Article

Deep learning-based drug-target interaction prediction

Ming Wen, Zhimin Zhang, Shaoyu Niu, Haozhi Sha, Ruihan Yang, Yonghuan Yun, and Hongmei Lu

J. Proteome Res., **Just Accepted Manuscript** • DOI: 10.1021/acs.jproteome.6b00618 • Publication Date (Web): 06 Mar 2017

Downloaded from http://pubs.acs.org on March 7, 2017

Journal of Proteome Research is published by the American Chemical Society. 1155 Sixteenth Street N.W., Washington, DC 20036

Deep learning-based drug-target interaction prediction

Ming Wen [1], Zhimin Zhang [1], Shaoyu Niu [1], Haozhi Sha [1], Ruihan Yang [1], Yonghuan Yun [2] and Hongmei Lu [1] *

[1] College of Chemistry and Chemical Engineering, Central South University, Changsha 410083, PR China; [2] Institute of Environment and Plant Protection, Chinese Academy of Tropical Agricultural Sciences, Haikou 571101, PR China.

ABSTRACT: Identifying interactions between known drugs and targets is a major challenge in drug repositioning. In silico prediction of drug target interaction (DTI) can speed up the expensive and time-consuming experimental work by providing most potent DTIs. In silico prediction of DTI can also provide insights about the potential drug-drug interaction and promote the exploration of drug side-effect. Traditionally, the performance of DTI prediction heavily depends on the descriptors used to represent the drugs and the target proteins. In this paper, to accurately predict new DTIs between approved drugs and targets without separating the targets into different classes, we developed a deep learning-based algorithmic framework named DeepDTIs. It firstly abstracts representations from raw input descriptors using unsupervised pre-training, and then applies known label pairs of interaction to build a classification model. Comparing with other methods, it is found that DeepDTIs reaches or outperforms other state-of-the-art methods. The DeepDTIs can be further used to predict whether a new drug targets to some existing targets or whether a new target interacts with some existing drugs.

KEYWORDS: Deep learning, Deep belief network, Feature extraction, Drug-target interaction prediction, Semi-supervised learning.

INTRODUCTION

Identification of interactions between drugs and targets is a key area in drug discovery and drug repositioning[1, 2]. Since approved drugs have clear availability and known safety profiles, the idea of recycling drugs to new indication could not only reduce drug development cost but also decrease the drug safety risk [3]. Although various biological assay technologies are available, there still remains the limitation of large scale drug-target interaction experiments. In addition, the very expensiveness of the experiment and few available public drug repositioning assay data make it necessary to develop appropriate computational tools which could precisely detect the interaction between drugs and targets.

Nowadays, many in silico approaches have been proposed to identify new drug-target interaction (DTI). Ligand-based and structure-based are two most used computational approaches[4]. The most known ligand-based method is to apply quantitative structure-activity relationship (QSAR) to predict the bioactivity of a molecule on a target. QSAR is based on hypothesis that molecules with similar structure have similar bioactivity[5]. Given a certain amount of targets, each target builds a predictive model using its known active molecules. Then these built models are used to screen all the drugs to predict the DTIs between drugs and targets [6]. Unfortunately, the performance of a built QSAR model is poor if the number of known active molecule of a target is not enough and most QSAR models are unspecific or predict activity against only one target. Some approaches have been applied to solve this problem such as built multitarget QSAR (mt-QSAR) classification model[7, 8]. Different from ligand-based method, structure-based method, i.e. molecular docking, uses the crystallographic structure of target to screen the small molecules[9, 10]. Molecular docking method is precise when the three-dimensional (3D) structure of a target is available. However, for most targets, especially for membrane proteins, like GPCRs, their 3D structure information is still unavailable up to date[11]. Recently, various network-based methods have been proposed to infer drug-target interactions. In a drug target interaction network, the drugs and targets are represented by nodes; the known interactions of drugs and targets corresponding to the lines which link the nodes. The new DTIs are inferred from the known network. For example, Cheng et al[12] developed a network based inference model (NBI) to infer new DTIs. NBI is only based on drug-target bipartite network topology similarity. A score function was used to score the association between a drug and a target. One disadvantage of NBI is that it could not be applied to the new drugs without any known target information in the training set. Some other networks can be integrated to improve the performance of the network methods. For example, Chen et al[13] used a heterogeneous network which integrated drug-drug similarity network, protein-protein similarity network and drug-target interaction network to develop an effective model named NRWRH. It obtained a significant performance improvement than using only one network. In addition, networks such as drug side-effect similarity network[14] and disease related molecular network[15] can also be applied to infer DTIs. More recently, with the increasing of experimental data, numerous machine learning methods have been applied to predict DTIs. The commonly used machine learning method is to build classification model. It takes drug target pairs (DTPs) as input. And the output is whether there is interaction between the drug target pair (DTP). The most applied machine learning model is binary classifiers such as random forest (RF)[16], support vector machine (SVM)[17] and artificial neural network (ANN)[18]. Deep learning method is a kind of ANN with multiple hidden layers and more sophisticated parameter training procedure. Deep learning method attracts a lot of attention for its relatively better performance and ability to learn representations of data with multiple levels of abstraction[19]. Deep learning has been applied in many fields of biology and chemistry. For example, Brendan J Frey et al[20] adapted a deep learning method named DeepBind to the task of predicting sequence specificities of DNA- and RNA-binding proteins. It was found that deep learning outperforms other state-of-the-art methods, even when training on in vitro data and testing on in vivo data. Jianlin Cheng et al[21] developed a deep learning network method (DN-Fold) and greatly improved the performance on protein fold recognition (predict if a given query-template protein pair belongs to the same structural fold).

Recently, with the development of many experimental instruments and technologies, such as high-throughput experiment and next generation sequencing, there is a tendency that integrates multiple resources to gather more information about DTIs. For example, Nascimento et al[1] integrated multiple heterogeneous information sources for the identification of new DTIs. Yamanishi et al[22] predicted DITs by kernels method, which incorporated multiple sources of information and measure drugs and targets similarity. The merit of integrating different resources is that it could detect information that the calculated features did not represent, or it could detect information that other resources do not keep. However, this may bring in error if the resources were not authority. In addition, it needs more domain knowledge to extract features from these resources. For example, with dozens of chemical structure and protein sequence descriptors in hand, it is hard to make decision which descriptor is useful in a certain task [23]. The ability to extract and organize information from the row data and to learn representations of data with multiple levels of abstraction makes deep learning method particularly meet the case of DTIs prediction.

Previously, most methods used the ‘golden standard’ dataset which first proposed by Y. Yamanishi et al to evaluate their proposed methods. In this dataset, the target had been divided into four categories - enzymes, ion channels, GPCRs and nuclear receptors. The corresponding protein numbers are 664, 204, 95 and 26, respectively. The number of the corresponding interactions are 1515, 776, 314 and 44, respectively. These datasets were widely used in a number of works[24, 25] and reached a relative high accuracy. However, given models built on different categories of targets, it cannot predict DTI when the original target and a new target of the drug belong to different categories. For example, serotonin and serotonergic are drugs which can interact with both 5-HT (a GPCR protein) and 59HT3A (an iron channel protein). 5-HT and 59HT3A are in different classes. If the GPCR class contains serotonin while the iron class does not contain, it could not predict serotonin - 59HT3A interaction using the GPCR class model. Thus, the built model has a relative low application domain and a global dataset should be reconsidered.

In this article, an effective deep learning method – deep belief network (DBN) was applied to accurately predict new DTIs between FDA approved drugs and targets without dividing the target into different classes. The developed approach is termed as DeepDTIs. The features of drugs and targets were automatically extracted from simple chemical sub-structure and sequence order information. To our knowledge, this is the first time to employ deep learning method to predict DTIs. We tested our method using independent test set and compared with some acceptable algorithm such as random forest (RF), Bernoulli Naïve Bayesian (BNB) and Decision Tree (DT)[26]. Moreover, our algorithm was also tested on an external EDTIs database (experimental drug target interactions database) which was extracted from DrugBank database. Finally, the prediction of all the possible interaction in the drug target space (DTS) was successfully carried out by DeepDTIs and the 10 most possible predicted DTIs have been partly validated by known experiments in the literatures.

