*Briefings in Bioinformatics*, 22(3), 2021, 1–15

**doi: 10.1093/bib/bbaa205**
Problem Solving Protocol

# DTI-MLCD: predicting drug-target interactions using multi-label learning with community detection method

Yanyi Chu, Xiaoqi Shan, Tianhang Chen, Mingming Jiang, Yanjing Wang, Qiankun Wang, Dennis Russell Salahub, Yi Xiong and Dong-Qing Wei

Corresponding authors. Yi Xiong and Dong-Qing Wei, State Key Laboratory of Microbial Metabolism, and School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China; Peng Cheng Laboratory, Vanke Cloud City Phase I Building 8, Xili Street, Nanshan District, Shenzhen, Guangdong, 518055, China. Tel: +86 21-34204573; E-mail: xiongyi@sjtu.edu.cn, dqwei@sjtu.edu.cn

## Abstract

Identifying drug-target interactions (DTIs) is an important step for drug discovery and drug repositioning. To reduce the experimental cost, a large number of computational approaches have been proposed for this task. The machine learning-based models, especially binary classification models, have been developed to predict whether a drug-target pair interacts or not. However, there is still much room for improvement in the performance of current methods. Multi-label learning can overcome some difficulties caused by single-label learning in order to improve the predictive performance. The key challenge faced by multi-label learning is the exponential-sized output space, and considering label correlations can help to overcome this challenge. In this paper, we facilitate multi-label classification by introducing community detection methods for DTI prediction, named DTI-MLCD. Moreover, we updated the gold standard data set by adding 15,000 more positive DTI samples in comparison to the data set, which has widely been used by most of previously published DTI prediction methods since 2008. The proposed DTI-MLCD is applied to both data sets, demonstrating its superiority over other machine learning methods and several existing methods. The data sets and source code of this study are freely available at https://github.com/a96123155/DTI-MLCD.

**Key words:** Drug-target interaction; multi-label learning; label correlation; community detection

**Yanyi Chu** is a Ph.D. candidate at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University. She works on drug discovery through machine learning methods.
**Xiaoqi Shan** is a master at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University working on the study of drug metabolism.
**Tianhang Chen** is currently a junior student at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University. His research interests are bioinformatics, data mining, and machine learning for integrating data in biology and medicine.
**Mingming Jiang** is a master student at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University. He works on deep learning in bioinformatics.
**Yanjing Wang** is a postdoctoral scholar at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University. She works on drug discovery through machine learning methods and molecular dynamics simulations.
**Qiankun Wang** is a Ph.D. candidate at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University. He works on MD simulation and machine learning in bioinformatics.
**Dennis Russell Salahub** is a full professor at the Department of Chemistry, University of Calgary, Fellow Royal Society of Canada, and Fellow of the American Association for the Advancement of Science.
**Yi Xiong** is an associate professor at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University. His main research interests focus on machine learning algorithms, and their applications in the protein sequence-structure–function relationship and biomedicine.
**Dong-Qing Wei** is a full professor at the School of Life Sciences and Biotechnology, Shanghai Jiao Tong University. His main research areas include structural bioinformatics and biomedicine.
**Submitted:** 23 May 2020; **Received (in revised form):** 6 August 2020

## Introduction

In order to reduce drug development cost, drug discovery (i.e., to find new candidate drugs) and drug repositioning (i.e., to find new indications for existing drugs) are two important strategies [1]. An important step to achieve these goals is to predict drug-target interactions (DTIs). In recent years, a large number of studies have applied the popular machine learning technology to realize intelligent medical treatments, which has accelerated the process of drug development to a certain extent.

Over the past decade, a great number of computational methods have been developed for the prediction of DTIs. There are some review articles [2–7] summarizing the progress of machine learning methods in the DTI prediction task, and the binary classification method is an important branch. For the binary classification methods [8–41], the drug-target pairs and whether or not the interactions exist between them are regarded as samples and labels, respectively. In addition to the binary classification methods, there exist network inference methods [42–55], matrix factorization methods [56–64], kernel-based methods [65–69], the restricted Boltzmann machine method [70], the collaborative filtering method [71], the clustering method [72], and the label propagation method [73], etc. It is worth noting that many of these latter methods can be considered as binary classification methods. For example [8], the network inference method regards the DTI prediction task as a bipartite network inference problem, and infers missing edges to achieve DTI 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.

The binary classification methods are trained on a benchmark data set which consists of positive and negative samples. If the unknown DTIs are treated as negative samples, it can bring noise since some unknown DTIs may be experimentally verified as positive DTIs in the future [74]. Moreover, following the multi-target multi-drug paradigm, a drug can interact with more than one target protein, and a target protein can interact with more than one drug. Therefore, the drug-target interaction prediction can be formulated as a multi-label classification task. From the machine learning point of view, the binary classification models do not consider the possible correlations among the labels, which may contain crucial information to increase the precision of the predictions [75].

To overcome 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. In multi-label classification, $m$ drugs (or $n$ targets) are regarded as samples, and $n$ targets (or $m$ drugs) are considered as labels. The samples (i.e., drugs or targets) are characterized as the input feature vectors. Then a multi-label learning algorithm is used to predict drug targets (or drugs that can interact with the specific target). The experimental results in this study demonstrated that it outperforms the traditional binary classification models, and its speed is much higher than that of the binary classification method, especially for large data sets. Until now, there are few applications to explore multi-label learning in the DTI prediction problem. DrugE-Rank [76] is a method using the 'Learning To Rank' paradigm to model the DTI prediction problem as a multi-label task. A study [77] uses multi-task deep neural networks for drug targets prediction, and firstly uses extended connectivity fingerprints with radius 12 as the drug representation. 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 used $k$-means for label division.

Moreover, the gold standard data set currently used in the field of DTI prediction is the data set collected by Yamanishi et al in 2008 [78], named Yamanishi_08. Over the past 12 years, a large number of new DTIs have been discovered, but they were not fully explored as training samples. As is well known, positive samples (i.e., DTIs) are essential for model construction. The incompleteness 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 predictions and the actual results. For this point, Keum and Nam [11] updated these data sets 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 new targets should also be considered.

This study updates the gold standard data set of drugs, targets, and DTIs as of December 2019. In addition, we propose the multi-label learning with community detection method for DTI 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 label space into multiple subspaces, then applies multi-label learning on each subspace, and finally performs DTI prediction. Comparisons with traditional machine learning methods and other previously published DTI prediction methods confirm the effectiveness of the proposed DTI-MLCD method. The workflow is shown in Figure 1.

## Materials and Methods

### Problem description

This study divides the DTI prediction problem into two subtasks: (a) drug repositioning, which predicts new targets for existing/old drugs, named $T_T$; (b) drug discovery, which predicts new drugs for known targets, named $T_D$. These two tasks are formulated as multi-label classification problems, described below and shown as Step 2 in Figure 1.

For task $T_D$, suppose $X_D = R^d$ and $Y_T = \{y_1, y_2, \ldots, y_p\}$ denote the $d$-dimensional drug instance space and the label space with $p$ possible target class labels. In $Y_T$, 0 means unknown or non-existing DTIs (i.e., negative data), and 1 means known DTIs (i.e., positive data). This task is to learn a function $f : X_D \rightarrow 2^{Y_T}$ from the multi-label training set $D = \left\{\left(x_{D,i}, y_{T,i}\right) | 1 \leq i \leq m\right\}$, where $m$ is the number of samples. For each sample $\left(x_{D,i}, y_{T,i}\right)$, $x_{D,i} \in X_D$, 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.

