Briefings in Bioinformatics, 2018, 1–21

doi: 10.1093/bib/bby002
Paper

Computational prediction of drug–target interactions using chemogenomic approaches: an empirical survey

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

Corresponding author. Xiao-Li Li, Institute for Infocomm Research (I2R), A *Star, 1 Fusionopolis Way, #21-01 Connexis, Singapore 138632. E-mail: xlli@i2r.a-star.edu.sg

Abstract

Computational prediction of drug–target interactions (DTIs) has become an essential task in the drug discovery process. It narrows down the search space for interactions by suggesting potential interaction candidates for validation via wet-lab experiments that are well known to be expensive and time-consuming. In this article, we aim to provide a comprehensive overview and empirical evaluation on the computational DTI prediction techniques, to act as a guide and reference for our fellow researchers. Specifically, we first describe the data used in such computational DTI prediction efforts. We then categorize and elaborate the state-of-the-art methods for predicting DTIs. Next, an empirical comparison is performed to demonstrate the prediction performance of some representative methods under different scenarios. We also present interesting findings from our evaluation study, discussing the advantages and disadvantages of each method. Finally, we highlight potential avenues for further enhancement of DTI prediction performance as well as related research directions.

Key words: drug-target interaction prediction; machine learning

Introduction

In silico prediction of interactions between drugs and their target proteins is desirable, as it effectively complements wet-lab experiments that are typically costly and laborious. The newly discovered drug–target interactions (DTIs) are critical for discovering novel targets interacting with existing drugs, as well as new drugs targeting certain disease-associated genes.

Drug repositioning, for instance, is the reuse of existing drugs for novel indications, that is, existing drugs may be used to treat diseases other than those that they were originally developed for [1]. As existing drugs have already been

Ali Ezzat is a Ph.D. student at the School of Computer Engineering in Nanyang Technological University (NTU), Singapore. He received the B.Sc. degree in computer science from Ain Shams University, Egypt, in 2006 and the M.Sc. degree in bioinformatics from Nanyang Technological University, Singapore, in 2013. He also worked as an IT professional at United OFOQ, Egypt, from 2008 to 2011. His research interests include machine learning, data mining and bioinformatics.

Min Wu is currently a Research Scientist in the Data Analytics Department at the Institute for Infocomm Research (I2R) under the Agency for Science, Technology and Research (A*STAR), Singapore. He received the B.Eng. from the University of Science and Technology of China (USTC), China in 2006 and his Ph.D. degree from Nanyang Technological University, Singapore in 2011. His current research interests include machine learning, data mining and bioinformatics.

Xiao-Li Li is currently a department head at the Institute for Infocomm Research, A*STAR, Singapore. He also holds adjunct professor positions at the National University of Singapore and Nanyang Technological University. His research interests include data mining, machine learning, AI, and bioinformatics. He has served as a (senior) PC member/workshop chair/session chair in leading data mining related conferences (including KDD, ICDM, SDM, PKDD/ECML, WWW, IJCAI, AAAI, ACL and CIKM) and as an editor of bioinformatics-related books. Xiaoli has published more than 160 high quality papers and won best paper/benchmark competition awards.

Chee-Keong Kwoh received the bachelor's degree in electrical engineering (first class) and the master's degree in industrial system engineering from the National University of Singapore in 1987 and 1991, respectively. He received the PhD degree from the Imperial College of Science, Technology and Medicine, University of London, in 1995. He is currently an associate professor at the School of Computer Engineering, Nanyang Technological University (NTU). His research interests include data mining, soft computing and graph-based inference, bioinformatics and biomedical engineering. He is a member of the Association for Medical and Bio-Informatics, Imperial College Alumni Association of Singapore.

Submitted: 6 October 2017; **Received (in revised form):** 21 December 2017

extensively studied (e.g. their bioavailability and safety profiles), repositioning them would significantly reduce costs and accelerate the drug discovery process, which made drug repositioning a popular strategy for drug discovery [2]. One famous example of a repositioned drug is that of Gleevec (imatinib mesylate), which was originally thought to interact only with the Bcr-Abl fusion gene associated with leukemia. Nevertheless, Gleevec was later found to also interact with PDGF and KIT, eventually leading it to be repositioned to treat gastrointestinal stromal tumors as well [3, 4]. This is one of many drug repositioning success stories that exist in the literature [5–10]. As demonstrated in the example of Gleevec, a drug's promiscuity (i.e. interaction with multiple targets) may contribute to its polypharmacology (i.e. having multiple therapeutic effects), which is clear motivation for attempting to discover new DTIs for existing drugs.

On the other hand, there is also a large number of small-molecule compounds that have not been used as drugs yet and, for the majority of them, their interaction profiles with proteins are still unknown. For example, the PubChem database currently houses >90 million compounds, most of which have unknown interaction profiles [11]. Detecting interactions (with disease-associated genes and target proteins) for these compounds would be useful for new drugs, as this would help narrow down prospective drug candidates to work with in the drug discovery process [12]. Moreover, detecting such interactions may provide insight by discovering off-targets that can cause undesirable side effects [13]. Therefore, prediction of DTIs is of great importance; it is essential for drug repositioning, assists with drug candidate selection and helps detect side effects in advance.

While experimental wet-lab techniques exist for predicting such interactions, they involve tedious and time-consuming work. This is where computational methods prove useful, as they may be used to efficiently predict potential interaction candidates with reasonable accuracy, thus narrowing down the DTI search space to be investigated by their wet-lab counterparts.

Currently, there are three major categories of computational methods for predicting DTIs. The first category is the ligand-based approaches, which leverage the concept that similar molecules tend to share similar properties and usually bind similar proteins [14]. In particular, they predict interactions using the similarity between the proteins' ligands [15]. However, the prediction results of ligand-based approaches may become unreliable when the number of known ligands per protein is insufficient [16].

The second category is the docking approaches, which take the 3 D structures of a drug and a protein and then run a simulation to determine whether they would interact [17–19]. However, there are proteins for which the 3 D structure is not known, so docking cannot be applied to them. For example, many drug targets are membrane proteins [20] for which the prediction of the 3 D structure is still challenging [21]. In addition, dealing with a receptor protein's flexibility can be challenging, as a large number of degrees of freedom need to be considered in the calculations.

The third category is the chemogenomic approaches, which use information from both the drug and target sides simultaneously to perform prediction. An advantage of chemogenomic approaches is that they can work with widely abundant biological data to perform prediction. For example, the information used for prediction in [22] consisted of chemical structure graphs and genomic sequences for the drugs and targets, respectively, which are available and easy to obtain from publicly accessible online databases.

In this survey, we focus on reviewing the more popular third category, the chemogenomic methods. The survey starts by describing the kinds of data required to perform the prediction task as well as how they may be obtained. Next, we classify the chemogenomic methods into five types and aimed to provide an overview of all the important prediction methods that belong to each of these types. Furthermore, we choose representative methods for each of the five types below, present a comprehensive comparison among them and discuss the advantages and disadvantages for these methods.

1. **Neighborhood models.** Neighborhood methods predict the interaction profile for a drug (or a target) based on its nearest neighbors' interaction information.
2. **Bipartite local models.** Bipartite local models (BLMs) first perform two sets of predictions individually, namely, one from the drug side and one from the target side, and then aggregate these predictions to generate the final prediction scores for given drug–target pairs.
3. **Network diffusion models.** Network diffusion methods investigate graph-based techniques (e.g. Random Walk) for influence propagation in drug–target networks and predict novel DTIs.
4. **Matrix factorization models.** Matrix factorization first learns the latent feature matrices for drugs and targets from the DTI matrix, and then multiplies these two latent feature matrices to reconstruct the interaction matrix for prediction.
5. **Feature-based classification models.** Drug–target pairs in training data are represented as feature vectors, which are then fed into machine learning models [e.g. Random Forest, Support Vector Machines (SVMs)] for predicting novel interactions.

In previous surveys [23, 24], it was commonplace to separate the prediction methods into only two categories, similarity-based methods and feature-based methods. However, as more prediction methods were being proposed by researchers, we decided to further divide the similarity-based methods into four categories, each with their unique characteristics. The new categorization of chemogenomic methods was found to be convenient and useful when representative methods were chosen from each of the categories and compared with each other in cross-validation (CV) experiments whose results are presented later in this study. Conclusions were drawn regarding the advantages and disadvantages of the prediction methods and their corresponding categories, which is useful information for practitioners as well as newcomers to the field.

Compared with previous reviews on this topic of DTI prediction [23–26], our survey is more comprehensive and up-to-date regarding the chemogenomic methods for predicting DTIs. In addition, we provide a novel categorization for the different chemogenomic approaches. Moreover, we describe the kinds of data that may be used in chemogenomic prediction tasks; however, note that we especially focus on listing software packages that generate features for representing drugs and targets (as opposed to online databases containing readily available information on DTIs). Furthermore, in the Supplementary Material of this study, we provide a comprehensive list of data sets that have been compiled by fellow researchers and used in previous work. We also perform an empirical comparison among various state-of-the-art methods from the different categories and discuss their advantages and limitations based on the results. One of the surveys, [23], also provided comparison results among different methods; however, as many new prediction methods have appeared since it was published in 2013, it is desirable to summarize more recent advanced methods. Finally, we discuss potential future trends as well as promising research directions that could be used to further improve DTI prediction.

In the recent review by Chen *et al*. [26], all online databases that store information on drugs and their targets were mentioned and described in detail (KEGG [27], DrugBank [28], etc.). Furthermore, a literature review on algorithms for DTI prediction was provided where the different algorithms are described and discussed. In addition to the algorithms, online Web servers for predicting interactions were described, and promising research directions in the field of drug discovery have been discussed as well. Our survey is similar to [26] in terms of reviewing the state-of-the-art methods and providing a list of potential future research directions. However, we provide a different categorization of the various prediction methods, and the future research directions proposed here differ from and complement those discussed in [26]. Finally, from the data perspective, while we do not focus on the databases from which data can be obtained, we make an effort here to list the different software packages that may be used to generate further descriptors for drugs and targets. We also provide, as Supplementary Material, a list of data sets that have been used in previous efforts in DTI prediction.

While targets exist in multiple forms, this survey primarily considers protein targets. As such, unless otherwise stated, all targets being referred to in this work are proteins.

The rest of this survey is organized as follows. ‘Data representation and types’ section first introduces the data for representing drugs, targets and their interactions. Then, ‘Methods’ section presents our novel categorization for various prediction methods in details. Next, ‘Empirical evaluation’ section demonstrates the empirical comparison results for various methods on benchmark data. Finally, ‘Avenues for improvement and further research’ section discusses future directions for DTI prediction.

Data representation and types

To train a classifier for predicting DTIs, a list of known DTIs is required. In other words, we want to predict which drugs and targets interact or not based on existing training data. Data for representing the drugs and targets involved are also needed. These required data are described in more detail below. Furthermore, we provide as Supplementary Material a complete list of publicly accessible data sets that have been used in previous efforts for predicting DTIs. An overview of a typical DTI prediction task is given in Figure 1.

Interaction data

Information on known DTIs needs to be gathered, as a classifier will be trained on these known interactions to predict the new interactions. Such information can be found in publicly available online databases that store information on drugs and their known targets. Examples of databases that have been used in previous work include KEGG [27], DrugBank [28], ChEMBL [29] and STITCH [30] (see [26] for an exhaustive list of such databases). The interaction data gathered from these databases are usually formatted into an interaction (adjacency) matrix between drugs and targets. This matrix corresponds to a bipartite graph where nodes represent drugs and targets, and edges connect drug–target pairs that interact.

Drug and target data

Types of data that are available for drugs and may be used for training DTI classifiers include—but are not limited to—graphical representations of drugs’ chemical structures [31], side effects [32], Anatomical Therapeutic Chemical (ATC) codes [33] and gene expression responses to drugs [34]. Other forms of data may further be extracted from the chemical structure graphs of drugs including substructure fingerprints as well as constitutional, topological and geometrical descriptors among other molecular properties (e.g. via the Rcpi [35], PyDPI [36] or Open Babel [37] packages).

As for targets, available data that can be obtained include genomic sequences [38], Gene Ontology (GO) information [39], gene expression profiles [40], disease associations [41] and protein–protein interaction (PPI) network information [42, 43] among others. Further, information may also be extracted from protein sequences, including amino acid composition, CTD (composition, transition and distribution) and autocorrelation descriptors (e.g. via the PROFEAT Web server [44]).

protein

Methods

Many (chemogenomic) DTI prediction methods have been developed over the past decade. We briefly describe them here

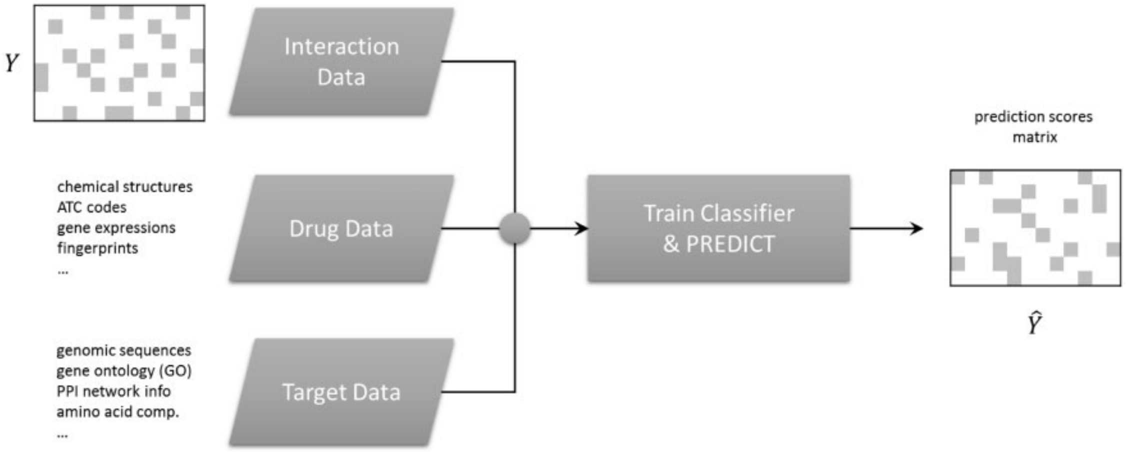

Figure 1. Flowchart of a standard DTI prediction task using a chemogenomic prediction method.

and categorize them based on the techniques that they use for prediction. Table 1 demonstrates our summarized categories of different methods for predicting DTIs.