METHODS

Dataset and Drug Target Space (DTS)

The drugs and targets data were extracted from the DrugBank database (http://www.drugbank.ca/)[27]. DrugBank database is a unique bioinformatics and cheminformatics resource that combines detailed drug data with comprehensive drug target information. The data used in present study is released in 2016-02-15. The interactions of drugs and targets were downloaded from Drug Target Identifiers category of Protein Identifiers in DrugBank download website (https://www.drugbank.ca/releases/latest#protein-identifiers). The approved drug structures and approved target sequences were downloaded from https://www.drugbank.ca/releases/latest#structures and https://www.drugbank.ca/releases/latest#target-sequences respectively. We manually discarded the drugs which are inorganic compounds (i.e. Lithium (DB01356)) or very small molecule compounds (i.e. Carbon dioxide (DB09157)). The descriptors of all approved drug structures and target sequences were calculated. If a downloaded drug target interaction pair has calculated protein and drug descriptors, it was used to build the model. Those datasets are also available at https://github.com/Bjoux2/DeepDTIs. The drug target space (DTS) is defined as all the possible drug-target pairs (DTPs). As shown in **Fig. 1**, the DTS were consisting of 1412 drugs and 1520 targets. DTS has 2146240 (that is, 1412×1520) DTPs. Among them, as depicted in **Fig. 1**, some pairs are positive DTIs (marked as YES), some pairs are not DTIs (marked as NO). Currently, there are 6262 DTPs which have known interaction, and the others are not known. The known 6262 DTPs were set as positive dataset. Because the number of no interaction pairs is far more than the number of interaction pairs, the negative dataset can be randomly selected from the DTS. In present study, we randomly select the 6262 DTPs from the DTS as negative dataset. Thus, the whole dataset has 12524 samples.

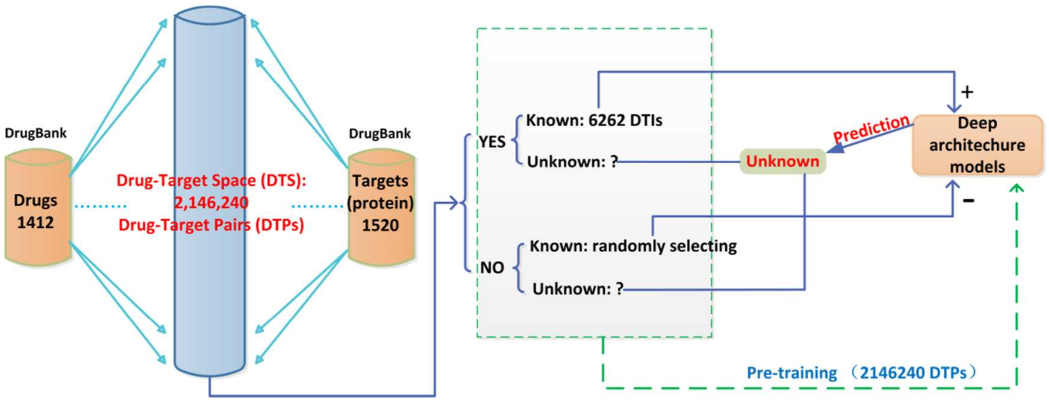


Fig. 1. The flowchart of DeepDTIs.

Experimental drug dataset

An external experimental drug target pairs (EDTPs) dataset was used to test our model. This dataset is derived from DrugBank. The download process and website are similar to the training dataset described above. An experimental drug target pair consists of an experimental drug and an experimental target or consists of an experimental drug and an approved target. The experimental drug is the one which has been shown experimentally bind to specific proteins (http://www.drugbank.ca/documentation). Though EDTPs are not true DTIs yet, they have a high probability to be true DTIs. The prediction of these EDTPs can partly evaluate the performance of our model. The EDTPs dataset contains 7352 experimental drug target interaction pairs. The EDTPs dataset consists of 2528 targets and 4383 experimental drugs. Among the 7352 pairs, 2444 pairs are set as EDTPs1 which consists of 504 targets which are included in the training dataset and 2003 drugs; 4908 pairs are set as EDTPs2 which consists of 2818 drugs and 2024 targets which are not included in the training dataset, In EDTPs2, these drugs and targets are new to that in the training set and were used to evaluate the applicability of the model.

Chemical structure and protein sequence representation.

The representation of chemical compound can be classified into two categories: molecular descriptors (MDs) and molecular fingerprints (MFs). MDs are experimentally-defined or theoretically-derived properties of a molecule. MFs are property profiles of a molecule. MFs are always represented by bit or count vectors with the vector elements indicating the existence or the frequencies of certain properties, respectively[28]. In the rest of this paper, to avoid ambiguity, we use feature to represent both descriptor and fingerprint.

In present study, we choose the most common and simple features – Extended Connectivity Fingerprints (ECFP) and protein sequence composition descriptors (PSC) for drugs and targets representation. ECFP are a class of topological fingerprints representing the presence of particular substructures[29]. ECFP describes features of substructures consisting of each atom and circular neighborhoods within a diameter range. We combined ECFP2, ECFP4 and ECFP6 (corresponding to diameters are 2, 4, 6, respectively) to ensure that we do not lose structure information. Each compound has 6144 fingerprints. The PSC consists of amino acid composition (AAC), dipeptide composition (DC) and tripeptide composition (TC). AAC is the statistic frequency of each amino acid. DC is the statistic frequency of every two amino acid combination. TC is the statistic frequency of every three amino acid combination. Each protein sequence has 8420 descriptors. Unlike any other features of molecule and protein, as described above, ECFP and PSC are two simple features which only contain molecular substructure and protein subsequence. They contain the information about molecule structure and sequence order. Finally, each DTP has 14564 features. The molecular fingerprints and protein descriptors were calculated using Open-Source Cheminformatics Software rdkit and protein descriptor generator propy[30], respectively.

RBM

Restricted Boltzmann Machine (RBM) is a graphical model that can learn a probability distribution from input data[31]. As is shown in **Fig. 2**, RBM consists of two layers – a visible layer and a hidden layer. Each visible unit is connected with the entire hidden unit. There are no visible-visible and hidden-hidden connections between the same layer. Given a RBM network, the energy $\mathrm{E}(v,h)$ of a RBM is defined as:

$$E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta}) = -\boldsymbol{b}'\mathbf{v} - \boldsymbol{c}'\mathbf{h} - \mathbf{h}'\boldsymbol{W}\mathbf{v} \tag{1}$$

where $\boldsymbol{\theta} = \{\boldsymbol{W},\boldsymbol{b},\boldsymbol{c}\}$. $\boldsymbol{W}$ represents the weights that connect hidden and visible units. $\boldsymbol{b}$ and $\boldsymbol{c}$ are the offsets of the visible and hidden layers, respectively. When parameter $\boldsymbol{\theta}$ is given, based on $E(\mathbf{v},\mathbf{h})$, the probability distribution of $(\mathbf{v},\ \mathbf{h})$ is:

$$P(\mathbf{v},\mathbf{h}|\boldsymbol{\theta}) = \frac{1}{z(\boldsymbol{\theta})} e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})} \tag{2}$$

$$z(\boldsymbol{\theta}) = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})} \tag{3}$$

where $z(\boldsymbol{\theta})$ is a normalizing factor called the partition function by analogy with physical systems. The probability that the network assigns to the visible layer $\mathbf{v}$ is given by summing over all possible hidden vectors.

$$P(\mathbf{v}|\boldsymbol{\theta}) = \frac{1}{z(\boldsymbol{\theta})} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})} \tag{4}$$

This function is also called likelihood function.

