Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

DTI-Voodoo: machine learning over interaction networks and ontology-based background knowledge predicts drug–target interactions

Tilman Hinnerichs [1,*] and Robert Hoehndorf [2]

[1]Department, Institution, City, Post Code, Country and
[2]Computational Bioscience Research Center, Computer, Electrical and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology, 4700 King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: *In silico* drug–target interaction (DTI) prediction is important for drug discovery and drug repurposing. Approaches to predict DTIs can proceed indirectly, top-down, using phenotypic effects of drugs to identify potential drug targets, or they can be direct, bottom-up and use molecular information to directly predict binding potentials. Both approaches can be combined with information about interaction networks.
Results: We developed DTI-Voodoo as a computational method that combines molecular features and ontology-encoded phenotypic effects of drugs with protein–protein interaction networks, and uses a graph convolutional neural network to predict DTIs. We demonstrate that drug effect features can exploit information in the interaction network whereas molecular features do not. DTI-Voodoo is designed to predict candidate drugs for a given protein; we use this formulation to show that common DTI datasets contain intrinsic biases with major affects on performance evaluation and comparison of DTI prediction methods. Using a modified evaluation scheme, we demonstrate that DTI-Voodoo improves substantially over state of the art DTI prediction methods.
Availability: DTI-Voodoo source code and data necessary to reproduce results are freely available at `https://github.com/THinnerichs/DTI-VOODOO`.
Contact: tilman.hinnerichs@kaust.edu.sa
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Identifying drug–target interactions (DTIs) is important in drug repurposing and repositioning, and in the exploration of novel drug candidates and their interaction partners. Computational methods are widely applied to predict DTIs and many computational methods have been developed. These methods can be broadly classified into "top-down" and "bottom-up" approaches. Top-down approaches start from observable characteristics resulting from a drug–target interaction, such as side-effects or the diseases treated by a drug, and infer likely molecular mechanisms (i.e., the interaction) using these observations. Bottom-up approaches start from molecular features such as molecular structure or fingerprints associated with drug and protein, and predict interactions from this information.

Both bottom-up and top-down approaches to drug–target interaction prediction bear some advantages and limitations. Generally, bottom-up methods face the challenge to predict whether a structure binds to a protein given their molecular properties; whether two entities interact depends not only on the molecular structure of the entities (where binding sites and molecular forces need to be determined for accurate prediction) but also properties namely where a protein is expressed or in which celltypes and anatomical structures. Top-down methods use information for DTI prediction such as side-effect similarity (Campillos *et al.*, 2008) that is largely complementary to knowledge gained from molecular properties. While methods that rely on molecular information are directly predicting whether two molecular could interact, top-down methods base on more indirect means and infer a DTI from observable effects resulting from the interaction.

Both approaches may be used in conjunction with network inference (Chen *et al.*, 2015). Biological networks used for DTI prediction include protein–protein interaction networks (Feng *et al.*, 2017; Lee and Nam, 2018) and networks including several other types of biological relations, including similarity between represented entities (Ding *et al.*, 2013; Gottlieb *et al.*, 2011). Network-based DTI prediction methods use the guilt-by-association principle (Oliver, 2000) and assume that a protein is a likely target for a drug if many of the protein's neighbors in the interaction network are targets of the drug (Gillis and Pavlidis, 2012). Network-based methods have been applied successfully to DTI prediction. However, if DTIs are taken as direct physical interactions between a drug and protein, it remains an unresolved question whether the network-based guilt-by-association hypothesis is true, or whether an interaction of a drug and protein dysregulates several of the protein's interaction partners, and therefore resulting in effects that are not direct interactions but only downstream consequences of an interaction.

Progress in machine learning using graph neural networks can allow us to test this hypothesis and combine both bottom-up and top-down features with a network in a single machine learning model. In particular, Graph Convolutional Networks (Kipf and Welling, 2016) and their variants operate on different types of kernels (Defferrard *et al.*, 2016; Bianchi *et al.*, 2019), including attention mechanisms (Veličković *et al.*, 2017), and different forms of exploring node neighborhoods (Klicpera *et al.*, 2018; Hamilton *et al.*, 2017) can combine different types of features and graph-based information. They have previously been applied for a number of tasks, including prediction of protein functions (Zitnik and Leskovec, 2017), cancer drug response (Liu *et al.*, 2020) and drug–target affinity prediction (Nguyen *et al.*, 2020).

Potential biases resulting from the underlying datasets (Pahikkala *et al.*, 2014) which may affect model evaluation and comparison pose a challenge for DTI prediction. Firstly, novel drugs are often developed by altering non-functional components of a drug, leading to two and more very similar drugs designed to target the same proteins (Overington *et al.*, 2006). This can result in a bias when it leads to hidden duplicates or highly similar compounds that are distributed among training and evaluation dataset, resulting in a better (measured) predictive performance than it would be expected when the model is applied to identify drugs that target a protein for which no drugs yet exist. Secondly, some proteins (which we call *hub proteins*) have significantly more known interactions with drugs than others. In the STITCH database, 5% of the proteins have 40% of the interactions, and similar distributions are present in other datasets (Wishart *et al.*, 2007, 2017); preferentially predicting these proteins may increase predictive performance while not reflecting the actual performance when applied to a new protein (i.e., a protein for which no interactions are known). These differences in the number of drugs targeting certain proteins may be the result of study bias where more "valuable" proteins have more drugs designed to target them due to their involvement in more common diseases (or diseases for which drugs can be more profitably marketed). This might affect common evaluation schemes (Wang and Kurgan, 2018) where it is possible to exploit these biases within DTI prediction ()

cite or delete

. van Laarhoven and Marchiori (2014) showed that several bias can be exploited on the dataset of Yamanishi *et al.* (2008).