In sections 'Neighborhood', 'Bipartite local models', 'Network diffusion' and 'Matrix factorization', the input data for the methods below consist of an interaction matrix $Y \in \mathbb{R}^{n \times m}$ showing which drugs and targets interact, a similarity matrix $S_d \in \mathbb{R}^{n \times n}$ for drugs and a similarity matrix $S_t \in \mathbb{R}^{m \times m}$ for targets. In section 'Feature-based classification', the similarity matrices are replaced by drug and target feature matrices, $F_d \in \mathbb{R}^{n \times p}$ and $F_t \in \mathbb{R}^{m \times q}$, for representing the drugs and targets, respectively.

Neighborhood

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

Nearest Profile and Weighted Profile

Nearest Profile and Weighted Profile are two methods that were introduced in [22]. Nearest Profile infers the interaction profile of a new drug or target from its nearest neighbor (i.e. the drug or target most similar to it). For example, the Nearest Profile for a new drug d_i is computed as:

$$\widehat{Y}(d_i) = S_d(d_i, d_{nearest}) \times Y(d_{nearest}), \tag{1}$$

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

$$\widehat{Y}(d_i) = \frac{\sum_{j=1}^{n} S_d(d_i, d_j) \times Y(d_j)}{\sum_{j=1}^{n} S_d(d_i, d_j)}. \tag{2}$$

In both of these methods, predictions from both the drug and target sides are averaged to obtain the final predictions.

Similarity Rank-based Predictor

Similarity Rank-based Predictor (SRP) [45] computes two tendency indices for each drug–target pair: one for the likelihood that it would interact and one for the likelihood that it would not. Given a drug d_i and a target t_j, its 'tendency-to-interact' index is computed as:

$$TI^+(d_i, t_j) = \sum_{p \in P^+(t_j)} \frac{S_d(d_i, d_p)}{R_d(d_i, d_p)}, \tag{3}$$

where $P^+(t_j) \subset \{d_1, d_2, \ldots d_n\}$ is the set of drugs that are known to interact with t_j, and $R_d(d_i, d_p)$ is the similarity rank of drug d_i to d_p among the n drugs. Another 'tendency-to-not-interact' index, $TI^-(d_i, t_j)$, is similarly computed as:

$$TI^-(d_i, t_j) = \sum_{q \in Q^-(t_j)} \frac{S_d(d_i, d_q)}{R_d(d_i, d_q)}, \tag{4}$$

where $Q^-(t_j)$ is the set of drugs known to not interact with t_j. An interaction likelihood score is then computed as the odds ratio:

$$C(d_i, t_j) = \frac{TI^+(d_i, t_j)}{TI^-(d_i, t_j)}. \tag{5}$$

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

Bipartite local models

BLMs perform two sets of predictions, namely, one from the drug side and one from the target side, and then aggregate these predictions to give the final prediction scores for the potential interaction candidates.

SVM-based BLMs

This pioneering effort [46] introduced the concept of BLM where a local model is trained for each drug (or target) to predict which targets (or drugs) would interact with it. In the case of [46], the local models were SVM classifiers. Predictions from the drug and target sides are then averaged to get the final results.

Table 1. The Categories of the different methods for predicting DTIs.

Categories	Methods	Category description
Neighborhood	Nearest Profile and Weighted Profile [22], SRP [45]	Neighborhood methods use relatively simple similarity functions to perform predictions
BLMs	Bleakley *et al.* [46], LapRLS [47], RLS-avg and RLS-kron [48], BLM-NII [49]	BLMs perform two sets of predictions, one from the drug side and one from the target side, and then aggregates these predictions to give the final prediction scores
Network diffusion	NBI [50], Wang *et al.* [51], NRWRH [52], PSL [53], DASPfind [54]	Network diffusion methods investigate graph-based techniques to predict new interactions
Matrix factorization	KBMF2K [55], PMF [56], CMF [57], WGRMF [58], NRLMF [59], DNILMF [60]	Matrix factorization finds two latent feature matrices that, when multiplied together, reconstruct the interaction matrix
Feature-based classification	He *et al.* [61], Yu *et al.* [62], Fuzzy KNN [63], Ezzat *et al.* [64], EnsemDT [65], SITAR [66], RFDT [78], PDTPS [81], ER-Tree [83], SCCA [84], MH-L1SVM [86]	Feature-based classification methods are those that need the drug–target pairs to be explicitly represented as fixed-length feature vectors

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

1. Ignoring t_j, a classifier is trained for d_i using the list of its known interactions with other targets (positive examples) as well as the list of targets not known to interact with d_i (negative examples). Interactions are labeled as $+1$, whereas noninteractions are labeled as -1. The trained classifier is used to predict for e_{ij}.
2. Ignoring d_i, a classifier is trained for t_j using the list of its known interactions with other drugs as well as the list of drugs not known to interact with t_j. Interactions are labeled as $+1$, whereas noninteractions are labeled as -1. The trained classifier then predicts for e_{ij}.
3. Predictions from both the drug and target sides (i.e. from both classifiers) are aggregated using the $\max(\cdot,\cdot)$ function.

Laplacian Regularized Least Squares

Laplacian Regularized Least Squares (LapRLS) [47] is another algorithm that is based on the BLM concept. The local models in LapRLS use regularized least squares to minimize an objective function that includes an error term as well as a graph regularization term. From the drug side, the objective function to be minimized is:

$$\min_{\alpha_d}(||Y - S_d\alpha_d||_F^2 + \beta_d Tr(\alpha_d^\top S_d L_d S_d \alpha_d)), \tag{6}$$

where $||\cdot||_F$ is the Frobenius norm, L_d is the normalized Laplacian obtained using S_d and β_d is a parameter. Note that the trace of a given matrix A is $Tr(A) = \sum_i A_{ii}$, and the expression $Tr(\alpha_d^\top S_d L_d S_d \alpha_d)$ is a graph regularization term, which helps model the manifold that is assumed to underlie the data. The manifold assumption (i.e. that data points lie on a low-dimensional nonlinear manifold) is one that is found to be usually true [67–69] and, therefore, modeling the manifold would be beneficial to the prediction performance. After obtaining α_d^* that minimizes the above function, the prediction matrix from the drug side is obtained as $\widehat{Y}_d = S_d\alpha_d^*$. A similar objective function is minimized from the target side to obtain $\widehat{Y}_t$, and then the final prediction matrix is obtained as:

$$\widehat{Y} = \frac{\widehat{Y}_d + \widehat{Y}_t}{2}. \tag{7}$$

Regularized Least Squares

Regularized Least Squares (RLS-avg) [48] uses kernel ridge regression to perform prediction. Furthermore, unlike the previous methods, Gaussian interaction profile (GIP) kernels are used to compute network similarity matrices for drugs and targets from the interaction matrix Y; network similarity between two drugs d_i and d_j is computed as $GIP_d(d_i, d_j) = \exp(-\gamma||Y(d_i) - Y(d_j)||^2)$ where γ is a parameter, and $Y(d_i)$ and $Y(d_j)$ are the interaction profiles of d_i and d_j, respectively. These network similarity matrices are then merged with S_d and S_t as:

$$K_d = \alpha S_d + (1-\alpha)GIP_d, \tag{8}$$

$$K_t = \alpha S_t + (1-\alpha)GIP_t, \tag{9}$$

where α is a parameter such that $0 \leq \alpha \leq 1$. K_d is a drug kernel that is formed via linear combination between the drug chemical similarity matrix S_d and the drug network similarity matrix GIP_d, whereas K_t is a target kernel that is formed via linear combination between the target sequence similarity matrix S_t and the target network similarity matrix GIP_t. The authors of this work hypothesized that incorporating network information (i.e. interaction information from the DTI network) into the prediction process as indicated above would lead to better prediction performance. Next, the prediction scores matrix is obtained as:

$$\widehat{Y} = \frac{1}{2}(K_d(K_d + \sigma I)^{-1}Y) + \frac{1}{2}(K_t(K_t + \sigma I)^{-1}Y^\top)^\top, \tag{10}$$

where σ is a regularization parameter. Note that Equation (10) shows predictions from both the drug and target sides being averaged to give the final scores.

Moreover, another algorithm was also introduced in [48], named RLS-kron, where the drug and target sides of the prediction have been merged into one by using the Kronecker product. Given that $K = K_d \otimes K_t$ is a kernel over drug–target pairs, the prediction scores matrix is obtained as:

$$vec(\widehat{Y}^\top) = K(K + \sigma I)^{-1} vec(Y^\top), \tag{11}$$

where $vec(Y^\top)$ is a column vector created by stacking the columns of $Y^\top$. As the matrix K would require too much memory and the computation of its inverse would be computationally intensive, the authors use a more efficient implementation [70] that is based on eigen decompositions.

Bipartite Local Models with Neighbor-based Interaction Profile Inferring

BLM algorithms exploiting local models achieved decent performance for DTI prediction. However, they had an outstanding issue that they are not able to train local models for drugs or targets that do not have any known interactions (i.e. new drugs or targets). To address this issue, Bipartite Local Models with Neighbor-based Interaction Profile Inferring (BLM-NII) [49], which is based on RLS-avg, introduces a preprocessing method denoted as NII to infer temporary interaction profiles for those novel drugs or targets.

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

$$Y(d_i) = \frac{\sum_{j=1}^{n} S_d(d_i, d_j) \times Y(d_j)}{\sum_{j=1}^{n} S_d(d_i, d_j)}, \tag{12}$$

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

$$\tilde{Y}(d_i) = \frac{Y(d_i) - \min(Y(d_i))}{\max(Y(d_i)) - \min(Y(d_i))}. \tag{13}$$

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

Regularized Least Squares with Weighted Nearest Neighbors

Another method based on RLS-kron [48] was introduced in [71] where RLS-kron was augmented with a preprocessing method, WNN, that has the same purpose as NII. For every new drug d_i, WNN is used to infer an interaction profile for it as:

$$Y(d_i) = \sum_{j=1}^{n} w_j Y(d_j), \tag{14}$$

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

Network diffusion

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

Network-based inference

To perform prediction, network-based inference (NBI) [50] applies network diffusion on the DTI bipartite network corresponding to the interaction matrix Y. Network diffusion is performed according to:

$$\widehat{Y} = WY, \tag{15}$$

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

$$W_{ij} = \frac{1}{\Gamma_{(i,j)}} \sum_{l=1}^{m} \frac{Y_{il} Y_{jl}}{k(t_l)}, \tag{16}$$

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

Heterogeneous graph inference

Another method that extends NBI is presented in [51]. In place of the basic bipartite network, network diffusion is performed on a heterogeneous network (as illustrated in Figure 2). The heterogeneous network augments the basic bipartite one by adding, between all pairs of drugs or targets, edges whose weights correspond to the pairwise similarities as indicated in S_d or S_t, respectively. Network diffusion in this method is done using the equation:

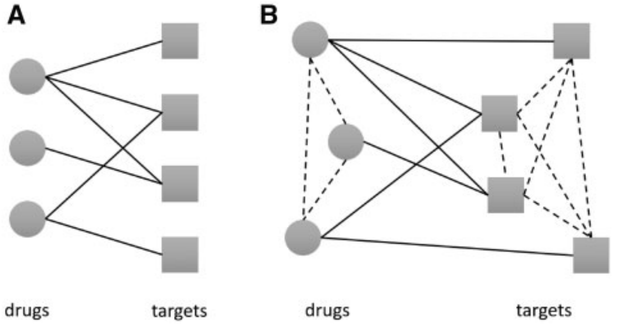

Figure 2. (A) Bipartite DTI network, (B) heterogeneous network that additionally includes drug and target pairwise similarities (the dashed lines).

$$Y^{(i+1)} = \alpha S_d Y^{(i)} S_t + (1-\alpha) Y^{(0)}, \tag{17}$$

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

$$S_d(d_i, d_j) = \frac{S_d(d_i, d_j)}{\sqrt{\sum_{k=1}^{n} S_d(d_i, d_k) \sum_{k=1}^{n} S_d(d_k, d_j)}}, \tag{18}$$

$$S_t(t_i, t_j) = \frac{S_t(t_i, t_j)}{\sqrt{\sum_{k=1}^{m} S_t(t_i, t_k) \sum_{k=1}^{m} S_t(t_k, t_j)}}. \tag{19}$$

Network-based Random Walk with Restart on the Heterogeneous network

Network-based Random Walk with Restart on the Heterogeneous network (NRWRH) [52] uses a heterogeneous network as well, and it predicts interactions via Random Walk on it. To perform Random Walk, NRWRH uses the transition matrix:

$$M = \begin{bmatrix} M_{DD} & M_{DT} \\ M_{TD} & M_{TT} \end{bmatrix}, \tag{20}$$

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

$$A^{(i+1)} = (1-r) M^{\top} A^{(i)} + r A^{(0)}, \tag{21}$$

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

Probabilistic Soft Logic

In addition to the above diffusion models, Probabilistic Soft Logic (PSL) [53] also uses a heterogeneous network similar to the one shown in Figure 2. As its name suggests, PSL uses probabilistic soft logic to perform prediction, i.e. it involves applying rules that use logical connectives, such as $\wedge$ (and), $\vee$ (or), and $\neg$ (not). Specifically, to determine if a drug and target interact, triad and tetrad relations (i.e. paths of length 3 and 4, respectively) involving the drug and target are searched for in the heterogeneous network and used for prediction of the potential interaction. Triad rules take the form of:

$$\begin{aligned} &\mathit{SimilarTarget}(t_1, t_2) \wedge \mathit{Interacts}(d, t_1) \\ &\quad \rightarrow \mathit{Interacts}(d, t_2) \end{aligned} \tag{22}$$

$$\begin{aligned} &\mathit{SimilarDrug}(d_1, d_2) \wedge \mathit{Interacts}(d_1, t) \\ &\quad \rightarrow \mathit{Interacts}(d_2, t) \end{aligned} \tag{23}$$

while tetrad rules take the form of:

$$\begin{aligned} &SimilarDrug(d_1, d_2) \wedge SimilarTarget(t_1, t_2) \\ &\wedge Interacts(d_1, t_1) \rightarrow Interacts(d_2, t_2). \end{aligned} \tag{24}$$

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

Furthermore, to avoid investigating the large number of all possible triad and tetrad relations, a technique called blocking is used beforehand where edges between pairs of drugs or targets (which correspond to pairwise similarities) are removed from the network if their weights (i.e. similarity values) are below some user-defined cutoff value.