### Data sets

Yamanishi_08 is the data set proposed in 2008, which is widely used in the field of DTI prediction as the gold standard data set. It collects and integrates DTI data from KEGG BRITE [79], BRENDA [80], SuperTarget [81], and DrugBank [82] databases released in 2008. It consists of four DTI data sets according to the classes of protein targets, which are nuclear receptor (NR), G-protein-coupled receptor (GPCR), ion channel (IC), and enzyme (E). To update these data sets, we collect new drugs, new targets, and new DTIs using KEGG BRITE, UniProt [83], and DrugBank databases in this study. The workflow contains two

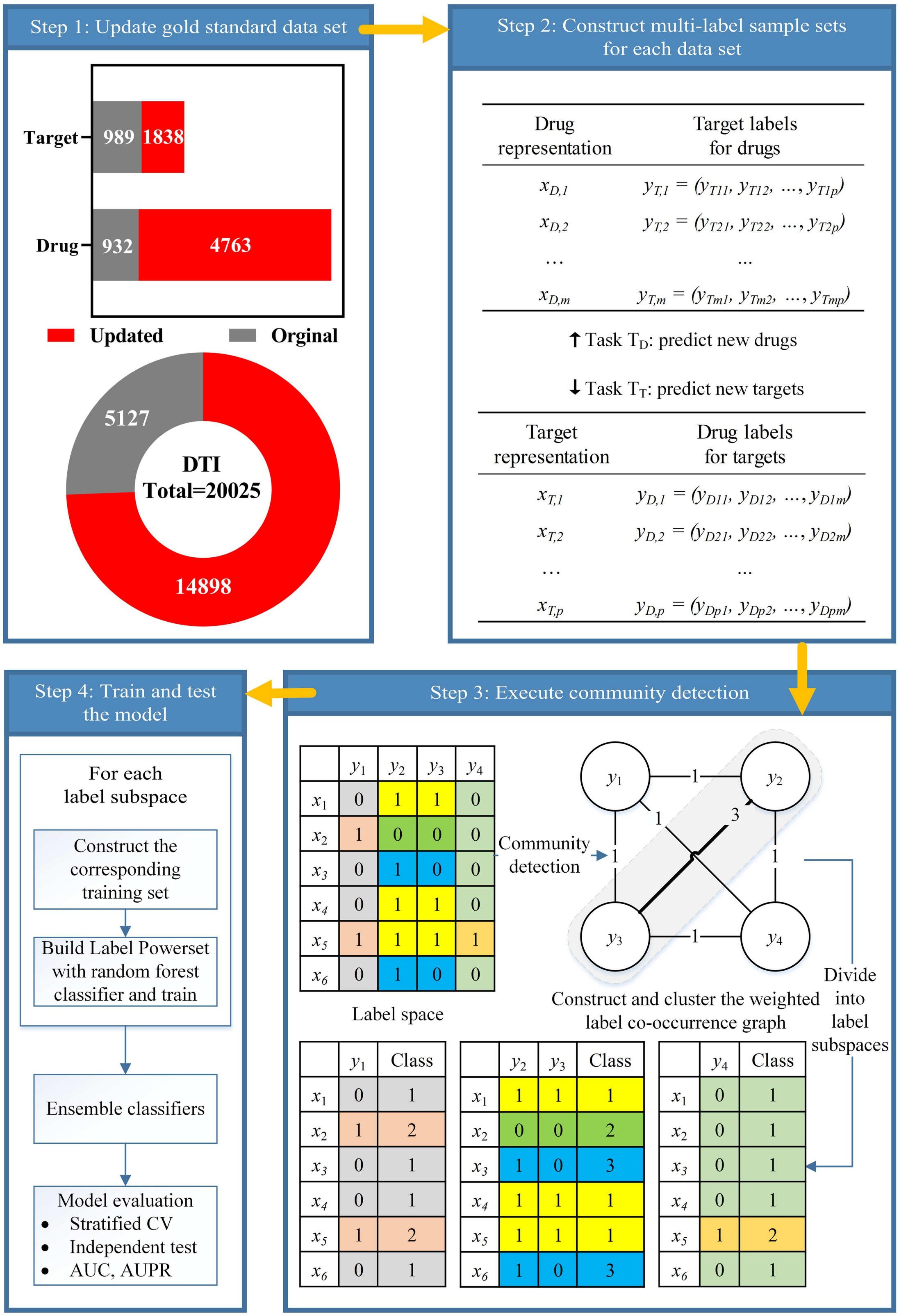



**Figure 1.** The flowchart of the proposed DTI-MLCD framework consists of four main steps: (1) update the gold standard data set, (2) construct multi-label sample set for drug and target data set, (3) execute community detection algorithm on a weighted label co-occurence graph to divide the label space into several subspaces, and (4) execute Label Powerset algorithm with random forest as base classifiers for each divided label subspaces. Then, assemble the classifiers to an integrated model. Finally, evaluate the model performance based on stratified cross-validation (CV), independent test, and AUC and AUPR.

**Table 1.** Statistics of the original and updated 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 | | $n_d$ | $n_t$ | $n_{interaction}$ | 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 |

main stages: 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, we 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, we 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 data set is freely accessible at https://github.com/a96123155/DTI-MLCD. Some statistics of the original gold standard and the newly updated four data sets are shown in Table 1.

In addition, the independent test set is built. For task $T_D$, 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. The independent test set of task $T_T$ is constructed similarly.

## Features

### *Drug representation*

Various types of representation can be used to describe drugs. In general, these can be categorized into two types: molecular descriptors (MDs), and 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 tools, this study treats them as different drug representations. The tools used in this study are CDK [84], Pybel [85], RDKit [86], and PaDEL [87]. 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 MD, called MD_MER. Currently, MFs are always divided into three categories [88]: (a) topological path-based fingerprint. The representative FP2 [89] (MF_FP2) is used in this study; (b) topological circular fingerprint. ECFP4 [90] (MF_EC4) and ECFP8 [90] (MF_EC8) are used as representatives; (III) substructure key-based fingerprint. MACCS [91] (MF_MAC) and PubChem fingerprint [92] (MF_PCP) are used because of their popularity in DTI prediction. Their dimensions are 1024, 2048, 2048, 167, and 881, respectively. In addition to the MDs and MFs, we also used the Word2vec-inspired feature [33] (W2V), which extracts semantic information from drug SMILES.

Further, we combine the above three types of features, since the complementarity among these three types of features may help enhance performance. In this process, we use the feature selection to obtain clean, highly complementary, and less redundant but combined features.

### *Target representation*

This study uses three types of sequence-derived features to represent protein targets. The first type of feature is Composition, Transition, and Distribution (CTD), which is represented as the 504-dimensional feature vector generated by using the PROFEAT web server [93]. The second type of feature (named PRO) is composed of 1437 default protein descriptors generated by PROFEAT. There are many studies using CTD [76] or PRO [12, 23, 24, 41] as the target representation method. 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 type of feature is the protein domain fingerprint (PDF), which is extracted from the PFAM v31.0 database [94]. For different data sets, we extracted different numbers of domains. The dimensions of feature vectors for the targets in NR, GPCR, IC, and E are 30, 61, 1404, and 2182, respectively. We also combined these three different types of features.

## Methods

Traditional binary classification (single-label learning) can be regarded as a degenerated version of multi-label learning since each sample is assigned to only one single label [95]. However, the generality of multi-label learning makes it 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. For this purpose, this study applies the community detection method from social networks to divide the whole label space into several smaller label subspaces. Next, each divided label subspace corresponds to a multi-label learning sub-problem, and multiple Label Powerset (LP) multi-label classifiers are joined to cover the entire label space. The base learner applied in LP is the random forest (RF) because of its simplicity, parallelism, and superior capabilities in DTI prediction tasks [12, 25, 27, 96]. 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 as Step 3 and Step 4 in Figure 1.

### Algorithms of multi-label learning

The development of multi-label learning algorithms is the key challenge in multi-label learning research, although there has been a boom in the various kinds of algorithms in the last decade. A simple categorization is described as follows.

