Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Interpretable Drug Target Prediction Using Deep Neural Representation

Kyle Yingkai Gao, Achille Fokoue, Heng Luo, Arun Iyengar, Sanjoy Dey, Ping Zhang
IBM Research AI, 1101 Kitchawan Road, Yorktown Heights, NY 10598
kyle.ygao@gmail.com, heng.luo@ibm.com, {achille, aruni, deysa, pzhang}@us.ibm.com

Abstract

The identification of drug-target interactions (DTIs) is a key task in drug discovery, where drugs are chemical compounds and targets are proteins. Traditional DTI prediction methods are either time consuming (simulation-based methods) or heavily dependent on domain expertise (similarity-based and feature-based methods). In this work, we propose an end-to-end neural network model that predicts DTIs directly from low level representations. In addition to making predictions, our model provides biological interpretation using two-way attention mechanism. Instead of using simplified settings where a dataset is evaluated as a whole, we designed an evaluation dataset from BindingDB following more realistic settings where predictions of unobserved examples (proteins and drugs) have to be made. We experimentally compared our model with matrix factorization, similarity-based methods, and a previous deep learning approach. Overall, the results show that our model outperforms other approaches without requiring domain knowledge and feature engineering. In a case study, we illustrated the ability of our approach to provide biological insights to interpret the predictions.

1 Introduction

The identification of drug-target interactions (DTI) is a key task in drug discovery, where drugs are chemical compounds and targets are proteins. The high profits on patented new drugs have motivated pharmaceutical labs to examine most potential interactions. Since experimental assays take time and are expensive, efficient computational methods for predicting and understanding potential DTIs are useful and urgently demanded.

Two major computational approaches have been investigated for DTI prediction: molecular docking and machine learning. Molecular docking using three-dimensional (3D) simulation is widely used for its reasonable accuracy and visual interpretability. However, it suffers from two significant shortcomings: first, getting 3D structures of proteins itself is a challenging task; second, large scale simulation can be time consuming.

Meanwhile, machine learning approaches are very promising as they enable large scale testing of candidates in a relatively short time. DTI prediction can be viewed as a binary classification problem, where the input is a pair of a drug candidate and a protein and the output label indicates whether there is an interaction between them. To the best of our knowledge, three main approaches have been investigated to represent DTI pairs: (1) chemical and biological descriptors such as molecule fingerprints or amino acid sequence descriptors [Faulon *et al.*, 2007]; (2) an aggregation of expert designed similarity measures to derive candidate pairs from known DTI pairs [Ding *et al.*, 2013; Yamanishi *et al.*, 2010]; and (3) representations learned from descriptors from (1), for example, using techniques such as Restricted Boltzmann machine or Autoencoder [Wen *et al.*, 2017; Chan *et al.*, 2016]. Although relatively effective, most methods are usually black-boxes and less biologically interpretable.

In this paper, we propose an interpretable end-to-end neural network model that predicts DTIs directly from low level representations. Specifically, the inputs of the model are raw amino acids sequences and chemical structures, and it produces structure-level interpretations in addition to the DTI predictions. As shown in Figure 1, we use long short term memory recurrent neural networks and graph-based convolutional neural network to project proteins and drug structures into dense vector spaces. A two-way attention mechanism (shown as α_{pi} and α_{di}) is used to calculate how the pair interacts and thus enables interpretability. Finally, the attention-based vector representations are used by a classifier, a simple sigmoid function in this paper, to make a prediction. We have also shown that our model is extensible to incorporate high-level information such as Gene Ontology annotations.

In the experiments, we pay special attention to the method of constructing testing examples. Our testing dataset is constructed in a way that simulates the practical situations, where, given a pair of drug and protein at testing time, the drug, the protein, or both of them may have not been observed at training time. Such experimental setting demands great generalization ability. Compared with previous methods, our model yields superior results, while using less feature engineering and domain expertise, specifically in the dif-

ficult cases that were not covered well by human-designed features and where neither the drug nor the protein from a testing pair is observed at training. At the end, we present a case study to demonstrate the visualizable interpretation and its usefulness during the drug discovery process.