Determine All Simple Paths, Find Interactions

Determine All Simple Paths, Find Interactions (DASPfind) [54] predicts an interaction between a drug *d* and a target *t* by finding all simple paths (i.e. that have no cycles) connecting them on the heterogeneous network. For the found simple paths, each path *p* has its score s_p computed by multiplying the weights on its edges. Finally, the scores are summed up to give the final prediction score for (*d*, *t*) as per the equation:

$$score = \sum_{p=1}^{z} (s_p)^{\alpha \times len(p)}, \tag{25}$$

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

Matrix factorization

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

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

As we are looking for missing interactions in the matrix Y, matrix factorization can be used as a matrix completion technique (i.e. the abovementioned transfer of interactions between drugs themselves and targets themselves), which makes it a good fit for the DTI prediction problem. An illustration of matrix factorization is given in Figure 3.

Kernelized Bayesian Matrix Factorization with Twin Kernels

Kernelized Bayesian Matrix Factorization with Twin Kernels (KBMF2K) [55] is, to our knowledge, the first in a number of methods that uses matrix factorization for predicting DTIs. It uses a Bayesian probabilistic formulation along with the concept of matrix factorization to perform prediction. Worded differently, it uses variational approximation to perform nonlinear dimensionality reduction, thus improving efficiency in terms of computation time.

As there are too many algorithmic details to mention, we only provide a minimal overview of the algorithm here. Please observe Figure 4 below which is inspired from a figure in [55].

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

Probabilistic Matrix Factorization

Probabilistic Matrix Factorization (PMF) [56] is another matrix factorization method that uses probabilistic formulations as well. Specifically, it models interactions via 'probabilistic linear models with Gaussian noise'. Unlike KBMF2K, however, it does not depend on or use similarity matrices between drugs or targets while performing prediction, and thus it achieves relatively lower performance than other matrix factorization techniques introduced here.

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

$$p(Y|A, B, \sigma^2) = \prod_{i=1}^{n} \prod_{j=1}^{m} [f(Y_{ij}|a_i b_j^\top, \sigma^2)]^{I_{ij}}, \tag{26}$$

where n and m are the numbers of drugs and targets, respectively, $f(x|\mu, \sigma^2)$ is the Gaussianly distributed probability density

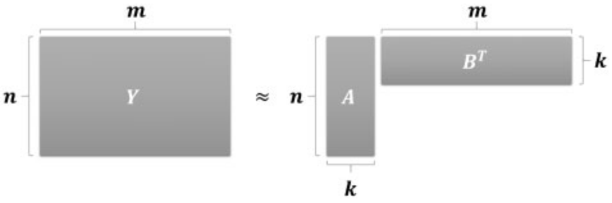

Figure 3. Illustration of matrix factorization. The goal is to find two latent feature matrices, A and B, that reconstruct the interaction matrix Y when multiplied together.

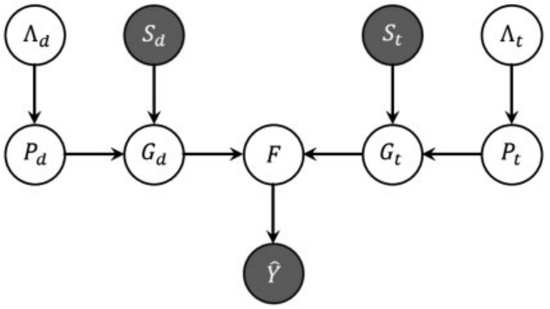

Figure 4. Minimal representation of the KBMF2K algorithm, which predicts DTIs from drug and target kernels, S_d and S_t.

function for x with mean μ and variance σ^2 and I_{ij} is an indicator function that is equal to 1 if Y_{ij} is known and 0 otherwise. Assuming zero-mean, spherical Gaussian priors on the latent vectors of A and B, the formula of the log-likelihood of A and B is derived using Bayes' rule as:

$$\begin{aligned}\ln(p(A,B|Y,\sigma^2,\sigma_A^2,\sigma_B^2)) = &-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}I_{ij}(Y_{ij}-a_ib_j^\top)^2 \\ &-\frac{1}{2\sigma_A^2}\sum_{i=1}^{n}a_ia_i^\top-\frac{1}{2\sigma_B^2}\sum_{j=1}^{m}b_jb_j^\top\end{aligned}. \quad (27)$$

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

Collaborative Matrix Factorization

Collaborative Matrix Factorization (CMF) [57] uses collaborative filtering for prediction. In addition to the standard goal of matrix factorization to find two matrices A and B where $3AB^\top \approx Y$, CMF proposes regularization terms to ensure that $AA^\top \approx S_d$ and $BB^\top \approx S_t$. CMF's objective function is given as:

$$\begin{aligned}\min_{A,B}\ &||W\odot(Y-AB^\top)||_F^2 + \quad \lambda_l(||A||_F^2+||B||_F^2) \\ &+ \quad \lambda_d||S_d-AA^\top||_F^2 + \quad \lambda_t||S_t-BB^\top||_F^2\end{aligned}, \quad (28)$$

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

Another variant for CMF, named MSCMF, involves using multiple similarities for both the drugs and targets. Besides the chemical structure similarity and genomic sequence similarity that are typically used for the drugs and targets, respectively, ATC similarity is also used for drugs, and GO and PPI network similarities are used for targets. The objective function for MSCMF is given as:

$$\begin{aligned}\min_{A,B}\quad &||W\odot(Y-AB^\top)||_F^2 \\ &+ \quad \lambda_l(||A||_F^2+||B||_F^2) \\ &+ \quad \lambda_d||\sum_{k=1}^{M_d}\omega_d^kS_d^k - AA^\top||_F^2 \\ &+ \quad \lambda_t||\sum_{k=1}^{M_t}\omega_t^kS_t^k - BB^\top||_F^2 \\ &+ \quad \lambda_\omega(||\omega_d||_F^2+||\omega_t||_F^2) \\ \\ \text{s.t.}\quad &|\omega_d| = |\omega_t| = 1\end{aligned}, \quad (29)$$

where M_d and M_t are the numbers of drug and target similarity matrices, respectively, and λ_ω is a parameter. ω_d and ω_t are weight vectors for linearly combining the drug and target similarity matrices, respectively. The fifth line in the above equation includes Tikhonov regularization terms for ω_d and ω_t, while the sixth line is a constraint that ensures the weights in each of ω_d and ω_t sum up to 1.

Weighted Graph Regularized Matrix Factorization

Weighted Graph Regularized Matrix Factorization (WGRMF) [58] is similar to CMF with the exception that WGRMF alternatively uses graph regularization terms to learn a manifold for label propagation. WGRMF's objective is given as:

$$\begin{aligned}\min_{A,B}\ &||W\odot(Y-AB^\top)||_F^2 + \quad \lambda_l(||A||_F^2+||B||_F^2) \\ &+ \quad \lambda_dTr(A^\top\tilde{\ell}_dA) + \quad \lambda_tTr(B^\top\tilde{\ell}_tB).\end{aligned} \quad (30)$$

where $Tr(\cdot)$ is the trace of a matrix, and $\tilde{\ell}_d$ and $\tilde{\ell}_t$ are the normalized graph Laplacians that are obtained from S_d and S_t, respectively. Before computing the graph Laplacians, S_d and S_t are sparsified by keeping only a predefined number of nearest neighbors for each drug and target, respectively. Kindly refer to [72–74] for more details on graph regularization.

The weight matrix W here has the same role as in CMF; by setting $W_{ij}=0$ for unknown drug–target pairs (i.e. test set instances), they would not contribute toward the prediction of interactions. The weight matrix W is important, as, otherwise, these test instances would count as noninteractions (i.e. as negative instances) and may unfavorably affect predictions.

Neighborhood Regularized Logistic Matrix Factorization

Neighborhood Regularized Logistic Matrix Factorization (NRLMF) [59] performs prediction via using the idea of logistic matrix factorization (LMF) [75]. In other words, it models the probability of an interaction between drug d_i and target t_j as the logistic function:

$$p_{ij} = \frac{\exp(a_ib_j^\top)}{1+\exp(a_ib_j^\top)}, \quad (31)$$

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

$$\begin{aligned}\min_{A,B}\quad &\sum_{i=1}^{n}\sum_{j=1}^{m}(1+cY_{ij}-Y_{ij})\ln[1+\exp(a_ib_j^\top)] - cY_{ij}a_ib_j^\top \\ &+\lambda_d||A||_F^2 \ + \ \lambda_t||B||_F^2 \\ &+\alpha Tr(A^\top\ell_dA) + \ \beta Tr(B^\top\ell_tB)\end{aligned}, \quad (32)$$

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

Dual-Network Integrated Logistic Matrix Factorization

Dual-Network Integrated Logistic Matrix Factorization (DNILMF) [60] can be considered an extension of NRLMF. DNILMF additionally incorporates network-based similarity in a way that is similar to how it is done in RLS-avg and RLS-kron. Unlike RLS-avg and RLS-kron, however, all kernels (i.e. similarity matrices) undergo a kernel diffusion step beforehand. For a drug (or target) kernel, a local similarity matrix is generated by keeping the similarities to the nearest *k* neighbors for each drug (or target) while discarding the rest, and then the local similarity matrix is diffused with the global similarity matrix over a number of iterations.

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

$$L_t(i,j) = \begin{cases} \dfrac{S_t(i,j)}{\sum_{k\in N_i} S_t(i,k)} & , \quad j \in N_i \\ 0 & , \quad otherwise \end{cases} ,$$
$$L_{t,GIP}(i,j) = \begin{cases} \dfrac{GIP_t(i,j)}{\sum_{k\in N_i} GIP_t(i,k)} & , \quad j \in N_i \\ 0 & , \quad otherwise \end{cases} \tag{33}$$

where N_i denotes the nearest neighbors of target t_j, and *k* is a parameter specifying the number of nearest neighbors to consider. Owing to the equations above, similarities to targets outside the list of nearest neighbors are set to 0. The local matrices, L_t and $L_{t,GIP}$, are then used to update the global matrices, S_t and GIP_t, as:

$$\begin{aligned} S_t^{(h+1)} &= (L_t) GIP_t^{(h)} (L_t)^\top \\ GIP_t^{(h+1)} &= (L_{t,GIP}) S_t^{(h)} (L_{t,GIP})^\top \end{aligned} \tag{34}$$

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

Predictions are obtained using the objective function:

$$\begin{aligned} \min_{A,B} \textstyle\sum_{ij}((1+cY_{ij}-Y_{ij})\ln[1+\exp(\alpha AB^\top + \beta K_d AB^\top + \gamma AB^\top K_t)] \\ -cY_{ij}(\alpha AB^\top + \beta K_d AB^\top + \gamma AB^\top K_t)) \\ +\frac{\lambda_d}{2}||A||_F^2 + \frac{\lambda_t}{2}||B||_F^2, \end{aligned} \tag{35}$$

which is based on the modified logistic function:

$$p = \frac{\exp(\alpha AB^\top + \beta K_d AB^\top + \gamma AB^\top K_t)}{1+\exp(\alpha AB^\top + \beta K_d AB^\top + \gamma AB^\top K_t)}, \tag{36}$$

where α, β, γ, λ_d and λ_t are parameters. Note that, in contrast to NRLMF's logistic function from Equation (31), the above logistic function incorporates information from the similarity matrices K_d and K_t (which were obtained via kernel diffusion). The matrices, *A* and *B*, that minimize the objective function in Equation (35) are used to obtain the final predictions matrix as $Y = AB^\top$.

As both NRLMF and DNILMF are based on logistic matrix factorization, their objective functions [from Equations (32) and (35)] resemble one another. However, while NRLMF uses graph regularization to make use of the similarity matrices S_d and S_t in prediction, DNILMF instead obtains the diffused kernels K_d and K_t, incorporates them into the logistic function and uses them in the objective function from Equation (35).

Feature-based classification

Feature-based classification methods are those that need drug–target pairs to be explicitly represented as fixed-length feature vectors. Given a drug feature vector $d = [d_1, d_2, \ldots, d_p]$ and a target feature vector $t = [t_1, t_2, \ldots, t_q]$, the drug–target pair would typically be represented by the concatenated feature vector $d \oplus t = [d_1, d_2, \ldots, d_p, t_1, t_2, \ldots, t_q]$. In addition to the feature vector, each drug–target pair has a label to show whether it is a known interaction (i.e. positive class) or a noninteraction (i.e. negative class). With the feature vectors and labels, various supervised machine learning methods can thus be developed for predicting DTIs as illustrated in Figure 5.