The first category is the algorithm adaptation method, which works by fitting the existing algorithm to data and directly tackles the multi-label data. The representative algorithm is Multi-Label $k$-Nearest Neighbor (ML$k$NN) [97]. ML$k$NN is a lazy learning method based on the traditional $k$-Nearest Neighbor algorithm. It is now widely used in multi-label classification prediction tasks and has achieved satisfactory performance [98, 99]. The second category is the problem transformation method, which works by fitting data to a well-established algorithm and transforming multi-label learning problems into other learning techniques. Binary Relevance (BR) [100], Classifier Chains (CC) [101] and Label Powerset (LP) [102] are representative algorithms in this category. BR transforms the 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. However, there may exist correlations among labels in many fields, which is a limitation of the BR. CC is based on BR to exploit label correlations. 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 labels in the classifier chain is always random. Unlike BR and CC, LP transforms the multi-label learning task into a 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 unable to predict the unknown 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 shortcomings 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 [103], which can be seen as combining ensemble learning with LP. This is the design principle of random $k$-labelsets (RA$k$EL) [103]. 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 assembles 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 $k$, without considering the whole structure of the 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 [104]. Especially, the community detection method, which divides the label space in a data-driven manner, has well been applied to multiple benchmark data sets for multi-label learning [104]. Thus, this study discusses the application of five classic community detection algorithms in DTI prediction.

### Execution of community detection

The process of community detection is to find tightly connected community structures in complex network structures, that is, to discover clusters of nodes in the network [105]. 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 built based on the weighted co-occurrence graph derived from the training data.

*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 as Step 3 in Figure 1.

*Algorithms of community detection.* *The fast greedy algorithm (FGA)* [106] is a modularity-based algorithm based on the greedy approach. It treats each node as a singleton community at the beginning. Then, it iteratively searches the maximization of modularity. With each iteration, this method merges two communities to achieve the greatest contribution to modularity. When the modularity can no longer increase as the community merges, it is defined as converged.

The *multi-level algorithm (MLA)* [107] is also a modularity-based algorithm with a different greedy approach for the modularity optimization. At the beginning, a different community is assigned to each node. Then, by moving a node to the community where one of its neighbors is located, the greatest contribution to modularity is achieved. The above steps are repeated until modularity is not increased by any movement. Each community is considered as a single node, and then the process enters the next level. When there is only one node or the modularity can no longer be increased, the algorithm will stop.

The *label propagation algorithm (LPA)* [108] is a diffusion-based algorithm based on the graph semi-supervised learning algorithm, which simulates the diffusion of flow on a network through the diffusion of labels. At the beginning, each node is assigned a unique label (or community). Next, the label of every node is updated iteratively with the majority label assigned to its neighbors. The update order for each iteration is random. The convergence criterion of the algorithm is reached when all node labels are consistent with the most frequent labels in their neighborhood.

The *walk trap algorithm (WTA)* [109] is a node similarity-based algorithm based on random walks. One intuition is that when performing short distance random walks on a graph, it is easy to fall into the same community. At the beginning, each node is considered as a community. Then the random walk distance between all communities with connected edges is calculated. Next, two communities that are connected and have the shortest random walk distance are merged. The above steps are repeated until all nodes are put into the same community.

The *infomap algorithm (IMA)* [110] is a compression-based algorithm based on random walks. It believes that a good community division should make the average description length of the information 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.

**Table 2.** The merits, demerits and computational complexity of five community detection algorithms for the network with $N$ nodes and $E$ edges

| Algorithm | Merits | Demerits | Computational complexity |
|---|---|---|---|
| FGA | Fast. | Resolution-limit and coarse results, usually used as a first approximation. | $O(N\log^2(N))$ |
| WTA | Stable performance in small and large networks. | Slower, inaccurate compared to IMA and MLA. | $O(EN^2)$ |
| LPA | Simple, performs accurate in small networks. It scales computing time better on network size in log–log scale. | Large variance with unstable results, requires large number of initializations and slow, inaccurate in large networks. | $O(E)$ |
| MLA | Faster and have reasonable computation speeds on large networks. Relatively accurate in small and large networks. | Likely to provide the wrong number of communities for large networks. | $O(N\log N)$ |
| IMA | Theoretically sound and accurate in small networks. | Slow especially in large networks. Very likely to provide the wrong number of communities for large networks. | $O(E)$ |

Obviously, these algorithms are implemented based on different definitions of the community [111]. In this study, we tested these five typical community detection algorithms and summarized their advantages, disadvantages and computational complexity through related researches [104, 111–113] (Table 2).

### Performance evaluation

The performance evaluation metrics of multi-label learning are much more complex than binary classification [95]. Following previous research, this study adopts AUC and AUPR as performance evaluation metrics that are convenient for comparison with other methods. AUC is the area under the receiver operating characteristic curve based on different recall and false positive rate under the condition of different classified cutoff values. AUPR is the area under the precision-recall curve based on different precision and recall under the condition of different classified cutoff values. It is worthwhile to note that AUPR is a reliable metric as a severe punishment on false positive instances for highly imbalanced data. Therefore, the discussion in this article focuses on AUPR.

### Stratified cross-validation (SCV)

Cross-validation is a typical method to do model selection. For multi-label data, many labels have class imbalance characteristics [114] that each data set has a large number of label sets, and most label sets only contain a small number of samples (Table 3). In this case, the random partitioning strategy used in standard cross-validation may result in some labels without positive samples in a divided subset. Such a subset will not only affect the accuracy of the model, but may also cause a computational error.

To overcome the above dilemma, a stratified sampling strategy in cross-validation is a proven solution [114, 115], called stratified cross-validation (SCV). Furthermore, the 10-fold SCV has proven to be the best method in model selection from the perspective of statistical inference [115]. To ensure the confidence of the results, we performed 5 simulations on 10-fold SCV using different random seeds.

### Hypothesis test

When comparing multiple algorithms on a set of data sets, Demšar [116] recommends using the non-parametric Friedman rank test [117, 118] which is based on a ranking algorithm. 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 [119], named Friedman-Nemenyi test. This method can indicate whether there is a significant difference between the two algorithms based on the significance level $\alpha$.

## Results and Discussion

### Selecting drug representation

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

To achieve this goal, an experiment is conducted on the basic learning algorithm of LP for each updated data set, and the same parameter settings were used. The AUPR and AUC are shown in Table 4. However, AUPR is the focus as it is 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 it describes more information. For MFs, MF_EC4 is the best MF among all four data sets, and it has been proved that it is sufficient to describe molecules [120]. Further, the result reveals that 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. Table 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. For NR and GPCR, MF_EC4 was used. For IC and E, we used the combination of MF_EC4 and MD_MER.

### Selecting target representation

We have adopted the same strategy as for the drugs, that is, there is no best target representation method, only the most suitable feature representation in a specific situation. Therefore,