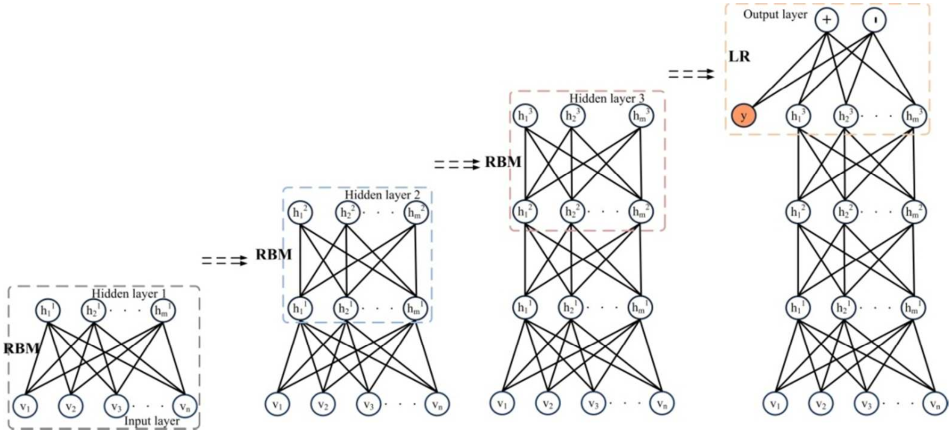


Fig. 2. DBN architecture. DBN is consisted by stacking many RBMs

A RBM model can be learnt by performing stochastic gradient descent (SGD) on the empirical negative log-likelihood of the training data. The loss function is defined as the negative log-likelihood function.

$$\boldsymbol{L}(\theta, T) = -\frac{1}{N}\sum_{\mathbf{v}\in T}\log P(\mathbf{v}|\boldsymbol{\theta}) \tag{5}$$

where T is a set of samples used in SGD. Then we update θ by:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \frac{\partial \boldsymbol{L}(\theta,T)}{\partial \boldsymbol{W}} \tag{6}$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \frac{\partial \boldsymbol{L}(\theta,T)}{\partial \boldsymbol{b}} \tag{7}$$

$$\boldsymbol{h} \leftarrow \boldsymbol{h} - \frac{\partial \boldsymbol{L}(\theta,T)}{\partial \boldsymbol{h}} \tag{8}$$

DBN

Deep Belief Networks (DBN) is a neural network which is consisted by stacking RBMs and trained in a greedy manner[32, 33]. **Fig. 2** is the architecture of DBN model. It consists of 5 layers. The first layer is the input layer which is the calculated features. The second, third and fourth layer are hidden layer. The last layer is output layer. Every adjacent two layers (except the last two layers) make up a RBM. DBN is graphical model that learn to extract a deep hierarchical feature of the training data. It models the joint distribution between training sample vector $\boldsymbol{x}$ and the l hidden layers as follows:

$$P(\boldsymbol{x}, \boldsymbol{h}^1, \dots, \boldsymbol{h}^l) = \left(\prod_{k=0}^{l-2} P(\boldsymbol{h}^k \middle| \boldsymbol{h}^{k+1})\right) P(\boldsymbol{h}^{l-1}, \boldsymbol{h}^l)$$

where $\boldsymbol{x} = \boldsymbol{h}^0$, $P(\boldsymbol{h}^{k-1}|\boldsymbol{h}^k)$ is a conditional distribution for the visible units conditioned on the hidden units of the RBM at level k, and $P(\boldsymbol{h}^{l-1}, \boldsymbol{h}^l)$ is the visible-hidden joint distribution in the top level RBM.

The training procedure of DBN can be separated into two consecutive processes: the greedy layer-wise unsupervised training process and the supervised fine-tuning process. The greedy layer-wise unsupervised training process is as follows:

1. Initializing parameter $\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{c}$ by using random generator.
2. Train the first and second layer as a RBM. Using the raw input vector $\boldsymbol{x}$ as its visible layer.
3. Train the second and third layer as a RBM, taking the second layer as visible layer and obtain the representation of third layer. Iterated for the desired number of layers.

The supervised fine-tuning process is as follows:

1. Using the output of the last hidden layer of the DBN as the input of the logistic regression classifier (LR).
2. Fine-tune all the RBM and LR parameters via supervised SGD of the DBN log-likelihood cost.

Measurement of prediction quality

Four frequently used evaluation metrics – area under the receiver operator characteristic curve (AUC), accuracy (ACC), true positive rate (TPR, sensitivity/recall) and false positive rate (TNR, specificity) – are used to assess the performance of the model in this study. The calculation formulas of ACC, TPR and TNR are following:

$$\mathrm{ACC} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\mathrm{TPR} = \frac{TP}{TP + FN}$$

$$\mathrm{TNR} = \frac{TN}{TN + FP}$$

where TP, FP, TN and FN represent true positive, false positive, true negative and false negative, respectively. In a two-class prediction problem, the outcomes are labeled either as positive (p) or negative (n). If the prediction and actual value are all p, it is called a TP; if the prediction value is p while the actual value is n, it is called a FP. Conversely, if the prediction and actual value are all n, it is called a TN; if the prediction value is n while the actual value is p, it is called a FN.

Implementation

The DBN algorithm was implemented in Python (version 2.7), using the famous DeepLearningTutorials package (https://github.com/lisa-lab/DeepLearningTutorials). The algorithm is coded based on Theano[34]. The algorithm is accelerated on the GPU (GEFORCE GTX-TITAN-X 6GD5) using CUDA. The operate system is Ubuntu Kylin 15.04 with 4.4 GHz Intel core i7 processor and 32G memory. The code is freely available on the web at https://github.com/Bjoux2/DeepDTIs.

RESULTS

Determining architectures of DeepDTIs

DeepDTIs has 3 hyper-parameters: (i) the mini-batches size (*mbs*); (ii) the pre-training learning rate (*plr*) and (iii) the fine-tune learning rate (*flr*). The architecture of DeepDTIs is mainly determined by 2 factors: (i) the number hidden layers (*nhl*) and (ii) the number of nodes of each layer (*nnl*). These hyper-parameters and factors were determined by grid-search on the validation error (In the rest of this paper, to avoid ambiguity, we use parameters to represent both hyper-parameters and factors mentioned above.). In grid-search, *mbs* is in [10, 32, 64, and 128]; the *plr* is in [10^{-1}, 10^{-2}, and 10^{-3}]; *flr* is in [10^{-1}, 10^{-2}, 10^{-3}, and 10^{-5}]; *nhl* is in [2, 3, 4, 5, and 6]. To reduce the number of points in the grid, we arbitrarily set the *nnl* as 2000. Thus, totally, 300 (4×3×5×5×1) models were built to optimize these parameters. To determine whether the effect of each parameter has a linear relationship with the performance of DeepDTIs, the parameters versus the model performances were plotted. **Fig. 3(a)** is the plot of the *nhl* versus the performance of 300 models, each point represents the performance of the model in a certain *nhl*. The line in the plot represents the first quartile (ranked from largest to smallest) model at different *nhl*. From **Fig. 3(a)**, there is no linear relationship between the model performances and *nhl*. Similar to *nhl*, *plr*, *flr* and *mbs* (**Fig. 3(b), 3(c) and 3(d)**) do not have the apparent relationships between model performances and parameters. Every value of a parameter can achieve a high performance. Thus, it is not easy to determine the model architecture and training parameter by experience and a grid-search strategy is recommended to optimize the model architecture and training parameter. The grid-search result was available at Supplementary Table S1. The optimized *plr*, *flr*, *nhl* and *mbs* are 0.1, 0.1, 4 and 128, respectively.

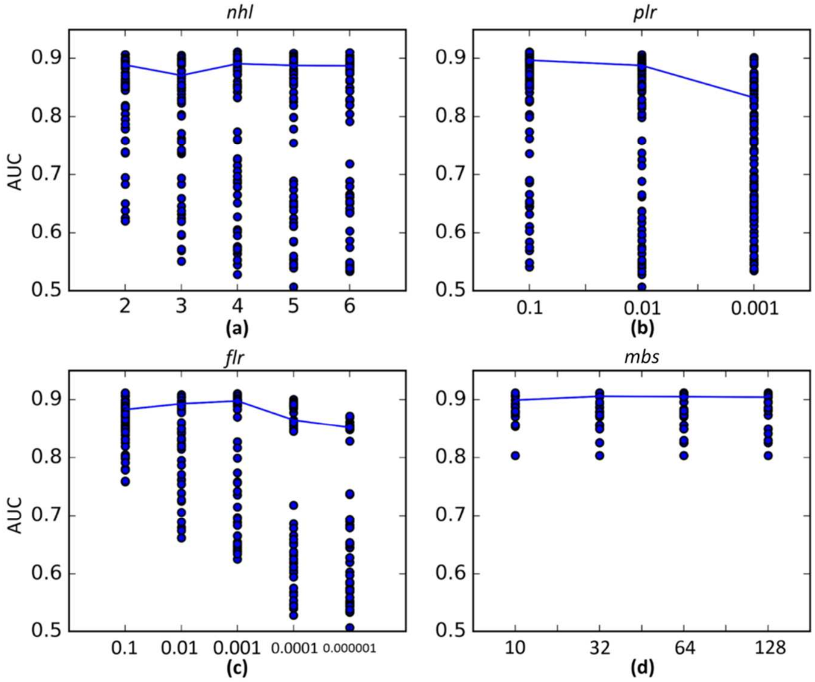


Fig. 3. the plot of parameter values versus the model performances.