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

Incremental and forward feature selection

In [61], drugs were represented by a number of common functional groups that are found in drugs' chemical structures, while targets were represented by pseudo amino acid composition. An innovative feature selection procedure was additionally introduced in this work for the sake of improving the prediction performance by using a better feature set.

The feature selection procedure starts by ranking features using the mRMR (minimum Redundancy Maximum Relevance) algorithm [76]. Incremental feature selection is then applied on the ranked features, i.e. the ranked features are added to the selected feature set in order, one by one, until the prediction performance on a temporary validation set stops improving. Finally, the set of selected features is further filtered by applying forward feature selection to it. After the feature selection phase is complete, a nearest neighbor algorithm is then applied to obtain final predictions.

Random Forest and SVMs

Random Forest and SVM models were proposed for predicting interactions in [62]. Assuming that the data consist of n_d drugs and n_t targets, this means that there are $n_d \times n_t$ drug–target pairs in total. When the dimensionality of the data is high (i.e. drug–target pairs are represented by many features), it becomes challenging, if not infeasible, to use the entire data set of all drug–target pairs as training data to train a classification model. Therefore, the set of noninteractions, which is far bigger than the set of known interactions, is undersampled until its size is equal to that of the set of interactions.

Drug and target features were generated using the DRAGON (http://www.talete.mi.it/) and PROFEAT [44] packages, respectively. Drug features generated by DRAGON include

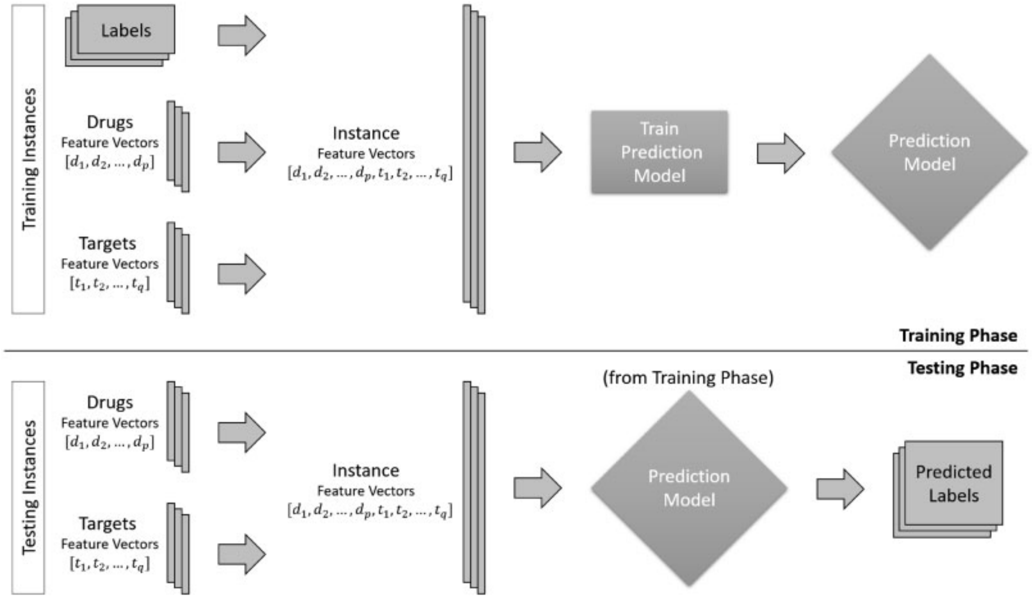

Figure 5. Illustration of how feature-based prediction models are created. In the training phase, feature vectors for the training instances (i.e. the drug–target pairs) are generated by concatenating the feature vectors of the involved drugs and targets. Along with their labels, the training instances are used to train the prediction model. In the testing phase, feature vectors are generated for the testing instances, and the prediction model (from the training phase) is used to predict for the testing instances.

constitutional and topological descriptors, eigenvalue-based indices and 2D autocorrelations among others. On the other hand, target features generated by PROFEAT include CTD and autocorrelation descriptors, amino acid composition and so on.

Fuzzy K-nearest neighbors

Fuzzy *K*-nearest neighbors (Fuzzy KNN) [63] models each training instance as belonging to two classes (i.e. positive and negative classes) with different membership values. For each test instance, its membership value to each of the two classes is computed by a kind of weighted average of its similarities to its nearest *K* neighbors, and the higher of the two values decides which class it belongs to. The drugs were represented by FP2 fingerprints that were generated using the Open Babel [37] package, while the targets were represented using pseudo amino acid composition.

Decision tree ensemble with oversampling

An ensemble technique was introduced in [64] to predict interactions. Drug descriptors were computed using the Rcpi package [35], whereas target descriptors were generated via the PROFEAT Web server [44]. Similar to the Random Forest used in [62], an ensemble of decision trees is trained, and feature subspacing is applied (i.e. a subset of the features is randomly sampled for each decision tree). However, in contrast to Random Forest, which performs bagging on the same sampled group of negatives, a different set of negatives is randomly sampled for each decision tree, which means better coverage of the negative class in the data and including more of it in the training process. In addition, clustering is used to look for small disjuncts in the interacting class that are then oversampled to reinforce them. This is to deal with an issue in the data known as the within-class imbalance.

Decision tree ensemble with dimensionality reduction

EnsemDT [65] is another ensemble technique that is similar to the one presented in [64]. However, EnsemDT does not oversample small disjuncts in the interacting class. Instead, it uses dimensionality reduction. That is, dimensionality reduction is applied to the drug and target feature vectors before concatenating them to form the instances. Three dimensionality reduction techniques were investigated, namely, Singular Value Decomposition (SVD), Partial Least Squares [77] and Laplacian Eigenmaps [69]. While dimensionality reduction is commonly used to improve the computational efficiency (i.e. reducing the running time), these techniques were found to improve the prediction performance as well.

Rotation Forest-based Predictor of Drug–Target Interactions

Rotation Forest-based Predictor of Drug–Target Interactions (RFDT) [78] uses yet another ensemble learning technique to predict DTIs. In particular, a variant based on Rotation Forest [79] was used. For each base classifier, the feature set is randomly divided into *K* roughly equal subsets (where *K* is a parameter). In other words, the feature matrix $X \in \mathbb{R}^{n \times p}$ is split into *K* submatrices such that each submatrix has around *p*/*K* columns where *n* is the number of instances, and *p* is the number of features. Bagging is then applied to the training set, that is a subset of the training examples is randomly sampled to form the training set for the current base classifier. Next, principal component analysis (PCA) is applied to each of the submatrices separately, and then the resulting features from each submatrix are combined to form a diagonal block matrix called the rotation matrix. Finally, the feature matrix *X* is multiplied by the rotation matrix, and the resulting matrix is used as the training set along with the corresponding labels to train the base classifier. This procedure is repeated for all base classifiers constituting the ensemble.

Using a rotation matrix (that is constructed by dividing the feature set into *K* randomized subsets) and bagging are both ways of injecting diversity into the ensemble. Increased diversity within the ensemble is known to improve the overall prediction performance [80].

In this study, PubChem fingerprints (i.e. binary vectors indicating presence or absence of 881 common substructures) are used to represent drugs. Targets, on the other hand, are represented using autocovariance vectors that were generated using the targets' genomic sequences; specifically, a position-specific scoring matrix (PSSM) was computed for each target (using its sequence), and then the PSSMs were used to generate autocovariance vectors for representing the targets.

Predicting Drug Targets with Protein Sequence

Predicting Drug Targets with Protein Sequence (PDTPS) [81] is similar to *RFDT* in that it makes use of PSSMs to represent targets. However, in place of autocovariance, it instead computes bi-gram probabilities from the PSSMs. In addition, PCA is later applied to reduce the dimensionality of the features. For prediction, PDTPS uses Relevance Vector Machines (RVMs). Experimental results showed that the proposed method was successful.

RVM [82] is a machine learning method that is functionally identical to SVM. However, unlike SVM, it uses Bayesian learning to make use of probabilistic formulations in prediction. Prediction models trained via RVM are typically sparse (i.e. compact and interpretable), while, at the same time, they are able to produce results that are comparable with (and exceed) those of SVM.

Extremely Randomized Trees

In [83], Extremely Randomized Trees (ER-Tree) are used to perform prediction. In regular Decision Tree-based ensembles, each Decision Tree follows certain rules for (i) selecting attributes to use for tree-splitting and (ii) determining cutoff points within the attributes. In ER-Tree, randomization is explicitly introduced into the training process by random selection of attributes and cutoff points. This explicit randomization helps strongly reduce the variance of the tree-based models, thus improving prediction performance. Furthermore, bagging is avoided (i.e. the entire training set is used) to keep the bias as low as possible.

In ER-Tree, for each base classifier, K attributes are chosen at random. Each of the K attributes then has a cutoff point randomly generated for it, that is each attribute a has its minimum and maximum values, a_{min} and a_{max}, determined, and the cutoff point is randomly generated from the interval $[a_{min}, a_{max}]$. The different K splits are then each evaluated by the formula:

$$Score(s, N) = \frac{2I_c^s(N)}{H_c(N) + H_s(N)}, \tag{37}$$

where N is the current tree node (i.e. before the split s), N_L and N_R are the left and right child nodes of N, respectively, $H_c(N)$ is the classification entropy at N, $H_s(N)$ is the split entropy and $I_c^s(N)$ is the mutual information of the split outcome and the classification. Specifically, $H_c(N)$, $H_s(N)$ and $I_c^s(N)$ are computed as:

$$\begin{aligned} H_c(N) &= -\sum_{i=1}^{C} p_i \, log_2 \, p_i, \\ H_s(N) &= -\left(\frac{|N_L|}{|N|} log_2 \frac{|N_L|}{|N|} + \frac{|N_R|}{|N|} log_2 \frac{|N_R|}{|N|}\right), \\ I_c^s(N) &= H_c(N) - \frac{|N_L|}{|N|} H_c(N_L) - \frac{|N_R|}{|N|} H_c(N_R). \end{aligned} \tag{38}$$

where C is the number of different classes (two in our case), and $|N|$ is the number of examples at node N. The split with the highest score is chosen for this iteration as:

$$s^* = \arg_{s_i} \max_{i=1...K} Score(s_i, N). \tag{39}$$

This step is recursively repeated for the two child nodes, N_L and N_R, and so on until this base classifier is trained. This procedure is repeated for all the base classifiers, forming the ensemble.

In terms of data representation, drugs are represented as PubChem fingerprints, while targets were represented using Pseudo Substition Matrix Representation (pseudo-SMR).

Similarity-based Inference of drug–TARgets (SITAR)

Unlike typical feature-based classification methods, which concatenate both drug and target feature vectors to represent drug–target pairs, Similarity-based Inference of drug–TARgets (SITAR) [66] represents each instance (drug–target pair) as a vector of its similarities to the positives in the data. In particular, the similarity between two drug–target pairs (d, t) and (d', t') is computed using the geometric mean as:

$$S((d, t), (d', t')) = S_d(d, d')^r \cdot S_t(t, t')^{(1-r)}, \tag{40}$$

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

Chemical substructures–protein domains correlation model

In [84], drugs are represented as PubChem fingerprints (binary vectors indicating the absence/presence of 881 common chemical substructures), while targets are represented as domain fingerprints (binary vectors indicating the absence/presence of 876 protein domains obtained from the Pfam database [85]).

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

$$\begin{aligned} &\max_{\alpha,\beta} \quad \alpha^\top D^\top Y T \beta \\ &s.t. \quad ||\alpha||_2^2 \le 1, \quad ||\beta||_2^2 \le 1, \\ &\quad ||\alpha||_1 \le c_1\sqrt{u}, \quad ||\beta||_1 \le c_1\sqrt{v} \end{aligned} \tag{41}$$

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

When used to predict DTIs, SCCA produced results that are comparable with those of SVM. However, unlike SVM, which is focused only on prediction, SCCA is an interpretable classifier that, having been trained, can be inspected for learned rules that may contain useful insights. As stated above, SCCA emphasizes learning sparse weight vectors, which makes it possible to inspect these weight vectors for biological insights; the nonzero elements in the learned weight vector would correspond to the most significant chemical structures and protein domains that govern DTIs.

SVMs and minwise hashing

In [86], drugs were represented as PubChem fingerprints (881 chemical substructures), and proteins were represented as domain fingerprints (4, 137 Pfam domains). Given a drug vector $\Phi(C)$ and a protein vector $\Phi(P)$, a compound–protein pair fingerprint $\Phi(C, P)$ is then obtained by the tensor product of $\Phi(C)$ and $\Phi(P)$ as:

$$\Phi(C, P) = \Phi(C) \otimes \Phi(P), \tag{42}$$

resulting in a binary vector that is 3 644 697 elements long. Dimensionality reduction is then achieved by applying minwise hashing [87] to the compound–protein fingerprints to convert them to compact fingerprints to make the algorithm scalable to large data sets.

Linear SVM is used as the classifier. Two variants have been considered: one with an L_2 regularization term (MH-L2SVM) and another with an L_1 regularization term (MH-L1SVM). The two variants were found to produce similar prediction performance. However, the learned weight vector from the MH-L1SVM is more interesting because the number of features extracted was much smaller than that of MH-L2SVM (i.e. less features to inspect for insights).

Finally, using the inverse operation of the minwise operation mentioned above, the weight vector learned using the compact fingerprints is converted into a final weight vector for the original fingerprint. This final weight vector is then inspected for biological interpretation.