2 Related Work

Predicting drug-target binding has been an interesting topic to the pharmaceutical industry and researchers. Molecular docking [Luo *et al.*, 2016; Trott and Olson, 2010] is widely used to predict the binding mode and score given the 3D structure inputs of a drug molecule and a target. In order to make the binding prediction, the docking program looks for the optimal binding position of the drug molecule inside the binding pocket of the target and estimates the binding score according to predefined force fields. Though molecular docking is a popular tool for high throughput screening, it takes time for large scale predictions and it is limited by the availability of the 3D structures of the protein targets.

Machine learning methods have been implemented to predict DTIs. [Ding *et al.*, 2013] reviewed similarity-based machine learning methods based on sequence, protein-protein interaction (PPI) network and Gene Ontology (GO) semantic information. Their machine learning approaches include nearest neighbor, bipartite local models, pairwise kernel method, kernelized Bayesian matrix factorization, network-based inference and so on. [Faulon *et al.*, 2007] developed support vector machine (SVM) models to predict DTIs and catalytic effect based on chemical structures and enzymatic reactions. [Yamanishi *et al.*, 2010] integrated chemical, genomic and pharmacological data together to predict drug-target binding via similarity-based methods.

With the increasing popularity of deep learning, researchers are adopting deep neural models to predict DTIs. [Wen *et al.*, 2017] developed a Deep Belief Networks (DBN) model consisting of stacked Restricted Boltzmann machines (RBM). For a DTI pair, the inputs are Extended Connectivity Fingerprints of the drug and sequence composition descriptors of the protein. First the DBN is pre-trained in an unsupervised manner using only the training feature vectors, and then it is fine tuned with both feature vectors and labels from the training dataset. Instead of using RBM and DBN, [Chan *et al.*, 2016; Wang *et al.*, 2018] used stacked Autoencoder for representation learning and SVM or rotation forest for classification.

Although aforementioned (deep) machine learning methods have proved to be able to make relatively effective predictions, the lack of interpretability limits their practicality from biological perspectives. Recently, differentiable representation learning methods that can be directly applied on low-level representations enable the potential of interpretable DTI predictions. For example, [Altae-Tran *et al.*, 2017; Duvenaud *et al.*, 2015] explored using graph convolutional network to model chemical structures, and while it is intuitive to apply recurrent neural network (RNN) on protein sequences [Pollastri *et al.*, 2002], [Schwaller *et al.*, 2017] also used RNN to model SMILES strings, which are sequential encoding of chemical structures.

To the best of our knowledge, this paper presents the first end-to-end deep machine learning work that produces interpretable DTI predictions directly from low level representations. By learning directly from molecular structures and protein sequences, our approach saves the effort of designing biochemical descriptors or similarity measures, both of which can be expensive processes of feature engineering.

3 Model

3.1 Problem Formulation

On the one hand, a protein sequence consists of a list of amino acids $p = (a_1, \ldots, a_n)$, where a_i is one of the 23 types of amino acids (20 standard, 2 additional, and 1 for unknown). Additionally, each protein has a set of gene ontology (GO) annotations [Ashburner *et al.*, 2000] $GO_p = \{g_1, \ldots, g_m\}$ that provides high level information (e.g., protein function).

On the other hand, a drug is represented by a SMILES [Weininger, 1988] sequence, which essentially encodes a chemical structure graph $d = \{V, E\}$, where V is a set of atoms and E is a set of chemical bonds that bind two atoms as undirected edges. We use RDKit[1] to transform SMILES string to chemical structure graph.

The goal of DTI prediction is to learn a model that takes a pair (p, d) as input and outputs $y \in \{0, 1\}$, where $y = 1$ indicates an interaction between p and d.

3.2 Recurrent Neural Network

In the situation where proteins are represented by amino acid sequences and drugs are represented by SMILES strings, we use recurrent neural network (RNN) to project sequential input to dense vector representations. Specifically, because, in reality, protein sequences fold in 3 dimensional space and because SMILES strings are contextual by design, both of which can be viewed as long-distance dependencies, we use long short term memory (LSTM) RNNs [Hochreiter and Schmidhuber, 1997] for their ability to memorize long term information. At each time step t, the LSTM unit takes the t-th input token embedding $x_t \in \mathbb{R}^M$, the hidden states from the previous time step $h_{(t-1)} \in \mathbb{R}^H$, and the cell states from previous step $c_{(t-1)} \in \mathbb{R}^H$. Then it produces new hidden and cell state h_t, c_t:

$$h_t, c_t = LSTM(x_t, h_{t-1}, c_{t-1})$$

Here, M and H are two hyper parameters for the dimension of the embedding space and the dimension of the hidden space respectively. We initialize $h_0 = c_0 = \mathbf{0}_H$ as a vector of zeros. Suppose the input tokens belong to a vocabulary $V = \{t_1, \cdots, t_{|V|}\}$, the input embeddings are obtained as $x_i = W_v^T \mathbf{I}_i$ where $W_v \in \mathbb{R}^{|V| \times M}$ is a learnable parameter and $\mathbf{I}_i \in \mathbb{R}^{|V| \times 1}$ is a vector whose i-th value is one and all others zero.