We developed DTI-Voodoo as a method for predicting DTIs. We use an ontology-based machine learning method (Chen *et al.*, 2020) to encode phenotypic consequences of DTIs and deep learning methods to encode molecular features. We combine both using a protein interaction network which we exploit with the aid of a graph neural network. We use this model to test whether molecular or phenotype features benefit from the network information and find that only phenotype features localize on the graph whereas molecular features do not. We further evaluate and compare DTI-Voodoo against several DTI prediction methods and demonstrate a substantial improvement of DTI-Voodoo over the state of the art in predicting drugs that target a protein. We also identify and characterize several biases in both training and evaluating DTI prediction methods, and make recommendations on how to avoid them. DTI-Voodoo is available as Free Software at `https://github.com/THinnerichs/DTI-VOODOO`.

2 Methods

2.1 Problem Description

DTI-Voodoo aims to solve the following problem: for a given drug and a given protein we want to determine whether those interact or not. We do not differentiate between types of interaction such as activation and inhibition, and do not predict the strength of the interaction. We treat all drug–protein pairs without a known interaction as negatives and therefore formulate the problem as a binary classification task.

2.2 Datasets

We obtain a dataset consisting of 12,884 human proteins with 340,627 links from STRING (Szklarczyk *et al.*, 2014). For the drug-target interactions, we use 229,870 links from the STITCH database (Szklarczyk *et al.*, 2015). As both STRING and STITCH provide confidence scores for each association, we filtered them as advised by a threshold of 700, therefore retaining only high-confidence interactions.

We utilize the PhenomeNET ontology (Hoehndorf *et al.*, 2011), an ontology integrating ontologies such as the Human Phenotype Ontology (Köhler *et al.*, 2018), Gene Ontology (Ashburner *et al.*, 2000; Carbon *et al.*, 2020), Mammalian Phenotype Ontology (Smith and Eppig, 2009) and several others. We obtained side effects and their links to drugs from SIDER (Kuhn *et al.*, 2015); SIDER contains side effects encoded using identifiers from the MedDRA database (Mozzicato, 2009). We mapped side effects to the PhenomeNET ontology using the *Phenomebrowser.net*, which provides a SPARQL query endpoint for the mentioned resources.

For comparative evaluation, we use the gold standard dataset introduced by Yamanishi *et al.* (2008) consisting of 1,923 interactions between 708 drugs and 1,512 proteins, and the BioSnap dataset (Zitnik *et al.*, 2018) which consists of 5,017 drug nodes, 2,324 gene nodes and 15,138 edges.

We only use proteins in our analysis that have at least one link in STRING or one association in PhenomeNET, and drugs with at least one side effect. Therefore, the intersection between these resources yields 1,428 drugs and 7,368 human proteins with 32,212 interactions for STITCH, 1,837 interactions between 680 drugs and 1,458 proteins for Yamanishi, and 6,498 links between 949 drugs and 2,221 proteins for BioSnap dataset. We provide links to and methods for obtaining and processing the necessary data on Github.

2.3 Model

Our model combines "top-down" and "bottom-up" information for drug–target identification. We consider an approach to be "top-down" when observable characteristics of either a drug (such as a drug effect) or protein (such as a protein function, or phenotypes resulting from a loss of function) are used to provide information about a molecular mechanisms; we consider an approach "bottom-up" when structural or other molecular information is used to determine a mechanism. In order to build a method that incorporates both top-down and bottom-up features, we first create a model for each type of feature separately. As features for the bottom-up model, we use features derived from molecular structures of drugs from the *SmilesTransformer* (Honda *et al.*, 2019) and molecular features for proteins from *DeepGOPlus* (Kulmanov and Hoehndorf, 2019). *SmilesTransformer* is an autoencoder trained over the SMILES strings, and therefore captures

(some aspects of) the molecular organization of each drug in an unsupervised manner. *DeepGOPlus* provides features derived from protein amino acid sequences which are useful to predict protein function.

As phenotypes and functions are encoded through ontologies, we use DL2Vec (Chen *et al.*, 2020) to obtain ontology based representations for use as top-down features. DL2vec constructs a graph by introducing nodes for each ontology class and edges for ontology axioms, followed by random walks starting from each node in the graph. These walks are encoded using a Word2vec (Mikolov *et al.*, 2013) model. Therefore, DL2Vec generates representations that enable to encode drug effects or protein functions while preserving their semantic neighborhood within that graph.

2.3.1 Half-twin neural networks and feature transformation

As we want to learn from the similarity of drug side effects and protein phenotypes, we use a deep half-twin neural network with a contrastive loss using cosine similarity. A half-twin neural network aims to learn a similarity between two embeddings of variable but same dimension. As the original feature space may have varying dimensionality, we first process them using a fully connected neural network layer which takes as input an embedding and outputs a representation of a particular size, i.e., we use this layer as a trainable feature transformation and apply it to reduce the representation size of the embeddings for drugs and proteins separately. An example structure for both types of features can be found in Supplementary Figure 2. The use of this trainable feature transformation layer enables flexible experimentation as both ontology and molecular feature for both drugs and proteins are reduced to the same dimensionality for varying sizes of inputs; this allows for a high amount of modularity across different experimental setups by adding different kinds of features into the model. Additionally, the generated features may be used for other tasks. We follow the results of *DL2vec* (Chen *et al.*, 2020) and use $\sigma := \mathrm{LeakyReLU}$ as activation function which leads to improved performance compared to other activation functions.

2.3.2 Graph convolutional layers

We include these molecular and ontology-based sub-models within a graph neural network (GNN) (Kipf and Welling, 2016). The graph underlying the GNN is based on the protein–protein interaction (PPI) graph. The PPI dataset is represented by a graph $G = (V, E)$, where each protein is represented by a vertex $v \in V$, and each edge $e \in E \subseteq V \times V$ represents an interaction between two proteins. Additionally, we introduce a mapping $x : V \to \mathbb{R}^d$ projecting each vertex v to its node feature $x_v := x(v)$, where d denotes the dimensionality of the node features.