**Table 3.** Statistics for labels of eight 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 |

**Table 4.** The performance among different drug representations

| Representations | AUC | | | | AUPR | | | |
|---|---|---|---|---|---|---|---|---|
| | NR | GPCR | IC | E | NR | GPCR | IC | E |
| W2V[a] | 0.9171 | 0.9570 | 0.8921 | 0.8577 | 0.5798 | 0.5748 | 0.5055 | 0.1879 |
| MD_PYB[b] | 0.9380 | 0.9454 | 0.8914 | 0.8426 | 0.6487 | 0.4229 | 0.4899 | 0.1874 |
| MD_CDK[b] | 0.9541 | 0.9555 | 0.9105 | 0.8471 | 0.7495 | 0.5893 | 0.6292 | 0.2854 |
| MD_RDK[b] | 0.9562 | 0.9733 | 0.9236 | 0.8810 | 0.7634 | 0.6992 | 0.6755 | 0.3581 |
| MD_PAD[b] | 0.9611 | 0.9604 | 0.9336 | 0.8552 | 0.7839 | 0.6163 | 0.7119 | 0.3939 |
| MD_MER[b] | 0.9614 | 0.9717 | 0.9338 | 0.8579 | 0.7888 | 0.7015 | 0.7189 | 0.3992 |
| MF_FP2[c] | 0.9581 | 0.9769 | 0.9275 | 0.8742 | 0.7814 | 0.7470 | 0.7032 | 0.3917 |
| MF_MAC[c] | 0.9560 | 0.9736 | 0.9226 | 0.8749 | 0.7662 | 0.7213 | 0.6966 | 0.3781 |
| MF_PCP[c] | 0.9626 | 0.9745 | 0.9302 | 0.8588 | 0.7971 | 0.7552 | 0.7008 | 0.3854 |
| MF_EC4[c] | 0.9614 | 0.9755 | 0.9261 | 0.8683 | 0.8082 | 0.7667 | 0.7056 | 0.3939 |
| MF_EC8[c] | 0.9612 | 0.9755 | 0.9261 | 0.8683 | 0.8081 | 0.7663 | 0.7056 | 0.3939 |
| EC4, W2V[d] | 0.9556 | 0.9744 | 0.9231 | 0.8672 | 0.7700 | 0.7303 | 0.6784 | 0.3785 |
| W2V, MER[d] | 0.9614 | 0.9718 | 0.9329 | 0.8682 | 0.7841 | 0.7010 | 0.7183 | 0.3819 |
| EC4, MER[d] | 0.9620 | 0.9736 | 0.9328 | 0.8688 | 0.7952 | 0.7157 | 0.7193 | 0.4099 |
| EC4, W2V, MER[d] | 0.9620 | 0.9742 | 0.9325 | 0.8701 | 0.7910 | 0.7183 | 0.7190 | 0.4089 |

[a]The word2vec-inspired feature, which extracts semantic information from drug SMILES.
[b]They are molecular descriptors (MDs) that generated by different tools.
[c]They are molecular fingerprints (MFs).
[d]They are feature combinations of the above three types of features. The EC4 is short for MF_EC4, MER is MD_MER, both of them are the best representations in MDs and MFs.

we also compare target representation methods for four updated data sets and select the most suitable features for each data set according to AUPR.

According to Table 5, 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 according to the highest AUPR. For NR, the most suitable target representation method is CTD. For IC, it is PDF. For GPCR and E, it is the combination of CTD and PDF.

## 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 assembles them into a multi-label learning method. We explain the superiority of DTI-MLCD from two perspectives.

The first is the comparison of label partitioning algorithms. For the 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 DTI prediction problem [75]. So we use $k$-means as the baseline 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. The silhouette coefficient [121] is a measure of label division quality to calculate the goodness of a clustering technique. The $k$ value that maximizes the silhouette coefficient will be used as the optimal number of clusters. Table 6 (task $T_D$) and Table 7 (task $T_T$) indicate that the community detection algorithm is superior to $k$-means. Further, to

**Table 5.** The performance among different target representations

| Representations | AUC | | | | AUPR | | | |
|---|---|---|---|---|---|---|---|---|
| | NR | GPCR | IC | E | NR | GPCR | IC | E |
| CTD[a] | 0.5752 | 0.7896 | 0.9320 | 0.8650 | 0.2704 | 0.3554 | 0.6790 | 0.3322 |
| PRO[a] | 0.5789 | 0.7928 | 0.9321 | 0.8647 | 0.2656 | 0.3490 | 0.6876 | 0.3472 |
| PDF[b] | 0.5713 | 0.7613 | 0.9451 | 0.8568 | 0.1227 | 0.2063 | 0.7342 | 0.5424 |
| CTD, PDF[c] | 0.5750 | 0.7950 | 0.9405 | 0.8894 | 0.2403 | 0.3591 | 0.7356 | 0.5330 |
| PRO, PDF[c] | 0.5801 | 0.7950 | 0.9366 | 0.8868 | 0.2594 | 0.3563 | 0.7312 | 0.5174 |

[a]They are descriptors obtained by PROFEAT.
[b]The protein domain fingerprint.
[c]They are feature combinations of the above two types of features.

**Table 6.** The results of the proposed methods and other classical machine learning methods for task $T_D$ (i.e., predicting new drugs)

| Algorithm | AUC | | | | AUPR | | | |
|---|---|---|---|---|---|---|---|---|
| | NR | GPCR | IC | E | NR | GPCR | IC | E |
| FGA[a] | 0.9613 | 0.9738 | 0.9349 | 0.8840 | 0.8135 | 0.7721 | 0.7184 | 0.4148 |
| IMA[a] | 0.9611 | 0.9766 | 0.9358 | 0.8768 | 0.8129 | 0.7765 | 0.7194 | 0.4165 |
| LPA[a] | 0.9611 | 0.9763 | 0.9345 | 0.8833 | 0.8135 | 0.7755 | 0.7179 | 0.4173 |
| MLA[a] | 0.9614 | 0.9745 | 0.9347 | 0.8833 | 0.8134 | 0.7734 | 0.7186 | 0.4165 |
| WTA[a] | 0.9611 | 0.9744 | 0.9355 | 0.8839 | 0.8129 | 0.7722 | 0.7187 | 0.4184 |
| *k*-means[b] | 0.9629 | 0.9754 | 0.9352 | 0.8771 | 0.8128 | 0.7731 | 0.7178 | 0.4040 |
| ML*k*NN[c] | 0.9363 | 0.9575 | 0.8356 | 0.7962 | 0.6699 | 0.6340 | 0.1644 | 0.0454 |
| BR[c] | 0.9622 | 0.9814 | 0.9372 | 0.8771 | 0.8115 | 0.7307 | 0.6914 | 0.4040 |
| CC[c] | 0.9610 | 0.9767 | 0.9346 | 0.8664 | 0.8109 | 0.7219 | 0.6845 | 0.3822 |
| LP[c] | 0.9614 | 0.9755 | 0.9328 | 0.8688 | 0.8082 | 0.7667 | 0.7193 | 0.4099 |
| RA*k*EL[c] | 0.9532 | 0.9735 | 0.9306 | 0.8736 | 0.8004 | 0.7724 | 0.7048 | 0.4034 |
| RF[d] | 0.9626 | 0.9754 | 0.9423 | 0.8983 | 0.8102 | 0.7730 | 0.7113 | 0.3238 |
| ERT[d] | 0.9616 | 0.9688 | 0.9314 | 0.8786 | 0.8102 | 0.7571 | 0.7049 | 0.3546 |
| GNB[d] | 0.6818 | 0.7037 | 0.5015 | 0.5273 | 0.3732 | 0.3730 | 0.4197 | 0.0054 |

[a]FGA, IMA, LPA, MLA, and WTA are community detection algorithms in the proposed DTI-MLCD method.
[b]*k*-means is the baseline clustering method that is compared with community detection algorithms.
[c]ML*k*NN, BR, CC, LP, and RA*k*EL are classical multi-label methods that are compared with DTI-MLCD.
[d]RF, ERT, and GNB are classical binary classification methods that are compared with DTI-MLCD.

illustrate the biological explanation of the proposed methods, Figure 2 visualizes the results of six data-driven label partitioning methods that were applied to the NR data set. 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, *k*-means obtains only 4 communities, and the community structure is very different from community detection algorithms.