3.3 Convolutional Neural Network on Graph

When drugs are represented by chemical structural graphs, we use convolutional neural network (CNN) to project chemical structural graphs to dense vector representations. It is

[1] http://www.rdkit.org/

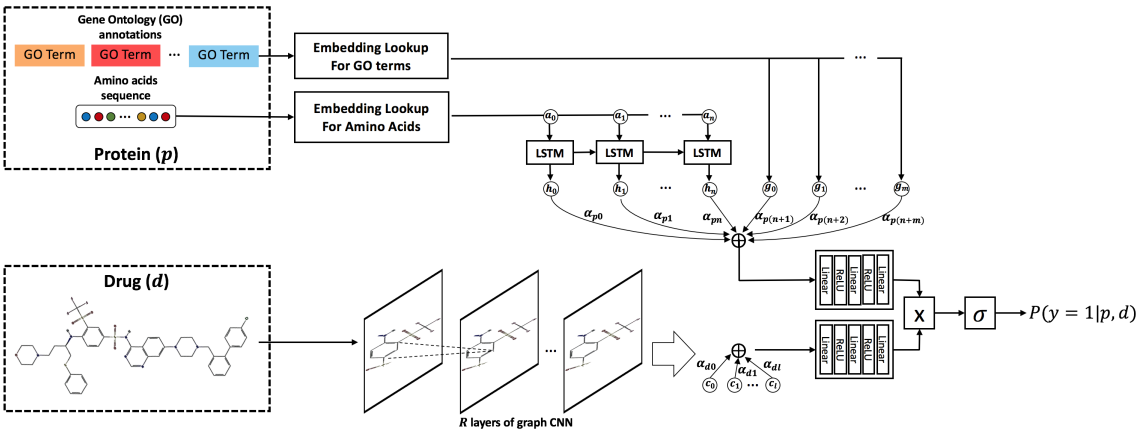

Figure 1: Overall data flow and neural network architecture.

more intuitive than using RNN to model them because it eliminates the step of linearizing the graph structures. [Duvenaud *et al.*, 2015] recently demonstrated that, as a differentiable generalization of circular fingerprint, a CNN based neural fingerprint provides better descriptive drug modeling in a data-driven manner.

Algorithm 1: Pseudocode of graph CNN.

Input: Molecule graph $G = (V, E)$, radius R, hidden weights $H_1^1 \dots H_R^5$
Output: A vector $\mathbf{r}_a$ for each atom a
Initialize: **for** *each node* $a \in V$ **do**
  $\mathbf{r}_a \leftarrow g(a)$ `// Look up initial feature vectors`
end
for $L = 1$ *to* R **do**
  for *each node* $a \in V$ **do**
    $\mathbb{N} = neighbors(a)$
    $\mathbf{v} \leftarrow \mathbf{r}_a + \sum_{u \in \mathbb{N}} \mathbf{r}_u$
    $\mathbf{r}_a \leftarrow \sigma(\mathbf{v} H_L^{|\mathbb{N}|})$
  end
end

Algorithm 1 shows the pseudo-code of our variant of the neural fingerprint algorithm that produces a dense vector representation for each atom in the molecule. We omit the steps required to calculate a single vector fingerprint, because instead we will exploit atom vectors with attention mechanism (Section 3.4) to enable interpretability on the drug side. At the initialization phase, as described in Section 4.2 of [Duvenaud *et al.*, 2015], the atom features are initialized as a 62 dimension sparse vector that indicates both chemical and topological properties of the atom. Then the algorithm iteratively applies convolutional operation on the graph R times and updates the atom vectors. The radius parameter R controls how many hops information can be propagated to. In this paper, $R = 3$.