A graph convolutional layer (Kipf and Welling, 2016) consists of a learnable weight matrix followed by an aggregation step, formalized by

$$\mathbf{X}' = \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{\Theta} \tag{1}$$

where for a given graph $G = (V, E)$, $\hat{A} = A + I$ denotes the adjacency matrix with added self-loops for each vertex, D is described by $\hat{D}_{ii} = \sum_{j=0} \hat{A}_{ij}$, a diagonal matrix displaying the degree of each node, and Θ denotes the learnable weight matrix. Added self-loops enforce that each node representation is directly dependent on its own preceding one. The number of graph convolutional layers stacked equals the radius of relevant nodes for each vertex within the graph.

The update rule for each node is given by a message passing scheme formalized by

$$\mathbf{x}'_i = \mathbf{\Theta} \sum_j^N \frac{1}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{x}_j \tag{2}$$

where both $\hat{d}_i, \hat{d}_j$ are dependent on the edge weights e_{ij} of the graph. With simple, single-valued edge weights such as $e_{ij} = 1 \; \forall (i, j) \in E$, all $\hat{d}_i$ reduce to d_i, i.e., the degree of each vertex i. We denote this type of graph convolutional neural layers with GCNConv.

While in this initial formulation of a GCNConv the node-wise update step is defined by the sum over all neighboring node representations, we can alter this formulation to other message passing schemes. We can rearrange the order of activation function σ, aggregation AGG, and linear neural layer MLP with this formulation as proposed by Li *et al.* (2020a):

$$\mathbf{x}'_i = \mathrm{MLP}\left(\mathbf{x}_i + \mathrm{AGG}\left(\left\{\sigma\left(\mathbf{x}_j + \mathbf{e_{ji}}\right) + \epsilon : j \in \mathcal{N}(i)\right\}\right)\right) \tag{3}$$

where we only consider $\sigma \in \{\mathrm{ReLU}, \mathrm{LeakyReLU}\}$. We denote this generalized layer type as GENConv following the notation of PyTorch Geometric (Fey and Lenssen, 2019). While the reordering is mainly important for numerical stability, this alteration also addresses the vanishing gradient problem for deeper convolutional networks (Li *et al.*, 2020a). Additionally, we can also generalize the aggregation function to allow different weighting functions such as learnable $\mathrm{SoftMax}$ or Power for the incoming signals for each vertex, substituting the averaging step in GCNConv. Hence, while GCNConv suffers from both vanishing gradients and signal fading for large scale and highly connected graphs, each propagation step in GENConv emphasizes signals with values close to 0 and 1. The same convolutional filter and weight matrix are applied to and learned for all nodes simultaneously. We further employ another mechanism to avoid redundancy and fading signals in stacked graph convolutional networks, using residual connections and a normalization scheme (Li *et al.*, 2019, 2020b). The residual blocks are reusable and can be stacked multiple times.

2.3.3 Combined prediction model

Combining half-twin and graph convolutional neural networks, we map all protein representations to their respective node features, initializing the graph convolutional update steps. The resulting representations are used for a similarity prediction. When combining ontology and molecular features with or without the graph model, we concatenate both protein features and both drugs features, before plugging them into the graph model for the similarity computation. An overview of the model architecture, combining both feature types, is shown in Supplementary Figure 4. Here the original representations are transformed by a dense layer and then used as input of a stack (with height 3) of residual graph convolutional blocks.

2.3.4 Hyperparameter tuning

As the number of drug-targets are sparse with respect to the amount of both drugs and proteins considered, the training, validation and testing datasets are imbalanced. As there are only $22,336$ links in the considered STITCH subset, the ratio

$$w := \frac{\#drugs \cdot \#proteins}{\#dti\_links} \approx 360, \tag{4}$$

consequently needs suitable compensation in the computed loss function and appropriate metrics for the evaluation.

Therefore, we weight all positive drug-protein pair samples with this ratio by introducing the following loss function with respect to binary cross-entropy:

$$l(x, y) = -w\left[y \cdot \log x + (1 - y) \cdot \log(1 - x)\right] \tag{5}$$

for a given prediction x and target y, and positive weight w defined by equation (4). We average this loss among all drug-protein pairs in the training set, leading to a stable environment for the *Adam* optimization algorithm (Kingma and Ba, 2014). We implemented a 5-fold cross validation split among the proteins. Furthermore, we used early stopping in the training process.

To find the best hyperparameter configuration for the proposed model, we performed a grid search to find the most expressive and non-redundant representation. We pretrained the bottom-up and the top-down model separately and aimed at best performing models with respect to our evaluation metrics. We optimized embedding sizes, depth of the neural network, optimizer, learning rate and layer types using an extensive, manual grid search. Starting from shallow feature transformations with an embedding size of 10, we scaled the network up to residual structures with up to 10 hidden layers leading to embeddings of size 4000, testing different network widths and learning rates for each configuration.

2.4 Evaluation and metrics

2.4.1 Splitting schemes

As DTI prediction is dependent on both drugs and proteins, there are multiple ways of determining training, validation and testing sets of pairs to evaluate each model. For cross-validation, we can perform the split over DTI pairs, over drugs and over proteins, respectively; when splitting over drugs or proteins, the entities (drugs or proteins) are separated and all their associations included in the split. Only when splitting by protein or drug are unseen entities guaranteed to be shown to the model in the validation and testing phase. The models resulting from the different splitting schemes may have different expressiveness and exploit different information in DTI prediction, as different information is known in the training and testing phase.

2.4.2 Metrics

To assess each model, we compute a variety of common metrics for binary classification. As the datasets are highly imbalanced, we use the area under the receiver operating characteristic curve (AUROC) on training, validation and testing split.

