology 7, no. 6 (2011): 327.

[130] Page, David, Vítor Santos Costa, Sriraam Natarajan, Aubrey Barnard, Peggy L. Peissig, and Michael Caldwell. "Identifying Adverse Drug Events by Relational Learning." In AAAI. 2012.

[131] Liu, Mei, Yonghui Wu, Yukun Chen, Jingchun Sun, Zhongming Zhao, Xue-wen Chen, Michael Edwin Matheny, and Hua Xu. "Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs." Journal of the American Medical Informatics Association 19, no. e1 (2012): e28-e35.

[132] Jin, Bo, Haoyu Yang, Cao Xiao, Ping Zhang, Xiaopeng Wei, and Fei Wang. "Multitask dyadic prediction and its application in prediction of adverse drug-drug interaction." In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[133] Xiao, Cao, Ping Zhang, W. Art Chaowalitwongse, Jianying Hu, and Fei Wang. "Adverse drug reaction prediction with symbolic latent Dirichlet allocation." In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[134] Campillos, Monica, Michael Kuhn, Anne-Claude Gavin, Lars Juhl Jensen, and

Peer Bork. "Drug target identification using side-effect similarity." Science 321, no. 5886 (2008): 263-266.

[135] Pauwels, Edouard, Véronique Stoven, and Yoshihiro Yamanishi. "Predicting drug side-effect profiles: a chemical fragment-based approach." BMC bioinformatics 12, no. 1 (2011): 169.

[136] Atias, Nir, and Roded Sharan. "An algorithmic framework for predicting side effects of drugs." Journal of Computational Biology 18, no. 3 (2011): 207-218.

[137] Fliri, Anton F., William T. Loging, Peter F. Thadeio, and Robert A. Volkmann. "Analysis of drug-induced effect patterns to link structure and side effects of medicines." Nature chemical biology 1, no. 7 (2005): 389.

[138] Mizutani, Sayaka, Edouard Pauwels, Véronique Stoven, Susumu Goto, and Yoshihiro Yamanishi. "Relating drug–protein interaction network with drug side effects." Bioinformatics 28, no. 18 (2012): i522-i528.

[139] Peng, Yigang, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images." IEEE transactions on pattern analysis and machine intelligence 34, no. 11 (2012): 2233-2246.

[140] Gu, Shuhang, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. "Weighted nuclear norm minimization and its applications to low level vision." International Journal of Computer Vision 121, no. 2 (2017): 183-208.

[141] Ching, John Y., Andrew K. C. Wong, and Keith C. C. Chan. "Class-dependent discretization for inductive learning from continuous and mixed-mode data." IEEE Transactions on Pattern Analysis and Machine Intelligence 17, no. 7 (1995): 641-651.

[142] Hu, Lun, and Keith CC Chan. "Extracting Coevolutionary Features from Protein Sequences for Predicting Protein-Protein Interactions." IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 14, no. 1 (2017): 155-166.

[143] Haberman, Shelby J. "The analysis of residuals in cross-classified tables." Biometrics (1973): 205-220.

[144] Chen, Bin, David Wild, and Rajarshi Guha. "PubChem as a source of

polypharmacology." Journal of chemical information and modeling 49, no. 9 (2009): 2044-2055.

[145] Kanehisa, Minoru, Susumu Goto, Miho Furumichi, Mao Tanabe, and Mika Hirakawa. "KEGG for representation and analysis of molecular networks involving diseases and drugs." Nucleic acids research 38, no. suppl_1 (2009): D355-D360.

[146] Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation." IEEE Transactions on pattern analysis and machine intelligence 22, no. 8 (2000): 888-905.

[147] Sun, Yizhou, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, and Tianyi Wu. "Rankclus: integrating clustering with ranking for heterogeneous information network analysis." In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, pp. 565-576. ACM, 2009.