While CNNs are usually applied on tensors, e.g. images, this algorithm is convolutional in the sense that it applies filters to each atom and its neighborhood to capture local signals, and then the aggregated local signals are pooled to get the final vector representation. Different from images where each pixel always has 8 neighbor pixels, an atom can have from one to five neighbor atoms. So instead of using one convolutional filter, the algorithm uses 5 types of linear filters $H^1 \dots H^5$ for atoms with corresponding number of neighbors.

3.4 Attentive Pooling Network

Neural networks with attention mechanism have been effectively applied to vision tasks (e.g., image captioning) and natural language processing tasks (e.g., machine translation), where the output components selectively focus on subsets of the input based on attention weights. Extending the one-way attention for pairwise inference, attentive pooling network [dos Santos *et al.*, 2016] provides a two-way attention mechanism that enables the input pairs to be aware of each other.

Suppose $P \in \mathbb{R}^{H_p \times L_p}$ is the context matrix of a given protein, where H_p, L_p are the dimension of the protein hidden space and the number of inputs, it can be formed in 3 ways as proteins have two input sources: (1) columns of the matrix P are the LSTM hidden vectors with the amino acids sequence as input so that L_p equals the number of amino acids in the sequence, (2) columns of P are GO annotations embeddings so that L_p equals the number of GO terms for the protein, and (3) the concatenation of both (1) and (2). Note that since there is no order between GO terms, for situation (2) and (3), the embeddings of GO terms are fed to the attention module directly without going through RNN.

Similarly, suppose $D \in \mathbb{R}^{H_d \times L_d}$ is the context matrix of a given drug, H_d, L_d being the dimension of the drug hidden space and the number of inputs. The columns of D can be (1) the LSTM hidden vectors with SMILES string as input so that L_d equals the number of tokens in the SMILES string or (2) the atom vectors obtained from graph CNN so that L_d equals the number of atoms in the molecule.

A soft alignment matrix $A \in \mathbb{R}^{L_p \times L_d}$ is calculated as

$$A = \tanh(P^T U D)$$

where $U \in \mathbb{R}^{H_p \times H_d}$ is a trainable parameter. For an intuitive example, when proteins are represented by amino acid

sequences and drug chemical structure graphs, A presents the interaction between each amino acid and each atom.

Next, the attention weights $\alpha_p \in \mathbb{R}^{L_p}, \alpha_d \in \mathbb{R}^{L_d}$, which can be interpreted as importance scores on the input units, are calculated by applying row-wise and column-wise max-pooling operations to A.

$$[\alpha_p]_i = \max_{1 \le j \le L_d} A_{i,j} \quad \text{and} \quad [\alpha_d]_j = \max_{1 \le i \le L_p} A_{i,j} \tag{1}$$

Finally, α_p and α_d are normalized by Softmax function, and the results of which are used as weights to weighted sums of the context vectors:

$$r_p = P \cdot \text{softmax}(\alpha_p) \quad \text{and} \quad r_d = D \cdot \text{softmax}(\alpha_d) \tag{2}$$

where the softmax function is defined as

$$[\text{softmax}(v)]_i = \frac{e^{v_i}}{\sum_j e^{v_j}} \tag{3}$$

3.5 Inference With Siamese Network

A Siamese network [Bromley *et al.*, 1994] has two input multi-layer networks and one output whose value corresponds to the similarity, possibility of interaction in the case of this work, between the input pair. As shown in Figure 1, two networks with 3 linear layers and 2 rectifier layers are used. To reduce the hyper-parameter space, we require all the linear layers to have the same input and output dimension H_s except the first one, whose input dimension corresponds to previous outputs.

The attention based vector representations r_p and r_d are fed separately into the two networks. Then we take the inner product of the outputs and use a sigmoid function to predict the probability that an interaction exists between a pair of protein and drug

$$v_p = f_p(r_p) \quad \text{and} \quad v_d = f_d(r_d) \tag{4}$$

$$P(y = 1|p, d) = \sigma(p, d) = \frac{1}{1 + e^{-v_p \cdot v_d}} \tag{5}$$

where f_p, f_d are the transformations of the siamese networks for protein and drugs respectively.

In a classification scenario, a hyper-parameter threshold δ is selected as classification boundary

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|p, d) > \delta \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

3.6 Training

Given a dataset $\mathcal{D} = \{(p_i, d_i, y_i)\}, i = 1 \cdots n$, the model can be trained by maximizing the likelihood of observing the training data, which is equivalent to minimizing the cross entropy loss function

$$\arg\min_{\Theta} -\sum_{i=1}^{n} y_i \log(\sigma(p_i, d_i)) + (1 - y_i) \log(1 - \sigma(p_i, d_i)) \tag{7}$$

where Θ is the set of neural network parameters.