Empirical evaluation

We performed a comprehensive empirical comparison among various methods, under three distinct CV settings in [25] as follows:

1. S1, where random drug–target pairs are left out as the test set;
2. S2, where entire drug profiles are left out as the test set; and
3. S3, where entire target profiles are left out as the test set.

S1 is the traditional setting for evaluation. Meanwhile, S2 and S3 are proposed to evaluate the ability of various methods to predict interactions for novel drugs and targets. Here, novel drugs and targets are those for which no interaction information is available. As such, additionally conducting experiments under S2 and S3 paints a fuller picture of how the different methods perform in various given situations. Illustrations of the different CV settings are provided in Figure 6.

In our experiments, we performed five repetitions of a 10-fold CV procedure under each of the above scenarios using AUPR [88] (area under the precision–recall curve) as the evaluation metric. That is, under each 10-fold CV procedure, the data set (the interaction data, specifically) is divided into 10 folds. The folds take turns being left out as the test set, and the prediction performance for each of them is evaluated in terms of AUPR. The computed AUPRs are then averaged to give the AUPR of the 10-fold CV. This process is repeated five times, and the AUPRs of the 10-fold CVs are averaged to give the final AUPR.

AUPR was used as the main evaluation metric in previous work in DTI prediction. Furthermore, in cases of class imbalance, the AUPR is more adequate because it severely penalizes highly ranked incorrect recommendations [89], which better reflects the aim of having accurate predictions at the top of the prediction lists. For these reasons, we use AUPR as the evaluation metric in our empirical comparison as well. In addition, in DTI prediction, the relative order of the labels is more important than the exact values of the prediction; thus, it makes more sense to use an evaluation metric that measures how well the different drug–target pairs are ranked.

Benchmark data set

Some of the most widely used data sets in the field of DTI prediction are those that are introduced in [22]. Specifically, they were four data sets concerning four different classes of target proteins, namely, enzymes (Es), ion channels (ICs), G protein-coupled receptors (GPCRs) and nuclear receptors (NRs). Interaction data were extracted from the KEGG database [27] (see Table 2 for some statistics on each of the data sets). In addition, each data set provides a drug similarity matrix S_d where the pairwise similarities between the drugs were computed using SIMCOMP [90] and a target similarity matrix S_t where the pairwise similarities between the targets are computed using normalized Smith–Waterman [91].

Selected methods

We include a subset of the methods mentioned in section ‘Methods’ such that the different categories are represented. As baseline methods, we selected the Nearest Profile and Weighted Profile from the neighbor-based methods. We further selected CMF and WGRMF from the matrix factorization methods. From the network-based methods, we selected Wang *et al.*’s method from section ‘Heterogeneous graph inference’ (which we will refer to as NBI+ from now on). As for BLMs, we selected Regularized Least Squares with Weighted Nearest Neighbors (RLS-WNN). In terms of prediction performance, the selected

Table 2. Statistics of each data set

Data sets	NR	GPCR	IC	E
Drugs	54	223	210	445
Targets	26	95	204	664
Interactions	90	635	1476	2926

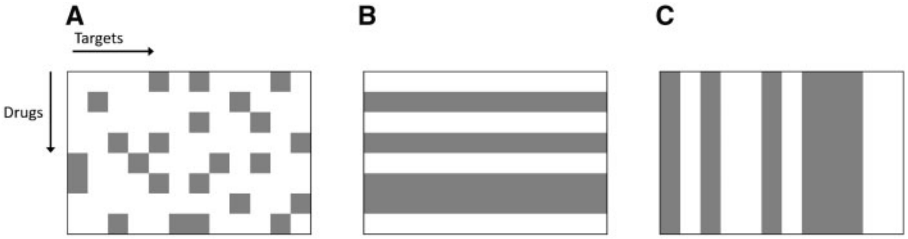

Figure 6. The different cross validation settings: (**A**) S1 involves leaving out random drug–target pairs from the interaction matrix Y to use as the test set, (**B**) S2 is the setting where entire drug profiles are left out and (**C**) S3 leaves out entire target profiles. Gray boxes represent left-out test instances.

methods are the best performing ones in their respective categories as reported in the publications where they appeared, which is why these methods in particular were selected to represent their categories. The source codes for all the selected methods are downloadable via the URL: https://github.com/alizat/Chemogenomic-DTI-Prediction-Methods.

The data sets in Table 2 are in the form of similarity matrices that were precomputed from nonvectorial data, that is, the raw data from which the matrices were derived (i.e. chemical structure graphs and genomic sequences) are not in the form of fixed-length feature vectors. Therefore, feature-based classification methods were not included in this comparison. However, we conducted a separate comparison among various feature-based methods on another benchmark data set introduced in [64]. Please refer to the Supplementary Material for the results of this comparison.

Parameters for all prediction methods have been tuned to give their optimal prediction performances under each of the cross validation settings. The optimal parameter values were obtained by grid search.

Results

The results of the different methods under S1, S2 and S3 CV settings are given in Tables 3, 4 and 5, respectively. We discuss the results below, stating advantages and disadvantages of each method as well as other general comments. Note that the results on the NR data set are particularly unstable because of its excessively small size [25]. As such, while we provide results for the NR data set, they are otherwise mostly ignored in the discussion below.

Table 3. AUPR results under S1

	NR	GPCR	IC	E
Nearest Profile	0.496 (0.012)	0.464 (0.009)	0.522 (0.005)	0.621 (0.003)
Weighted Profile	0.425 (0.012)	0.440 (0.010)	0.756 (0.003)	0.727 (0.001)
RLS-WNN	**0.729 (0.032)**	0.727 (0.018)	0.856 (0.011)	0.849 (0.006)
CMF	*0.639 (0.016)*	**0.754 (0.002)**	**0.937 (0.002)**	**0.883 (0.003)**
WGRMF	0.602 (0.038)	*0.737 (0.002)*	*0.923 (0.002)*	*0.877 (0.002)*
NBI+	0.287 (0.021)	0.255 (0.005)	0.162 (0.002)	0.206 (0.002)

Note: Best and second best AUPR results in each column are bold and italic, respectively. SDs are given in (parentheses).

Table 4. AUPR Results under S2

	NR	GPCR	IC	E
Nearest Profile	0.417 (0.031)	0.283 (0.017)	0.208 (0.013)	0.223 (0.007)
Weighted Profile	0.376 (0.022)	0.231 (0.005)	0.187 (0.005)	0.118 (0.002)
RLS-WNN	*0.545 (0.023)*	0.369 (0.007)	0.334 (0.010)	*0.393 (0.013)*
CMF	0.521 (0.027)	*0.407 (0.011)*	*0.353 (0.014)*	0.384 (0.012)
WGRMF	**0.570 (0.014)**	**0.427 (0.011)**	**0.367 (0.016)**	**0.413 (0.017)**
NBI+	0.267 (0.025)	0.201 (0.010)	0.112 (0.007)	0.110 (0.004)

Note: Best and second best AUPR results in each column are bold and italic, respectively. SDs are given in (parentheses).

Pair prediction case, S1

We draw two conclusions based on the results in Table 3. First, CMF is the overall best method under the S1 CV setting, followed by WGRMF. This shows that matrix factorization methods outperform other methods, which renders them as the most promising DTI prediction methods under S1. Second, Weighted Profile performs better than Nearest Profile in the IC and E data sets. The reason is likely that the IC and E data sets, being larger than the NR and GPCR data sets, have more neighbors to more accurately infer interactions from.

Drug prediction case, S2

Moving on to the S2 CV setting, it is obvious from the results in Table 4 that it is a more challenging setting than S1. According to insights obtained from a previous study on pair-input computational predictions [92], it is more difficult to predict new interactions for drugs (or targets) when they do not appear in the training set at all. This is in contrast to the S1 case where drug (or target) interaction profiles are only partially left out.

Going back to the results, WGRMF performed the best out of all the methods, followed by CMF. Again, matrix factorization methods seem to be doing well in general. WGRMF did better than CMF under S2 thanks to its graph regularization terms, which shows the usefulness of manifold learning in this less informative CV setting.

RLS-WNN, which uses network similarity, is able to give a reasonable prediction performance. This is thanks to the WNN preprocessing procedure that reinforces the learning process by inferring temporary profiles for the left-out drugs. Note that RLS-WNN computes network similarity in the form of GIP kernels that are used later in the algorithm. Naturally, the temporary profiles are better for computing network similarity than the initially empty profiles of the left-out drugs, which underscores the importance of preprocessing procedures like WNN when the incorporation of network similarity in training the classifiers is intended.

Target prediction case, S3

Finally, we reach the results for the S3 setting. As expected, the AUPR results of S3 are also lower than those obtained under S1, but they are consistently higher than those obtained under S2. This leads to the conclusion that target genomic sequence similarities are generally more reliable than drug chemical structure similarities, a conclusion that has been previously reached in [48].

As in the S2 case, the matrix factorization methods are generally superior, with WGRMF performing better than CMF thanks to its graph regularization terms. RLS-WNN gave a comparable performance. As for NBI+, similar to the S1 and S2

Table 5. AUPR Results under S3

	NR	GPCR	IC	E
Nearest Profile	0.393 (0.037)	0.444 (0.025)	0.589 (0.021)	0.647 (0.015)
Weighted Profile	0.379 (0.024)	0.327 (0.011)	0.721 (0.005)	0.673 (0.007)
RLS-WNN	**0.491 (0.032)**	0.574 (0.021)	0.763 (0.007)	0.778 (0.018)
CMF	*0.478 (0.017)*	*0.599 (0.033)*	*0.779 (0.011)*	*0.782 (0.013)*
WGRMF	0.464 (0.018)	**0.609 (0.032)**	**0.813 (0.007)**	**0.808 (0.018)**
NBI+	0.300 (0.020)	0.203 (0.006)	0.193 (0.006)	0.210 (0.007)

Note: Best and second best AUPR results in each column are bold and italic, respectively. SDs are given in (parentheses).

cases, it was unable to outperform the baseline methods, Nearest Profile and Weighted Profile. Thus, we conclude that, network-based methods are generally not the best choice for DTI prediction.

Discussions

Generally speaking, the matrix factorization methods are the best methods when it comes to predicting DTIs. In addition, the manifold assumption that points lie on or near to a low-dimensional manifold [67–69] appears to be successful in improving DTI prediction performance (as displayed by WGRMF). However, it seems that when prior information is available in abundance (the S1 setting), manifold learning becomes slightly less useful (as shown by CMF that did better than WGRMF under S1) but still useful nonetheless.

It is important to mention that while RLS-WNN did not beat the matrix factorization methods in the predictions, it is relatively a much faster algorithm. It is also more robust in terms of selecting values for its parameters—the matrix factorization methods have more parameters that are sensitive and need more fine-tuning. As such, when one goes about the task of predicting DTIs, it is always good idea to obtain initial predictions with RLS-WNN first. We also emphasize that all BLMs are generally fast and memory-efficient algorithms and that they should be the first algorithms to consider if the data set used is significantly larger than the ones used in this study.

Regarding the network-based method, NBI+, it did not do as well as the other methods. It may be that the properties of the DTI networks are not favorable for use with such a network-based method. Examples of such properties are the low average number of interactions known per drug or target in the network and the presence of a considerable number of undiscovered interactions among the noninteractions (which can negatively influence predictions). Furthermore, they do not do well in predicting new interactions for orphan drugs for which no interactions are previously known. The problem is even more challenging when the interaction that we try to predict is with an orphan target as well; this is because the path on the network between the orphan drug and target would be too indirect and would thus be given a low prediction score. Finally, it has been stated in a previous survey [26] that predictions from network-based methods tend to be biased toward those drugs with more associated targets (or targets with more associated drugs) and that it is generally nontrivial to predict 'an interaction between a drug in one subnetwork and a target in another'.

On the other hand, network-based methods still have a place in DTIs prediction. As an example, the pioneering network-based method, NRWRH [52], generated a heterogeneous network (as in Figure 2) on which a Random Walk was performed to obtain predictions, which is an elegant idea indeed. Augmenting the heterogeneous network with more information (e.g. by adding extra drug and target pairwise similarities) may help remedy the issues that network-based methods face in predicting interactions for orphan drugs or targets to some extent. It may also be helpful to draw inspiration from previous work on generating functional linkage networks (FLNs) [93–96]. FLNs are networks of functional associations between genes, and they have been successfully used in research related to investigating gene-related functions and diseases. Constructing FLNs requires gathering of information from multiple heterogeneous sources of varying quality and completeness and that may occasionally correlate highly with each other; such experience in constructing FLNs can be transferred to the generation of heterogeneous DTI networks on which network-based methods can be applied to predict new DTIs with better accuracy.

Now, we move on to an issue that is related to experimental design. As mentioned earlier, drug–target pairs are left out as test instances to see how well they are predicted by the different prediction methods. This is done by setting the values of the test instances to 0 (i.e. set $Y_{ij} = 0$ for test instances). The issue here is that known noninteractions and test instances would be both be represented by the same 0 value, which may not be ideal. However, giving a unique representation for noninteractions to separate them from test instances is not straightforward. In [48], it was found via experimentation that representing noninteractions by any value that is far from 0 (e.g. -1) is generally not a good idea. This is mainly because of the severe imbalance in the data (i.e. much more noninteractions than there are interactions); supposing, for example, that noninteractions are represented as -1, classifiers would focus more on predicting noninteractions correctly at the expense of predicting interactions correctly. While some algorithms (e.g. CMF and WGRMF) partially circumvent the representation issue by using a weight matrix W that prevents test instances from contributing in the predictions, most (if not all) previous work in DTI prediction has represented test instances by setting them to 0. Note that this issue does not apply to feature-based classification methods where test instances are simply excluded from the training set used to train the classifier, and then the trained classifier is used to perform predictions on the test instances.