Overall performance

The whole dataset (12524 samples) was split into three subsets – training set, validation set and test set, with the ratio 0.6 (7514 samples), 0.2 (2505 samples) and 0.2 (2505 samples), respectively. The training set is used to pre-train and fine-tune the model. The validation set is used to optimize the parameters. The test set is used to evaluate the model. In addition, to assess the robustness of each method and avoid the chance results obtained from data which randomly generated from DTS, we randomly generated 10 negative data and the average results were given. The performance of DBN model is listed in Table 1. The AUC, accuracy, sensitivity and specificity of test set are 0.9158, 0.8588, 0.8227 and 0.8953, respectively. In addition, to compare our model with other machine learning methods, Bernoulli Naive Bayesian (BNB) , Decision Trees (DT)[26] and RF were used to build classification models. The performances of these methods are also listed in Table 1. From Table 1, DBN is better than BNB and DT in AUC, ACC, TPR and TNR (P value < 0.05). Because the number of positive DTIs is much fewer than that of negative in DTS and the purpose of the model is to predict the true positive DTIs, TPR is a more important evaluation metric among the four evaluation metrics. Comparing with RF, though the AUC of DBN is not much better than RF (0.58% higher, P value < 0.05 DeepDTI versus RF), the TPR is much better than RF (1.71% higher, P value < 0.05 DeepDTI versus RF). **Fig. 4** is the plot of TPR, TNR, ACC and AUC of 4 methods. From **Fig. 4**, the performance fluctuated within a narrow range in 10 times and indicates that the random data split procedure is stable. Overall, DBN gained the best performance in TPR, TNR, ACC and AUC. This indicates that the built DBN model is reliable and can be further applied for novel DTIs prediction.

Table 1. the overall performance of BNB, DT, RF and DeepDTIs.

	TPR	TNR	ACC	AUC
BNB[a]	0.5913±0.0123	0.8654±0.0122	0.7272±0.0620	0.7544±0.0077
DT[a]	0.7949±0.0123	0.7418±0.0335	0.7684±0.0147	0.7683±0.0153
RF[b]	0.8056±0.0105	0.8636±0.0108	0.8342±0.0068	0.9100±0.0053
DBN	0.8227±0.0065	0.8953±0.0130	0.8588±0.0049	0.9158±0.0059

[a]The hyper-parameters are set default. [b]The hyper-parameter max feature is optimized in log2(q), sqrt(q), q/3, q/2 and q, where q is the number of variables. The other hyper-parameters are set default. Given a list X which contains the 10 prediction results using 10 negative datasets, the final result is represented as mean(X) ± max(max(X)-mean(X), mean(X)-min(X)).

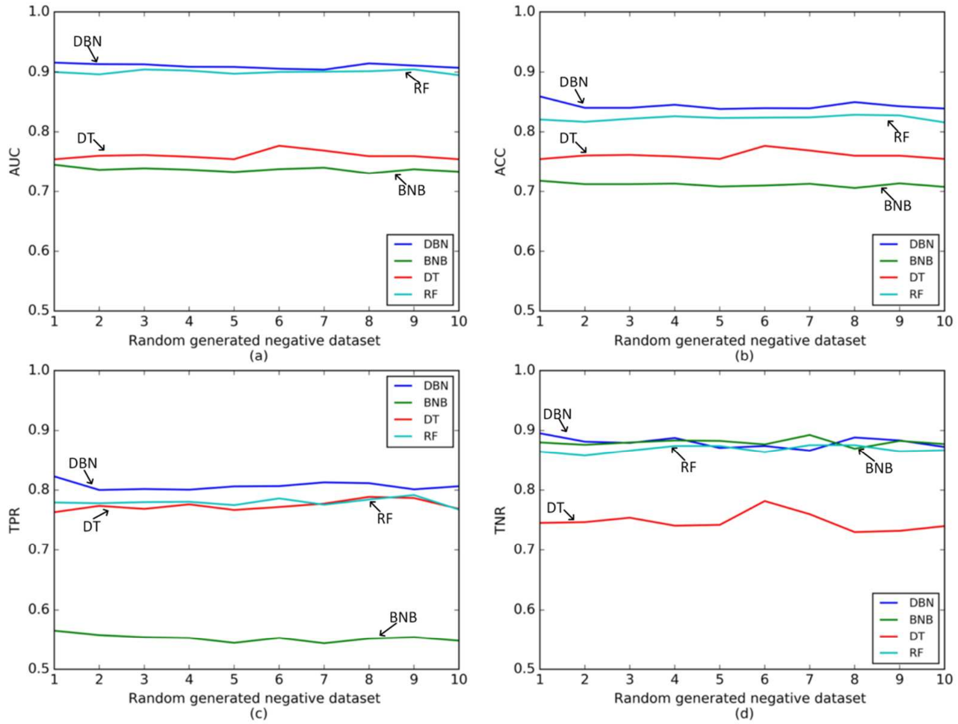


Fig. 4. TPR, TNR, ACC and AUC of 4 methods in 10 times using random data generation procedure.

Performance of EDTPs and applicability

The prediction probability distribution of BNB, RF and DBN on EDTPs dataset is shown in **Fig. 5**. The larger the prediction probability, the more credible the drug target pair has a positive interaction. Because DT is a single tree model, it could not calculate the prediction probability and its result is not shown in **Fig. 5**. As is shown in **Fig. 5**, the prediction probability of BNB and DBN tends to close to 0 or 1, whereas RF tends to close to 0.3. The prediction results of BNB, RF and DBN are list in table 2. If we set the prediction probability of an experimental drug-target pair larger than 0.5 as positive DTI, DBN and RF have similar recall (21.8% and 21.4% respectively) in EDTPs. RF has a higher recall than DBN in EDTPs1 (Table 2). However, from **Fig. 5**, we can see that most positive DTIs predicted by RF are not very certain because their prediction probabilities are close to 0.5. If we set the prediction probability of an experimental drug-target pair larger than 0.9 as positive DTI. DBN obtain the best recall (much higher than RF and BNB) in all EDTPs1, EDTPs2 and EDTPs dataset. Besides, the drugs and targets of EDTPs2 are new to that in the training dataset, this indicates that DeepDTIs can be used to predict whether a new drug targets to some existed targets or whether a new target is interact with some existed drugs.