However, although, in this paper, we use a dataset with both positive and negative pairs as described in section 4, negative pairs ($y_i = 0$) are usually not available for similar tasks especially when a dataset is from a knowledge graph that stores only existing triples. Therefore, we employ a pairwise ranking loss [Bordes *et al.*, 2011] that, for each given protein p, maximizes the margin between interacting drugs and non-interacting drugs, i.e. ranking positive drugs higher than negative drugs as much as possible.

$$\arg\min_{\Theta} \sum_{p} \sum_{d \in N^+(p)} \sum_{d' \in N^-(p)} \max(0, \gamma + \sigma(p, d') - \sigma(p, d)) \tag{8}$$

where $\gamma > 0$ is a hyper-parameter that specifies the width of the margin, and $N^+(p)$ and $N^-(p)$ give the set of drugs that interact with p and those do not interact with p respectively. In this setting, negative examples can be generated by sampling pseudo-negative drugs with heuristic criteria if a dataset does not have any.

4 Experiments

4.1 Dataset

BindingDB [Gilson *et al.*, 2016] is a public, web-accessible database of measured binding affinities, focusing chiefly on the interactions of small molecules (drugs/drug candidates) and proteins (targets/target candidates). We took a snapshot of BindingDB that contains 1.3 million data records, each of which contains information such as the identifiers of involved entities, the observed experiment results, etc. By the following criteria we construct a binary classification dataset[2] with 39,747 positive examples and 31,218 negative examples.

1. Record has chemical identifier (PubChem CID), and the small molecule has chemical structure represented by SMILES[3].
2. Record has protein identifier (Uniprot ID), and the protein has both sequence representation and Gene Ontology annotations [Ashburner *et al.*, 2000].
3. Record has IC50 value, a primary measure of binding effectiveness.
4. The chemical molecule weight is less than 1,000Da, due to our focus on small molecule drugs.
5. By following the activity threshold discussion in [Wang *et al.*, 2016], record is positive if its IC50 is less than 100nm, negative if IC50 greater than 10,000nm.

Suppose $\mathbb{P}_{\text{train}}$ and $\mathbb{D}_{\text{train}}$ are the sets of proteins and drugs that are observed in the training dataset. [Pahikkala *et al.*, 2014] suggested that there are 4 experimental settings under which the model can be learned and applied to predict the label between a drug (candidate) and a protein target d and p:

1. Both p and d are observed in the training dataset: $p \in \mathbb{P}_{\text{train}}$ and $d \in \mathbb{D}_{\text{train}}$;
2. The protein p is observed in the training dataset but the drug d is not: $p \in \mathbb{P}_{\text{train}}$ and $d \notin \mathbb{D}_{\text{train}}$;
3. The drug d is observed in the training dataset but the gene p is not: $p \notin \mathbb{P}_{\text{train}}$ and $d \in \mathbb{D}_{\text{train}}$;

[2] https://github.com/IBM/InterpretableDTIP

[3] Simplified molecular-input line-entry system [Weininger, 1988]

Dataset	Protein	Drug	Positive	Negative
Train	758	43,160	28,240	21,915
Dev	472	5,077	2,831	2,776
Test	466	5,016	2,706	2,802

Table 1: The number of distinct proteins, drugs, known positive pairs, and known negative pairs of the training, development, and testing datasets.

4. Neither p nor d is observed in the training dataset: $p \notin \mathbb{P}_{\text{train}}$ and $d \notin \mathbb{D}_{\text{train}}$.

We split proteins and drugs into those that should be observed in training and those that should not with four experimental settings; we then allocate DTI pairs into training, development, and testing datasets. Statistics of the datasets are shown in Table 1.

4.2 Training Details

During training, the parameters are initialized randomly from an uniform distribution $\Theta \sim (-0.08, 0.08)$. In each step, with batch size equals 32, a batch of proteins or drugs is randomly selected from the training data. Then we retrieve positive and negative interactions that involve them as input to the model. Dropout is applied on the output of embedding lookups and between Siamese network layers. According to Equation 5 and Equation 8, we use Adam gradient descent optimization with initial learning rate equals to 0.001 to train the parameters. We train the model for 30 epochs, where each epoch consists of 100 steps. The model is evaluated using the validation dataset after each epoch, and the one with the best ROC score is reported.