We calculate the AUROC by computing true positive rate at various false positive rate thresholds and use trapezoidal approximations to estimate the area under the curve. We refer to this measure as MacroAUC.

We also calculate the MicroAUC score. For given lists D and P of drugs and proteins, respectively, and a set of known interactions $Int := \{(d_i, p_i)\}$, *MicroAUC* is calculated as the average per entity *AUROC* score. For example, the protein-centric score can be formalized as: given labels $l : D \times P \to \{0, 1\}$ and predictions $y : D \times P \to [0, 1]$, we define

$$\text{MicroAUC}'_p(l, y) := \underset{p \in P}{mean}\left(\{\text{AUROC}(\{(l(d_i, p), y(d_i, p))|d_i \in D\})\}\right)$$

In some cases, the $MicroAUC$ score may not be defined as in some datasets some proteins or drugs have no interactions, leading to an infeasible $TPR = 0$ for all thresholds and an undefined $AUROC$ score for that entity. As this is quite common for DTI datasets, we do not omit but impute the $MicroAUC$ interpolating linearly for those entities using the accuracy for this subset:

$$\text{MicroAUC}_p(l, y) := \begin{cases} MicroAUC'_p(l, y) & \text{if } \sum_{d_i \in D} l(d_i, p) \neq 0 \\ Accuracy(l, y) & otherwise \end{cases}$$

Note that drugs and proteins can be interchanged in this formulation. We refer to the different measures as protein-centric microAUC ($MicroAUC_p$) and a drug-centric microAUC ($MicroAUC_d$).

Further, we choose the AUROC as primary metric to compare different methods over other measures such as the area under precision recall curve (AUPRC); AUPRC is sensitive to data imbalances (Jeni *et al.*, 2013) and therefore more challenging to apply to comparing different DTI prediction methods.

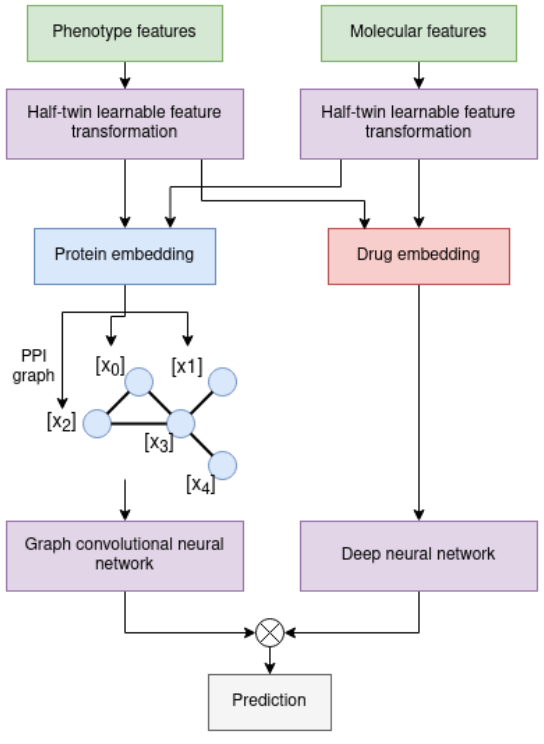

Figure 1. Full DTI prediction model based on the pretrained learnable feature transformations (LFT) for either molecular structure or ontology based features. The transformed protein representations are added to each corresponded protein as node features for the graph convolutional steps.

3 Results

3.1 DTI-Voodoo: computational model to identify drugs that target a protein

We developed DTI-Voodoo as a computational model to predict drug–target interactions. Specifically, given a protein, DTI-Voodoo will identify and rank drugs that likely target this protein. DTI-Voodoo combines two types of features: structural information for drugs and proteins that can be used to determine if the drug and protein physically interact, and information about phenotypic effects of drugs and changes in protein function that may "localize" on an interaction network (i.e., neighboring nodes will share some of these features or are phenotypically similar). As structural features, DTI-Voodoo uses structural representations of drugs from the SMILES transformer (Honda *et al.*, 2019) and representations of protein amino acid sequences from DeepGOPlus (Kulmanov and Hoehndorf, 2019). DTI-Voodoo learns representations of drug effects and protein functions using the ontology-based machine learning method DL2Vec (Chen *et al.*, 2020) and ontology-based annotations of drugs and proteins.

We construct a graph with proteins as nodes and protein-protein interactions as edges, mapping the protein features to each target as node features. DTI-Voodoo then propagates information among the PPI network utilizing graph convolutional steps, calculates the similarity of drug and protein representations, and predicts whether there is an interaction. The full workflow scheme is depicted in Figure 1

We evaluate our model's ability to identify drug–target interactions using different approaches and datasets. First, we perform a cross-validation over proteins and validate our results. A cross-validation over proteins aims to evaluate how the model performs when tasked to identify drugs

that may target a "novel" protein, i.e., one not seen during training, or a protein for which a drug that targets it should be predicted.

	(a) STITCH results				(b) Yamanishi results			
DTI-Voodoo results	PPI graph				PPI graph			
	without		with		without		with	
	Macro AUC	Micro AUC_p	Macro AUC	Micro AUC_p	Macro AUC	Micro AUC	Macro AUC	Micro AUC_p
MolPred	0.69	0.65	0.69	0.67	0.66	0.67	0.66	0.64
OntoPred	0.88	0.87	0.92	0.93	0.80	0.79	0.83	0.82
DTI-Voodoo	0.89	0.90	**0.93**	**0.94**	0.83	0.82	**0.84**	**0.84**

Table 1. Results for DTI-Voodoo on the STITCH and Yamanishi datasets evaluated with 5-fold cross-validation. We call the model using only molecular features MolPred and the model using only ontology-based features OntoPred. DTI-Voodoo combines both types of features.

