

# Predicting drug-target interactions using multi-label learning with community detection method (DTI-MLCD)

Yanyi Chu<sup>1,2,3</sup>, Xiaoqi Shan<sup>4</sup>, Dennis R. Salahub<sup>3</sup>, Yi Xiong<sup>1,2,\*</sup>, and Dong-Qing Wei<sup>1,2,\*</sup>

<sup>1</sup> State Key Laboratory of Microbial Metabolism, and School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China; <sup>2</sup> Peng Cheng Laboratory, Vanke Cloud City Phase I Building 8, Xili Street, Nanshan District, Shenzhen, Guangdong, 518055, China; <sup>3</sup> University of Calgary

## Abstract

Identifying drug-target interactions (DTIs) is an important step for drug discovery and drug repositioning. To reduce heavily experiment cost, booming machine learning has been applied to this field and developed many computational methods, especially binary classification methods. However, there is still much room for improvement in the performance of current methods. Multi-label learning can reduce difficulties faced by binary classification learning with high predictive performance, and has not been explored extensively. The key challenge it faces is the exponential-sized output space, and considering label correlations can help it. Thus, we facilitate the multi-label classification by introducing community detection methods for DTIs prediction, named DTI-MLCD. On the other hand, we updated the gold standard data set proposed in 2008 and still in use today. The proposed DTI-MLCD is performed on the gold standard data set before and after the update, and shows the superiority than other classical machine learning methods and other benchmark proposed methods, which confirms the efficiency of it. The data and code for this study can be found at <https://github.com/a96123155/DTI-MLCD>.

**Keywords:** Drug-target interaction, dataset updating, multi-label learning, label correlation, community detection

## 1. Introduction

For drug development, drug discovery (i.e. finding potential new drugs) and drug repositioning (i.e. obtaining old drugs with new efficacy) are two important strategies with heavy cost [2], and an important step to achieve them is predicting DTIs. In recent years, many studies have applied the popular machine learning technology to realize intelligent medical treatment, which has accelerated the process of drug development to a certain extent. For DTIs prediction, the use of machine learning techniques can not only reduce the experimental scope of experimental research but also play a guiding role in experimental research.

There are many review articles [3-7] summarizing the progress of machine learning methods in the field of DTIs prediction in recent years, and the binary classification method is an important branch. For the binary classification method

[8-41], drug-target pairs and interactions are treated as samples and labels, respectively. It describes the drug-target pair by encoding drugs and targets as the feature vector, then, predicts DTIs by building a binary classifier. In addition to the binary classification methods, there are network inference methods [42-55], matrix factorization methods [56-63], kernel-based methods [64-68], restricted Boltzmann machine method [69], collaborative filtering method [70], clustering method [71], label propagation method [72], etc. It is worth noting that many of these other methods can be attributed to the binary classification method in a sense. For example, the network inference method regards the DTIs prediction problem as the bipartite network inference problem, and infers missing edges to achieve DTIs prediction. If the missing edges are regarded as negative samples and the existing edges are regarded as positive samples, it is converted into a binary classification problem.

For the binary classification method, it requires the participation of positive and negative samples, so unknown DTIs are often treated as negative samples. This negative sample construction strategy will not only introduce noise but also cause data imbalance as a large number of negative samples. Besides, it is also faced with excessive computational load and overfitting due to the redundant feature space and extremely high feature dimensions. For example, 10 drugs and 10 targets will be combined into  $10 \times 10 = 100$  samples, and the same drug or target in different samples has the same feature vector, that is, the feature vector of each drug or target will appear 10 times in the feature space of 100 samples. To reduce the above difficulties, the application of multi-label learning to DTI prediction problems is worth exploring. The multi-label classification problem trains a model that maps the input feature vector to more than one label. Transform the above binary classification example into a multi-label classification problem, described as: 10 drugs (or targets) as samples, and 10 targets (or drugs) as labels. The input feature is only a description of 10 drugs (or targets). Then use the multi-label learning algorithm to predict drug targets (or drugs that can interact with the specific target). Obviously, unlike the binary classification problem, it only requires information about the drug (or target) to predict a series of DTIs. The experiments in this study prove that its performance is very competitive with the binary classification problem, and its speed is much higher than that of the binary classification method, especially for large data sets. Until now, there are few applications and a lot of space for exploring multi-label learning applied in the DTIs prediction problem. DrugE-Rank [73] is a method using the “Learning To Rank” paradigm to model the DTIs prediction problem as a multi-label task. A study [74] uses multi-task deep neural networks for drug targets prediction, and firstly uses extended connectivity fingerprints with radius 12 as drug representation. Moreover, to overcome the training difficulties caused by too many labels in multi-label learning, Pliakos et al. [75] proposed three multi-label learning methods for DTI prediction, which use k-means for label division.

On the other hand, the gold standard data set currently used in the field of DTIs prediction is the data set collected by Yamanishi in 2008 [76], named Yamnishi\_08. After 12 years, a large number of new DTIs have been discovered, but they were not considered. As we all know, positive samples (i.e. DTIs) are essential for method

construction. The missing of positive samples not only introduces error in the modeling process, but also hides a great risk of false negatives during the model evaluation, making the unknown bias between research results and the actual results. For this point, Keum and Nam [11] updated these datasets among the original drugs and targets. However, in reality, it cannot be limited to the original drugs and targets, and the DTI between new drugs and targets should also be considered.

This study updates the gold standard data set of drugs, targets, and DTIs. In addition, we proposed the multi-label learning with community detection method for DTIs prediction (DTI-MLCD) and tested it on four original and updated gold standard data sets. The proposed DTI-MLCD first uses the community detection algorithm to divide the target space into multiple subspaces, then applies multi-label learning on each subspace, and finally performs DTIs prediction. Comparison with traditional machine learning methods and other benchmark DTIs prediction methods confirms the effectiveness of the proposed DTI-MLCD method.

## 2. Material and Methods

### 2.1. Problem description

This study divides the DTIs prediction problem into two sub-tasks: (a) drug discovery, which predicts new drugs, named  $T_D$ ; (b) drug repositioning, which predicts new targets, named  $T_T$ . These two tasks are regarded as multi-label classification problems, described below.

For task  $T_D$ , suppose  $X_D = \mathbb{R}^d$  and  $Y_T = \{y_1, y_2, \dots, y_p\}$  denote the  $d$ -dimensional drug instance space and the label space with  $p$  possible target class labels. This task is to learn a function  $f: X_D \rightarrow 2^{Y_T}$  from the multi-label training set  $D = \{(x_{D,i}, y_{T,i}) | 1 \leq i \leq m\}$ , where  $m$  is the number of samples. For each sample  $(x_{D,i}, y_{T,i})$ ,  $x_{D,i} \in X_D$ , it is a  $d$ -dimensional feature vector and  $y_{T,i} \in Y_T$  is the label set associated with  $x_{D,i}$ . For drug instances of the test set, the multi-label classifier  $f(\cdot)$  predicts the proper labels. The task  $T_T$  can be defined by analogy.

### 2.2. Data sets

Yamnishi\_08 derives from the KEGG BRITE [77], BRENDA [78], SuperTarget [79], and DrugBank [80] databases. It consists of four DTI datasets. These datasets are different according to the protein targets, namely nuclear receptor (NR), G-protein-coupled receptor (GPCR), ion channel (IC), and enzyme (E). To update these datasets, we collect new drugs, new targets, and new DTIs using KEGG BRITE, UniProt [81], and DrugBank databases in this study. The steps are divided into two parts: data integration and data cleaning. Data integration is achieved through web crawler technology. First, the DTI data corresponding to the 4 types of targets is obtained from the KEGG BRITE database and merged with Yamanishi\_08 to prevent the loss of information in the SuperTarget and BRENDA databases. Then, use the

UniProt database as the connection database of KEGG BRITE and DrugBank, search the DrugBank database for each target obtained in the previous step, and add drugs and corresponding DTIs that are not in KEGG BRITE and Yamanishi\_08. Next, search all known drugs one by one to maximize the DTI integrity of existing drugs and targets. After obtaining the integrated data, we deleted useless, invalid, and redundant data, including non-small molecule drugs (such as biotechnology drugs), mixed drugs, drugs with the same or unknown structure, and drugs with unknown end groups in the structure. It is worth noting that all drugs in the updated data set are approved drugs. The code for updating the dataset has been published on the GitHub page. Some statistics of the original gold standard and new updating four data sets are shown in Table 1.

**Table 1.** Statistics of the original and updating four data sets. The NR is short for the nuclear receptor, GPCR for the G-protein-coupled receptor, IC for the ion channel, and E for the enzyme. Besides, the n represents the amount, D represents degree, and the subscripts d and t represent drug and target, respectively.