[148] Huang, Hong, Yuxiao Dong, Jie Tang, Hongxia Yang, Nitesh V. Chawla, and Xiaoming Fu. "Will Triadic Closure Strengthen Ties in Social Networks?." ACM Transactions on Knowledge Discovery from Data (TKDD) 12, no. 3 (2018): 30.

[149] Huang, Hong, Jie Tang, Lu Liu, JarDer Luo, and Xiaoming Fu. "Triadic closure pattern analysis and prediction in social networks." IEEE Transactions on Knowledge and Data Engineering 27, no. 12 (2015): 3374-3389.

[150] Huang, Hong, Jie Tang, Sen Wu, and Lu Liu. "Mining triadic closure patterns in social networks." In Proceedings of the 23rd international conference on World wide web, pp. 499-504. ACM, 2014.

[151] Fortunato, Santo. "Community detection in graphs." Physics reports 486, no. 3-5 (2010): 75-174.

[152] Leskovec, Jure, Kevin J. Lang, and Michael Mahoney. "Empirical comparison of algorithms for network community detection." In Proceedings of the 19th international conference on World wide web, pp. 631-640. ACM, 2010.

[153] Newman, Mark EJ. "Modularity and community structure in networks." Proceedings of the national academy of sciences 103, no. 23 (2006): 8577-8582.

[154] Clauset, Aaron, Mark EJ Newman, and Cristopher Moore. "Finding community

structure in very large networks." Physical review E 70, no. 6 (2004): 066111.

[155] Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks." Journal of statistical mechanics: theory and experiment 2008, no. 10 (2008): P10008.

[156] Wang, Qinna, and Eric Fleury. "Uncovering overlapping community structure." In Complex networks, pp. 176-186. Springer, Berlin, Heidelberg, 2011.

[157] Ahn, Yong-Yeol, James P. Bagrow, and Sune Lehmann. "Link communities reveal multiscale complexity in networks." nature 466, no. 7307 (2010): 761.

[158] Airoldi, Edoardo M., David M. Blei, Stephen E. Fienberg, and Eric P. Xing. "Mixed membership stochastic blockmodels." Journal of Machine Learning Research 9, no. Sep (2008): 1981-2014.

[159] Zhou, Yang, Hong Cheng, and Jeffrey Xu Yu. "Graph clustering based on structural/attribute similarities." Proceedings of the VLDB Endowment 2, no. 1 (2009): 718-729.

[160] Zhou, Yang, Hong Cheng, and Jeffrey Xu Yu. "Clustering large attributed graphs: An efficient incremental approach." In Data Mining (ICDM), 2010 IEEE 10th International Conference on, pp. 689-698. IEEE, 2010.

[161] Blei, David M. "Probabilistic topic models." Communications of the ACM 55, no. 4 (2012): 77-84.

[162] Sun, Yizhou, Jiawei Han, Jing Gao, and Yintao Yu. "itopicmodel: Information network-integrated topic modeling." In Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on, pp. 493-502. IEEE, 2009.

[163] MacKay, David JC, and David JC Mac Kay. Information theory, inference and learning algorithms. Cambridge university press, 2003.

[164] Streich, Andreas P., Mario Frank, David Basin, and Joachim M. Buhmann. "Multi-assignment clustering for Boolean data." In Proceedings of the 26th annual international conference on machine learning, pp. 969-976. ACM, 2009.

[165] Tian, Yuanyuan, Richard A. Hankins, and Jignesh M. Patel. "Efficient

aggregation for graph summarization." In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 567-580. ACM, 2008.

[166] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the royal statistical society. Series B (methodological) (1977): 1-38.

[167] Mcauley, Julian, and Jure Leskovec. "Discovering social circles in ego networks." ACM Transactions on Knowledge Discovery from Data (TKDD) 8, no. 1 (2014): 4.

[168] Qi, Guo-Jun, Charu C. Aggarwal, and Thomas Huang. "Community detection with edge content in social media networks." In Data Engineering (ICDE), 2012 IEEE 28th International Conference on, pp. 534-545. IEEE, 2012.