Table 2. the prediction results of BNB, RF and DBN in EDTPs1, EDTPs2 and EDTPs dataset

Methods	dataset	NOS[a]	Recall[b] (p>0.5)[c]	Recall (p>0.9)[d]
BNB	EDTPs1	2444	202 (8.3%)	184 (7.5%)
	EDTPs2	4908	214 (4.4%)	236 (4.8%)
	EDTPs	7352	438 (5.9%)	398 (5.4%)
RF	EDTPs1	2444	956 (39.1%)	80 (3.3%)
	EDTPs2	4908	622 (12.7%)	0 (0%)
	EDTPs	7352	1578 (21.4%)	80 (1.1%)
DBN	EDTPs1	2444	618 (25.3%)	419 (17.2%)
	EDTPs2	4908	988 (20.1%)	582 (11.9%)
	EDTPs	7352	1606 (21.8%)	1001 (13.6%)

[a]Number of sample. [b]The Recall is defined as the predicted positive pairs / NOS. [c]Drug-target pair is considered as DTIs if the prediction probability is larger 0.5. [d]Drug-target pair is considered as DTIs if the prediction probability is larger 0.9

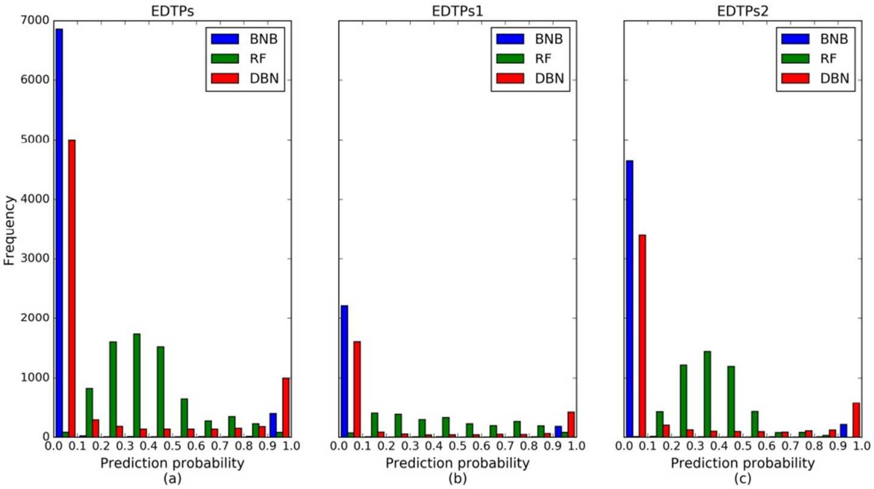


Fig. 5. The prediction probability distribution of BNB, RF and DBN in EDTPs, EDTPs1 and EDTPs2 dataset.

New predicted interactions

After confirming the reliability of our built model, we predicted all the remaining two million unknown pairs in DTS, and ranked them by their probability. All the predicted DTIs that the probability greater than 0.9 are available at Supplementary Table S2. The distribution of prediction probability is shown in **Fig. 6**. As is shown in **Fig. 6**, most pairs were predicted as no interaction. 88.15% pairs have prediction probabilities less than 0.5. If the threshold is set as 0.9, only 5.53% pairs were predicted to be positive DTIs. This conforms to the fact that the number of no interaction pairs is far more than the number of interaction pairs Table 3 shows the list of the top 10 probability predicted DTIs by DBN. Among the top 10 predicted DTIs, 4 of them were found in STITCH or literature. One drug, Drospirenone (DB01395), was found to exhibit a low relative binding affinity to the glucocorticoid receptor (P04150) in literature [35]. In the remaining predicted 5 DTIs, though we did not find any experimental evidence from databases and literatures, they still have potentiality to be true positive DTIs. For examples, Ziprasidone (DB00246) was found to exhibit 5-hydroxytryptamine receptor 1A, 2A, 1B, 2C and 1D which have high sequence homology with 5-hydroxytryptamine receptor 1C. These results on new predictions indicated that the DBN model is practically useful in prediction of novel DTIs and have potential applications in drug repositioning.

Table 3. Top 10 probability scoring DTIs predicted by our model.

Rank	Pair	Description	Evidence
1	DB00246, P08909	Ziprasidone,5-hydroxytryptamine receptor 1C (2C) (HTR2C)	STITCH
2	DB09016, P21728	Butriptyline, D(1A) dopamine receptor (DRD1)	-
3	DB00894, P04150	Testolactone, Glucocorticoid receptor (NR3C1)	-
4	DB01395, P04150	Drospirenone, Glucocorticoid receptor (NR3C1)	No interaction

5	DB00404, P28476	Alprazolam, Gamma-aminobutyric acid receptor subunit rho-2 (GABRR2)	STITCH
6	DB00363, P08909	Clozapine, 5-hydroxytryptamine receptor 1C (2C) (HTR2C)	STITCH
7	DB00246, P41595	Ziprasidone, 5-hydroxytryptamine receptor 2B (HTR2B)	-
8	DB00831, P21728	Trifluoperazine, D(1A) dopamine receptor (DRD1)	-
9	DB00990, P04150	Exemestane, Glucocorticoid receptor (NR3C1)	-
10	DB04839, P04150	Cyproterone acetate, Glucocorticoid receptor (NR3C1)	

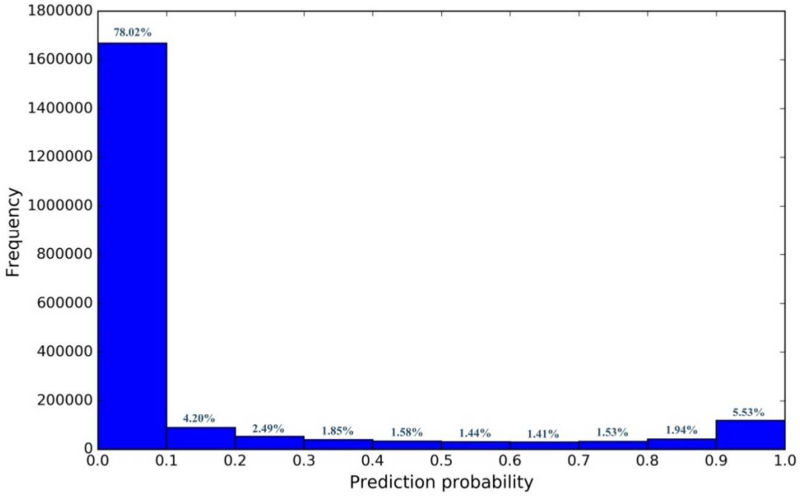


Fig. 6. Prediction probability distribution of DTS by DBN

The influence of unsupervised pre-training

The success of a machine learning algorithm generally depends on the descriptors used to represent and describe data. This is because different descriptors can entangle and hide more or less the different explanatory factors of variation behind data [36]. This encourages experts to design more powerful descriptors to help algorithms perform better on a certain task. The importance of descriptor certainly highlights the weakness of traditional learning algorithms: they are unable to extract and organize the discriminative information from the data [36]. Experts design more powerful descriptors is a way to complement the algorithm weakness by taking advantage of human ingenuity and prior knowledge. Moreover, unsupervised learning along with supervised learning is particularly beneficial to DTIs prediction. Among the DTS, only small part of DTPs are known DTIs (less than 0.3%) and the others are unknown, and the number of no interaction pairs is far more than the number of interaction pairs. Thus, it is hard to use only 0.3% samples to represent the whole sample space and the applicability of the model may bias. In DeepDTIs, DBN applied all the samples in the data space to learning the distribution of the data space. DBN can capture the posterior distribution of the explanatory factors for the observed input. With the goal of yielding more abstract and useful representations, DBN's hidden layers are formed by the composition of multiple non-linear transformations of the data. **Fig. 7** is the plot of accuracy of logistic regression (LR) model in 5 layers in our DBN model. We utilized the transformed data in each layer as input data and applied LR to train a classification model. The test set is then used to evaluate the performance of LR models. As is shown in **Fig. 7**, except hidden layer 1, the accuracy of hidden layer 2, 3 and 4 are much better than raw input data. With the increasing of hidden layer depth, the accuracy of LR is increasing. This indicates that the pre-training procedure in DBN has a strong ability to abstract information from raw input dataset.

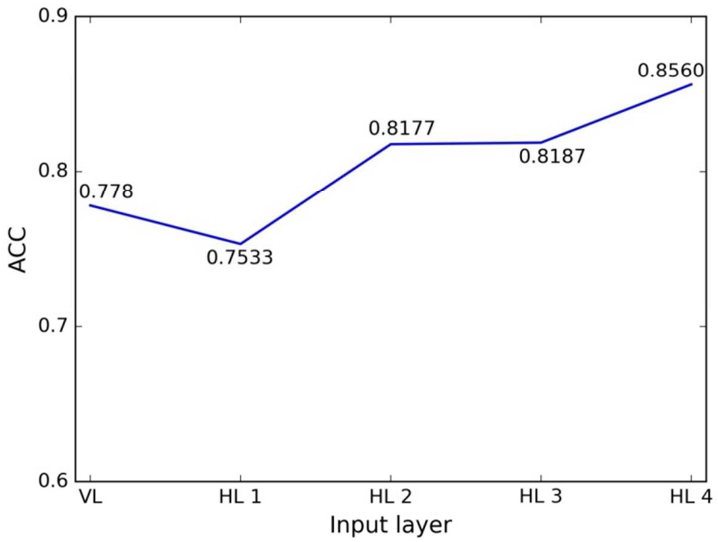


Fig. 7. The plot of accuracy of logistic regression (LR) model in 5 layers. VL represents visible layer (raw input dataset). HL represents hidden layer. The transformed dataset in each layer was used to train a logistic regression model and the accuracy of test dataset is used to evaluate the performance of each model.