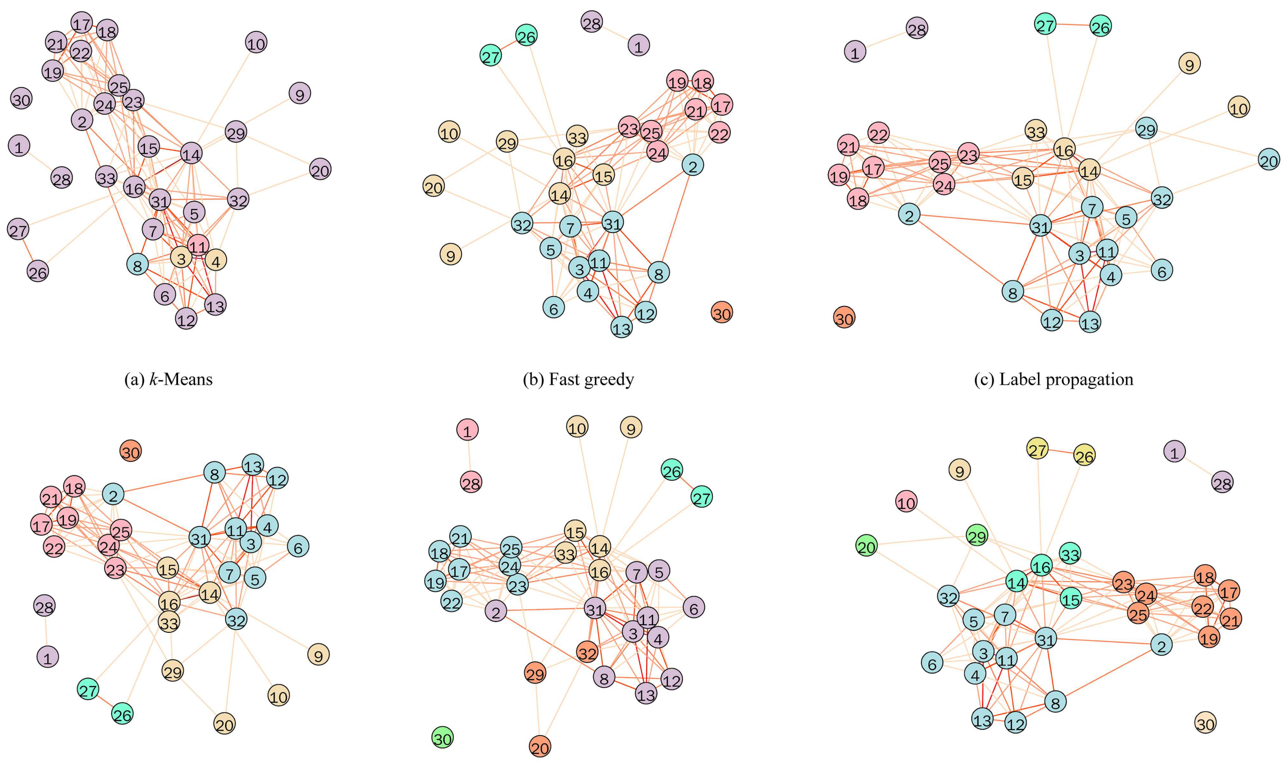
On the other hand, we discuss the pathway and classification of three communities through the KEGG database, and the details shown in Table 8. The (1, 28) and (26, 27) are communities obtained by all six algorithms, and (20, 29, 32) are only available in IMA. For each of the first two communities, the 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 significance for biological interpretation. This also confirms the classical assumption that similar targets tend to combine similar drugs.

The other aspect we discuss is to compare the DTI-MLCD algorithm with other classic machine learning algorithms, (a) multi-label algorithms: ML*k*NN, BR, CC, LP, and RA*k*EL, 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 6 (task $T_D$) and Table 7 (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 label sets and individual labels have very few samples (see Table 3 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, our experimental results demonstrated that its long calculation time and large required memory will make it difficult to achieve optimal performance through fine-tuning (Supplementary Table S1 and S2). Further, the Friedman-Nemenyi test with a significance level of 0.05 confirmed the significant differences among methods. All 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.

**Table 7.** The results of the proposed methods and other classical machine learning methods for task $T_T$ (i.e., predicting new targets)

| Algorithm | AUC | | | | AUPR | | | |
|---|---|---|---|---|---|---|---|---|
| | NR | GPCR | IC | E | NR | GPCR | IC | E |
| FGA[a] | 0.5715 | 0.8027 | 0.9489 | 0.8593 | 0.2311 | 0.3702 | 0.7468 | 0.5669 |
| IMA[a] | 0.5748 | 0.8002 | 0.9476 | 0.8598 | 0.2409 | 0.3683 | 0.7663 | 0.5669 |
| LPA[a] | 0.5759 | 0.8048 | 0.9459 | 0.8591 | 0.2494 | 0.3785 | 0.7518 | 0.5670 |
| MLA[a] | 0.5657 | 0.8062 | 0.9478 | 0.8640 | 0.2177 | 0.3759 | 0.7609 | 0.5677 |
| WTA[a] | 0.5745 | 0.8002 | 0.9463 | 0.8642 | 0.2401 | 0.3746 | 0.7574 | 0.5673 |
| *k*-means[b] | 0.5611 | 0.7893 | 0.9382 | 0.8639 | 0.2383 | 0.3693 | 0.7174 | 0.5668 |
| ML*k*NN[c] | 0.5470 | 0.7351 | 0.9094 | 0.8053 | 0.1811 | 0.2751 | 0.6414 | 0.3112 |
| BR[c] | 0.5617 | 0.7892 | 0.9382 | 0.8639 | 0.2352 | 0.3694 | 0.7174 | 0.5673 |
| CC[c] | 0.5647 | 0.7580 | 0.9183 | 0.8563 | 0.2360 | 0.2424 | 0.6475 | 0.5152 |
| LP[c] | 0.5752 | 0.7927 | 0.9403 | 0.8568 | 0.2704 | 0.3670 | 0.7429 | 0.5651 |
| RA*k*EL[c] | 0.5642 | 0.7902 | 0.9395 | 0.8640 | 0.2352 | 0.3714 | 0.7242 | 0.5670 |
| RF[d] | 0.6764 | 0.7610 | 0.9511 | 0.8775 | 0.2445 | 0.3104 | 0.7419 | 0.5652 |
| ERT[d] | 0.5804 | 0.7179 | 0.9459 | 0.8404 | 0.2632 | 0.3410 | 0.7650 | 0.5462 |
| GNB[d] | 0.4451 | 0.6566 | 0.5006 | 0.5347 | 0.2149 | 0.3770 | 0.3107 | 0.0035 |

[a]FGA, IMA, LPA, MLA, and WTA are community detection algorithms in the proposed DTI-MLCD method.
[b]*k*-means is the baseline clustering method that is compared with community detection algorithms.
[c]ML*k*NN, BR, CC, LP, and RA*k*EL are classical multi-label methods that are compared with DTI-MLCD.
[d]RF, ERT, and GNB are classical binary classification methods that are compared with DTI-MLCD.



(a) *k*-Means (b) Fast greedy (c) Label propagation

(d) Multi-level (e) Infomap (f) Walk trap

**Figure 2**. The label partition results that community detection algorithms and baseline *k*-means method applied in the label space of the nuclear receptor data set.

## Comparison to other DTI prediction methods on Yamanishi_08 data sets

We compare the proposed method against three state-of-the-art methods for DTI prediction. NetLapRLS [66], BLM-NII, and DDR [27]. NetLapRLS introduces the drug-target network information into the manifold Laplacian regularized least squares method which uses the concept of the bipartite local model. It avoids the dilemma caused by negative sample construction through a semi-supervised setting. BLM-NII exploits a bipartite local model with neighbor-based interaction profile inferring on a bipartite network of DTIs, which adds a preprocessing component to infer training data from neighbors' interaction profiles. DDR executes the graph-mining technique first to acquire the comprehensive feature vectors and then applies the random forest model by using different graph-based features

**Table 8.** The details for three communities A: (1, 28), B: (26, 27), and C: (20, 29, 32). The numbers represent the nodes in Figure 2

| Community node | Gene | Details |
|---|---|---|
| A1<br>A28 | LXRA<br>LXRB | Classification:<br>(1) Liver X receptor like receptor<br>(2) Cys4 thyroid hormone-like transcription factor<br>Pathway: Insulin resistance |
| B26<br>B27 | THRA<br>THRB | Classification:<br>(1) Cys4 thyroid hormone-like transcription factor<br>(2) Thyroid hormone like receptor<br>Pathway:<br>(1) Neuroactive ligand-receptor interaction<br>(2) Thyroid hormone signaling pathway |
| C20 | RORA | Classification:<br>(1) Cys4 thyroid hormone-like transcription factor<br>(2) Thyroid hormone like RAR-related orphan receptor |
| C29 | VDR | Classification:<br>(1) Cys4 thyroid hormone-like transcription factor<br>(2) Thyroid hormone like vitamin D3 like receptor |
| C32 | CAR | Classification:<br>(1) Cys4 thyroid hormone-like transcription factor<br>(2) Thyroid hormone like vitamin D3 like receptor<br>(3) constitutive androstane receptor |

**Table 9.** The results of the proposed methods and three existed DTI prediction methods for task $T_D$ (i.e., predicting new drugs)

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

[a]FGA, IMA, LPA, MLA, and WTA are community detection algorithms in the proposed DTI-MLCD method.
[b]BLM-NII, NetLapRLS, and DDR are existed DTI prediction methods that are compared with DTI-MLCD.