Data sets		$n_d$	$n_t$	$n_{\text{interaction}}$	Density (%)	$D_d$	$D_t$	$D_d=1$ (%)	$D_t=1$ (%)
NR	Original	54	26	90	6.41	1.67	3.46	72.22	30.77
	Updated	541	33	886	4.96	1.64	26.85	65.99	18.18
GPCR	Original	223	95	635	3.00	2.85	6.68	47.53	35.79
	Updated	1680	156	5383	2.05	3.20	34.51	46.13	14.74
IC	Original	210	204	1476	3.45	7.03	7.24	38.57	11.27
	Updated	765	238	6385	3.51	8.35	26.83	21.70	8.82
E	Original	445	664	2926	0.99	6.58	4.41	39.78	43.37
	Updated	1777	1411	7371	0.29	4.15	5.22	45.24	37.99

## 2.3. Features

### 2.3.1. Drug representation

Many representations can be used to describe drugs, in general, these can be categorized into two types: molecular descriptors (MDs), molecular fingerprints (MFs). To explore the drug representation that is most suitable for this study, we used some open source tools commonly used in DTI prediction to generate MDs and MFs. For the MDs or MFs generated by different software, this study treats them as different drug representations. The tools used in this study are CDK [82], Pybel [83], RDKit [84], and PaDEL [85]. The MDs generated by the above tools are called MD\_CDK, MD\_PYB, MD\_RDK, and MD\_PAD. Their dimensions are 275, 24, 196, and 1875, respectively. Further, we combine these four types of MDs as a new type of MDs, called MD\_MER. Currently, MFs are always divided into three categories [86]: (a) topological path-based fingerprint. The representative FP2 [87] (MF\_FP2) used in this study; (b) topological circular fingerprint. ECFP4 [88] (MF\_EC4) and ECFP8 [88] (MF\_EC8) are used as their representativeness; (III) substructure key-based fingerprint. MACCS [89] (MF\_MAC) and PubChem fingerprint [90] (MF\_PCP) are used as their popularity in DTIs prediction. The dimension of them is 1024, 2048,

2048, 167, and 881, respectively. In addition to the MDs and MF, we also used the Word2vec-inspired feature [33] (W2V), which extracts word information from drug SMILES.

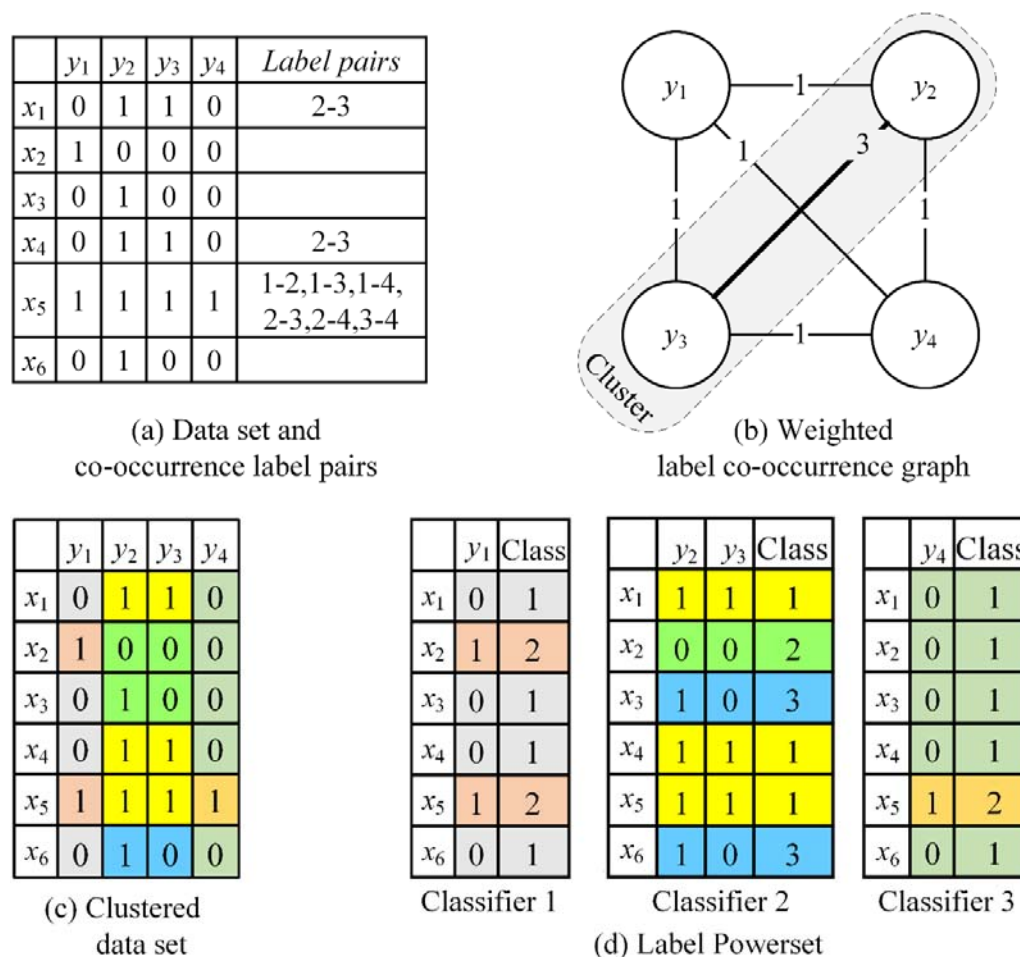
Further, we perform the feature combination of the above three types of features, because the complementarity between these three types of features may help enhance performance. In this process, we fuse the feature selection to obtain clean, highly complementary, and less redundant combined features.

### 2.3.2. Target representation

This study uses three types of target sequence-derived representations commonly used in DTIs prediction studies. The first is Composition, Transition, and Distribution (CTD), which is represented as the 504-dimensional feature vector obtained by PROFEAT web server [91]. The second is 1437 default protein descriptors generated by PROFEAT, named PRO. Besides CTD, it also includes amino acid composition, dipeptide composition, autocorrelation, quasi-sequence-order, amphiphilic pseudo-amino acid composition, and total amino acid properties. The third is the protein domain fingerprint (PDF), which is extracted from the PFAM v31.0 database [92]. For different data sets, we extracted different numbers of domains. The feature vector dimension of targets in NR, GPCR, IC, and E is 30, 61, 1404, and 2182, respectively. In addition, the feature combination is also performed.

## 2.4. Methods

The traditional supervised learning can be regarded as a degenerated version of multi-label learning as each sample is confined to have only one single label. However, the generality of multi-label learning makes harder to design the algorithm. The exponential-sized output space is the core issue of learning, i.e. there are  $2^m$  possible label sets for  $m$  labels. And exploiting label correlations or executing label space partition can help it. For this purpose, this study applies the community detection method from social networks to divide label space. Next, each divided label subspace corresponds to a multi-label learning sub-problem, and multiple Label Powerset (LP) multi-label classifiers are jointed to cover the entire label space. The base learner applied in LP is random forest (RF) because of its simplicity, parallelism, and superior capabilities, etc. In this section, we will introduce the typical algorithms of multi-label learning and community detection. The execution steps of the proposed DTI-MLCD method are shown in Figure 1.



**Figure 1.** The execution steps of the proposed DTI-MLCD method.

#### 2.4.1. Algorithms of multi-label learning

The multi-label learning algorithm development is the key challenge in multi-label learning researches, although the algorithm has boomed in a big explosion in the past 10 years. A simple categorization is adopted as follows.

The first category is the algorithm adaptation method, which works by fitting the existing algorithm to data and directly tackle the multi-label data. The representative algorithm is Multi-Label  $k$ -Nearest Neighbor (ML $k$ NN) [93]. ML $k$ NN is a lazy learning method based on the traditional  $k$ -Nearest Neighbor. It is now widely used in multi-label classification prediction tasks and has achieved satisfactory results [94, 95]. The second category is the problem transformation method, which works by fitting data to the well-established algorithm and transforming multi-label learning problems into the other learning technique. Binary Relevance (BR) [96], Classifier Chains (CC) [97] and Label Powerset (LP) [98] are representative algorithms in this category. BR tackles multi-label learning problem into multiple independent binary classification problems, where one binary classifier corresponds to one label. It is based on the assumption that labels are independent of each other, and each classifier of this method only recognizes the characteristics related to one label, but cannot

identify the characteristics related to all labels. Thus, it is not valid in many fields in reality, which is also the limitation of the BR. CC is proposed based on BR and considers label correlation. It converts the multi-label learning problem into a chain of binary classification problems. The main idea is to add the labels of all previous classifiers to the feature vector of the next training set and pass them to the next classifier. Obviously, the order of labels has a great influence on the prediction result. However, the order of the classifiers in the classifier chain is always random. Unlike BR and CC, LP transforms the multi-label learning task into the multi-class or single-label classification task. In other words, LP models the joint distribution of labels. It treats each label subset in the multi-label training set as a class of a multi-class task, and the prediction will be one of these subsets. Although LP is simple, it has two impractical points that tend to cause over-fitting. One is incompleteness. It can only predict label sets appearing in the training set, and is powerless for other label sets. The other is inefficiency. As the number of labels increases, it may face high complexity because of the increase in the number of label subsets, and the high imbalance of samples in each class or subset.