The validation set is also used to select model configurations, hyperparameters, and the classification boundary δ. As mentioned in Section 3.4, there are 3 ways to represent proteins and 2 ways to represent drugs. We found that the best model uses both amino acid sequences and GO annotations for proteins, and it uses atom vectors from graph CNN for drugs. We use gradient-boosted-tree search from scikit-optimize[4] to find effective hyperparameters. All the space dimensions are selected from powers of two that are between 8 and 64, embedding dropout probability from 0 to 0.2, Siamese dropout probability from 0 to 0.5, and training margin γ from 0.0001 to 0.5. All ranges are inclusive. The values of hyperparameters of the best model are shown in Table 2, and the best classification boundary is $\delta = 0.4995$.

[4]https://scikit-optimize.github.io/

Protein	Sequence	Embedding Size	16
		Hidden Dimension	16
		Embedding Dropout	0.1
	GO	Embedding Size	16
		Embedding Dropout	0.1
Drug	Graph CNN	Hidden Dimension	64
Siamese		Hidden Size	32
		Dropout	0.1
γ		0.0005	

Table 2: Hyper parameters for the best model.

4.3 Baselines

We compare our system with 3 baselines: matrix factorization, a similarity-based machine learning method, and a previous deep learning based method.

Matrix Factorization Instead of viewing DTI prediction as a classification task, it can also be treated as a collaborative filtering task. Given the IC50 values of the known pairs in our dataset, the task is to predict the IC50 values of other pairs. From this perspective, matrix factorization has been widely used and proved effective [Koren *et al.*, 2009]. We use the implementation from LIBMF[5] as the first baseline.

Note that although this baseline does not use content-based information from amino acids sequences and molecules, it uses extra information from the IC50 values, including those of neither positive nor negative examples, from BindingDB. The raw IC50 values are truncated at an upper threshold 100,000 and fed into LIBMF in their logarithmic values.

Similarity Based Method We also compare our approach against a similarity based prediction approach. In particular, we use as our second baseline a state-of-the-art similarity based prediction system adapted to DTI predictions: **Tiresias** [Fokoue *et al.*, 2016]. Tiresias system takes as input a pair of drug and protein, builds a feature vector using the statistics of a large number of similarity measures against known DTI pairs, and outputs the prediction using a logistic regression classification model.

Deep Learning Based Method At last, we compare our approach with [Wen *et al.*, 2017] as mentioned in Section 2. Its hyperparameters, including the number of units of RBM layers, the number of pre-train and fine tuning epochs, and the learning rates are obtained by the same gradient-boosted-tree search as for our model. We refer this method as **DBN** in the rest of this paper.

5 Results

5.1 Effectiveness

Before diving into the results, recall that different amounts and levels of information are exposed to the systems. Since matrix factorization is not content-based, it does not learn from any low-level representation and cannot handle unobserved instance, and so it is only evaluated in the first setting. However, it uses extra information from the raw IC50 values and uses more training examples whose IC50 values (larger than 100nm and smaller than 10,000nm) fall outside of our classification range. Tiresias, as the state-of-the-art similarity-based system, uses a set of expert designed similarity measures based on protein and drug properties. For deep learning approaches, DBN uses middle-level features from expert designed molecular fingerprints and protein descriptors, whereas our best model (E2E) learns from molecular chemical structures, amino acids, and GO annotations in an