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 9, all the proposed methods in task $T_D$ outperform the three methods in terms of AUPR. For task $T_T$ (Table 10), the proposed methods outperform the three 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. These results indicate that all the proposed methods are performed better than the three other methods. Moreover, they are significantly better than DDR and NetLapRLS with significance levels of 0.05 and 0.1, respectively.

## Independent test

We conduct independent tests of the proposed DTI-MLCD method according to the data set before and after the update. The model for the independent test is trained on the Yamanishi_08 data set. The results are shown in Table 11 (task $T_D$) and Table 12 (task $T_T$).

**Table 10.** The results of the proposed methods and three existed DTI prediction methods for task $T_T$ (i.e., predicting new targets)

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

[a]FGA, IMA, LPA, MLA, and WTA are community detection algorithms in the proposed DTI-MLCD method.
[b]BLM-NII, NetLapRLS, and DDR are existed DTI prediction methods that are compared with DTI-MLCD.

**Table 11.** The results of independent tests on Yamanishi_08 data set for task $T_D$. The column Algorithm contains five community detection algorithms of the proposed DTI-MLCD method

| 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 | 0.4000 | 0.3012 | 0.1353 |
| LPA | 0.8157 | 0.9000 | 0.8257 | 0.8430 | 0.5334 | 0.3982 | 0.3013 | 0.1375 |
| MLA | 0.8174 | 0.8944 | 0.8246 | 0.8455 | 0.5331 | 0.3928 | 0.2776 | 0.1378 |
| WTA | 0.8174 | 0.8920 | 0.8230 | 0.8427 | 0.5331 | 0.3935 | 0.2890 | 0.1363 |

**Table 12.** The results of independent tests on Yamanishi_08 data set for task $T_T$. The column Algorithm contains five community detection algorithms of the proposed DTI-MLCD method

| 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 | 0.1077 |
| 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 | 0.3840 | 0.0076 | 0.2127 | 0.1048 |
| MLA | 0.8228 | 0.6255 | 0.7395 | 0.7339 | 0.3787 | 0.0080 | 0.2119 | 0.1071 |
| WTA | 0.8224 | 0.6080 | 0.7363 | 0.6814 | 0.3787 | 0.0074 | 0.2142 | 0.1052 |

## Conclusion

This study updated the gold standard data set Yamanishi_08, and proposed DTI-MLCD for DTI prediction, which is a new multi-label learning framework empowered by community detection. This framework explore five community detection algorithms to conduct label partitioning. This study conducted experiments on both Yamanishi_08 data set and our updated data set. On Yamanishi_08 data set, the DTI-MLCD shows higher performance than several existed methods. In our updated data set, DTI-MLCD is superior to classic machine learning algorithms. In addition, this study also constructed the independent tests on new and old data sets. On the other hand, the results of the five community detection algorithms used in this framework are superior to the baseline $k$-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.

### Key Points

- For drug discovery and drug repositioning, predicting DTIs is highly important, especially using computational methods such as machine learning methods. The dominant issues in the prediction of DTIs are the absence of positive samples and the unsatisfactory performance with large computational cost. We have tackled these issues.
- The quality of the benchmark data set is crucial to the performance of a DTI prediction method. Since the gold standard data sets often used in the previous studies was proposed in 2008, we updated the gold standard data set and added about 15,000 positive DTI samples in the present work.
- The proposed DTI-MLCD method is a multi-label classification framework. It transforms the DTI prediction problem from traditional binary classification into multi-label classification, and introduces the community detection method with the label correlations considered. For different data sets, the most suitable drug

(or target) representation method is different. Therefore, different feature representations are adopted for different tasks under different data sets in DTI-MLCD.
- DTI-MLCD achieves competitive performance with the binary classification method, and avoids its disadvantages, such as excessive computational load and missing information about the correlations among labels. Moreover, DTI-MLCD can predict a series of DTIs for a drug or target at once.
- DTI-MLCD is superior to other classic machine learning algorithms and some previously published DTI prediction methods, which indicates its usefulness and capability.

## Supplementary data

Supplementary data mentioned in the text are available to subscribers in *BRIBIO* online.

## Funding

This work was supported by the grants from the Key Research Area Grant 2016YFA0501703 of the Ministry of Science and Technology of China, the National Natural Science Foundation of China (Contract No. 61832019, 61503244), the Science and Technology Commission of Shanghai Municipality (Grant: 19430750600), the Natural Science Foundation of Henan Province (162300410060), as well as SJTU JiRLMDS Joint Research Fund and Joint Research Funds for Medical and Engineering and Scientific Research at Shanghai Jiao Tong University (YG2017ZD14, ZH2018QNA41, YG2019GD01, YG2019ZDA12).