To overcome the shortcoming of LP while retaining its simplicity, the idea that dividing the label space into multiple subspaces and applying the LP algorithm in these subspaces has been proposed [99], which can be seen as combining ensemble learning with LP. This is the design principle of random  $k$ -labelsets (RA $k$ EL) [99]. RA $k$ EL divides the overall label set into multiple size- $k$  label subsets randomly and implements LP on each label subspace to ensure computational efficiency. Then, it ensembles several LP classifiers to guarantee the completeness of the prediction. However, an obvious disadvantage of RA $k$ EL is the random partition strategy, which makes the label correlation controlled only by  $k$ , and not considered training data.

To consider the correlation among labels informatively, the data-driven clustering algorithm is used instead of the random partition strategy. Moreover, it has been confirmed that the data-driven method is superior to random selection for the label space division in multi-label classification problems [100]. Especially, the community detection method has been well applied to multiple benchmark data sets for multi-label learning, it divides the label space in a data-driven manner [100]. Thus, this study discusses the application of five classic community detection algorithms in DTIs prediction.

## **2.4.2. Execution of community detection**

Community detection is to find tightly connected community structures in complex network structures, that is, to discover clusters of nodes in the network [100]. In this study, the goal of using the community detection method is to divide label space with a data-driven approach. For this purpose, the community detection method is used based on the weighted co-occurrence graph which is constructed based on training data.

### *2.4.2.1. Construct the weighted label co-occurrence graph*

Defining the weighted undirected co-occurrence graph, where vertices represent

the label set, edges represent label pairs that occur together at least once in the training label set, and the weight assigned to each edge is defined as the number of samples that have both labels. The visualization of the weighted label co-occurrence graph is shown in Figure 1 (a) and (b).

#### 2.4.2.2. Algorithms of community detection

There are many algorithms for community detection. This study uses only five typical algorithms.

The modularity-based approach is a very important branch of the community detection algorithm. It works through maximizing the modularity to implement label space division. Modularity [101] is a measure to describe the quality of the community partition. A relatively good partition has a higher degree of similarity in the community. However, finding the maximum of modularity is NP-hard [102], so we employ three approximation-based techniques instead.

*The fast greedy algorithm (FGA)* [103] is based on the greedy algorithm to search the maximization of modularity, which iteratively merges communities from a single instance. With each iteration, this method merges two communities to achieve the greatest contribution to modularity. When the modularity value of the current community can no longer increase as the community merges, it is defined as convergence.

*The multi-level algorithm (MLA)* [104] is a bottom-up algorithm. In the beginning, each vertex is a separate community, and the vertices move iteratively between the communities by maximizing the local contribution of the vertices to the overall modularity. When modularity is not increased by any movement, each community in the original graph shrinks to a vertex while maintaining the total weight of adjacent edges, and then the process enters the next level. When communities shrink to vertices and the modularity can no longer be increased, the algorithm will stop.

In addition to the modularity-based algorithm, we also use three other algorithms concerning flow.

*The label propagation algorithm (LPA)* [105] is based on the graph semi-supervised learning algorithm, which simulates the diffusion of flow on the network through the diffusion of labels. In the graph, each vertex is assigned a unique label. Next, the tag of every vertex is updated iteratively with the majority tag assigned to the neighbors of the elements. The update order for each iteration is random. The convergent criterion of the algorithm is when all vertex tags are consistent with the most frequent tags in their neighborhood.

*The walk trap algorithm (WTA)* [106] is a bottom-up approach based on random walks. One intuition is that when performing random walks on the graph, it is easy to fall into the dense connection of the graph, which can be regarded as a community. Consider each node as a community, and then calculate the random walk distance or flow distance between all communities with connected edges. Then, take two communities that are connected and have the shortest random walk distance to merge, recalculate the distance between the communities, and then iterate until all nodes are put into the same community.

The *infomap algorithm (IMA)* [107] believes that a good community division should make the average description length of the flow the shortest. It divides the graph by calculating the minimum value of the map equation, where the map equation corresponds to the length of the information description corresponding to the partition.

## 2.5. Performance evaluation

The performance evaluation metrics of multi-label learning are much complex than binary classification [108]. Following the previous researches, this study adopts AUC and AUPR as performance evaluation metrics. It is convenient for comparison with other methods. It is worthy to note that the AUPR is more reliable metrics as a severe punishment on false positive instances for high imbalanced data. Therefore, the discussion in this article focuses on AUPR.

## 2.6. Stratified cross-validation (SCV)

Cross-validation is a typical method to do model selection. For multi-label data, many labels have class imbalance characteristics [109] that each data set has a large number of label sets, and most label sets only contain a small number of samples (Table 2). In this case, the random partitioning strategy used in standard cross-validation may result in some labels having no positive samples in a divided subset. Such a subset will not only affect the accuracy of the model, but may also cause the computational error.

To overcome the above dilemma, a stratified sampling strategy in cross-validation is a proven solution [109, 110], called stratified cross-validation (SCV). Furthermore, the 10-fold SCV has been proved the best method in model selection from the perspective of statistical inference [110]. To ensure the confidence of the results, we performed 5 simulations on 10-fold SCV using different random seeds.

**Table 2.** Statistics for labels of four multi-label data sets. The data in the table is the number of corresponding row and column headings. For the Data sets column, the NR is short for nuclear receptor, GPCR for G-protein-coupled receptor, IC for ion channel, and E for enzyme. For the Tasks column, the  $T_D$  is predicting new drugs,  $T_T$  is predicting new targets.

Tasks	Data sets	Label sets	Samples per label set			Samples per label		
			min	mean	max	min	mean	max
$T_D$	NR	77	1	7.0	132	1	26.8	159
	GPCR	352	1	4.8	135	1	34.5	249
	IC	280	1	2.7	67	1	26.8	144
	E	692	1	2.5	102	1	5.2	154
$T_T$	NR	31	1	1.1	2	1	1.6	9
	GPCR	138	1	1.1	7	1	3.2	34
	IC	179	1	1.3	20	1	8.3	123
	E	713	1	2.0	154	1	4.1	293

## 2.7. Hypothesis test

When comparing multiple algorithms on a set of data sets, Demšar [111] recommends using the non-parametric Friedman rank test [112, 113] which based on algorithm ranking. However, the Friedman rank test can only tell us whether there is a significant difference among algorithms, but cannot specify which algorithms have performance differences. Therefore, post-hoc analysis is needed to locate specific algorithms with differences. For the Friedman rank test, the commonly used post-hoc test method is the Nemenyi test [114], named Friedman-Nemenyi test. This method can indicate whether there is a significant difference between the two algorithms based on the significance level  $\alpha$ .

## 3. Results and Discussion

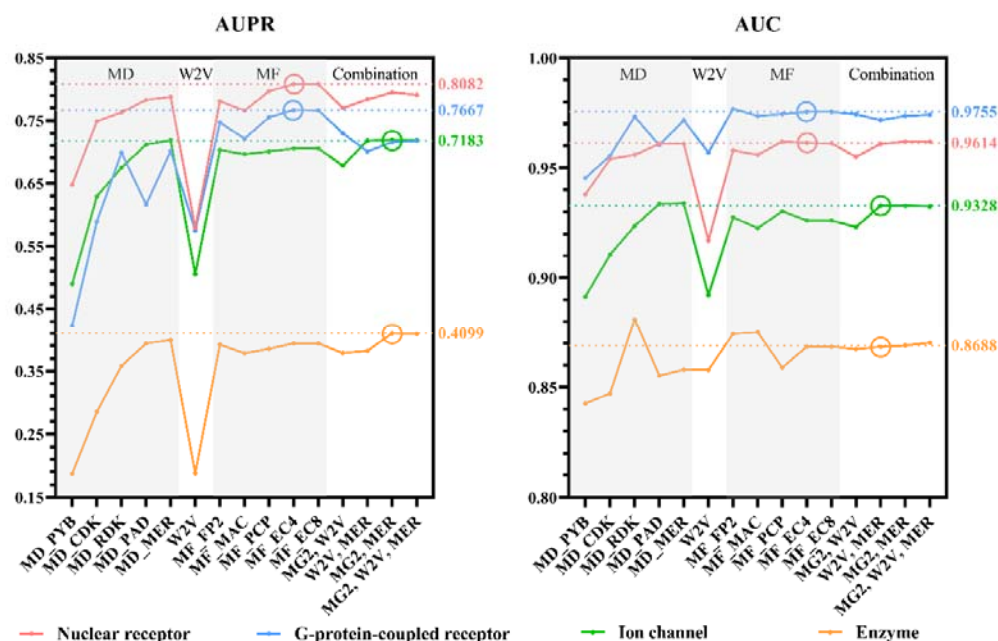
### 3.1. Selecting drug representation

We guess that for different data sets, the most suitable drug representation method is different. So far no other articles have explored this, and our following experiments prove this conjecture. And this phenomenon makes us apply different feature representation methods in different data sets.