Avenues for improvement and further research

In this section, we give examples on how researchers attempted to improve DTI prediction performance and occasionally provide some suggestions of our own for ideas on how to improve as well.

Using more information

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

One such work was [97] where many drug and target kernels were used, and a multiple kernel learning method was developed to take them as input and determine how best to merge them to provide the best predictions. An interesting thing about this work is that some of the kernels were produced from the same information source. From target protein sequences, normalized Smith–Waterman, mismatch and spectrum kernels were created (via the KeBABS package [98]). From the drug side, the Rchemcpp [99] package was used to obtain spectrum, Lambda-k, Marginalized, MinMax and Tanimoto kernels from drugs' chemical structures. For more on multiple kernel learning in general, the reader is referred to [100]. Furthermore, Table 6 provides a list of software packages that exist for extracting drug and target features from their chemical structures and genomic sequences, respectively.

Another work that used multiple kernels was [101]. Drugs were represented as FP2 fingerprints. As for targets, different representations were generated including those based on

Table 6. Software packages to compute features for drugs and targets

Package	Link
ChemCPP	http://chemcpp.sourceforge.net
RDKit	http://www.rdkit.org/
PyDPI [36]	https://sourceforge.net/projects/pydpicao/
OpenBabel [37]	http://openbabel.org/
Rcpi [35]	http://bioconductor.org/packages/release/bioc/html/Rcpi.html
Rchemcpp [99]	http://shiny.bioinf.jku.at/Analoging/
KeBABS [98]	http://www.bioinf.jku.at/software/kebabs/
PROFEAT [44]	http://bidd2.nus.edu.sg/cgi-bin/profeat2016/main.cgi

autocovariance, entropy, discrete wavelet and substitution matrices among several others. An ensemble of SVM classifiers was trained, one classifier for each target descriptor type. Each drug–target pair was represented by concatenating the FP2 fingerprint of the drug with the target's descriptor. Predictions from the different SVM classifiers were summed up to give the final predictions.

Secondary structure information of proteins [102] is something that has not been used often. It is a type of information that can be extracted from protein sequences, which is a good thing, as genomic sequences are always available for proteins. Known drug–disease and protein–disease associations may also be used as other sources of information [103]; however, these data have also not been used frequently for DTI prediction.

Determination of 3D structures of membrane proteins (via wet-lab techniques) is becoming more feasible over time [21], so we may eventually witness the use of protein structure information in global-scale DTI prediction. This would possibly trigger the emergence of software packages that would routinely be used to generate descriptors for a protein from its structure (instead of from its sequence). These features may yield better prediction performance, as it is widely accepted that the protein's structure is what dictates its function. Until such software packages appear, researchers can experiment to find features that are most useful to extract from proteins' structures.

Finally, for a comprehensive overview of the different ways to integrate multiple information sources simultaneously for improving prediction performance, we refer the reader to [104].

Ensemble learning

There are two types of ensemble methods: heterogeneous ensembles and homogeneous ensembles.

Heterogeneous ensembles consist of different learners that have different induction biases. These learners are typically trained using the same data. A procedure known as stacking is usually used where the results from the different learners are concatenated to form feature vectors that are then used to train yet another meta-learner, which gives the final predictions [80]. The improvement in prediction performance is intended to be obtained from the diversity induced by the different inductive biases of the learners constituting the ensemble.

A heterogeneous ensemble for DTI prediction was previously developed [105] that consisted of four methods: Weighted Profile, RLS-avg, LapRLS and NBI. After predicting with these methods, an SVM meta-learner is trained with their results and then used to give the final prediction. The ensemble showed improved prediction performance over all the constituent methods. Another heterogeneous ensemble, DrugE-Rank [106], also uses a number of different learners as in [105], but instead of the SVM meta-learner, it uses a ranking algorithm, LambdaMART [107], to give the final predictions.

In terms of possible future work regarding heterogeneous ensemble methods, a technique that has not been used in previous work is ensemble pruning. That is, a subset of the base learners is used to constitute the ensemble. This would lead to smaller ensembles and, subsequently, to better computational efficiency because of the lower number of base learners performing predictions. In addition, it was shown that these smaller ensembles obtained via ensemble pruning can also have better generalization performance [108].

Homogeneous ensembles, on the other hand, consist of learners of the same type. For example, Random Forest [109] is a homogeneous ensemble method that consists of many decision trees. To obtain improved prediction performance, the diversity that would help achieve this can come from different sources. An example is bagging, which induces diversity by randomly sampling with replacement; for each learner, a subset of the training examples is randomly sampled to train it (i.e. each learner uses a different training set). Another example is feature subspacing, which induces diversity by randomly sampling a subset of the features for each learner, and so on.

A homogeneous ensemble method based on decision trees was introduced in [110]. It randomly projects the features matrix (representing the different drug–target pairs) into a lower dimensionality matrix. This reduces the dimensionality of the data (thus improving computational speed) as well as injects diversity into the ensemble leading to gains in prediction performance. Examples of other homogeneous ensembles include [62, 64, 65], which have been described earlier in section 'Methods'

Besides bagging and feature subspacing, there are other ways to generate diversity in the base learners that have not yet been used in homogeneous ensemble methods for DTI prediction. One such way is to use different parameter settings for each of the base learners [80]. Another way is to randomly flip the labels of some training examples (i.e. convert from 1 to 0 or vice versa) [111]. These tricks may be used to enhance the prediction performance further.

Deep learning

The use of deep learning has been steadily increasing in drug discovery [112, 113]. The reason for this is that deep learning has the potential to build complex models that are able to learn difficult concepts and thus outperform other competing methods. In addition, as it has the ability to extract useful features from the input features, we believe that deep learning methods would especially shine when it comes to merging different sources of information. The two main limitations that were holding deep learning from being popular were: (1) a lot of training data are needed to train the complex model being generated, and (2) a lot of computational power is needed to perform the training. However, as time goes on, these two issues are becoming less of drawbacks because of the accumulation of more data to work with as well as the emerging of more high-performance computing resources.

In [25], the authors suggested the use of a CV setting, S4, where drugs and targets used in training do not appear in the test set, and it is known to be a challenging setting indeed. In the experiments conducted in [25], only trivial interactions were predicted successfully under S4. We believe that deep learning—with its ability to obtain useful deep representations of the drugs, targets and interactions—has the potential to do much better than other state-of-the-art methods in predicting interactions under S4. This is yet to be confirmed in future work.

A number of efforts regarding DTI prediction have made use of deep learning to improve prediction performance. Deep learning techniques that have been used in DTI prediction include restricted Boltzmann machines [114], deep neural networks [115, 116, 117], stacked auto-encoders [118, 119] and deep belief networks [120]. As of yet, none of the deep learning methods developed for DTI prediction have attempted to simultaneously use multiple heterogeneous sources of drug and target information. It would be interesting to see efforts that attempt to do so in future work.

Absence of reliable negatives

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

Biased SVM is a variant of SVM that was used in [121] to give different weights to the positive and negative classes in the data. Positive examples, being more reliable, are given higher weights than the negative examples. The weights are tuned to give the best possible prediction performance.

PUDT [122] is a DTI prediction method that uses positive unlabeled learning to deal with the issue of unconfident negatives. Sets of unlabeled drug–target pairs are labeled as reliable negative and likely negative. An SVM classifier is then trained where, similar to biased SVM, weights are given to these negative classes (along with the positive class) and are tuned to give a good prediction performance.

In [123], multiple drug and target similarities were obtained and then merged together via the following equation:

$$S_{ij} = 1 - \prod_{n} (1 - S_{ij}^{(n)}). \tag{43}$$

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

In [124], the BioLip [125] and BindingDB [126] databases are searched for interactions with a binding affinity $< 10\mu M$ to be used as negatives.

Rather than using binary values to represent interactions and noninteractions, the use of data sets where continuous values correspond to drug–target affinities has been previously proposed [25]. Examples of such data sets include those introduced in [127] and [128] which, respectively, contain kinase disassociation constants and kinase inhibition constants—constants with lower values correspond to higher affinities and vice versa. It is suggested in [25] that such data sets be used as benchmarking data sets in future DTI prediction efforts. It is an idea worth considering, as these data sets provide a more accurate representation of reality than traditional binary-valued data sets. Using such data sets would also implicitly eliminate the issue of reliable negatives discussed above. We suspect that this is a trend that will increase in the future, as data of this kind become more abundant.

Big data

Over 90 million chemical compounds are currently stored in PubChem [129], while the conducted BioAssays have only covered about 2.4 million compounds (i.e. targets are now known for these compounds) [11]. It is unlikely that future BioAssay experiments will cover the remaining compounds anytime in the foreseeable future. Virtual screening of these compounds is thus inevitable. However, as the size of the data is exceptionally large, big data technologies (e.g. cloud computing) will need to be used.

However, adjustments to algorithms (or, possibly, novel algorithms altogether) will also need to be made to handle data of such size. For example, the work done in [86] is a step in this direction—minwise hashing was used to obtain a compact representation for the drug–target pairs, which reduces the data dimensionality, and the reduced dimensionality helps lower both the space and time complexities. Another work that aims for scalability is [130] where a memory-efficient tree structure is developed to query large databases for similar drug–target pairs.

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

Network visualization

As an exploratory analysis aid, network visualization may be used to display the DTI bipartite network. Inspecting the network visually may provide clues or insights that could otherwise be difficult to reach.

For example, it may be easier to determine why certain interactions tend to get low prediction scores by carefully observing the visualized network for hints. The user may consider using edge width or coloring edges with a color scale to indicate how high or low their prediction scores are. This particular example is illustrated in Figure 7.

Many tools exist for visualizing networks. Two tools that are used in visualizing DTI networks are NodeXL [131] and Cytoscape [132].

Noncoding RNAs

While this work primarily focuses on target proteins, there is another type of target—noncoding RNAs (ncRNAs)—for which drugs have been successfully developed. ncRNAs are RNAs that do not code for proteins, and they consist of multiple subcategories including microRNAs (miRNAs), long noncoding RNAs and intronic RNAs among several others. To give a few examples, drugs based on miRNAs have been used to treat Hepatitis C virus [133] and Alport nephropathy [134], while others based on intronic RNAs have been used to treat Duchenne muscular dystrophy [135] and Usher syndrome [136]. Each of the different types of ncRNAs has unique behaviors and mechanisms, thus presenting various challenges and opportunities, all of which are discussed with examples in a recent overview [137].

We are expecting more research involving ncRNAs in the future. Worthy of mentioning is the NRDTD database [138] that

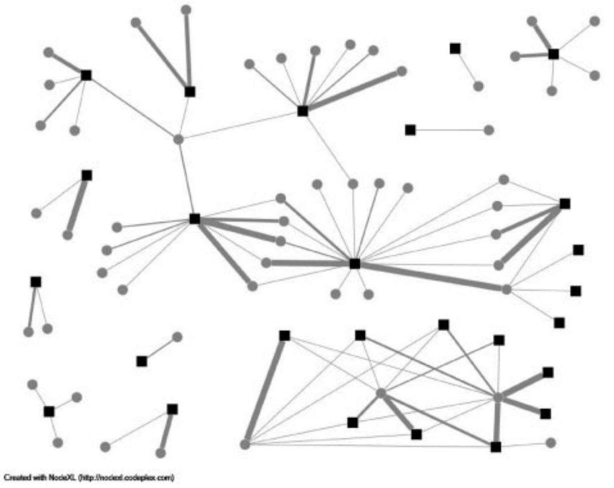

Figure 7. Visualization of the NR DTI network where circles and squares represent drugs and targets, respectively. An S1 CV experiment was performed, and the final averaged prediction scores are represented by the thickness of the edges.

has been recently set up to store information on ncRNAs and their binding drugs. It is likely that research into ncRNAs as drug targets will witness the frequent use of this database.

Evaluation metrics

An important part of DTI prediction that deserves some attention is how the prediction performance of the different classifiers is evaluated. We take as an example the AUPR metric, which was used in this study to compare between the different prediction methods. The AUPR score was computed by first pooling the prediction values for all the drug–target pairs and then sorting them to compute the AUPR.

However, as we may also be interested in how well-ranked the predictions are for each drug/target separately, computing the AUPR differently may be worth considering. Specifically, a per-drug AUPR may be calculated by computing a separate AUPR for each drug—i.e. sort each drug's predicted targets and compute the AUPR for each drug separately—and then average all the drug AUPRs to get the per-drug AUPR. A per-target AUPR may also be obtained in a similar fashion. The per-drug and per-target AUPRs may possibly be better in reflecting the aspects of the prediction performance that we are really interested in testing.

Another evaluation metric that has been previously used in recommender systems is the mean percentile ranking (MPR) [75, 139]. MPR is typically used in cases where there is a lack of negative feedback data (which is analogous to the lack of confident noninteractions in our case). It can also be considered a per-drug or per-target metric, so it would serve a similar purpose to that of the per-drug and per-target AUPRs. Taking the per-drug MPR as an example, it is defined as:

$$MPR_d = \frac{\sum_{dt} Y_{dt} rank_d(t)}{\sum_{dt} Y_{dt}}, \tag{44}$$

where $Y_{dt} = 1$ if drug d and target t interact and $Y_{dt} = 0$ otherwise, and $rank_d(t)$ is the predicted rank of target t among all the targets for likelihood of interaction with drug d. For example, this metric may be adequately used for testing the prediction performance of classifiers under the S2 CV setting. Similarly, a per-target MPR may be used in the S3 CV setting as well. For a more exhaustive discussion of the implications involved in the selection of the evaluation metric as well as the advantages and disadvantages of each of the different metrics, the reader is referred to [140].