## References

1. Breckenridge A. Clinical pharmacology and therapeutics. *BMJ* 1995;**310**:377–80.
2. Chen X, Yan CC, Zhang X, *et al*. Drug-target interaction prediction: databases, web servers and computational models. *Brief Bioinform* 2016;**17**:696–712.
3. Chen R, Liu X, Jin S, *et al*. Machine learning for drug-target interaction prediction. *Molecules* 2018;**23**:2208.
4. Zhang W, Lin W, Zhang D, *et al*. Recent advances in the machine learning-based drug-target interaction prediction. *Curr Drug Metab* 2019;**20**:194–202.
5. Anusuya S, Kesherwani M, Priya KV, *et al*. Drug-target interactions: prediction methods and applications. *Current Protein and Peptide Science* 2018;**19**:537.
6. Zhao Q, Yu H, Ji M, *et al*. Computational model development of drug-target interaction prediction: a review. *Current Protein and Peptide Science* 2019;**20**:492–4.
7. Maryam B, Elyas S, Kai W, *et al*. Machine learning approaches and databases for prediction of drug-target interaction: a survey paper. *Brief Bioinform* 2020.
8. Bleakley K, Yamanishi Y. Supervised prediction of drug-target interactions using bipartite local models. *Bioinformatics* 2009;**25**:2397–403.
9. Mei JP, Kwoh CK, Yang P, *et al*. Drug-target interaction prediction by learning from local information and neighbors. *Bioinformatics* 2013;**29**:238–45.
10. Li Z, Han P, You Z, *et al*. In silico prediction of drug-target interaction networks based on drug chemical structure and protein sequences. *Sci Rep* 2017;**7**:1–13.
11. Keum J, Nam H. SELF-BLM: prediction of drug-target interactions via self-training SVM. *PLoS One* 2017;**12**:e171839.
12. Ezzat A, Wu M, Li X, *et al*. Drug-target interaction prediction via class imbalance-aware ensemble learning. *BMC Bioinformatics* 2016;**17**.
13. Ding Y, Tang J, Guo F. Identification of drug-target interactions via multiple information integration. *Inform Sci* 2017;**418-419**:546–60.
14. Liu H, Sun J, Guan J, *et al*. Improving compound-protein interaction prediction by building up highly credible negative samples. *Bioinformatics* 2015;**31**:i221–9.
15. Peng L, Zhu W, Liao B, *et al*. Screening drug-target interactions with positive-unlabeled learning. *Sci Rep* 2017;**7**:1–17.
16. Meng FR, You ZH, Chen X, *et al*. Prediction of drug-target interaction networks from the integration of protein sequences and drug chemical structures. *Molecules* 2017; **22**.
17. Mousavian Z, Khakabimamaghani S, Kavousi K, *et al*. Drug-target interaction prediction from PSSM based evolutionary information. *J Pharmacol Toxicol Methods* 2016;**78**:42–51.
18. Tabei Y, Pauwels E, Stoven V, *et al*. Identification of chemogenomic features from drug–target interaction networks using interpretable classifiers. *Bioinformatics* 2012;**28**:487–94.
19. Yamanishi Y, Kotera M, Kanehisa M, *et al*. Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics* 2010;**26**:246–54.
20. Mahmud SMH, Chen W, Jahan H, *et al*. 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**:48699–714.
21. Zhang J, Zhu M, Chen P, *et al*. DrugRPE: random projection ensemble approach to drug-target interaction prediction. *Neurocomputing* 2017;**228**:256–62.
22. Rayhan F, Ahmed S, Shatabda S, *et al*. iDTI-ESBoost: identification of drug target interaction using evolutionary and structural features with boosting. *Sci Rep* 2017;**7**:1–18.
23. Ezzat A, Wu M, Li X, *et al*. Drug-target interaction prediction using ensemble learning and dimensionality reduction. *Methods* 2017;**129**:81–8.
24. Sharma A, Rani R. BE-DTI': ensemble framework for drug target interaction prediction using dimensionality reduction and active learning. *Comput Methods Programs Biomed* 2018;**165**:151–62.
25. Shi H, Liu S, Chen J, *et al*. Predicting drug-target interactions using lasso with random forest based on evolutionary information and chemical structure. *Genomics* 2019;**111**:1839–52.
26. Wang L, You ZH, Chen X, *et al*. RFDT: a rotation Forest-based predictor for predicting drug-target interactions using drug structure and protein sequence information. *Current Protein and Peptide Science* 2018;**19**:445–54.
27. Olayan RS, Ashoor H, Bajic VB. DDR: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics* 2018;**34**:1164–73.
28. Mousavian Z, Khakabimamaghani S, Kavousi K, *et al*. Drug-target interaction prediction from PSSM based evolutionary information. *J Pharmacol Toxicol Methods* 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**: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**:e1007129.
31. Lim J, Ryu S, Park K, *et al.* Predicting drug-target interaction using a novel graph neural network with 3D structure-embedded graph representation. *J Chem Inf Model* 2019;**59**:3981–8.
32. Tian K, Shao M, Wang Y, *et al.* Boosting compound-protein interaction prediction by deep learning. *Methods* 2016;**110**:64–72.
33. Zhang Y, Wang X, Kaushik AC, *et al.* SPVec: a Word2vec-inspired feature representation method for drug-target interaction prediction. *Front Chem* 2020;**7**:895.
34. Wen M, Zhang Z, Niu S, *et al.* Deep-learning-based drug-target interaction prediction. *J Proteome Res* 2017;**16**:1401–9.
35. Xie L, He S, Song X, *et al.* Deep learning-based transcriptome data classification for drug-target interaction prediction. *BMC Genomics* 2018;**19**:667.
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, *et al.* A computational-based method for predicting drug-target interactions by using stacked autoencoder deep neural network. *J Comput Biol* 2018;**25**:361–73.
38. Rayhan F, Ahmed S, Mousavian Z, *et al.* FRnet-DTI: deep convolutional neural networks with evolutionary and structural features for drug-target interaction. *arXiv preprint arXiv* 2018;**1806**:07174.
39. Gao KY, Fokoue A, Luo H *et al.* Interpretable Drug Target Prediction Using Deep Neural Representation. In: *International Joint Conference on Neural Networks*. 2018, p. 3371–7.
40. Chan KC, You Z-H. Large-scale prediction of drug-target interactions from deep representations. In: *International Joint Conference on Neural Networks*. 2016, p. 1236–43. IEEE.
41. Bahi M, Batouche M. Drug-target interaction prediction in drug repositioning based on deep semi-supervised learning. In: *IFIP International Conference on Computational Intelligence and Its Applications*. Springer, 2018, 302–13.
42. Cheng F, Liu C, Jiang J, *et al.* Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol* 2012;**8**:e1002503.
43. Chen X, Liu M, Yan G. Drug-target interaction prediction by random walk on the heterogeneous network. *Mol Biosyst* 2012;**8**:1970–8.
44. Fu G, Ding Y, Seal A, *et al.* Predicting drug target interactions using meta-path-based semantic network analysis. *BMC Bioinformatics* 2016;**17**:160.
45. Luo Y, Zhao X, Zhou J, *et al.* A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat Commun* 2017;**8**:1–13.
46. Wu Z, Cheng F, Li J, *et al.* SDTNBI: an integrated network and chemoinformatics tool for systematic prediction of drug-target interactions and drug repositioning. *Brief Bioinform* 2016;**18**:333–47.
47. Li Z, Huang M, Zhong W, *et al.* Identification of drug-target interaction from interactome network with 'guilt-by-association' principle and topology features. *Bioinformatics* 2016;**32**:1057–64.
48. Alaimo S, Pulvirenti A, Giugno R, *et al.* Drug-target interaction prediction through domain-tuned network-based inference. *Bioinformatics* 2013;**29**:2004–8.
49. Seal A, Ahn Y, Wild DJ. Optimizing drug-target interaction prediction based on random walk on heterogeneous networks. *J Chem* 2015;**7**: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**:520–31.
51. Emig D, Ivliev A, Pustovalova O, *et al.* Drug target prediction and repositioning using an integrated network-based approach. *PLoS One* 2013;**8**.
52. Lu Y, Guo Y, Korhonen A. Link prediction in drug-target interactions network using similarity indices. *BMC Bioinformatics* 2017;**18**:39.
53. Yu W, Yan Y, Liu Q, *et al.* Predicting drug-target interaction networks of human diseases based on multiple feature information. *Pharmacogenomics* 2013;**14**: 1701–7.
54. Alaimo S, Bonnici V, Cancemi D, *et al.* 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**:1–11.
55. Re M, Valentini G. Network-based drug ranking and repositioning with respect to DrugBank therapeutic categories. *IEEE/ACM Trans Comput Biol Bioinform* 2013;**10**: 1359–71.
56. Gönen M. Predicting drug-target interactions from chemical and genomic kernels using Bayesian matrix factorization. *Bioinformatics* 2012;**28**:2304–10.
57. Liu Y, Wu M, Miao C, *et al.* Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. *PLoS Comput Biol* 2016;**12**:e1004760.
58. Ezzat A, Zhao P, Wu M, *et al.* Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Trans Comput Biol Bioinform* 2017;**14**:646–56.
59. Zheng X, Ding H, Mamitsuka H *et al.* 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, p. 1025–33. ACM.
60. Hao M, Bryant SH, Wang Y. Predicting drug-target interactions by dual-network integrated logistic matrix factorization. *Sci Rep* 2017;**7**:1–11.
61. Peska L, Buza K, Koller J. Drug-target interaction prediction: a Bayesian ranking approach. *Comput Methods Programs Biomed* 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**:440.
63. Cobanoglu MC, Liu C, Hu F, *et al.* Predicting drug-target interactions using probabilistic matrix factorization. *J Chem Inf Model* 2013;**53**:3399–409.
64. Bagherian M, Kim RB, Jiang C, *et al.* Coupled matrix-matrix and coupled tensor-matrix completion methods for predicting drug-target interactions. *Brief Bioinform* 2020.
65. Jacob L, Vert JP. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics* 2008;**24**:2149–56.
66. Xia Z, Wu L, Zhou X, *et al.* Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Syst Biol* 2010;**4**(Suppl 2):S6.