[5]https://www.csie.ntu.edu.tw/ cjlin/libmf/

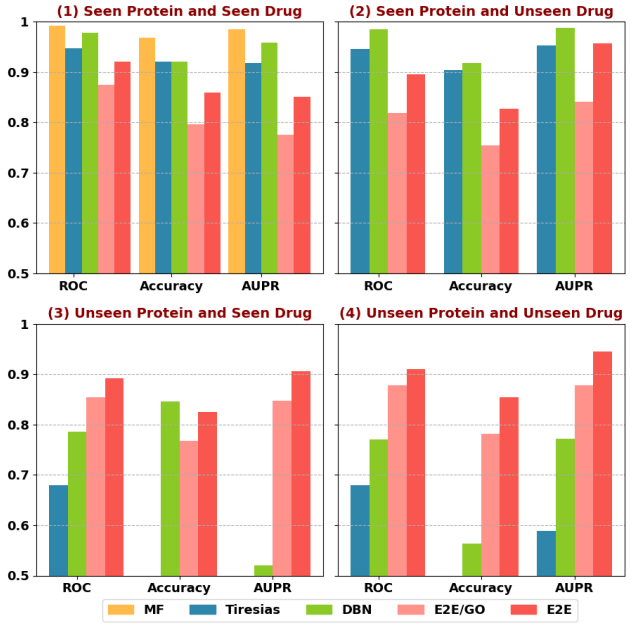

Figure 2: Performance comparison of our model (E2E), our model without using GO annotation (E2E/GO), matrix factorization (MF), similarity-based approach (Tiresias), and deep learning based approach (DBN). For each system three metrics are reported: area under receiver operating characteristic (ROC) curve, accuracy, and area under precision recall curve (AUPR). The four charts correspond to the experimental settings as mentioned in Section 4.1. Note that the accuracy scores of Tiresias do not show in the plot (3) and (4) because they are lower than the lower bound of the y-axis.

end-to-end manner. For a fair comparison with DBN, we also include a version of our model without using GO annotations (E2E/GO).

Figure 2 presents the performance comparison of our model and the baselines. As mentioned in Section 4.1, the systems are evaluated in four different testing datasets, to which the systems have different visibility of entities in the training data. In terms of overall performance, our best system (E2E) is the only one that consistently performs well across all datasets and all metrics (all greater than 0.8), whereas the performance of the baselines drops dramatically when the proteins are not observed in the training data. If we aggregated the 4 cases, our method outperforms other baselines in all metrics, e.g. the average AUPR of E2E is 0.91 while that of DBN is 0.81.

The baselines are more effective than our model when the tested proteins are observed. However, when the tested proteins are not observed, Tiresias fails as its similarity measures for proteins are not effective, DBN loses to its counterpart E2E/GO in most cases except for its accuracy in the third setting, and, finally, our best model E2E introduces more improvement on top of E2E/GO by exploiting its flexibility of incorporating high level information.

5.2 Case Study for Interpretability

The key advantage of our model over all baselines is its interpretability. Before conducting costly lab tests on potential

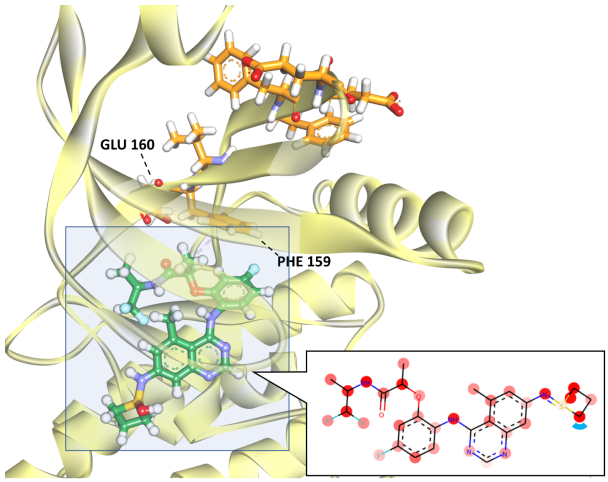

Figure 3: The interaction between SCHEMBL16362922 and the MAP kinase-interacting serine/threonine-protein kinase 2. The protein is shown in yellow and the small molecule is shown in green. The predicted top seven contributing residues by our method are highlighted in orange. The 2D view of the small molecule shows the contributing weights of each atom proportional to the saturation of red. The blue fan represents a solvent accessible surface.

interactions, it is helpful for researchers to gain insights by telling them where to look at.

To demonstrate that, we conducted a case study of a top predicted interaction between chemical (PubChem ID: 117793281) and protein (UniProt ID: Q9HBH9) by comparing our interpretation with that obtained by molecular docking, which is currently more biologically interpretable. We use molecular docking to generate their structural interaction pattern and the result was analyzed in Discovery Studio Visualizer 2017 R2 (Figure 3). We found that, according to the attention mechanism, the top seven contributing protein residues are either within or surrounding the protein binding pocket, and two of them (GLU 160 and PHE 150 as labeled in Figure 3) may closely interact with the drug. From the drug's perspective, among the most contributing atoms (marked in saturated red), three are nitrogen atoms, which have potential to formulate hydrogen bonds, and one is a carbon atom that has a relatively big solvent accessible surface, indicating potential solvent interaction can be formulated. Thus, our model gives reasonable cues on the factors for the binding, which may have broad pharmaceutical applications.