To achieve this goal, the experiment is conducted on the basic learning algorithm LP for each updated data set, and the same parameter settings were used. The AUPR and AUC are shown in Figure 2. However, AUPR is the focus as its more reliable, and its lower value is more valuable than high AUC for discussion and comparison.

For MDs, on the four data sets, as the dimension of drug representation increases, the prediction performance tends to be higher because of the more information it describes. For MFs, MF\_EC4 is the best MF among all four data sets, and it has been proved that is sufficient to describe chemical molecules [115]. Further, the result reveals the topological circular fingerprint is better than the other two categories in this study. Next, the feature combination procedure has been performed. There are 4 combinations of MD\_MER, MF\_EC4, and W2V. Figure 4 indicates that the performance of any drug representation after adding W2V was lower than that without W2V.

For different data sets, this study selects the drug representation with the best AUPR as the feature vector. NR and GPCR use MF\_EC4, IC and E use the combination of MF\_EC4 and MD\_MER.



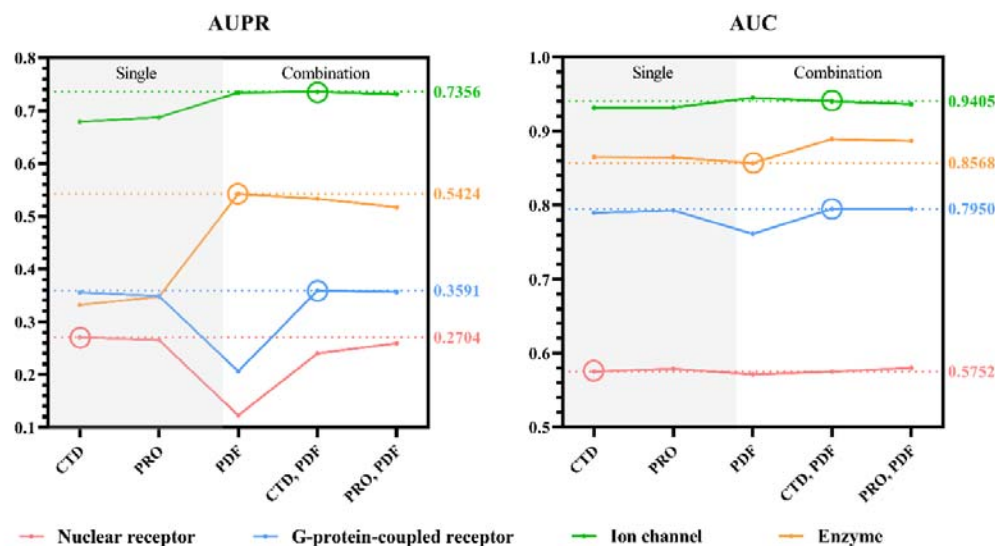
**Figure 2.** The performance among different drug representations.

### 3.2. Selecting target representation

We have adopted the same strategy as drugs, that is, there is no best target representation method, only the most suitable feature representation in a specific situation. Therefore, we also compare target representation methods under four updated data sets and select the most suitable features for each data set according to AUPR.

According to Figure 3, it is obvious that the performance of CTD and PRO is close, probably because both of them are generated by the PROFEAT web server, and CTD is a subset of PRO. Further, for the combination of CTD or PRO with PDF, the performance is also close. Besides, on the NR and GPCR data sets, PDF appears to be a significant trough. Because the protein domain information is too little to fully describe the target. Also, its lower dimension than CTD and PRO makes it have little effect on the performance of feature combinations. On the contrary, on the IC and E data sets, the performance of PDF is significantly improved compared to CTD and PRO as its rich protein domain information. Therefore, PDF dominates the performance of feature combinations.

Finally, we chose the most suitable target representation method for each data set. For NR, the most suitable target representation method is CTD, for IC is PDF, for GPCR and E is the combination of CTD and PDF.



**Figure 3.** The performance among different target representations.

### 3.3. The DTI-MLCD and classical machine learning methods in updated data sets

This study proposed the DTI-MLCD method which applies five data-driven community detection algorithms as label partitioning methods and ensembles them into the multi-label learning method. We explain the superiority of DTI-MLCD from two aspects. The first is the comparison of label partitioning algorithms. For data-driven label partitioning method,  $k$ -means is always used due to its simplicity and popularity, and has been applied with  $k \in \{2, 4, 8, 16, 32\}$  to solve the DTIs

prediction problem [75]. So we use  $k$ -means as the benchmark label partition method to compare with community detection algorithms. To be more convincing, we expanded the value range of  $k$  from 2 to the number of the label set, and used the silhouette coefficient score as a measure of label division quality. The  $k$  value that maximizes the silhouette coefficient score will be used as the optimal number of clusters. The other is to compare the DTI-MLCD algorithm with other classic machine learning algorithms, (a) multi-label algorithms:  $MLkNN$ , BR, CC, LP, and RAKEL, and (b) binary classification algorithms: RF, extremely randomized Trees (ERT), and Gaussian naïve Bayes (GNB).

The results of the above methods on the four updated data sets are listed in Table 3 (task  $T_D$ ) and Table 4 (task  $T_T$ ), respectively. In the results, the proposed DTI-MLCD is superior to other machine learning methods in most cases. The reason why LP performs better than DTI-MLCD on the NR data set under the  $T_T$  task is that NR has few label sets, and both each label set and each label has very few samples (see Table 2 for details), but it has 541 labels. Therefore, only a single LP algorithm can achieve better results, but after adding the community detection algorithm, it will cause overfitting. On the other hand, although the binary classification methods RF and ERT

have achieved competitive results with DTI-MLCD, however, this research experiment proves that its long calculation time will make it difficult to achieve optimal performance through fine-tuning. Further, the Friedman-Nemenyi test with a significance level of 0.05 confirmed the significant differences among methods. All the five proposed methods are at the forefront of the ranking, and the overall performance of FGA is slightly lower than the other four proposed methods.

To illustrate the biological explanation of the proposed methods, Figure 4 visualizes the results of six data-driven label partitioning methods that applied to the NR dataset. Although the community structures obtained by different community detection algorithms have their own characteristics, they also have certain similarities. FGA, LPA, and MLA divide 33 labels into 6 communities. Especially, the community structure of FGA and MLA is the same, noted that both FGA and MLA belong to the modularity-based algorithm. In addition, for the random walk-based algorithm, the number of communities obtained by WTA and IMA is relatively large. Moreover, the  $k$ -means obtains only 4 communities, and the community structure is very different from community detection algorithms.

On the other hand, we discuss the pathway and classification of three communities through the KEGG database, and the details shown in Table 5. The (1, 28) and (26, 27) are communities obtained by all six algorithms, and (20, 29, 32) are only available in IMA. For each of the first two communities, the two vertices belong to the same classification and pathway. The three vertices in the third community have similarities and differences. Therefore, we can think that the label clustering obtained by the community detection algorithm has a certain biological interpretation significance. This also confirms the classical assumption that similar targets tend to combine similar drugs.

**Table 3.** The results of the proposed methods and other classical machine learning methods for task  $T_D$  (i.e. predicting new drugs). WTA, IMA, LPA, MLA, and FGA are community detection algorithms in the proposed DTI-MLCD method.  $k$ -means is the benchmark method that can replace community detections. BR, CC, LP, RAkEL, MLkNN, RF, ET, and GNB are baseline methods.