[169] Mullard, Asher. "New drugs cost US $2.6 billion to develop." (2014): 877.

[170] Scannell, Jack W., Alex Blanckley, Helen Boldon, and Brian Warrington. "Diagnosing the decline in pharmaceutical R&D efficiency." Nature reviews Drug discovery 11, no. 3 (2012): 191..

[171] Breuza, Lionel, Sylvain Poux, Anne Estreicher, Maria Livia Famiglietti, Michele Magrane, Michael Tognolli, Alan Bridge, Delphine Baratin, and Nicole Redaschi. "The UniProtKB guide to the human proteome." Database 2016 (2016).

[172] Günther, Stefan, Michael Kuhn, Mathias Dunkel, Monica Campillos, Christian Senger, Evangelia Petsalaki, Jessica Ahmed et al. "SuperTarget and Matador: resources for exploring drug-target relationships." Nucleic acids research 36, no. suppl_1 (2007): D919-D922.

[173] Chen, Xing, Biao Ren, Ming Chen, Quanxin Wang, Lixin Zhang, and Guiying Yan. "NLLSS: predicting synergistic drug combinations based on semi-supervised learning." PLoS computational biology 12, no. 7 (2016): e1004975.

[174] Chen, Xing, Chenggang Clarence Yan, Xu Zhang, and Zhu-Hong You. "Long non-coding RNAs and complex diseases: from experimental results to computational models." Briefings in bioinformatics 18, no. 4 (2016): 558-576.

[175] Huang, Yu-An, Xing Chen, Zhu-Hong You, De-Shuang Huang, and Keith CC

Chan. "ILNCSIM: improved lncRNA functional similarity calculation model." Oncotarget 7, no. 18 (2016): 25902.

[176] Tax, David MJ, and Robert PW Duin. "Support vector data description." Machine learning 54, no. 1 (2004): 45-66.

[177] Tax, D. M. J. "DDTools, the data description toolbox for MATLAB, version 2.1. 2." Delft University of Technology, Delft, Netherlands (2015).

[178] Vapnik, Vladimir. Statistical learning theory. 1998. Vol. 3. Wiley, New York, 1998.

[179] Petrik, Marek, Gavin Taylor, Ron Parr, and Shlomo Zilberstein. "Feature selection using regularization in approximate linear programs for Markov decision processes." arXiv preprint arXiv:1005.1860 (2010)..

[180] Liang, Jiye, Feng Wang, Chuangyin Dang, and Yuhua Qian. "A group incremental approach to feature selection applying rough set technique." IEEE Trans. Knowl. Data Eng. 26, no. 2 (2014): 294-308.

[181] Elad, Michael. "Sparse and redundant representation modeling—What next?." IEEE Signal Processing Letters 19, no. 12 (2012): 922-928.

[182] Taguchi, Y. H., Mitsuo Iwadate, and Hideaki Umeyama. "Principal component analysis-based unsupervised feature extraction applied to in silico drug discovery for posttraumatic stress disorder-mediated heart disease." BMC bioinformatics 16, no. 1 (2015): 139.

[183] Zhao, Zhendong, Gang Fu, Sheng Liu, Khaled M. Elokely, Robert J. Doerksen, Yixin Chen, and Dawn E. Wilkins. "Drug activity prediction using multiple-instance learning via joint instance and feature selection." In BMC bioinformatics, vol. 14, no. 14, p. S16. BioMed Central, 2013.

[184] Wen, Ming, Zhimin Zhang, Shaoyu Niu, Haozhi Sha, Ruihan Yang, Yonghuan Yun, and Hongmei Lu. "Deep-learning-based drug–target interaction prediction." Journal of proteome research 16, no. 4 (2017): 1401-1409.

[185] Kwon, Sunyoung, and Sungroh Yoon. "DeepCCI: End-to-end deep learning for chemical-chemical interaction prediction." In Proceedings of the 8th ACM International

Conference on Bioinformatics, Computational Biology, and Health Informatics, pp. 203-212. ACM, 2017.