6 Conclusion

We have presented an interpretable end-to-end deep learning architecture to predict drug-target interactions from low level representations. Experimental evaluation shows that this approach overall outperforms all baselines, and it is the only one capable of generalizing well to new proteins (i.e., not seen in the training data), which is critical for drug discovery given that only a small fraction of proteins are known to be targets of chemical compounds. Furthermore, we have illustrated, via a case study, the ability of our approach to provide biological insights to understand the nature of predicted interactions.

References

[Altae-Tran *et al.*, 2017] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017.

[Ashburner *et al.*, 2000] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. Learning structured embeddings of knowledge bases. In *AAAI*, volume 6, page 6, 2011.

[Bromley *et al.*, 1994] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.

[Chan *et al.*, 2016] Keith CC Chan, Zhu-Hong You, et al. Large-scale prediction of drug-target interactions from deep representations. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 1236–1243. IEEE, 2016.

[Ding *et al.*, 2013] Hao Ding, Ichigaku Takigawa, Hiroshi Mamitsuka, and Shanfeng Zhu. Similarity-based machine learning methods for predicting drug–target interactions: a brief review. *Briefings in Bioinformatics*, 15(5):734–747, 2013.

[dos Santos *et al.*, 2016] Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *CoRR, abs/1602.03609*, 2016.

[Duvenaud *et al.*, 2015] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[Faulon *et al.*, 2007] Jean-Loup Faulon, Milind Misra, Shawn Martin, Ken Sale, and Rajat Sapra. Genome scale enzyme–metabolite and drug–target interaction predictions using the signature molecular descriptor. *Bioinformatics*, 24(2):225–233, 2007.

[Fokoue *et al.*, 2016] Achille Fokoue, Mohammad Sadoghi, Oktie Hassanzadeh, and Ping Zhang. Predicting drug-drug interactions through large-scale similarity-based link prediction. In *International Semantic Web Conference*, pages 774–789. Springer, 2016.

[Gilson *et al.*, 2016] Michael K Gilson, Tiqing Liu, Michael Baitaluk, George Nicola, Linda Hwang, and Jenny Chong. Bindingdb in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic acids research*, 44(D1):D1045–D1053, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[Luo *et al.*, 2016] Heng Luo, William Mattes, Donna L Mendrick, and Huixiao Hong. Molecular docking for identification of potential targets for drug repurposing. *Current topics in medicinal chemistry*, 16(30):3636–3645, 2016.

[Pahikkala *et al.*, 2014] Tapio Pahikkala, Antti Airola, Sami Pietilä, Sushil Shakyawar, Agnieszka Szwajda, Jing Tang, and Tero Aittokallio. Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, 16(2):325–337, 2014.

[Pollastri *et al.*, 2002] Gianluca Pollastri, Darisz Przybylski, Burkhard Rost, and Pierre Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, 47(2):228–235, 2002.

[Schwaller *et al.*, 2017] Philippe Schwaller, Theophile Gaudin, David Lanyi, Costas Bekas, and Teodoro Laino. "found in translation": Predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *CoRR*, abs/1711.04810, 2017.

[Trott and Olson, 2010] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[Wang *et al.*, 2016] Zhonghua Wang, Lu Liang, Zheng Yin, and Jianping Lin. Improving chemical similarity ensemble approach in target prediction. *Journal of Cheminformatics*, 8(20), 2016.

[Wang *et al.*, 2018] Lei Wang, Zhu-Hong You, Xing Chen, Shi-Xiong Xia, Feng Liu, Xin Yan, Yong Zhou, and Ke-Jian Song. A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network. *Journal of Computational Biology*, 25(3):361–373, 2018.

[Weininger, 1988] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[Wen *et al.*, 2017] Ming Wen, Zhimin Zhang, Shaoyu Niu, Haozhi Sha, Ruihan Yang, Yonghuan Yun, and Hongmei Lu. Deep-learning-based drug–target interaction prediction. *Journal of Proteome Research*, 16(4):1401–1409, 2017.

[Yamanishi *et al.*, 2010] Yoshihiro Yamanishi, Masaaki Kotera, Minoru Kanehisa, and Susumu Goto. Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics*, 26(12):i246–i254, 2010.