Algorithm	AUC				AUPR			
	NR	GPCR	IC	E	NR	GPCR	IC	E
FGA	0.9613	0.9738	0.9349	0.8840	<b>0.8135</b>	0.7721	0.7184	0.4148
IMA	0.9611	0.9766	0.9358	0.8768	0.8129	<b>0.7765</b>	<b>0.7194</b>	0.4165
LPA	0.9611	0.9763	0.9345	0.8833	<b>0.8135</b>	0.7755	0.7179	0.4173
MLA	0.9614	0.9745	0.9347	0.8833	0.8134	0.7734	0.7186	0.4165
WTA	0.9611	0.9744	0.9355	0.8839	0.8129	0.7722	0.7187	<b>0.4184</b>
$k$ -means	0.9629	0.9754	0.9352	0.8771	0.8128	0.7731	0.7178	0.4040
BR	0.9622	0.9814	0.9372	0.8771	0.8115	0.7307	0.6914	0.4040
CC	0.9610	0.9767	0.9346	0.8664	0.8109	0.7219	0.6845	0.3822
LP	0.9614	0.9755	0.9328	0.8688	0.8082	0.7667	0.7193	0.4099
RAkEL	0.9532	0.9735	0.9306	0.8736	0.8004	0.7724	0.7048	0.4034
MLkNN	0.9363	0.9575	0.8356	0.7962	0.6699	0.6340	0.1644	0.0454

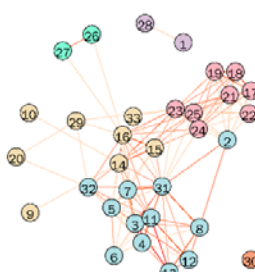
RF	0.9626	0.9754	0.9423	0.8983	0.8102	0.7730	0.7113	0.3238
ERT	0.9616	0.9688	0.9314	0.8786	0.8102	0.7571	0.7049	0.3546
GNB	0.6818	0.7037	0.5015	0.5273	0.3732	0.3730	0.4197	0.0054

**Table 4.** The results of the proposed methods and other classical machine learning methods for task  $T_T$  (i.e. predicting new targets). WTA, IMA, LPA, MLA, and FGA are community detection algorithms in the proposed DTI-MLCD method.  $k$ -means is the benchmark method that can replace community detections. BR, CC, LP, RAKEL, MLkNN, RF, ET, and GNB are baseline methods.

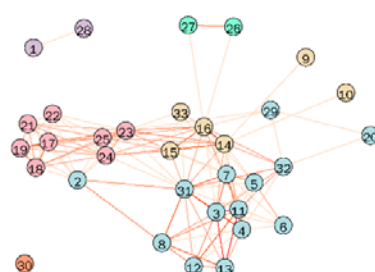
Algorithm	AUC				AUPR			
	NR	GPCR	IC	E	NR	GPCR	IC	E
FGA	0.5715	0.8027	0.9489	0.8593	0.2311	0.3702	0.7468	0.5669
IMA	0.5748	0.8002	0.9476	0.8598	0.2409	0.3683	<b>0.7663</b>	0.5669
LPA	0.5759	0.8048	0.9459	0.8591	0.2494	<b>0.3785</b>	0.7518	0.5670
MLA	0.5657	0.8062	0.9478	0.8640	0.2177	0.3759	0.7609	<b>0.5677</b>
WTA	0.5745	0.8002	0.9463	0.8642	0.2401	0.3746	0.7574	0.5673
$k$ -means	0.5611	0.7893	0.9382	0.8639	0.2383	0.3693	0.7174	0.5668
BR	0.5617	0.7892	0.9382	0.8639	0.2352	0.3694	0.7174	0.5673
CC	0.5647	0.7580	0.9183	0.8563	0.2360	0.2424	0.6475	0.5152
LP	0.5752	0.7927	0.9403	0.8568	<b>0.2704</b>	0.3670	0.7429	0.5651
RAKEL	0.5642	0.7902	0.9395	0.8640	0.2352	0.3714	0.7242	0.5670
MLkNN	0.5470	0.7351	0.9094	0.8053	0.1811	0.2751	0.6414	0.3112
RF	0.6764	0.7610	0.9511	0.8775	0.2445	0.3104	0.7419	0.5652
ERT	0.5804	0.7179	0.9459	0.8404	0.2632	0.3410	0.7650	0.5462
GNB	0.4451	0.6566	0.5006	0.5347	0.3149	0.3770	0.3107	0.0035



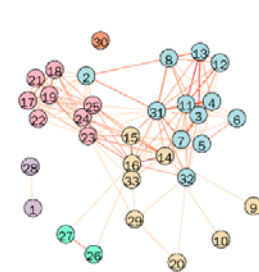
(a)  $k$ -Means



(b) Fast greedy



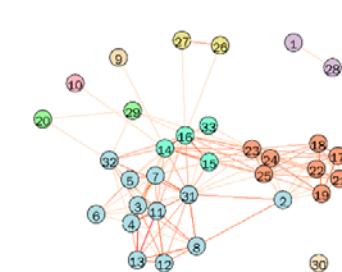
(c) Label propagation



(d) Multi-level



(e) Infomap



(f) Walk trap

**Figure 4.** The label partition results that community detection algorithms and benchmark  $k$ -means method applied in the label space of the nuclear receptor data set.

**Table 5.** The details for three communities A: (1, 28), B: (26, 27), and C: (20, 29, 32). The numbers represent the nodes in Figure 7.

Community	Gene	Details
A	1	LXRA
	28	LXRB
		Classification: (1) Liver X receptor like receptor (2) Cys4 thyroid hormone-like transcription factor Pathway: Insulin resistance
B	26	THRA
	27	THRB
		Classification: (1) Cys4 thyroid hormone-like transcription factor (2) Thyroid hormone like receptor Pathway: (1) Neuroactive ligand-receptor interaction (2) Thyroid hormone signaling pathway
C	20	RORA
		Classification: (1) Cys4 thyroid hormone-like transcription factor (2) Thyroid hormone like RAR-related orphan receptor
	29	VDR
		Classification: (1) Cys4 thyroid hormone-like transcription factor (2) Thyroid hormone like vitamin D3 like receptor
	32	CAR
		Classification: (1) Cys4 thyroid hormone-like transcription factor (2) Thyroid hormone like vitamin D3 like receptor
		(3) constitutive androstane receptor

### 3.4. The DTI-MLCD and benchmark methods in Yamanishi\_08 data sets

We compare the proposed method against three state-of-the-art methods for DTIs prediction. NetLapRLS [65], BLM-NII [116], and DDR [27]. NetLapRLS introduces the drug-target network information into the manifold Laplacian regularized the least square method which uses the concept of the bipartite local model. BLM-NII exploits bipartite local model with neighbor-based interaction profile inferring, which adds a preprocessing component to infer training data from neighbors' interaction profiles. DDR executes the graph mining technique firstly to acquire the comprehensive feature vectors and then applies the random forest model by using different graph-based features extracted from the drug-target heterogeneous graph. Since these methods are proposed on the Yamanishi\_08 data set, we perform the proposed DTI-MLCD method on this data set and compare it with other methods. All methods are carried out under the same experimental environment, such as SCV, random seeds, etc. And the results are obtained after fine-tuning. As reflected in Table 6, all the proposed methods in task  $T_D$  outperform the benchmark methods in terms of AUPR. For task  $T_T$  (Table 7), the proposed methods outperform benchmark methods in IC and E data sets while they are slightly inferior to BLM-NII in NR and GPCR. In order to comprehensively

test the superiority of the method proposed in this study, we conduct the Friedman-Nemenyi test for all 8 methods. This hypothesis test is performed on both AUPR and AUC for completeness although AUPR is more informative than AUC in this study. And the result reveals that all the proposed methods are ranked before the three benchmark methods. Moreover, they are significantly better than DDR and NetLapRLS with a significance level of 0.05 and 0.1, respectively.

**Table 6.** The results of the proposed methods and three benchmark methods for task  $T_D$  (i.e. predicting new drugs). WTA, IMA, LPA, MLA, and FGA are community detection algorithms in the proposed DTI-MLCD method.

Algorithm	AUC				AUPR			
	NR	GPCR	IC	E	NR	GPCR	IC	E
FGA	0.7829	0.8636	0.8220	0.8506	0.4990	0.4504	0.3887	<b>0.4105</b>
IMA	0.7830	<b>0.8698</b>	0.8223	0.8537	0.4992	<b>0.4593</b>	0.3857	0.4045
LPA	0.7785	0.8655	0.8197	<b>0.8563</b>	<b>0.5079</b>	0.4537	<b>0.3924</b>	0.4067
MLA	0.7829	0.8632	<b>0.8237</b>	0.8522	0.4990	0.4488	0.3885	0.4088
WTA	0.7828	0.8619	0.8219	0.8539	0.4989	0.4501	0.3860	0.4045
BLM-NII	<b>0.8042</b>	0.8496	0.8119	0.8204	0.4503	0.3415	0.3260	0.2690
NetLapRLS	0.7919	0.8281	0.7721	0.7933	0.4313	0.2456	0.2078	0.1287
DDR	0.6019	0.5678	0.4994	0.4768	0.2878	0.1907	0.1471	0.1336

**Table 7.** The results of the proposed methods and three benchmark methods for task  $T_T$  (i.e. predicting new targets). WTA, IMA, LPA, MLA, and FGA are community detection algorithms in the proposed DTI-MLCD method.