67. van Laarhoven T, Nabuurs SB, Marchiori E. Gaussian interaction profile kernels for predicting drug-target interaction. *Bioinformatics* 2011;**27**:3036–43.
68. Shang F, Jiao LC, Liu Y. Integrating spectral kernel learning and constraints in semi-supervised classification. *Neural Processing Letters* 2012;**36**:101–15.
69. Nascimento ACA, Prudêncio RBC, Costa IG. A multiple kernel learning algorithm for drug-target interaction prediction. *BMC Bioinformatics* 2016;**17**:46.
70. Wang Y, Zeng J. Predicting drug-target interactions using restricted Boltzmann machines. *Bioinformatics* 2013;**29**: i126–34.
71. Koohi A, Prediction of drug-target interactions using popular Collaborative Filtering methods, In: *International Conference on Bioinformatics*. 2013, p. 58–61. IEEE.
72. Zhang X, Li L, Ng MK, *et al*. Drug-target interaction prediction by integrating multiview network data. *Comput Biol Chem* 2017;**69**:185–93.
73. Zhang W, Chen Y, Li D. Drug-target interaction prediction through label propagation with linear Neighborhood information. *Molecules* 2017;**22**:2056.
74. Chen H, Zhang Z. A semi-supervised method for drug-target interaction prediction with consistency in networks. *PLoS One* 2013;**8**:e62975.
75. Pliakos K, Vens C, Tsoumakas G. Predicting drug-target interactions with multi-label classification and label partitioning. *IEEE/ACM Trans Comput Biol Bioinform* 2019;1.
76. Yuan Q, Gao J, Wu D, *et al*. DrugE-rank: improving drug-target interaction prediction of new candidate drugs or targets by ensemble learning to rank. *Bioinformatics* 2016;**32**:i18–27.
77. Ceci M, Hollmén J, Todorovski L *et al*. *Machine Learning and Knowledge Discovery in Databases : European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II*. Cham: Springer International Publishing AG, 2017.
78. Yamanishi Y, Araki M, Gutteridge A, *et al*. Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 2008;**24**: i232–40.
79. Kanehisa M, Araki M, Goto S, *et al*. KEGG for linking genomes to life and the environment. *Nucleic Acids Res* 2007;**36**: D480–4.
80. Schomburg I, Chang A, Ebeling C, *et al*. BRENDA, the enzyme database: updates and major new developments. *Nucleic Acids Res* 2004;**32**:D431–3.
81. Günther S, Kuhn M, Dunkel M, *et al*. SuperTarget and matador: resources for exploring drug-target relationships. *Nucleic Acids Res* 2007;**36**:D919–22.
82. Wishart DS, Knox C, Guo AC, *et al*. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res* 2007;**36**:D901–6.
83. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res* 2019;**47**:D506–15.
84. Willighagen EL, Mayfield JW, Alvarsson J, *et al*. The chemistry development kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *J Chem* 2017;**9**:33.
85. O'Boyle NM, Morley C, Hutchison GR. Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit. *Chem Cent J* 2008;**2**:5.
86. Landrum G. rdkit/rdkit: 2019 03 4 (Q1 2019) Release. 2019, URL https://doi. org/10.5281/zenodo;3366468.
87. He Y, Liew CY, Sharma N, *et al*. PaDEL-DDPredictor: open-source software for PD-PK-T prediction. *J Comput Chem* 2013;**34**:604–10.
88. Riniker S, Landrum GA. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *J Chem* 2013;**5**:26.
89. O'Boyle NM, Banck M, James CA, *et al*. Open babel: an open chemical toolbox. *J Chem* 2011;**3**:33.
90. Rogers D, Hahn M. Extended-connectivity fingerprints. *J Chem Inf Model* 2010;**50**:742–54.
91. Durant JL, Leland BA, Henry DR, *et al*. Reoptimization of MDL keys for use in drug discovery. *J Chem Inf Comput Sci* 2002;**42**:1273–80.
92. Chen B, Wild D, Guha R. PubChem as a source of polypharmacology. *J Chem Inf Model* 2009;**49**:2044–55.
93. Li ZR, Lin HH, Han LY, *et al*. PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res* 2006;**34**:W32–7.
94. El-Gebali S, Mistry J, Bateman A, *et al*. The Pfam protein families database in 2019. *Nucleic Acids Res* 2019;**47**:D427–32.
95. Zhang M, Zhou Z. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 2014;**26**:1819–37.
96. Cao DS, Zhang L, Tan G, *et al*. Computational prediction of drug-target interactions using chemical, biological, and network features. *Qsar & Combinatorial Science* 2014;**33**:669–81.
97. Zhang M, Zhou Z. ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recognition* 2007;**40**:2038–48.
98. Liu G, Li G, Wang Y, *et al*. Modelling of inquiry diagnosis for coronary heart disease in traditional Chinese medicine by using multi-label learning. *BMC Complement Altern Med* 2010;**10**:37.
99. Zhang T, Dai H, Liu LA, *et al*. Classification models for predicting cytochrome P450 enzyme-substrate selectivity. *Molecular Informatics* 2012;**31**:53–62.
100. Zhang M, Li Y, Liu X, *et al*. Binary relevance for multi-label learning: an overview. *Front Comp Sci* 2018;**12**:191–202.
101. Read J, Pfahringer B, Holmes G, *et al*. Classifier chains for multi-label classification. *Machine learning* 2011;**85**:333.
102. SHEN C, Zhi-hai W, SUN Y. A multi-label classification algorithm based on label clustering. *Computer engineering & Software* 2014;**5**.
103. Tsoumakas G, Katakis I, Vlahavas I. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering* 2010;**23**:1079–89.
104. 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* 2016;**18**: 282.
105. Chen M, Kuzmin K, Szymanski BK. Community detection via maximization of modularity and its variants. *IEEE Transactions on Computational Social Systems* 2014;**1**:46–65.
106. Clauset A, Newman ME, Moore C. Finding community structure in very large networks. *Physical Review E* 2004;**70**:66111.
107. Blondel VD, Guillaume JL, Lambiotte R, *et al*. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008;**2008**(10):0–0.
108. Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 2007;**76**:36106.

109. Pons P, Latapy M. Computing communities in large networks using random walks. In: *International symposium on computer and information sciences*. 2005, p. 284–93. Springer.
110. Rosvall M, Axelsson D, Bergstrom CT. The map equation. *The European Physical Journal Special Topics* 2009;**178**:13–23.111.
111. Yang Z, Algesheimer R, Tessone CJ. A comparative analysis of community detection algorithms on artificial networks. *Sci Rep* 2016;**6**.
112. Orman GK, Labatut V, Cherifi H. Comparative evaluation of community detection algorithms: a topological approach. *Journal of Statistical Mechanics: Theory and Experiment* 2012;**2012**:8001.
113. Rotta R, Noack A. Multilevel local search algorithms for modularity clustering. *ACM Journal of Experimental Algorithms* 2011;**16**.
114. Sechidis K, Tsoumakas G, Vlahavas I. On the stratification of multi-label data. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2011, p. 145–58. Springer.
115. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *International Joint Conference on Artificial Intelligence*, Montreal, Canada. 1995, p. 1137–45.
116. Demsar J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 2006;**7**:1–30.
117. 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**:675–701.
118. Friedman M. A comparison of alternative tests of significance for the problem of $m$ rankings. *Annals of Mathematical Statistics* **11**:86–92.
119. Nemenyi P. *Distribution-free multiple comparisons, unpublished Ph. D*', Ph. D. Dissertation, thesis. Princeton, New Jersey: Princeton University, 1963.
120. Rogers D, Hahn M. Extended-connectivity fingerprints. *J Chem Inf Model* 2010;**50**:742–54.
121. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 1987;**20**(1):53–65.