Conclusion and outlook

Drug repositioning involves many computational techniques that are used in various circumstances depending on the current level of available knowledge on the target disease [141]. Comprehensive surveys providing overviews on these computational approaches have been published previously [142, 143]. Of these approaches, we gave an overview of DTI prediction, which is an important task in drug discovery. Indeed, many Web servers have been developed to facilitate this task for practitioners who wish to perform it on a global scale [26]. Examples of such Web servers include DINIES [144], BalestraWeb [145] and SuperPred [146] among several others [54, 147–150].

In this work, we started by describing the data required for the task of drug–DTI prediction and gave examples of different kinds of data that may be used. Next, we gave an up-to-date overview of the different state-of-the-art methods that are trained with said data and then used to predict new interactions. We then performed an empirical comparison between a number of pre-selected methods to show their prediction performances under different scenarios. Finally, we provided a list of avenues for further improvement of the prediction performance.

Research on chemogenomic DTI prediction has been conducted for about a decade now, starting with pioneering works such as [151–153] up until the current day. Research on chemogenomic methods for predicting DTIs is expected to continue for several years with contributions involving deep learning concepts, multiview learning and possibly unprecedented clever features for representing drugs and/or targets. In addition, as algorithms get more sophisticated over time, big data technologies (e.g. Spark) may enter the picture.

From the data perspective, there is the issue of data sets being of a binary nature, i.e. given an interaction matrix Y where $Y_{ij} = 1$ if drug d_i and target t_j interact and 0 otherwise. This brings forth a significant problem. Some of the 0's in Y may be interactions that are yet undiscovered, which may throw off the training process for the different classifiers. Another point is that, in reality, drug–target pairs have binding affinities that vary over a spectrum (interactions are not binary on/off). Data sets with continuous values representing drug–target binding affinities (as opposed to discrete 0 and 1 values) have been previously proposed [127, 128], and we expect the trend of using such continuous-valued data sets to eventually catch on, as it is more useful and more meaningful (i.e. better represents reality) than the binary data sets that have been used in the majority of previous work in DTI prediction.

So far, the majority of the work has been concerned with protein targets. However, it is expected that ncRNAs will eventually snatch some of the spotlight. Many ncRNA-targeting drugs have been developed, and much more are expected to appear as our understanding of how ncRNAs operate improves [137]. As for global-scale DTI prediction using machine learning algorithms (as exemplified in this survey), it is possible that we will witness efforts that attempt to do so in the near future. Such efforts would use a repository such as NRDTD [138] that stores extensive information on known drug–ncRNA interactions.

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bib.

Funding

This work was supported by the Agency for Science, Technology and Research (A*STAR), Singapore.

References

1. Ashburn TT, Thor KB. Drug repositioning: Identifying and developing new uses for existing drugs. *Nat Rev Drug Discov* 2004;**3**(8):673–83.
2. Novac N. Challenges and opportunities of drug repositioning. *Trends Pharmacol Sci* 2013;**34**(5):267–72.
3. Frantz S. Drug discovery: playing dirty. *Nature* 2005; **437**(7061):942–3.
4. McLean SR, Gana-Weisz M, Hartzoulakis B, *et al.* Imatinib binding and cKIT inhibition is abrogated by the cKIT kinase domain I missense mutation Val654Ala. *Mol Cancer Ther* 2005;**4**(12):2008–15.
5. Pepin J, Guern C, Milord F, *et al.* Difluoromethylornithine for arseno-resistant Trypanosoma brucei gambiense sleeping sickness. *Lancet* 1987;**330**(8573):1431–3.
6. Chong CR, Chen X, Shi L, *et al.* A clinical drug library screen identifies astemizole as an antimalarial agent. *Nat Chem Biol* 2006;**2**(8):415–16.
7. Miguel DC, Yokoyama-Yasunaka JKU, Andreoli WK, *et al.* Tamoxifen is effective against Leishmania and induces a rapid alkalinization of parasitophorous vacuoles harbouring Leishmania (Leishmania) amazonensis amastigotes. *J Antimicrob Chemother* 2007;**60**(3):526–34.
8. Chow WA, Jiang C, Guan M. Anti-HIV drugs for cancer therapeutics: back to the future? *Lancet Oncol* 2009;**10**(1):61–71.
9. Gloeckner C, Garner AL, Mersha F, *et al.* Repositioning of an existing drug for the neglected tropical disease Onchocerciasis. *Proc Natl Acad Sci USA* 2010;**107**(8):3424–9.
10. Aronson JK. Old drugs–new uses. *Br J Clin Pharmacol* 2007; **64**(5):563–5.
11. Wang Y, Bryant SH, Cheng T, *et al.* PubChem BioAssay: 2017 update. *Nucleic Acids Res* 2017;**45**(D1):D955.
12. Yao L, Evans JA, Rzhetsky A. Novel opportunities for computational biology and sociology in drug discovery: corrected paper. *Trends Biotechnol* 2010;**28**(4):161–70.
13. Keiser MJ, Setola V, Irwin JJ, *et al.* Predicting new molecular targets for known drugs. *Nature* 2009;**462**(7270):175–81.
14. Johnson, MA, Maggiora, GM. *Concepts and Applications of Molecular Similarity*. New York, NY: Wiley, 1990.
15. Keiser MJ, Roth BL, Armbruster BN, *et al.* Relating protein pharmacology by ligand chemistry. *Nat Biotechnol* 2007;**25**(2):197–206.
16. Jacob L, Vert J-P. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics* 2008; **24**(19):2149–56.
17. Li H, Gao Z, Kang L, *et al.* TarFisDock: a web server for identifying drug targets with docking approach. *Nucleic Acids Res* 2006;**34(Suppl 2)**:W219–24.
18. Cheng AC, Coleman RG, Smyth KT, *et al.* Structure-based maximal affinity model predicts small-molecule druggability. *Nat Biotechnol* 2007;**25**(1):71–5.
19. Pujadas G, Vaque M, Ardevol A, *et al.* Protein-ligand docking: a review of recent advances and future perspectives. *Curr Pharm Anal* 2008;**4**(1):1–19.
20. Yildirim MA, Goh K-I, Cusick ME, *et al.* Drug–target network. *Nat Biotechnol* 2007;**25**(10):1119–26.
21. Opella SJ. Structure determination of membrane proteins by nuclear magnetic resonance spectroscopy. *Annu Rev Anal Chem* 2013;**6**:305–28.
22. 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**(13):i232–40.
23. Ding H, Takigawa I, Mamitsuka H, Zhu S. Similarity-based machine learning methods for predicting drug–target interactions: a brief review. *Brief Bioinform* 2014;**15**(5):734–47.
24. Mousavian Z, Masoudi-Nejad A. Drug-target interaction prediction via chemogenomic space: learning-based methods. *Expert Opin Drug Metab Toxicol* 2014;**10**(9):1273–87.
25. Pahikkala T, Airola A, Pietilä S, *et al.* Toward more realistic drug–target interaction predictions. *Brief Bioinform* 2015;**16**: 325–37.
26. Chen X, Yan CC, Zhang X, *et al.* Drug-target interaction prediction: databases, web servers and computational models. *Brief Bioinform* 2016;**17**(4):696–712.
27. Kanehisa M, Goto S, Sato Y, *et al.* KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res* 2012;**40**(D1):D109–14.
28. Knox C, Law V, Jewison T, *et al.* DrugBank 3.0: a comprehensive resource for 'Omics' research on drugs. *Nucleic Acids Res* 2011;**39(Suppl 1)**:D1035–41.
29. Gaulton A, Bellis LJ, Bento AP, *et al.* ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res* 2011;**40**:D1100–7.
30. Kuhn M, Szklarczyk D, Pletscher-Frankild S, *et al.* STITCH 4: integration of protein chemical interactions with user data. *Nucleic Acids Res* 2014;**42**(D1):D401–7.
31. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 1988;**28**(1):31–6.
32. Kuhn M, Campillos M, Letunic I, *et al.* A side effect resource to capture phenotypic effects of drugs. *Mol Syst Biol* 2010;**6**(1): 343.
33. Skrbo A, Begović B, Skrbo S. Classification of drugs using the ATC system (Anatomic, Therapeutic, Chemical classification) and the latest changes. *Med Arh* 2004;**58(1 Suppl 2)**: 138–41.
34. Lamb J. The connectivity map: a new tool for biomedical research. *Nat Rev Cancer* 2007;**7**(1):54–60.
35. Cao D-S, Xiao N, Xu Q-S, *et al.* Rcpi: R/Bioconductor package to generate various descriptors of proteins, compounds and their interactions. *Bioinformatics* 2015;**31**(2):279–81.
36. Cao D-S, Liang Y-Z, Yan J, *et al.* PyDPI: freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies. *J Chem Inf Model* 2013;**53**(11):3086–96. PMID: 24047419.
37. O'Boyle NM, Banck M, James CA, *et al.* Open Babel: an open chemical toolbox. *J Cheminform* 2011;**3**(1):33.
38. Jain E, Bairoch A, Duvaud S, *et al.* Infrastructure for the life sciences: design and implementation of the UniProt website. *BMC Bioinformatics* 2009;**10**(1):136.
39. Ashburner M, Ball CA, Blake JA, *et al.* Gene ontology: tool for the unification of biology. *Nat Genet* 2000;**25**(1):25–9.
40. Emig D, Ivliev A, Pustovalova O, *et al.* Drug target prediction and repositioning using an integrated network-based approach. *PLoS One* 2013;**8**(4):e60618.
41. Zong N, Kim H, Ngo V, *et al.* Deep mining heterogeneous networks of biomedical linked data to predict novel drug-target associations. *Bioinformatics* 2017;**33**:2337–44.

42. Cannataro M, Guzzi PH, Veltri P. Protein-to-protein interactions: technologies, databases, and algorithms. *ACM Comput Surv* 2010;**43**(1):Article no. 1.
43. Klingström T, Plewczynski D. Protein-protein interaction and pathway databases, a graphical review. *Brief Bioinform* 2011;**12**(6):702–13.
44. Zhang P, Tao L, Zeng X, *et al*. A protein network descriptor server and its use in studying protein, disease, metabolic and drug targeted networks. *Brief Bioinform* 2017;**18**:1057–70.
45. Shi J-Y, Yiu S-M. SRP: a concise non-parametric similarity-rank-based model for predicting drug-target interactions. In: *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Washington, DC, USA: IEEE, 2015, 1636–41.
46. Bleakley K, Yamanishi Y. Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics* 2009;**25**(18):2397–403.
47. Xia Z, Wu L-Y, Zhou X, *et al*. Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Syst Biol* 2010;**4**(2):S6.
48. van Laarhoven T, Nabuurs SB, Marchiori E. Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics* 2011;**27**(21):3036–43.
49. Mei J-P, Kwoh C-K, Yang P, *et al*. Drug-target interaction prediction by learning from local information and neighbors. *Bioinformatics* 2013;**29**(2):238–45.
50. 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**(5):e1002503.
51. Wang, W, Yang, S, Li, J. Drug target predictions based on heterogeneous graph inference. *Pac Symp Biocomput* 2013;**18**: 53–64.
52. Chen X, Liu M-X, Yan G-Y. Drug-target interaction prediction by random walk on the heterogeneous network. *Mol Biosyst* 2012;**8**(7):1970–8.
53. Fakhraei S, Huang B, Raschid L, *et al*. Network-based drug-target interaction prediction with probabilistic soft logic. *IEEE/ACM Trans Comput Biol Bioinform* 2014;**11**(5):775–87.
54. Ba-alawi W, Soufan O, Essack M, *et al*. DASPfind: new efficient method to predict drug–target interactions. *J Cheminform* 2016;**8**(1):15.
55. Gönen M. Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics* 2012;**28**(18):2304–10.
56. Cobanoglu MC, Liu C, Hu F, *et al*. Predicting drug–target interactions using probabilistic matrix factorization. *J Chem Inf Model* 2013;**53**(12):3399–409.
57. 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. Chicago, IL, USA: ACM, 2013, 1025–33.
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**(3):646–56.
59. Liu Y, Wu M, Miao C, *et al*. Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. *PLoS Comput Biol* 2016;**12**(2):e1004760.
60. Hao M, Bryant SH, Wang Y. Predicting drug-target interactions by dual-network integrated logistic matrix factorization. *Sci Rep* 2017;**7**:40376.
61. He Z, Zhang J, Shi X-H, *et al*. Predicting drug-target interaction networks based on functional groups and biological features. *PLoS One* 2010;**5**(3):e9603.
62. Yu H, Chen J, Xu X, *et al*. A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data. *PLoS One* 2012;**7**(5):e37608.
63. Xiao X, Min J-L, Wang P, *et al*. iGPCR-Drug: a web server for predicting interaction between GPCRs and drugs in cellular networking. *PLoS One* 2013;**8**(8):e72234.
64. Ezzat A, Wu M, Li X-L, *et al*. Drug-target interaction prediction via class imbalance-aware ensemble learning. *BMC Bioinformatics* 2016;**17**(19):267–76.
65. Ezzat A, Wu M, Li X-L, *et al*. Drug-target interaction prediction using ensemble learning and dimensionality reduction. *Methods* 2017;**129**: 81–8.
66. Perlman L, Gottlieb A, Atias N, *et al*. Combining drug and gene similarity measures for drug-target elucidation. *J Comput Biol* 2011;**18**(2):133–45.
67. Tenenbaum JB, de Silva V., Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science* 2000;**290**(5500):2319–23.
68. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science* 2000;**290**(5500):2323–6.
69. Belkin M, Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Dietterich TG, Becker S, Ghahramani Z (eds), *Advances in Neural Information Processing Systems*, Vol. **14**. Cambridge, MA: MIT Press, 2002, 585–91.
70. Raymond R, Kashima H. Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. *Mach Learn Knowl Discov Databases* 2010;131–47.
71. van Laarhoven T, Marchiori E. Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile. *PLoS One* 2013;**8**(6):e66952.
72. Gu Q, Zhou J, Ding C. *Collaborative Filtering: Weighted Nonnegative Matrix Factorization Incorporating User and Item Graphs*. Columbus, OH, USA: Society for Industrial and Applied Mathematics, 2010, 199–210.
73. Cai D, He X, Han J, *et al*. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans Pattern Anal Mach Intell* 2011;**33**(8):1548–60.
74. Shang F, Jiao LC, Wang F. Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognit* 2012;**45**(6):2237–50. Brain Decoding.
75. Johnson CC. Logistic matrix factorization for implicit feedback data. *Adv Neural Inf Process Syst* 2014;**27**.
76. Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 2005;**27**(8):1226–38.
77. Jong S. d. SIMPLS: an alternative approach to partial least squares regression. *Chemometr Intell Lab Syst* 1993;**18**(3): 251–63.
78. Wang L, You Z-H, Chen X, *et al*. RFDT: a rotation forest-based predictor for predicting drug-target interactions using drug structure and protein sequence information. *Curr Protein Pept Sci* 2016;**18**(999):1.
79. Zhang C-X, Zhang J-S. A variant of rotation forest for constructing ensemble classifiers. *Pattern Anal Appl* 2010;**13**(1): 59–77.
80. Zhou Z-H, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.
81. Meng F-R, You Z-H, Chen X, *et al*. Prediction of drug–target interaction networks from the integration of protein sequences and drug chemical structures. *Molecules* 2017; **22**(7):1119.