We trained, validated and finally tested all considered models on the STITCH dataset using a 5-fold cross-validation over a protein split; we then selected the best-performing models (with respect to $MicroAUC_p$, see Section 2.4.2), and retrain them from scratch in a 5-fold protein-split cross-validation on the Yamanishi benchmark dataset to avoid validation overfitting and yield more realistic testing results. To evaluate the influence of the different features separately, and to determine whether they "localize" on the PPI graph (and therefore can be exploited successfully by the graph neural networks), we train and evaluate models with different types of features, and with and without inclusion of the PPI graph, separately. We comparing the molecular (MolPred) and phenotype-based (OntoPred) prediction model, and a combination of both where we concatenate both types of features. Table 1 shows the results of these experiments.

We find that the model using ontology-based features (*OntoPred*) is showing better performance on STITCH compared to using only molecular features. We also observe that only the model using ontology-based features results in increased performance when incorporating the PPI graph. This increase can be observed with different graph neural network architectures and configurations (Supplementary Table 1). While the *GCNConv* and *GENConv* architecture already shows some minor improvement, the use of *ResGraphConv* results in larger performance improvements. *ResGraphConv* blocks add a large amount of additional learnable parameters to the network, leading to more expressive power. To test whether the observed improvement is due to the number of learnable parameters added or the result of better exploiting the information about PPIs, we experiment with a graph model in which all graph convolutional neural layers in the residual blocks are removed, resulting in a model with similar parameters but without the ability to use graph-based information. This pruned network, with no information on the protein–protein interactions, reached very similar results to the original *OntoPred* model and showed no improvement (Supplementary Table 1).

The improvements when including the graph are only provided by the *GENConv* graph convolution scheme which includes the *ResGraphConv* blocks; *GCNConv* and other graph convolutional methods fail to achieve any gain in comparison to the plain *OntoPred* performance even when combined with the residual blocks. The discrepancy between *GENConv* and other graph convolutional methods may be the result of numerical instability and fading signals (Li *et al.*, 2020b).

Our results demonstrate that the inclusion of graph information can increase performance when ontology-based features are used but not when molecular features are used alone. This observation allows us to conclude that information about protein functions localizes on the graph whereas molecular features do not.

3.2 Protein-centric evaluation

The goal of DTI-Voodoo is to find candidate drugs that target a specific protein; however, so far, we do not evaluate this application but rather how DTI-Voodoo would perform in finding plausible drug–target interactions among all possible interactions (since we use the MacroAUC as our main evaluation measure). This evaluation does not correspond to the application of DTI-Voodoo in finding drugs that target a specific protein. To provide a better estimate on how DTI-Voodoo performs for individual protein targets, we use micro-averages between proteins and compute the *MicroAUC* (see Section 2.4.2); to determine *MicroAUC*, we average the performance (true and false positive rates) per protein instead of across all drug–protein pairs; the resulting measure can therefore better estimate how DTI-Voodoo performs when tasked with finding a drug that targets a specific protein.

Furthermore, we hypothesize that it may be possible for a machine learning model to exploit biases in drug–target interaction data to achieve relatively high prediction performance without obtaining a biologically meaningful signal. For example, hub proteins may have a large number of interactions, or certain drugs interact with many proteins, and preferentially predicting these interactions may increase predictive performance even in the absence of any biological features. To test this hypothesis, we design a "naïve" baseline model that predicts the same list of proteins for each drug based only on the number of known drug–target interactions for a protein. Specifically, given lists D and P of drugs and proteins and a set of known interactions $\mathcal{I} := \{(d_i, p_i)\}$, we construct an interaction matrix $M_{int} \in \{0,1\}^{|D|\times|P|}$ with

$$M_{ij} = \begin{cases} 1 & \text{if } (d_i, p_j) \in \mathcal{I} \\ 0 & otherwise \end{cases}$$

describing for all drug–protein pairs whether there is a known interaction or not. We now rank all proteins $p_j \in P$ descending by their number of drug interactors by summing over the columns of M_{ij} and ranking these sums:

$$f : P \to \mathbb{N} \text{ with } f : p_j \mapsto \sum_{i=1}^{|D|} M_{ij}$$

Our "naïve" predictor P_k predicts all drugs to interact with the top k targets with respect to the introduced ranking:

$$P_k : D \times P \to \{0,1\} \text{ with } P_k : d_i, p_j \mapsto \begin{cases} 1 & \text{if } p_j \in Top_k(P) \\ 0 & otherwise \end{cases}$$

with the only hyperparameter k.

The prediction P_k is not dependent on the drug d_i and will predict the same ranked list of drugs for all proteins; consequently, this naïve predictor does not rely on any biological features and will not predict any novel information about interactions between drugs and proteins; the naïve predictor only exploits imbalances in the evaluation set to make predictions that may perform well. The way in which we formulated the naïve predictor, it is not applicable for a protein split cross-validation as the number of interactions for each protein in the validation set is unknown.

We apply this naive predictor on both the STITCH and Yamanishi datasets, using the full datasets as well as a 5-fold cross-validation over drugs and over drug–protein pairs to compare the prediction results directly to DTI-Voodoo. For each fold, we gradually increase k to determine the best performance for each fold. Using the full dataset, drug–target split, and drug split, we obtain the following MacroAUC results: for the STITCH

database, we obtain a performance of 0.76 on the whole dataset, 0.70 for the drug–target pairs and 0.73 in case of the drug splitting scheme; on the Yamanishi dataset, we obtain MacroAUC scores of 0.88, 0.84 and 0.85, for the total dataset, drug–target pair and drug split, respectively. The naïve predictor shows higher performance on the Yamanishi dataset than on STITCH, and a substantial gain in comparison to an expected random predictor on both datasets. In the following, we utilize this naïve predictor as baseline to compare its performance to state of the art models and DTI-Voodoo.