Algorithm	AUC				AUPR			
	NR	GPCR	IC	E	NR	GPCR	IC	E
FGA	0.4961	0.7458	0.9104	0.9285	0.3472	0.2943	0.7047	0.7861
IMA	0.4929	0.7429	<b>0.9114</b>	0.9214	0.3457	0.2919	0.7027	0.7875
LPA	0.4925	0.7509	0.9105	0.9214	0.3398	0.2969	0.7082	<b>0.7877</b>
MLA	0.4998	0.7481	0.9098	<b>0.9286</b>	0.3487	0.2942	<b>0.7093</b>	0.7868
WTA	0.4923	0.7495	0.9103	0.9217	0.3460	0.3010	0.7046	0.7873
BLM-NII	<b>0.5042</b>	<b>0.7777</b>	0.9093	0.9193	<b>0.3726</b>	<b>0.3078</b>	0.7028	0.7570
NetLapRLS	0.4986	0.7425	0.9082	0.9161	0.2793	0.2515	0.6543	0.7064
DDR	0.4932	0.6290	0.5784	0.6965	0.2365	0.2288	0.3108	0.5026

### 3.5. Independent test

We conduct independent tests of the proposed DTI-MLCD method according to the data set before and after the update. First, build the independent test set. Drugs and their DTIs that do not exist in the Yamanishi\_08 data set but exist in the updated data set will be used as independent test samples for task  $T_D$ . Similarly, independent test samples of task  $T_T$  is constructed. Then, conduct independent tests on the model which trained on the Yamanishi\_08 data set. The results are shown in Table 8 (task  $T_D$ ) and Table 9 (task  $T_T$ ).

**Table 8.** The results of independent tests on Yamanishi\_08 data sets for task  $T_D$ .

Algorithm	AUC				AUPR			
	NR	GPCR	IC	E	NR	GPCR	IC	E
FGA	0.8174	0.8941	0.8238	0.8457	0.5331	0.3953	0.2795	0.1369
IMA	0.8172	0.9020	0.8262	0.8426	0.5331	<b>0.4000</b>	0.3012	0.1353
LPA	0.8157	0.9000	0.8257	0.8430	<b>0.5334</b>	0.3982	<b>0.3013</b>	0.1375
MLA	0.8174	0.8944	0.8246	0.8455	0.5331	0.3928	0.2776	<b>0.1378</b>
WTA	0.8174	0.8920	0.8230	0.8427	0.5331	0.3935	0.2890	0.1363

**Table 9.** The results of independent tests on Yamanishi\_08 data sets for task  $T_T$ .

Algorithm	AUC				AUPR			
	NR	GPCR	IC	E	NR	GPCR	IC	E
FGA	0.8224	0.6130	0.7353	0.7348	0.3787	0.0076	0.2090	<b>0.1077</b>
IMA	0.8224	0.6135	0.7323	0.6834	0.3787	0.0075	0.2144	0.1057
LPA	0.8223	0.6107	0.7383	0.6809	<b>0.3840</b>	0.0076	0.2127	0.1048
MLA	0.8228	0.6255	0.7395	0.7339	0.3787	<b>0.0080</b>	0.2119	0.1071
WTA	0.8224	0.6080	0.7363	0.6814	0.3787	0.0074	<b>0.2142</b>	0.1052

#### 4. Conclusion

This study updated the gold standard data set Yamanishi\_08, and proposed DTI-MLCD for DTIs prediction, which is a new multi-label learning framework empowered by community detection. This framework has 5 effective models corresponding to five community detection algorithms to do label partitioning. This study conducted experiments on the gold standard data set before and after the update. On the original data set, the DTI-MLCD is compared with other benchmark methods, and its superiority is confirmed. In the updated data set, DTI-MLCD is superior to other classic machine learning algorithms. In addition, this study also constructed independent tests with new and old data sets. On the other hand, the results of the five community detection algorithms used in this framework are not significantly different. Moreover, they are superior to the benchmark  $k$ -means algorithm in performance and interpretability.

In the future, we will solve the problem of label imbalance and construct positive and negative samples in the form of semi-supervised learning to improve the performance of the framework in predicting DTIs.

#### 5. References

1. Chen M, Kuzmin K, Szymanski BK: **Community detection via maximization of modularity and its variants.** *IEEE Transactions on Computational Social Systems* 2014, **1**(1):46-65.
2. Breckenridge A: **Clinical pharmacology and therapeutics.** *BMJ* 1995, **310**(6976):377-380.
3. Chen X, Yan CC, Zhang X, Zhang X, Dai F, Yin J, Zhang Y: **Drug-target interaction prediction: databases, web servers and computational models.** *BRIEF BIOINFORM* 2016, **17**(4):696-712.
4. Chen R, Liu X, Jin S, Lin J, Liu J: **Machine Learning for Drug-Target Interaction Prediction.**

*MOLECULES* 2018, **23**(9).

5. Zhang W, Lin W, Zhang D, Wang S, Shi J, Niu Y: **Recent advances in the machine learning-based drug-target interaction prediction.** *CURR DRUG METAB* 2018.

6. Anusuya S, Keshewani M, Priya KV, Vimala A, Shanmugam G, Velmurugan D, Gromiha MM: **Drug-Target Interactions: Prediction Methods and Applications.** *CURR PROTEIN PEPT SC* 2018, **19**(6):537.

7. Zhao Q, Yu H, Ji M, Zhao Y, Chen X: **Computational Model Development of Drug-Target Interaction Prediction: A Review.** *Curr Protein Pept Sci* 2019, **20**(6):492-494.

8. Bleakley K, Yamanishi Y: **Supervised prediction of drug-target interactions using bipartite local models.** *BIOINFORMATICS* 2009, **25**(18):2397-2403.

9. Mei JP, Kwok CK, Yang P, Li XL, Zheng J: **Drug-target interaction prediction by learning from local information and neighbors.** *BIOINFORMATICS* 2013, **29**(2):238-245.

10. Li Z, Han P, You Z, Li X, Zhang Y, Yu H, Nie R, Chen X: **In silico prediction of drug-target interaction networks based on drug chemical structure and protein sequences.** *SCI REP-UK* 2017, **7**(1).

11. Keum J, Nam H: **SELF-BLM: Prediction of drug-target interactions via self-training SVM.** *PLOS ONE* 2017, **12**(2):e171839.

12. Ezzat A, Wu M, Li X, Kwok C: **Drug-target interaction prediction via class imbalance-aware ensemble learning.** *BMC BIOINFORMATICS* 2016, **17**(S19).

13. Ding Y, Tang J, Guo F: **Identification of drug-target interactions via multiple information integration.** *INFORM SCIENCES* 2017, **418-419**:546-560.

14. Liu H, Sun J, Guan J, Zheng J, Zhou S: **Improving compound-protein interaction prediction by building up highly credible negative samples.** *BIOINFORMATICS* 2015, **31**(12):i221-i229.

15. Peng L, Zhu W, Liao B, Duan Y, Chen M, Chen Y, Yang J: **Screening drug-target interactions with positive-unlabeled learning.** *SCI REP-UK* 2017, **7**(1).

16. Meng FR, You ZH, Chen X, Zhou Y, An JY: **Prediction of Drug-Target Interaction Networks from the Integration of Protein Sequences and Drug Chemical Structures.** *MOLECULES* 2017, **22**(7).

17. Mousavian Z, Khakabimamaghani S, Kavousi K, Masoudinejad A: **Drug-target interaction prediction from PSSM based evolutionary information.** *J PHARMACOL TOX MET* 2016, **78**:42-51.

18. Tabei Y, Pauwels E, Stoven V, Takemoto K, Yamanishi Y: **Identification of chemogenomic features from drug – target interaction networks using interpretable classifiers.** *BIOINFORMATICS* 2012, **28**(18):487-494.

19. Yamanishi Y, Kotera M, Kanehisa M, Goto S: **Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework.** *BIOINFORMATICS* 2010, **26**(12):246-254.

20. Mahmud SMH, Chen W, Jahan H, Liu Y, Sujana NI, Ahmed S: **iDTi-CSsmoteB: Identification of Drug-Target Interaction Based on Drug Chemical Structure and Protein Sequence Using XGBoost With Over-Sampling Technique SMOTE.** *IEEE ACCESS* 2019, **7**:48699-48714.

21. Zhang J, Zhu M, Chen P, Wang B: **DrugRPE: Random projection ensemble approach to drug-target interaction prediction.** *NEUROCOMPUTING* 2017, **228**:256-262.

22. Rayhan F, Ahmed S, Shatabda S, Farid DM, Mousavian Z, Dehzangi A, Rahman MS:

**iDTI-ESBoost: Identification of Drug Target Interaction Using Evolutionary and Structural Features with Boosting.** *SCI REP-UK* 2017, 7(1).

23. Ezzat A, Wu M, Li X, Kwoh C: **Drug-target interaction prediction using ensemble learning and dimensionality reduction.** *METHODS* 2017, 129:81-88.

24. Sharma A, Rani R: **BE-DTI': Ensemble framework for drug target interaction prediction using dimensionality reduction and active learning.** *COMPUT METH PROG BIO* 2018, 165:151-162.