82. Tipping ME. Sparse bayesian learning and the relevance vector machine. *J Mach Learn Res* 2001;**1**:211–44.
83. Huang Y-A, You Z-H, Chen X. A systematic prediction of drug-target interactions using molecular fingerprints and protein sequences. *Curr Protein Pept Sci* 2016, in press.
84. Yamanishi Y, Pauwels E, Saigo H, *et al*. Extracting sets of chemical substructures and protein domains governing drug-target interactions. *J Chem Inform Model* 2011;**51**(5): 1183–94.
85. Finn RD, Bateman A, Clements J, *et al*. Pfam: the protein families database. *Nucleic Acids Res* 2014;**42**(D1):D222–30.
86. Tabei Y, Yamanishi Y. Scalable prediction of compound-protein interactions using minwise hashing. *BMC Syst Biol* 2013;**7(Suppl 6)**:S3.
87. Broder AZ, Charikar M, Frieze AM, *et al*. Min-wise independent permutations. *J Comput Syst Sci* 2000;**60**(3):630–59.
88. Raghavan V, Bollmann P, Jung GS. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans Inform Syst* 1989;**7**(3):205–29.
89. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. Pittsburgh, PA, USA: ACM, 2006, 233–40.
90. Hattori M, Okuno Y, Goto S, *et al*. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *J Am Chem Soc* 2003;**125**(39):11853–65. PMID: 14505407.
91. Smith TF, Waterman MS. Identification of common molecular subsequences. *J Mol Biol* 1981;**147**(1):195–7.
92. Park Y, Marcotte EM. Flaws in evaluation schemes for pair-input computational predictions. *Nat Methods* 2012;**9**(12): 1134–6.
93. Linghu B, Franzosa EA, Xia Y. Construction of functional linkage gene networks by data integration. *Methods Mol Biol* 2013;**939**:215–32.
94. Bogdanov P, Macropol K, Singh AK. Function annotation in gene networks. In: *Functional Coherence of Molecular Networks in Bioinformatics*. Springer, New York, 2012, 49–67.
95. Hegde SR, Manimaran P, Mande SC. Dynamic changes in protein functional linkage networks revealed by integration with gene expression data. *PLoS Comput Biol* 2008;**4**(11):e1000237.
96. Karaoz U, Murali TM, Letovsky S, *et al*. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc Natl Acad Sci USA* 2004;**101**(9):2888–93.
97. Nascimento ACA, Prudêncio RBC, Costa IG. A multiple kernel learning algorithm for drug-target interaction prediction. *BMC Bioinformatics* 2016;**17**(1):46.
98. Palme J, Hochreiter S, Bodenhofer U. KeBABS: An R package for kernel-based analysis of biological sequences. *Bioinformatics* 2015;**31**(15):2574.
99. Klambauer G, Wischenbart M, Mahr M, *et al*. Rchemcpp: A web service for structural analoging in ChEMBL, Drugbank and the Connectivity Map. *Bioinformatics* 2015;**31**(20):3392.
100. Gönen M, Alpayd E. Multiple kernel learning algorithms. *J Mach Learn Res* 2011;**12**:2211–68.
101. Nanni L, Lumini A, Brahnam S. A set of descriptors for identifying the protein-drug interaction in cellular networking. *J Theor Biol* 2014;**359**:120–8.
102. Lin K, Simossis VA, Taylor WR, *et al*. A simple and fast secondary structure prediction method using hidden neural networks. *Bioinformatics* 2005;**21**(2):152–9.
103. Wang W, Yang S, Zhang X, *et al*. Drug repositioning by integrating target information through a heterogeneous network model. *Bioinformatics* 2014;**30**(20):2923–30.
104. Li Y, Wu F-X, Ngom A. A review on machine learning principles for multi-view biological data integration. *Brief Bioinform* 2016, doi: 10.1093/bib/bbw113.
105. Zhang R. An ensemble learning approach for improving drug–target interactions prediction. In: *Proceedings of the 4th International Conference on Computer Engineering and Networks*. SH, China: Springer, 2015, 433–442.
106. 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**(12): i18–27.
107. Burges CJC. From ranknet to lambdarank to lambdamart: an overview. Microsoft Research Technical Report, MSR-TR-2010-82, vol. 11(23–581), 2010, 81.
108. Zhou Z-H, Wu J, Tang W. Ensembling neural networks: many could be better than all. *Artif Intell* 2002;**137**(1–2): 239–63.
109. Breiman L. Random forests. *Mach Learn* 2001;**45**(1):5–32.
110. Zhang J, Zhu M, Chen P, *et al*. DrugRPE: random projection ensemble approach to drug-target interaction prediction. *Neurocomputing* 2017;**228**:256–62.
111. Breiman L. Randomizing outputs to increase prediction accuracy. *Mach Learn* 2000;**40**(3):229–42.
112. Gawehn E, Hiss JA, Schneider G. Deep learning in drug discovery. *Mol Inform* 2016;**35**(1):3–14.
113. Ekins S. The next era: deep learning in pharmaceutical research. *Pharm Res* 2016;**33**(11):2594–603.
114. Wang Y, Zeng J. Predicting drug-target interactions using restricted Boltzmann machines. *Bioinformatics* 2013;**29**(13): i126–34.
115. Wang C, Liu J, Luo F, *et al*. Pairwise input neural network for target-ligand interaction prediction. In: *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Belfast, UK: IEEE, 2014, 67–70.
116. Tian K, Shao M, Wang Y, *et al*. Boosting compound-protein interaction prediction by deep learning. *Methods* 2016;**110**: 64–72.
117. Wan F, Zeng J. Deep learning with feature embedding for compound-protein interaction prediction. *bioRxiv* 2016, https://doi.org/10.1101/086033.
118. Hu P-W, Chan KCC, You Z-H. Large-scale prediction of drug-target interactions from deep representations. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. Vancouver, BC, Canada: IEEE, 2016, 1236–43.
119. Wang L, You Z-H, Chen X, *et al*. A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network. *J Comput Biol* 2017, in press.
120. Wen M, Zhang Z, Niu S, *et al*. Deep learning-based drug-target interaction prediction. *J Proteome Res* 2017;**16**:1401–9.
121. Cheng Z, Zhou S, Wang Y, *et al*. Effectively identifying compound-protein interactions by learning from positive and unlabeled examples. *IEEE/ACM Trans Comput Biol Bioinform* 2016, doi: 10.1109/TCBB.2016.2570211.
122. Lan W, Wang J, Li M, *et al*. Predicting drug–target interaction using positive-unlabeled learning. *Neurocomputing* 2016;**206**: 50–7.
123. Liu H, Sun J, Guan J, *et al*. Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics* 2015;**31**(12):i221–9.
124. Coelho ED, Arrais JP, Oliveira JL. Computational discovery of putative leads for drug repositioning through drug-target interaction prediction. *PLoS Comput Biol* 2016;**12**(11): e1005219.

125. Yang J, Roy A, Zhang Y. BioLiP: A semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic Acids Res* 2013;**41**(D1):D1096–103.

126. Chen X, Liu M, Gilson MK. BindingDB: A web-accessible molecular recognition database. *Comb Chem High Throughput Screen* 2001;**4**(8):719–25.

127. Davis MI, Hunt JP, Herrgard S, *et al*. Comprehensive analysis of kinase inhibitor selectivity. *Nat Biotechnol* 2011;**29**(11): 1046–51.

128. Metz JT, Johnson EF, Soni NB, *et al*. Navigating the kinome. *Nat Chem Biol* 2011;**7**(4):200–2.

129. Bolton EE, Wang Y, ThiessenPA, *et al*. PubChem: Integrated platform of small molecules and biological activities. Annual Reports in Computational Chemistry, Vol. 4. Amsterdam: Elsevier, 2008, 217–241.

130. Tabei Y, Kishimoto A, Kotera M, *et al*. Succinct interval-splitting tree for scalable similarity search of compound-protein pairs with property constraints. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, 176–184.

131. Smith MA, Shneiderman B, Milic-Frayling N, *et al*. Analyzing (social media) networks with NodeXL. In: *Proceedings of the Fourth International Conference on Communities and Technologies*. ACM, University Park, PA, 255–264.

132. Lopes CT, Franz M, Kazi F, *et al*. Cytoscape Web: an interactive web-based network browser. *Bioinformatics* 2010;**26**(18): 2347–8.

133. Thakral S, Ghoshal K. miR-122 is a unique molecule with great potential in diagnosis, prognosis of liver disease, and therapy both as miRNA mimic and antimir. *Curr Gene Ther* 2015;**15**(2):142–50.

134. Gomez IG, MacKenna DA, Johnson BG, *et al*. Anti–microrna-21 oligonucleotides prevent alport nephropathy progression by stimulating metabolic pathways. *J Clin Investig* 2015; **125**(1):141.

135. Kole R, Krieg AM. Exon skipping therapy for duchenne muscular dystrophy. *Adv Drug Deliv Rev* 2015;**87**:104–7.

136. Lentz JJ, Jodelka FM, Hinrich AJ, *et al*. Rescue of hearing and vestibular function by antisense oligonucleotides in a mouse model of human deafness. *Nat Med* 2013;**19**(3): 345–50.

137. Matsui M, Corey DR. Non-coding RNAs as drug targets. *Nat Rev Drug Discov* 2017;**16**(3):167–79.

138. Chen X, Sun Y-Z, Zhang D-H, *et al*. NRDTD: a database for clinically or experimentally supported non-coding RNAs and drug targets associations. *Database* 2017, doi: 10.1093/database/bax057.

139. Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. In: *ICDM 2008 Eighth IEEE International Conference on Data Mining*. Pisa, Italy: IEEE, 2008, 263–72.

140. Herlocker JL, Konstan JA, Terveen LG, *et al*. Evaluating collaborative filtering recommender systems. *ACM Trans Inform Sys* 2004;**22**(1):5–53.

141. Jin G, Wong STC. Toward better drug repositioning: prioritizing and integrating existing methods into efficient pipelines. *Drug Discov Today* 2014;**19**(5):637–44.

142. Shameer K, Readhead B, Dudley JT. Computational and experimental advances in drug repositioning for accelerated therapeutic stratification. *Curr Topics Med Chem* 2015;**15**(1): 5–20.

143. Li J, Zheng S, Chen B, *et al*. A survey of current trends in computational drug repositioning. *Brief Bioinform* 2016;**17**(1):2–12.

144. Yamanishi Y, Kotera M, Moriya Y, *et al*. DINIES: drug-target interaction network inference engine based on supervised analysis. *Nucleic Acids Res* 2014;**42**(W1):W39–45.

145. Cobanoglu MC, Oltvai ZN, Lansing Taylor D, *et al*. BalestraWeb: Efficient online evaluation of drugtarget interactions. *Bioinformatics* 2015;**31**(1):131.

146. Nickel J, Gohlke B-O, Erehman J, *et al*. SuperPred: Update on drug classification and target prediction. *Nucleic Acids Res* 2014;**42**(W1):W26–31.

147. Nagamine N, Shirakawa T, Minato Y, *et al*. Integrating statistical predictions and experimental verifications for enhancing protein-chemical interaction predictions in virtual screening. *PLoS Comput Biol* 2009;**5**(6):1–11, 06.

148. Gfeller D, Grosdidier A, Wirth M, *et al*. SwissTarget Prediction: a web server for target prediction of bioactive small molecules. *Nucleic Acids Res* 2014;**42**(W1):W32–8.

149. Liu Y-t, Li Y, Huang Z-f, *et al*. Multi-algorithm and multi-model based drug target prediction and web server. *Acta Pharmacol Sin* 2014;**35**(3):419–31.

150. 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**(3):S4.

151. Nagamine N, Sakakibara Y. Statistical prediction of protein-chemical interactions based on chemical structure and mass spectrometry data. *Bioinformatics* 2007;**23**(15):2004–12.

152. Faulon J-L, Misra M, Martin S, *et al*. Genome scale enzyme–metabolite and drug–target interaction predictions using the signature molecular descriptor. *Bioinformatics* 2008;**24**(2): 225–33.

153. Campillos M, Kuhn M, Gavin A-C, *et al*. Drug target identification using side-effect similarity. *Science* 2008;**321**(5886): 263–6.