CONCLUSION

Drug-target interactions (DTIs) are important to current drug discovery processes. Identifying interaction of known drug with target helps to infer drug indications, adverse drug reactions, drug-drug interactions and drug mode of actions. In this study, we proposed DeepDTIs – a deep learning approach to predict DTIs. Our approach uses a DBN model to effectively abstract raw input vectors and accurately predict DTIs. Results on one test dataset and one external EDTPs dataset showed that our algorithm can achieve relatively high prediction performance. Further analysis of novel predicted DTIs indicated that our approach can infer a list of novel DTIs, which is practically useful for drug repositioning.

Supplementary Information

Supplementary Table S1: Grid-search result of parameters in DeepDTIs.

Supplementary Table S2: The predicted DTIs with probability greater than 0.9.

Corresponding Author

*E-mail: hongmeilu@csu.edu.cn. Tel: 0731-8830830

ACKNOWLEDGEMENTS AND FUNDING

The authors thank Mr. Hualiang Zeng for helping revise the paper. The authors gratefully thank the National Natural Science Foundation of China for support of the projects (Grant No. 81402853, 21175157, 21375151 and 21305163) and also supported by the Fundamental Research Funds for the Central University of Central

South University (Grants No. 2015zzts163). The studies meet with the approval of the university's review board.

REFERENCES

1. Nascimento, A. C.; Prudêncio, R. B.; Costa, I. G., A multiple kernel learning algorithm for drug-target interaction prediction. *BMC Bioinformatics* **2016,** 17, (1), 1.

2. Chen, X.; Yan, C. C.; Zhang, X.; Zhang, X.; Dai, F.; Yin, J.; Zhang, Y., Drug–target interaction prediction: databases, web servers and computational models. *Briefings in bioinformatics* **2015,** 17, (4), 696-712.

3. Novac, N., Challenges and opportunities of drug repositioning. *Trends in pharmacological sciences* **2013,** 34, (5), 267-272.

4. Yamanishi, Y.; Kotera, M.; Kanehisa, M.; Goto, S., Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics* **2010,** 26, (12), i246-i254.

5. Zanni, R.; Galvez-Llompart, M.; Galvez, J.; Garcia-Domenech, R., QSAR multi-target in drug discovery: a review. *Current computer-aided drug design* **2013,** 10, (2), 129-136.

6. González-Díaz, H.; Prado-Prado, F.; García-Mera, X.; Alonso, N.; Abeijón, P.; Caamano, O.; Yáñez, M.; Munteanu, C. R.; Pazos, A.; Dea-Ayuela, M. A., MIND-BEST: Web Server for Drugs and Target Discovery; Design, Synthesis, and Assay of MAO-B Inhibitors and Theoretical− Experimental Study of G3PDH Protein from Trichomonas gallinae. *Journal of proteome research* **2011,** 10, (4), 1698-1718.

7. Vina, D.; Uriarte, E.; Orallo, F.; González-Díaz, H., Alignment-Free Prediction of a Drug− Target Complex Network Based on Parameters of Drug Connectivity and Protein Sequence of Receptors. *Molecular pharmaceutics* **2009,** 6, (3), 825-835.

8. Durán, F. J. R.; Alonso, N.; Caamaño, O.; García-Mera, X.; Yañez, M.; Prado-Prado, F. J.; González-Díaz, H., Prediction of multi-target networks of neuroprotective compounds with entropy indices and synthesis, assay, and theoretical study of new asymmetric 1, 2-rasagiline carbamates. *International journal of molecular sciences* **2014,** 15, (9), 17035-17064.

9. Chen, Y.; Zhi, D., Ligand–protein inverse docking and its potential use in the computer search of protein targets of a small molecule. *Proteins: Structure, Function, and Bioinformatics* **2001,** 43, (2), 217-226.

10. Kitchen, D. B.; Decornez, H.; Furr, J. R.; Bajorath, J., Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery* **2004,** 3, (11), 935-949.

11. Periole, X.; Knepp, A. M.; Sakmar, T. P.; Marrink, S. J.; Huber, T., Structural determinants of the supramolecular organization of G protein-coupled receptors in bilayers. *Journal of the American Chemical Society* **2012,** 134, (26), 10959-10965.

12. Cheng, F.; Liu, C.; Jiang, J.; Lu, W.; Li, W.; Liu, G.; Zhou, W.; Huang, J.; Tang, Y., Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol* **2012,** 8, (5), e1002503.

13. Chen, X.; Liu, M.-X.; Yan, G.-Y., Drug–target interaction prediction by random

walk on the heterogeneous network. *Molecular BioSystems* **2012,** 8, (7), 1970-1978.

14. Campillos, M.; Kuhn, M.; Gavin, A.-C.; Jensen, L. J.; Bork, P., Drug target identification using side-effect similarity. *Science* **2008,** 321, (5886), 263-266.

15. Yang, K.; Bai, H.; Ouyang, Q.; Lai, L.; Tang, C., Finding multiple target optimal intervention in disease-related molecular network. *Molecular Systems Biology* **2008,** 4, (1), 228.

16. Cao, D. S.; Zhang, L. X.; Tan, G. S.; Xiang, Z.; Zeng, W. B.; Xu, Q. S.; Chen, A. F., Computational Prediction of Drug□ Target Interactions Using Chemical, Biological, and Network Features. *Molecular Informatics* **2014,** 33, (10), 669-681.

17. Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G., Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *Journal of Chemical Information and Computer Sciences* **2003,** 43, (6), 1882-1889.

18. Romero-Durán, F. J.; Alonso, N.; Yañez, M.; Caamaño, O.; García-Mera, X.; González-Díaz, H., Brain-inspired cheminformatics of drug-target brain interactome, synthesis, and assay of TVP1022 derivatives. *Neuropharmacology* **2016,** 103, 270-278.

19. Yann, L. C.; Yoshua, B.; Geoffrey, H., Deep learning. *Nature* **2015,** 521, (7553), 436-44.

20. Alipanahi, B.; Delong, A.; Weirauch, M. T.; Frey, B. J., Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology* **2015,** 33, (8), 831–838.

21. Jo, T.; Hou, J.; Eickholt, J.; Cheng, J., Improving protein fold recognition by deep learning networks. *Scientific reports* **2015,** 5, (17573), 1-11.

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

23. Sawada, R.; Kotera, M.; Yamanishi, Y., Benchmarking a Wide Range of Chemical Descriptors for Drug-Target Interaction Prediction Using a Chemogenomic Approach. *Molecular Informatics* **2014,** 33, (11-12), 719-731.

24. Cao, D.-S.; Liu, S.; Xu, Q.-S.; Lu, H.-M.; Huang, J.-H.; Hu, Q.-N.; Liang, Y.-Z., Large-scale prediction of drug–target interactions using protein sequences and drug topological structures. *Analytica chimica acta* **2012,** 752, 1-10.

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

26. Quinlan, J. R., Induction of decision trees. *Machine learning* **1986,** 1, (1), 81-106.

27. Wishart, D. S.; Knox, C.; Guo, A. C.; Cheng, D.; Shrivastava, S.; Tzur, D.; Gautam, B.; Hassanali, M., DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research* **2008,** 36, (suppl 1), D901-D906.

28. Dong, J.; Cao, D.-S.; Miao, H.-Y.; Liu, S.; Deng, B.-C.; Yun, Y.-H.; Wang, N.-N.; Lu, A.-P.; Zeng, W.-B.; Chen, A. F., ChemDes: an integrated web-based platform for molecular descriptor and fingerprint computation. *Journal of Cheminformatics* **2015,** 7, (1), 1-10.