25. Shi H, Liu S, Chen J, Li X, Ma Q, Yu B: **Predicting drug-target interactions using Lasso with random forest based on evolutionary information and chemical structure.** *GENOMICS* 2018.

26. Wang L, You ZH, Chen X, Yan X, Liu G, Zhang W: **RFDT: A Rotation Forest-based Predictor for Predicting Drug-Target Interactions Using Drug Structure and Protein Sequence Information.** *Curr Protein Pept Sci* 2018, 19(5):445-454.

27. Olayan RS, Ashoor H, Bajic VB: **DDR: efficient computational method to predict drug-target interactions using graph mining and machine learning approaches.** *BIOINFORMATICS*.

28. Mousavian Z, Khakabimamaghani S, Kavousi K, Masoudi-Nejad A: **Drug-target interaction prediction from PSSM based evolutionary information.** *J PHARMACOL TOX MET* 2016, 78:42-51.

29. Lee H, Kim W: **Comparison of Target Features for Predicting Drug-Target Interactions by Deep Neural Network Based on Large-Scale Drug-Induced Transcriptome Data.** *PHARMACEUTICS* 2019, 11(8):377.

30. Lee I, Keum J, Nam H: **DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences.** *PLOS COMPUT BIOL* 2019, 15(6):e1007129.

31. Lim J, Ryu S, Park K, Choe YJ, Ham J, Kim WY: **Predicting Drug-Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation.** *J CHEM INF MODEL* 2019, 59(9):3981-3988.

32. Tian K, Shao M, Wang Y, Guan J, Zhou S: **Boosting compound-protein interaction prediction by deep learning.** *METHODS* 2016, 110:64-72.

33. Zhang Y, Wang X, Kaushik AC, Chu Y, Shan X, Zhao M, Xu Q, Wei D: **SPVec: A Word2vec-Inspired Feature Representation Method for Drug-Target Interaction Prediction.** *FRONT CHEM* 2020, 7.

34. Wen M, Zhang Z, Niu S, Sha H, Yang R, Yun Y, Lu H: **Deep-Learning-Based Drug-Target Interaction Prediction.** *J PROTEOME RES* 2017, 16(4):1401-1409.

35. Xie L, He S, Song X, Bo X, Zhang Z: **Deep learning-based transcriptome data classification for drug-target interaction prediction.** *BMC GENOMICS* 2018, 19(S7).

36. You J, McLeod RD, Hu P: **Predicting drug-target interaction network using deep learning model.** *COMPUT BIOL CHEM* 2019, 80:90-101.

37. Wang L, You Z, Chen X, Xia S, Liu F, Yan X, Zhou Y, Song K: **A Computational-Based Method for Predicting Drug-Target Interactions by Using Stacked Autoencoder Deep Neural Network.** *J COMPUT BIOL* 2018, 25(3):361-373.

38. Rayhan F, Ahmed S, Mousavian Z, Farid DM, Shatabda S: **FRnet-DTI: Deep Convolutional Neural Networks with Evolutionary and Structural Features for Drug-Target Interaction**. 2018.
39. **Interpretable Drug Target Prediction Using Deep Neural Representation**.
40. **Large-Scale Prediction of Drug-Target Interactions from Deep Representations**.
41. **Drug-Target Interaction Prediction in Drug Repositioning Based on Deep Semi-Supervised Learning**.
42. Cheng F, Liu C, Jiang J, Lu W, Li W, Liu G, Zhou W, Huang J, Tang Y: **Prediction of drug-target interactions and drug repositioning via network-based inference**. *PLOS COMPUT BIOL* 2012, **8**(5):e1002503.
43. Chen X, Liu M, Yan G: **Drug–target interaction prediction by random walk on the heterogeneous network**. *MOL BIOSYST* 2012, **8**(7):1970.
44. Fu G, Ding Y, Seal A, Chen B, Sun Y, Bolton E: **Predicting drug target interactions using meta-path-based semantic network analysis**. *BMC BIOINFORMATICS* 2016, **17**(1):160.
45. Luo Y, Zhao X, Zhou J, Yang J, Zhang Y, Kuang W, Peng J, Chen L, Zeng J: **A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information**. *NAT COMMUN* 2017, **8**(1).
46. Wu Z, Cheng F, Li J, Li W, Liu G, Tang Y: **SDTNBI: an integrated network and chemoinformatics tool for systematic prediction of drug – target interactions and drug repositioning**. *BRIEF BIOINFORM* 2016, **18**(2):333-347.
47. Li Z, Huang M, Zhong W, Liu Z, Xie Y, Dai Z, Zou X: **Identification of drug-target interaction from interactome network with 'guilt-by-association' principle and topology features**. *BIOINFORMATICS* 2016, **32**(7):1057-1064.
48. Alaimo S, Pulvirenti A, Giugno R, Ferro A: **Drug-target interaction prediction through domain-tuned network-based inference**. *BIOINFORMATICS* 2013, **29**(16):2004-2008.
49. Seal A, Ahn Y, Wild DJ: **Optimizing drug–target interaction prediction based on random walk on heterogeneous networks**. *J CHEMINFORMATICS* 2015, **7**(1):40.
50. Yan X, Zhang S, Zhang S: **Prediction of drug–target interaction by label propagation with mutual interaction information derived from heterogeneous network**. *MOL BIOSYST* 2016, **12**(2):520-531.
51. Emig D, Ivliev A, Pustovalova O, Lancashire L, Bureeva S, Nikolsky Y, Bessarabova M: **Drug Target Prediction and Repositioning Using an Integrated Network-Based Approach**. *PLOS ONE* 2013, **8**(4).
52. Lu Y, Guo Y, Korhonen A: **Link prediction in drug-target interactions network using similarity indices**. *BMC BIOINFORMATICS* 2017, **18**(1):39.
53. Yu W, Yan Y, Liu Q, Wang J, Jiang Z: **Predicting drug–target interaction networks of human diseases based on multiple feature information**. *PHARMACOGENOMICS* 2013, **14**(14):1701-1707.
54. Alaimo S, Bonnici V, Cancemi D, Ferro A, Giugno R, Pulvirenti A: **DT-Web: a web-based application for drug-target interaction and drug combination prediction through domain-tuned network-based inference**. *BMC SYST BIOL* 2015, **9**(3):1-11.

55. Re M, Valentini G: **Network-Based Drug Ranking and Repositioning with Respect to DrugBank Therapeutic Categories.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2013, **10**(6):1359-1371.
56. Gönen M: **Predicting drug-target interactions from chemical and genomic kernels using Bayesian matrix factorization.** *BIOINFORMATICS* 2012, **28**(18):2304-2310.
57. Liu Y, Wu M, Miao C, Zhao P, Li X: **Neighborhood regularized logistic matrix factorization for drug-target interaction prediction.** *PLOS COMPUT BIOL* 2016, **12**(2):e1004760.
58. Ezzat A, Zhao P, Wu M, Li X, Kwoh C: **Drug-Target Interaction Prediction with Graph Regularized Matrix Factorization.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2017, **14**(3):646-656.
59. Zheng X, Ding H, Mamitsuka H, Zhu S: **Collaborative matrix factorization with multiple similarities for predicting drug-target interactions.** In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*:2013: ACM; 2013: 1025-1033.
60. Wang Y: **Predicting drug-target interactions by dual-network integrated logistic matrix factorization.** *SCI REP-UK*, **7**:40376.
61. Peska L, Buza K, Koller J: **Drug-target interaction prediction: A Bayesian ranking approach.** *COMPUT METH PROG BIO* 2017, **152**:15-21.
62. Bolgár B, Antal P: **VB-MK-LMF: fusion of drugs, targets and interactions using variational Bayesian multiple kernel logistic matrix factorization.** *BMC BIOINFORMATICS* 2017, **18**(1).
63. Cobanoglu MC, Liu C, Hu F, Oltvai ZN, Bahar I: **Predicting Drug-Target Interactions Using Probabilistic Matrix Factorization.** *J CHEM INF MODEL* 2013, **53**(12):3399-3409.
64. Jacob L, Vert JP: **Protein-ligand interaction prediction: an improved chemogenomics approach.** *BIOINFORMATICS* 2008, **24**(19):2149-2156.
65. Xia Z, Wu L, Zhou X, Wong STC: **Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces.** *BMC SYST BIOL* 2010, **4 Suppl 2**(Suppl 2):S6.
66. van Laarhoven T, Nabuurs SB, Marchiori E: **Gaussian interaction profile kernels for predicting drug-target interaction.** *BIOINFORMATICS* 2011, **27**(21):3036-3043.
67. Shang F, Jiao LC, Liu Y: **Integrating Spectral Kernel Learning and Constraints in Semi-Supervised Classification.** *NEURAL PROCESS LETT* 2012, **36**(2):101-115.
68. Nascimento ACA, Prudêncio RBC, Costa IG: **A multiple kernel learning algorithm for drug-target interaction prediction.** *BMC BIOINFORMATICS* 2016, **17**(1).
69. Wang Y, Zeng J: **Predicting drug-target interactions using restricted Boltzmann machines.** *BIOINFORMATICS* 2013, **29**(13):i126-i134.
70. Koohi A: **Prediction of drug-target interactions using popular Collaborative Filtering methods.** In: *2013-01-01* 2013: IEEE; 2013: 58-61.
71. Zhang X, Li L, Ng MK, Zhang S: **Drug-target interaction prediction by integrating multiview network data.** *COMPUT BIOL CHEM* 2017, **69**:185-193.
72. Zhang W, Chen Y, Li D: **Drug-Target Interaction Prediction through Label Propagation with Linear Neighborhood Information.** *MOLECULES* 2017, **22**(12):2056.
73. Yuan Q, Gao J, Wu D, Zhang S, Mamitsuka H, Zhu S: **DrugE-Rank: improving drug-target**

**interaction prediction of new candidate drugs or targets by ensemble learning to rank.** *BIOINFORMATICS* 2016, **32**(12):i18-i27.