For comparison with the state of the art methods, we chose the best performing methods for drug–target interaction prediction that were previously evaluated on the Yamanishi benchmark dataset. These methods include DTIGEMS+ (Thafar *et al.*, 2020) and DTI-CDF (Chu *et al.*, 2019) which have showed superior results in comparison to numerous works. Furthermore, we added DTINet (Luo *et al.*, 2017) as method for comparison which has been used to develop a number of methods such as NeoDTI (Wan *et al.*, 2019) with similar methodology.

We evaluate all models on their recommended splitting scheme choice, hyperparameters and folds in cross-validation, measuring their respective AUROC. We further evaluate each model by performing a protein-wise cross-validation determining the MacroAUC and $MicroAUC_p$. For this evaluation, we allow sub-sampling of negatives for the training process but not for the validation and testing phase as real world applications of these models would have to deal with possibly imbalanced data.

Approach	Original Splitting scheme	Original scheme Macro AUC	Protein split Macro AUC	Protein split Micro AUC_p
Naive predictor	Drugs	0.85	–	–
DTINet	DP pairs	0.91	0.74	0.67
DTIGEMS+	DP pairs	**0.93**	0.72	0.68
DTI-CDF	Proteins	0.85	**0.85**	0.79
DTI-Voodoo	Proteins	0.84	0.84	**0.84**

Table 2. Comparison of DTI-Voodoo with state of the art drug–target interaction prediction methods on the Yamanishi dataset; we evaluated the original and the protein-based split in a cross-validation

.

The results of our experiments are summarized in Table 2; we calculated the performance of all compared methods over their original splitting scheme and over a protein split. We find that there is a large difference in performance when evaluating over a drug–target pair split compared to a protein split, with generally higher performance achieved when using the drug–target pair split. Second, when evaluating the same methods over a protein split, we find a substantial performance difference in comparison to the splitting scheme used in the original evaluation of each method. DTI-CDF was originally evaluated on all three splitting schemes underlining this point (Chu *et al.*, 2019). While DTI-Voodoo provides comparable performance to the naïve predictor and DTI-CDF in terms of MacroAUC, it yields considerably better results with respect to MicroAUC$_p$. We also find that methods that are trained using a protein split generally result in higher MicroAUC$_p$ than methods trained using a drug–target pair split, indicating that they may generalize better to unseen protein targets whereas methods trained on a drug–target split potentially exploit hidden biases and therefore generalize less well.

As the difference in performance with different splitting schemes is quite large, we further evaluated additional drug–target interaction and drug–target affinity prediction methods that were trained and evaluation on other datasets. Following the results of MolTrans (Huang *et al.*, 2020), we reevaluated DeepDTI (Wen *et al.*, 2017), DeepDTA (Öztürk *et al.*, 2018),

Approach	Original Splitting scheme	Original scheme Macro AUC	Protein split Macro AUC	Protein split Micro AUC_p
Naive predictor	DP pairs	0.79	–	–
DeepDTI	Drugs	0.88	0.76	0.70
DeepDTA	DP pairs	0.88	0.77	0.69
DeepConv-DTI	DP pairs	0.88	0.76	0.73
MolTrans	DP pairs	**0.90**	0.77	0.74
DTI-Voodoo	Proteins	0.85	**0.85**	**0.82**

Table 3. Comparison of DTI-Voodoo with state of the art drug–target interaction prediction methods on the BioSnap dataset; we evaluated the original and the protein-based split in a cross-validation.

.

DeepConv-DTI (Lee *et al.*, 2019), and MolTrans itself on the BioSnap dataset (Zitnik *et al.*, 2018) and compared it to our "naïve" predictor as well as DTI-Voodoo (see Table 3). MolTrans was evaluated over the drug–target pair and the protein split; we were able to reproduce the MolTrans results (Table 3), showing a substantial difference based on the splitting scheme. We additionally computed the $MicroAUC_p$ score for all considered methods, leading to similar results as observed on the Yaminishi dataset. As DTI-Voodoo considers less drugs than alternative approaches, we perform a two-sided T-test yielding $p < 0.0001$ for $\alpha < 0.01$ showing the significance of DTI-Voodoo's performance gain over other state-of-the-art methods.

Formulation correct? Confidence interval?

In all our experiments, DTI-Voodoo has the highest Micro AUC_p, demonstrating that DTI-Voodoo can improve over other methods in the task of identifying drugs that target a specific protein. Several methods achieve a higher Macro AUC than DTI-Voodoo, in particular when evaluated using a drug–target pair splitting scheme; our results with the "naïve" prediction method show that it may be possible that models trained on a drug–target split utilize certain biases in the dataset without necessarily producing novel biological insights.

4 Discussion

4.1 "Bottom-up" and "top-down" prediction of drug-target interactions

There are many computational methods to predict drug–target interactions. They can broadly be grouped in two types; the first, which we refer to as "bottom-up" approaches, start from molecular information about a drug and protein and predict an interaction based on their molecular properties; the second, which we refer to as "top-down" approaches, start from observable characteristics of an organism and infer drug–target interactions as the putative molecular mechanisms that explain these observations.

Another view on these two approaches is as direct and indirect ways to predict drug–target interactions. On one hand, molecular information can be used to directly determine whether two molecules (such as a drug and protein) have the ability to interact, whereas information about phenotypic consequences of a drug (drug effects) or disruption of a protein function can be used to indirectly suggest candidate drug–target interactions. Molecular features will be specific to a drug–target pair and we would not expect this information to propagate through a protein–protein interaction network; the main information about drug–target interactions that could be obtained from interactions between proteins is information about binding sites between proteins that may also be used by a drug molecule (i.e., information that protein P_1 binds to protein P_2 reveals information about the molecular structures of both P_1 and P_2). On the other hand,

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

phenotypic consequences of changes in protein function, or drug effects, are often a result of aberrant pathway or network activity and involve more than one protein; consequently, we expect these features to benefit more from including information about protein–protein interactions. Our results (Table 1) confirm this hypothesis and demonstrate that molecular features do not benefit from including the interaction network whereas the indirect, top-down features benefit from the propagation over the interaction network.