29. Rogers, D.; Hahn, M., Extended-connectivity fingerprints. *Journal of chemical information and modeling* **2010,** 50, (5), 742-754.

30. Cao, D.-S.; Xu, Q.-S.; Liang, Y.-Z., propy: a tool to generate various modes of Chou's PseAAC. *Bioinformatics* **2013,** 29, (7), 960-962.

31. Hinton, G. E., A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade*, Springer: 2012; pp 599-619.

32. Hinton, G. E.; Salakhutdinov, R. R., Reducing the dimensionality of data with neural networks. *Science* **2006,** 313, (5786), 504-507.

33. Hinton, G. E.; Osindero, S.; Teh, Y.-W., A fast learning algorithm for deep belief nets. *Neural computation* **2006,** 18, (7), 1527-1554.

34. Bastien, F.; Lamblin, P.; Pascanu, R.; Bergstra, J.; Goodfellow, I.; Bergeron, A.; Bouchard, N.; Warde-Farley, D.; Bengio, Y., Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590* **2012**.

35. Krattenmacher, R., Drospirenone: pharmacology and pharmacokinetics of a unique progestogen. *Contraception* **2000,** 62, (1), 29-38.

36. Bengio, Y.; Courville, A.; Vincent, P., Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **2013,** 35, (8), 1798-1828.

Figure captions:

Fig. 1. The flowchart of DeepDTIs.

Fig. 2. DBN architecture. DBN is consisted by stacking many RBMs

Fig. 3. the plot of parameter values versus the model performances.

Fig. 4. TPR, TNR, ACC and AUC of 4 methods in 10 times using random data generation procedure.

Fig. 5. The prediction probability distribution of BNB, RF and DBN in EDTPs, EDTPs1 and EDTPs2 dataset.

Fig. 6. Prediction probability distribution of DTS by DBN

Fig. 7. The plot of accuracy of logistic regression (LR) model in 5 layers. VL represents visible layer (raw input dataset). HL represents hidden layer. The transformed dataset in each layer was used to train a logistic regression model and the accuracy of test dataset is used to evaluate the performance of each model.

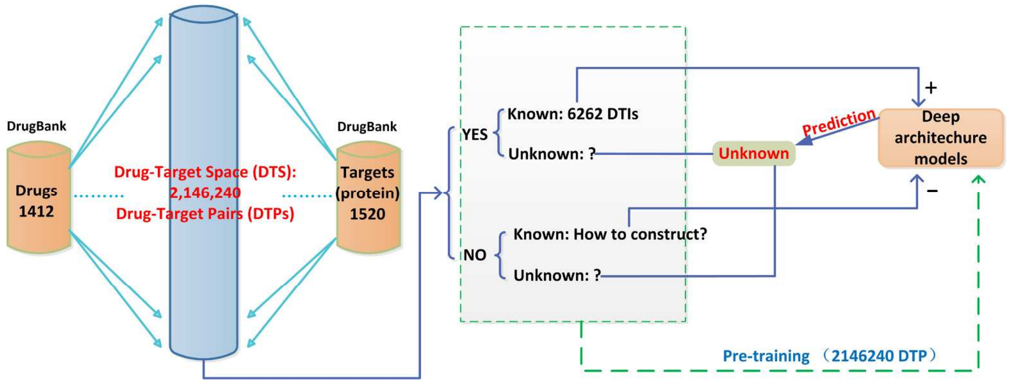


Fig.1

239x90mm (300 x 300 DPI)

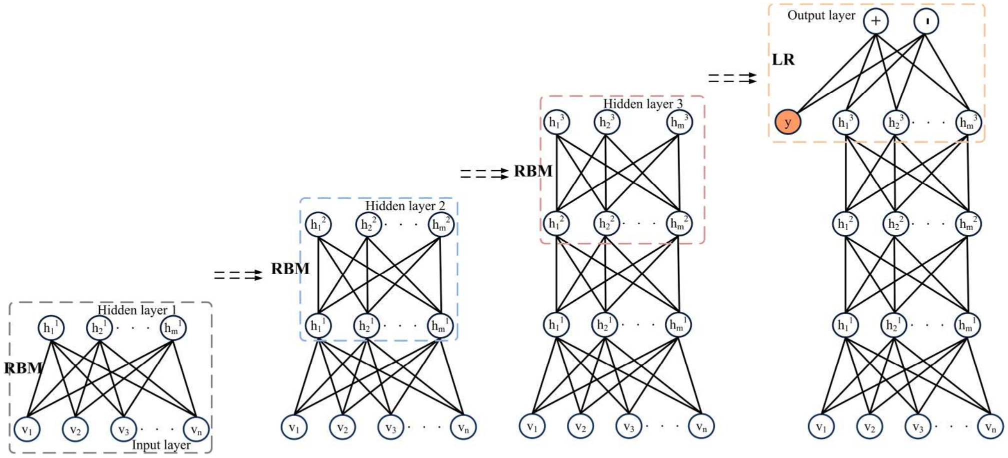

Output layer
+
-
LR
y
Hidden layer 3
RBM
Hidden layer 2
RBM
Hidden layer 1
RBM
Input layer
h_1^3 h_2^3 . . . h_m^3
h_1^2 h_2^2 . . . h_m^2
h_1^1 h_2^1 . . . h_m^1
v_1 v_2 v_3 . . . v_n

Fig.2

283x127mm (300 x 300 DPI)

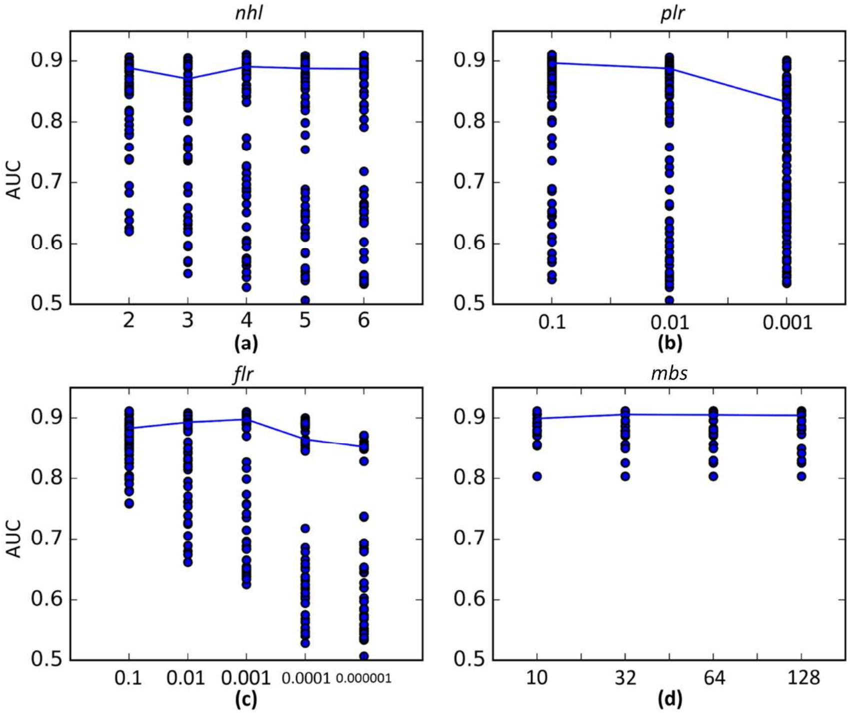
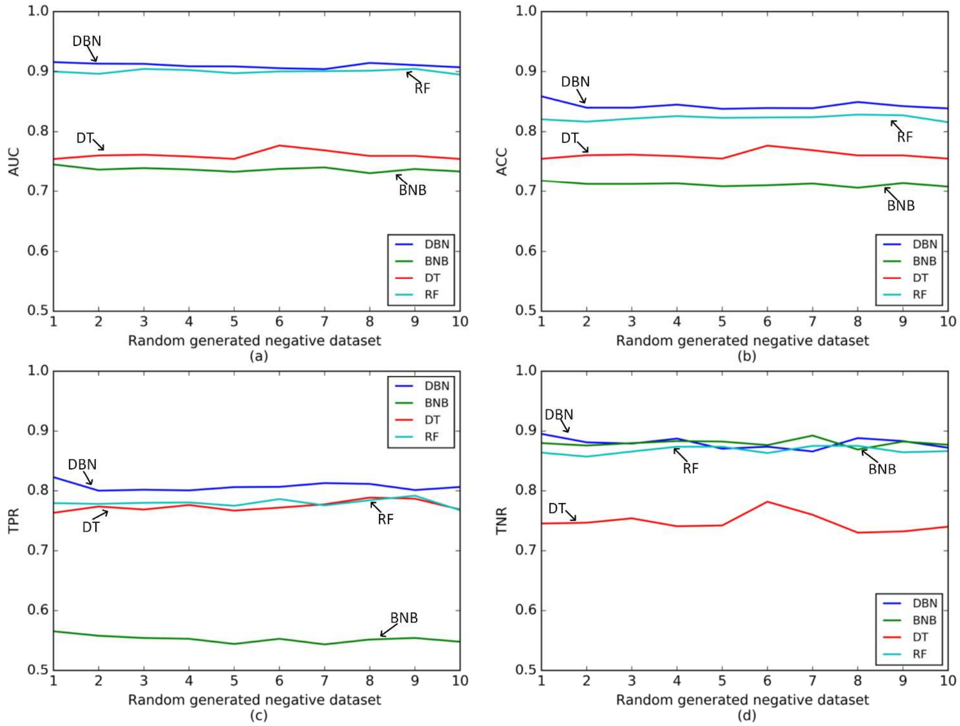