74. Ceci M, Hollmén J, Todorovski L, Vens C, Dzeroski S: **Machine Learning and Knowledge Discovery in Databases : European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II**, vol. 10535. Cham : Springer International Publishing AG; 2017.

75. Pliakos K, Vens C, Tsoumakas G: **Predicting drug-target interactions with multi-label classification and label partitioning.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2019:1.

76. Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M: **Prediction of drug-target interaction networks from the integration of chemical and genomic spaces.** *BIOINFORMATICS* 2008, **24**(13):i232-i240.

77. Kanehisa M, Araki M, Goto S, Hattori M, Hirakawa M, Itoh M, Katayama T, Kawashima S, Okuda S, Tokimatsu T: **KEGG for linking genomes to life and the environment.** *NUCLEIC ACIDS RES* 2007, **36**(suppl\_1):D480-D484.

78. Schomburg I, Chang A, Ebeling C, Gremse M, Heldt C, Huhn G, Schomburg D: **BRENDA, the enzyme database: updates and major new developments.** *NUCLEIC ACIDS RES* 2004, **32**(suppl\_1):D431-D433.

79. Günther S, Kuhn M, Dunkel M, Campillos M, Senger C, Petsalaki E, Ahmed J, Urdiales EG, Gewiss A, Jensen LJ: **SuperTarget and Matador: resources for exploring drug-target relationships.** *NUCLEIC ACIDS RES* 2007, **36**(suppl\_1):D919-D922.

80. Wishart DS, Knox C, Guo AC, Cheng D, Shrivastava S, Tzur D, Gautam B, Hassanali M: **DrugBank: a knowledgebase for drugs, drug actions and drug targets.** *NUCLEIC ACIDS RES* 2007, **36**(suppl\_1):D901-D906.

81. UniProt: **a worldwide hub of protein knowledge.** *NUCLEIC ACIDS RES* 2019, **47**(D1):D506-D515.

82. Willighagen EL, Mayfield JW, Alvarsson J, Berg A, Carlsson L, Jeliaskova N, Kuhn S, Pluskal T, Rojas-Chertó M, Spjuth O *et al*: **The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching.** *J CHEMINFORMATICS* 2017, **9**(1).

83. O'Boyle NM, Morley C, Hutchison GR: **Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit.** *CHEM CENT J* 2008, **2**:5.

84. Landrum G: **rdkit/rdkit: 2019 03 4 (Q1 2019) Release.** 2019. URL <https://doi.org/10.5281/zenodo.3366468>.

85. He Y, Liew CY, Sharma N, Woo SK, Chau YT, Yap CW: **PaDEL-DDPREDICTOR: open-source software for PD-PK-T prediction.** *J COMPUT CHEM* 2013, **34**(7):604-610.

86. Riniker S, Landrum GA: **Open-source platform to benchmark fingerprints for ligand-based virtual screening.** *J CHEMINFORMATICS* 2013, **5**(1):26.

87. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR: **Open Babel: An open chemical toolbox.** *J Cheminform* 2011, **3**:33.

88. Rogers D, Hahn M: **Extended-Connectivity Fingerprints.** *J CHEM INF MODEL* 2010, **50**(5):742-754.

89. Durant JL, Leland BA, Henry DR, Nourse JG: **Reoptimization of MDL keys for use in drug discovery.** *J Chem Inf Comput Sci* 2002, **42**(6):1273-1280.

90. Chen B, Wild D, Guha R: **PubChem as a source of polypharmacology.** *J CHEM INF MODEL*

2009, **49**(9):2044-2055.

91. Li ZR, Lin HH, Han LY, Jiang L, Chen X, Chen YZ: **PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence.** *NUCLEIC ACIDS RES* 2006, **34**(Web Server issue):W32-W37.

92. El-Gebali S, Mistry J, Bateman A, Eddy SR, Luciani A, Potter SC, Qureshi M, Richardson LJ, Salazar GA, Smart A *et al*: **The Pfam protein families database in 2019.** *NUCLEIC ACIDS RES* 2019, **47**(D1):D427-D432.

93. Zhang M, Zhou Z: **ML-KNN: A lazy learning approach to multi-label learning.** *PATTERN RECOGN* 2007, **40**(7):2038-2048.

94. Liu G, Li G, Wang Y, Wang Y: **Modelling of inquiry diagnosis for coronary heart disease in traditional Chinese medicine by using multi-label learning.** *BMC COMPLEM ALTERN M* 2010, **10**(1):37.

95. Zhang T, Dai H, Liu LA, Lewis DF, Wei D: **Classification models for predicting cytochrome P450 enzyme-substrate selectivity.** *MOL INFORM* 2012, **31**(1):53-62.

96. Zhang M, Li Y, Liu X, Geng X: **Binary relevance for multi-label learning: an overview.** *FRONT COMPUT SCI-CHI* 2018, **12**(2):191-202.

97. Read J, Pfahringer B, Holmes G, Frank E: **Classifier chains for multi-label classification.** In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*:2009: Springer; 2009: 254-269.

98. SHEN C, Zhi-hai W, SUN Y: **A Multi-Label Classification Algorithm Based on Label Clustering.** *Computer engineering & Software* 2014(8):5.

99. Tsoumakas G, Katakis I, Vlahavas I: **Random k-labelsets for multilabel classification.** *IEEE T KNOWL DATA EN* 2010, **23**(7):1079-1089.

100. Szymański P, Kajdanowicz T, Kersting K: **How Is a Data-Driven Approach Better than Random Choice in Label Space Division for Multi-Label Classification?** *ENTROPY-SWITZ* 2016, **18**(8):282.

101. **Finding and evaluating community structure in networks.**

102. Brandes U, Delling D, Gaertler M, Gorke R, Hoefer M, Nikoloski Z, Wagner D: **On Modularity Clustering.** *IEEE T KNOWL DATA EN* 2008, **20**(2):172-188.

103. Clauset A, Newman ME, Moore C: **Finding community structure in very large networks.** *PHYS REVE* 2004, **70**(6):66111.

104. **Fast unfolding of community hierarchies in large networks.**

105. Raghavan UN, Albert R, Kumara S: **Near linear time algorithm to detect community structures in large-scale networks.** *PHYS REV E* 2007, **76**(3):36106.

106. Pons P, Latapy M: **Computing communities in large networks using random walks.** In: *International symposium on computer and information sciences*:2005: Springer; 2005: 284-293.

107. Rosvall M, Axelsson D, Bergstrom CT: **The map equation.** *The European Physical Journal Special Topics* 2009, **178**(1):13-23.

108. Zhang M, Zhou Z: **A Review on Multi-Label Learning Algorithms.** *IEEE T KNOWL DATA EN* 2014, **26**(8):1819-1837.

109. **On the Stratification of Multi-Label Data.**

110. **A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.**

111. Demsar J: **Statistical Comparisons of Classifiers over Multiple Data Sets.** *J MACH LEARN*

*RES* 2006, **7**:1-30.

112. Friedman M: **The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance**. *Publications of the American Statistical Association* 1937, **32**(200):675-701.

113. Friedman, Milton: **A Comparison of Alternative Tests of Significance for the Problem of \$m\$ Rankings**. *Annals of Mathematical Statistics*, **11**(1):86-92.

114. Nemenyi P: **Distribution-free multiple comparisons**, unpublished Ph. D.: Ph. D. Dissertation, thesis, Princeton University, Princeton, New Jersey; 1963.

115. Rogers D, Hahn M: **Extended-connectivity fingerprints**. *J CHEM INF MODEL* 2010, **50**(5):742-754.

116. Mei J, Kwoh C, Yang P, Li X, Zheng J: **Drug-target interaction prediction by learning from local information and neighbors**. *BIOINFORMATICS* 2013, **29**(2):238-245.