There are other types of indirect features that could be added to our model. A common feature that may be added are drug indications which are predictive of drug–target interactions (Gottlieb *et al.*, 2011). However, we do not include them in our model as including drug indications would allow our model to make many trivial predictions based only on remembering which targets are often used for which indication; including network information would likely benefit predictions based on drug indications because different drugs may target the same pathway through different mechanisms.

Combining bottom-up and top-down approaches in a single model can follow different strategies. Interaction networks are used widely to determine indirect effects of molecular changes and predict drug–target interactions (Shahreza *et al.*, 2017). Our work relies on graph neural networks as a way to combine qualitative information about interactions with additional features (molecular interaction, phenotypic and functional features); even if only some of these features benefit from the information the graph provides, graph neural networks will allow further extension of our model with additional features in the future.

4.2 The challenge of evaluating drug–target interaction predictions

Why is having only having 20% of drugs reasonable -> significance test in results

One major component of our experiments was to determine how the information that is available to a machine learning model during training affects the performance of the model. Similarly to previous work (Huang *et al.*, 2020; Lee *et al.*, 2019), we find significant differences in predictive performance across different splitting schemes.

The most common scheme for drug–target interaction prediction is the split over drug-target pairs (Wang and Kurgan, 2018) where it may happen that most drugs and targets that are including in the model's validation and testing phase have also been included in the training phase (as part of other drug–target pairs). This scheme is prone to a number of biases. If the number of interactions for a drug or protein are imbalanced, i.e., some drugs or proteins have many more interactions than others, these will be seen more often during training and they will likely also have more interactions in the testing and validation sets; because some entities have more interactions, i.e., they are more likely to interact, any model that preferentially predicts these as interaction partners will improve its predictive performance. While this accurately captures the distribution, predicting based on biases in the number of interaction partners is not desirable when applying the model to novel entities. We have demonstrated that even a "naïve" classifier that makes predictions only by exploiting the imbalanced number of interaction partners can achieve performance close to state of the art methods (when measuring Macro AUC). When training a machine learning model on such imbalanced data, it will eventually overfit to this imbalance. Splitting by entity (protein or drug) can reduce the impact of this spurious correlation but not reduce it entirely, because similar entities will still exhibit similar interaction patterns. In our experiments, we observed the impact of splitting training and evaluation sets by protein as a decrease in overall performance (Macro AUC), providing some evidence that models trained using this splitting scheme are less sensitive to overfitting to this type of bias.

The way in which training and evaluation data is generated is related to how the model is evaluated. An evaluation based on Macro AUC evaluated the application scenario where a set of drugs and proteins are given, and out of all possible pairs, the more likely interactions need to be identified. However, this does not correspond to most scenarios for drug repurposing where a drug that targets a *specific* protein (e.g., a protein involved in a disease) needs to be identified. We use an evaluation measure based on micro-averages per protein (Micro AUC_p) to evaluate this scenario, and we often find substantial differences in predictive performance when evaluating with Macro AUC and Micro AUC_p; generally, models that are trained using a split over drug–target pairs perform worse in Micro AUC_p than models that use a protein-based split, further providing evidence that a drug–target split results in overfitting to dataset biases.

Finally, a potential source of differences in model performance is how negatives are identified and treated during evaluation (and training). There are few large sets of validated negative drug–target interactions; consequently, many models (including DTI-Voodoo) use all unknown interactions as negatives. As there are many more negative than positive interactions, negatives are then sub-sampled during training resulting in a training set that is balanced between positives and negatives (or a certain ratio is preserved). While this is a reasonable strategy to deal with imbalanced data, it may be problematic when the same sub-sampling is applied on the model's evaluation set because it over-simplifies the evaluation process. The performance differences is not usually visible when using ROC curves but results in unrealistically high precision and therefore high area under a precision-recall curve.

In summary, drug–target interaction prediction is not a single computational problem in bioinformatics but a set of related problems. Let P be a set of proteins P and D a set of drugs; one task can be to identify arbitrary pairs (p, d) with $p \in P$ and $d \in D$ that interact, another to identify a set of interacting drugs for each $p \in P$, and yet another to identify a set of proteins for each drug $d \in D$. The first task may be useful when no particular drug or protein is considered; the second task when searching for a drug that targets a specific (disease-associated) protein; and the third when aiming to find new applications for a given drug. The first task would best be evaluated using a Macro AUC, the second and third using a Micro AUC_p and Micro AUC_d.

5 Conclusions

We developed DTI-Voodoo as a machine learning model that combines molecular features and functional information with an interaction network using graph neural networks to predict drugs that may target specific proteins. In this task, DTI-Voodoo improves over several state of the art methods. We demonstrated that functional and phenotypic information localizes on the interaction network whereas molecular information does not. Moreover, we showed that drug–target interaction prediction datasets have some inherent biases that affect the performance of models. This led us to conclude that drug–target interaction prediction is not a single computational problem but a set of multiple problems. Experimental evaluation of drug–target interaction prediction methods must be carefully designed to reflect the problem the model aims to solve, and the interpretation of performance results should be aligned with the specific problem.

Acknowledgements

We acknowledge the use of computational resources from the KAUST Supercomputing Core Laboratory.

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

Funding

This work was supported by funding from King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. URF/1/3790-01-01 and URF/1/4355-01-01.

References

Ashburner, M. *et al.* (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, **25**(1), 25–29.

Bianchi, F. M. *et al.* (2019). Graph neural networks with convolutional arma filters.

Campillos, M. *et al.* (2008). Drug target identification using side-effect similarity. *Science*, **321**(5886), 263–266.