Fig. 3

289x225mm (300 x 300 DPI)



致青春·原来你还在这里

Fig. 4

257x200mm (300 x 300 DPI)

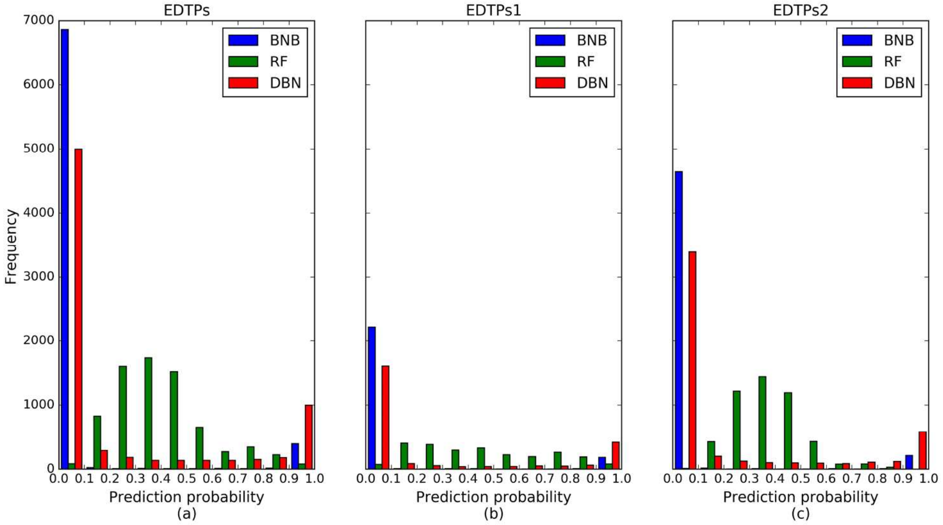


Fig.5

253x142mm (300 x 300 DPI)

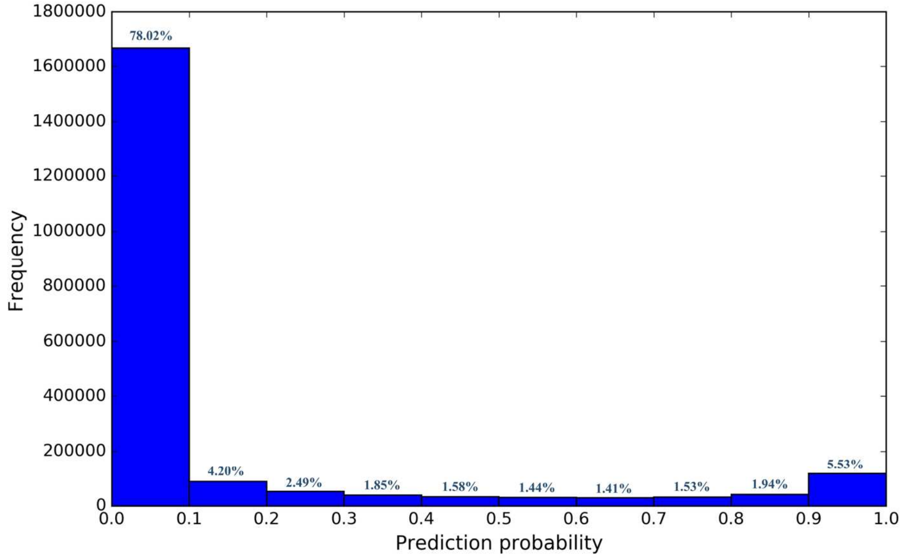


Fig.6

235x145mm (300 x 300 DPI)

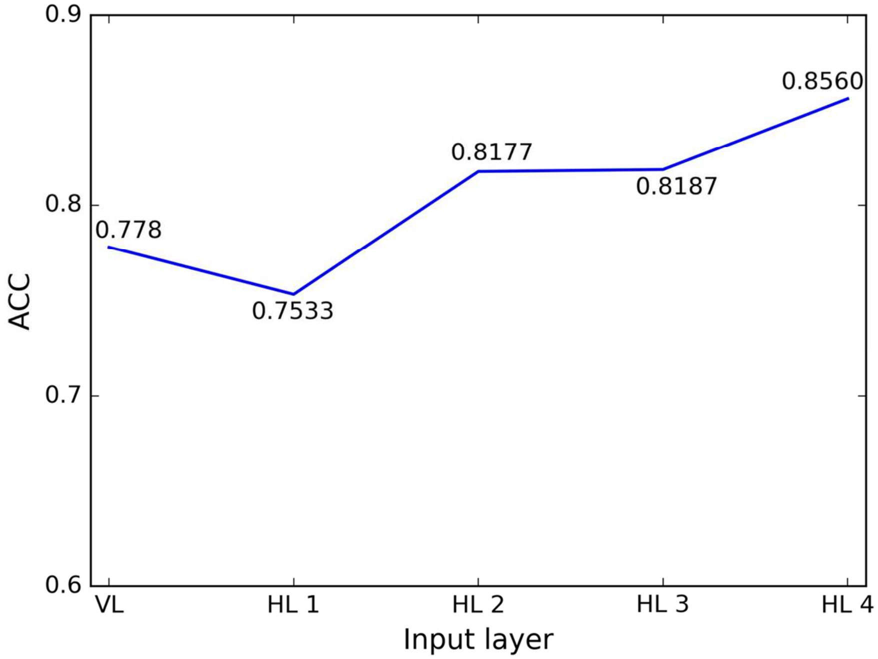
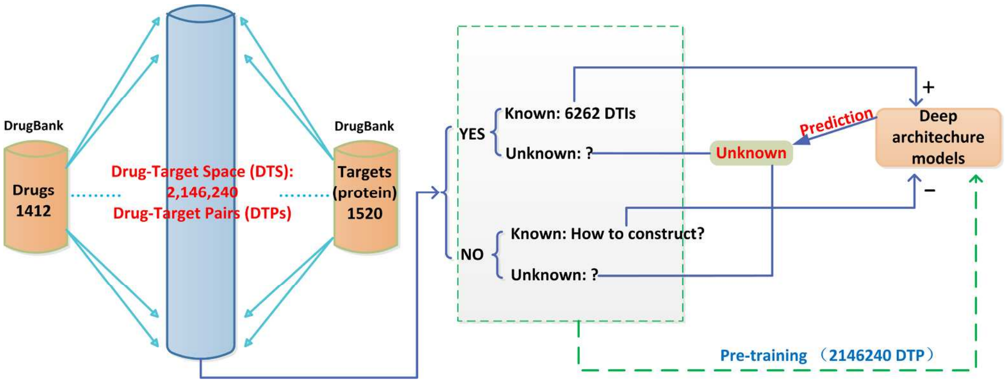


Fig.7

279x200mm (300 x 300 DPI)



For TOC only
239x90mm (300 x 300 DPI)