Carbon, S. *et al.* (2020). The gene ontology resource: enriching a GOld mine. *Nucleic Acids Research*, **49**(D1), D325–D334.

Chen, J. *et al.* (2020). Predicting candidate genes from phenotypes, functions and anatomical site of expression. *Bioinformatics*.

Chen, X. *et al.* (2015). Drug–target interaction prediction: databases, web servers and computational models. *Briefings in Bioinformatics*, **17**(4), 696–712.

Chu, Y. *et al.* (2019). DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features. *Briefings in Bioinformatics*, **22**(1), 451–462.

Defferrard, M. *et al.* (2016). Convolutional neural networks on graphs with fast localized spectral filtering.

Ding, H. *et al.* (2013). Similarity-based machine learning methods for predicting drug–target interactions: a brief review. *Briefings in Bioinformatics*, **15**(5), 734–747.

Feng, Y. *et al.* (2017). Drug target protein-protein interaction networks: A systematic perspective. *BioMed Research International*, **2017**, 1–13.

Fey, M. and Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Gillis, J. and Pavlidis, P. (2012). "guilt by association" is the exception rather than the rule in gene networks. *PLoS Computational Biology*, **8**(3), e1002444.

Gottlieb, A. *et al.* (2011). PREDICT: a method for inferring novel drug indications with application to personalized medicine. *Molecular Systems Biology*, **7**(1), 496.

Hamilton, W. L. *et al.* (2017). Inductive representation learning on large graphs.

Hoehndorf, R. *et al.* (2011). PhenomeNET: a whole-phenome approach to disease gene discovery. *Nucleic Acids Research*, **39**(18), e119–e119.

Honda, S. *et al.* (2019). Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery.

Huang, K. *et al.* (2020). MolTrans: Molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*.

Jeni, L. A. *et al.* (2013). Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks.

Klicpera, J. *et al.* (2018). Predict then propagate: Graph neural networks meet personalized pagerank.

Köhler, S. *et al.* (2018). Expansion of the human phenotype ontology (HPO) knowledge base and resources. *Nucleic Acids Research*, **47**(D1), D1018–D1027.

Kuhn, M. *et al.* (2015). The SIDER database of drugs and side effects. *Nucleic Acids Research*, **44**(D1), D1075–D1079.

Kulmanov, M. and Hoehndorf, R. (2019). DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*.

Lee, I. and Nam, H. (2018). Identification of drug-target interaction by a random walk with restart method on an interactome network. *BMC Bioinformatics*, **19**(S8).

Lee, I. *et al.* (2019). DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLOS Computational Biology*, **15**(6), e1007129.

Li, G. *et al.* (2019). Deepgcns: Can gcns go as deep as cnns?

Li, G. *et al.* (2020a). Deepergcn: All you need to train deeper gcns.

Li, G. *et al.* (2020b). Deepergcn: All you need to train deeper gcns.

Liu, Q. *et al.* (2020). DeepCDR: a hybrid graph convolutional network for predicting cancer drug response. *Bioinformatics*, **36**(Supplement_2), i911–i918.

Luo, Y. *et al.* (2017). A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information.

Mikolov, T. *et al.* (2013). Efficient estimation of word representations in vector space.

Mozzicato, P. (2009). MedDRA. *Pharmaceutical Medicine*, **23**(2), 65–75.

Nguyen, T. *et al.* (2020). GraphDTA: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*.

Oliver, S. (2000). Guilt-by-association goes global. *Nature*, **403**(6770), 601–602.

Overington, J. P. *et al.* (2006). How many drug targets are there? *Nature Reviews Drug Discovery*, **5**(12), 993–996.

Öztürk, H. *et al.* (2018). DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, **34**(17), i821–i829.

Pahikkala, T. *et al.* (2014). Toward more realistic drug-target interaction predictions. *Briefings in Bioinformatics*, **16**(2), 325–337.

Shahreza, M. L. *et al.* (2017). A review of network-based approaches to drug repositioning. *Briefings in Bioinformatics*, **19**(5), 878–892.

Smith, C. L. and Eppig, J. T. (2009). The mammalian phenotype ontology: enabling robust annotation and comparative analysis. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, **1**(3), 390–399.

Szklarczyk, D. *et al.* (2014). STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Research*, **43**(D1), D447–D452.

Szklarczyk, D. *et al.* (2015). STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data. *Nucleic Acids Research*, **44**(D1), D380–D384.

Thafar, M. A. *et al.* (2020). DTiGEMS: drug-target interaction prediction using graph embedding, graph mining, and similarity-based techniques. *Journal of Cheminformatics*, **12**(1).

van Laarhoven, T. and Marchiori, E. (2014). Biases of drug–target interaction network data. In *Pattern Recognition in Bioinformatics*, pages 23–33. Springer International Publishing.

Veličković, P. *et al.* (2017). Graph attention networks.

Wan, F. *et al.* (2019). NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions. *Bioinformatics*, **35**(1), 104–111.

Wang, C. and Kurgan, L. (2018). Review and comparative assessment of similarity-based methods for prediction of drug–protein interactions in the druggable human proteome. *Briefings in Bioinformatics*, **20**(6), 2066–2087.

Wen, M. *et al.* (2017). Deep-learning-based drug–target interaction prediction. *Journal of Proteome Research*, **16**(4), 1401–1409.

Wishart, D. S. *et al.* (2007). DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, **36**(suppl_1), D901–D906.

Wishart, D. S. *et al.* (2017). DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research*, **46**(D1), D1074–D1082.

Yamanishi, Y. *et al.* (2008). Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, **24**(13), i232–i240.

Zitnik, M. and Leskovec, J. (2017). Predicting multicellular function through multi-layer tissue networks.

Zitnik, M. *et al.* (2018). Biosnap datasets: Stanford biomedical network dataset collection. *Note: http://snap. stanford. edu/biodata Cited by*, **5**(1).

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture