Bioinformatics, 36(16), 2020, 4406–4414
doi: 10.1093/bioinformatics/btaa524
Advance Access Publication Date: 19 May 2020
Original Paper

Structural bioinformatics

TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments

Lifan Chen[1,2], Xiaoqin Tan[1,2], Dingyan Wang[1,2], Feisheng Zhong[1,2], Xiaohong Liu[1,3], Tianbiao Yang[1,2], Xiaomin Luo[1], Kaixian Chen[1,3], Hualiang Jiang[1,3,*] and Mingyue Zheng [1,*]

[1]Drug Discovery and Design Center, State Key Laboratory of Drug Research, Shanghai Institute of Materia Medica, Chinese Academy of Sciences, Shanghai 201203, China, [2]University of Chinese Academy of Sciences, Beijing 100049, China and [3]Shanghai Institute for Advanced Immunochemical Studies, School of Life Science and Technology, ShanghaiTech University, Shanghai 200031, China

*To whom correspondence should be addressed.

Associate Editor: Arne Elofsson

Received on February 26, 2020; revised on April 13, 2020; editorial decision on May 12, 2020; accepted on May 14, 2020

Abstract

Motivation: Identifying compound–protein interaction (CPI) is a crucial task in drug discovery and chemogenomics studies, and proteins without three-dimensional structure account for a large part of potential biological targets, which requires developing methods using only protein sequence information to predict CPI. However, sequence-based CPI models may face some specific pitfalls, including using inappropriate datasets, hidden ligand bias and splitting datasets inappropriately, resulting in overestimation of their prediction performance.

Results: To address these issues, we here constructed new datasets specific for CPI prediction, proposed a novel transformer neural network named TransformerCPI, and introduced a more rigorous label reversal experiment to test whether a model learns true interaction features. TransformerCPI achieved much improved performance on the new experiments, and it can be deconvolved to highlight important interacting regions of protein sequences and compound atoms, which may contribute chemical biology studies with useful guidance for further ligand structural optimization.

Availability and implementation: https://github.com/lifanchen-simm/transformerCPI.

Contact: hljiang@simm.ac.cn or myzheng@simm.ac.cn

1 Introduction

Identifying compound–protein interaction (CPI) plays an import role in discovering hit compounds (Vamathevan *et al.*, 2019). Conventional methods, such as structure-based virtual screening and ligand-based virtual screening, have been studied for decades and gained great success in drug discovery. However, some cases are not suitable to apply conventional screening methods, where the protein three-dimensional (3D) structure is unknown or the amount of known ligand dataset is too small. Therefore, Bredel and Jacoby (2004) introduced a novel perspective called chemogenomics to predict CPI without protein 3D structures. A variety of machine learning based algorithms have been proposed since then, which considers compound information and protein information at the same time in a unified model (Bleakley and Yamanishi, 2009; Cheng *et al.*, 2012; Gonen, 2012; Jacob and Vert, 2008; van Laarhoven *et al.*, 2011; Wang *et al.*, 2011; Wang and Zeng, 2013; Yamanishi *et al.*, 2008).

With the rapid development of deep learning, many types of end-to-end frameworks have been utilized in CPI research. In comparison with traditional machine leaning algorithms, end-to-end learning integrates representation learning and model training in a unified architecture simultaneously and no descriptors need to be defined and calculated before modeling. Although deep neural networks have been used in several CPI models, these current methods take predefined molecular fingerprints and protein descriptors as input features, which are fixed during training process and contain less information than that of end-to-end learning (Hamanaka *et al.*,

2017; Tian *et al.*, 2016; Wan and Zeng, 2016). Regarding the CPI problem as binary classification task, compounds can be considered as 1D sequences or molecular graphs (i.e. traditionally called 2D structures), and protein sequences can be regarded as 1D sequences. DeepDTA (Ozturk *et al.*, 2018) used convolutional neural networks (CNNs) to extract low-dimensional real-valued features of compounds and proteins, and then concatenated two feature vectors and pass through fully connected layers to calculate the final output. WideDTA (Öztürk *et al.*, 2019) and Conv-DTI (Lee *et al.*, 2019) followed the similar idea, and WideDTA utilized two extra features as well, ligand max common structures and protein motifs and domains, to improve model performance. From another perspective that regards compound structure as molecular graph, CPI–GNN (Tsubaki *et al.*, 2019) and GraphDTA (Nguyen *et al.*, 2019) used graph neural networks (Scarselli *et al.*, 2009) (GNNs) and graph CNNs (Kipf and Welling, 2016) (GCNs) instead of CNNs to learn the representation of compounds. In addition, recurrent neural networks were used to extract feature vectors of compounds and proteins in DeepAffinity (Karimi *et al.*, 2019), and Gao *et al.* (2018) and Zheng *et al.* (2020) also treated compounds and proteins as sequential information.

Due to the high relevance to chemical biology and pharmaceutical chemistry, many novel models based on deep learning or machine learning have been developed showing satisfactory performance on various datasets. However, much less efforts have been devoted to evaluate their generalization ability on external tests or practical applications. Since deep learning is a data-driven technique, it is critical to understand what the model really learns and to avoid the influences of unexpected factors. Recently, Google researchers put forward three pitfalls to avoid in machine learning (Riley, 2019), consisting of splitting data inappropriately, hidden variable and mistaking the objective. Inspired by these warnings in the AI industry, we wonder whether chemogenomics-based CPI modeling is facing similar problems, and three unique problems are summarized.

1.1 Using inappropriate datasets

Data are the core foundation of deep learning models, and in a way what a model learns mainly depends on the datasets it is fed, and inappropriate datasets make the model easily deviate from the goal. In chemogenomics-based CPI modeling, the general goal of modeling is to predict interaction between different proteins and different compounds based on a form of abstract representation of protein and ligand features. Therefore, interaction information is the key ingredient that the model should learn from the datasets. Considering chemogenomics-based CPI modeling as a binary classification task, a properly designed dataset should mainly consist of such instances that a specific ligand interacts with protein A but does not interact with protein B, which forces the model to learn protein information, or preferably the interaction features, to distinguish these instances. This proper designed dataset cannot be separated by other information rather than interaction features, such as ligand patterns. Previous chemogenomics-based CPI prediction models used inappropriate datasets to build deep learning models, such as DUD-E dataset (Mysinger *et al.*, 2012) and Human dataset (Liu *et al.*, 2015; Tsubaki *et al.*, 2019), where DUD-E dataset was collected with the intention to train structure-based virtual screening. Moreover, most ligands in DUD-E, MUV, Human and BindingDB only occur in one class, and negative samples were generated by algorithms that may introduce undetectable noise (Liu *et al.*, 2015; Mysinger *et al.*, 2012). These datasets can be separated by ligand information, and cannot guarantee that models learn protein information or interaction features.

1.2 Hidden ligand bias

Deep learning system is usually referred to as black-box models, thus it is difficult to interpret what exactly the model learns and based on which the model makes a prediction. Obtaining a better performance on the validation set and the test set usually means the end of the study, and fewer efforts were devoted to further investigate if the model learns in the manner as expected. The hidden ligand bias issue has been reported in DUD-E and MUV datasets (Sieg *et al.*, 2019), raising extensive concerns in the field of drug design. Structure-based virtual screening, 3D-CNN-based models (Chen *et al.*, 2019) and other models trained on DUD-E dataset (Sieg *et al.*, 2019) have been pointed out to make predictions mainly based on ligand patterns rather than interaction features, leading to mismatch between theoretical modeling and practical application. We wondered whether chemogenomics-based CPI modeling facing similar problem, and thus revisited a previous typical model CPI–GNN trained on Human dataset as an example to study the potential effects of hidden ligand bias.

Figure 1A shows the weight distribution plot of CPI–GNN model trained on Human dataset. The weights of CNN blocks used to extract protein features are significantly concentrated in zero, which indicates that little protein information has been considered when making prediction. In contrast, the weight distribution of GNN blocks utilized to extract compound features is wide and flat. We therefore argue that ligand information plays an overwhelming role as compared to protein information. Further training with ligand-only information and its comparison with the original model are elucidated in Figure 1B, where the dataset was randomly split for 10 times, and two models were evaluated on 10 different trails. The *P*-value in a two-sample *t*-test for the difference of AUC distribution is greater than 0.05, suggesting that using ligand information alone may achieve competitive performance to the original CPI–GNN model using both ligand and protein information. Thus, CPI–GNN model mainly learns how to classify different ligands rather than different CPI pairs, which increases the risk that a ligand is

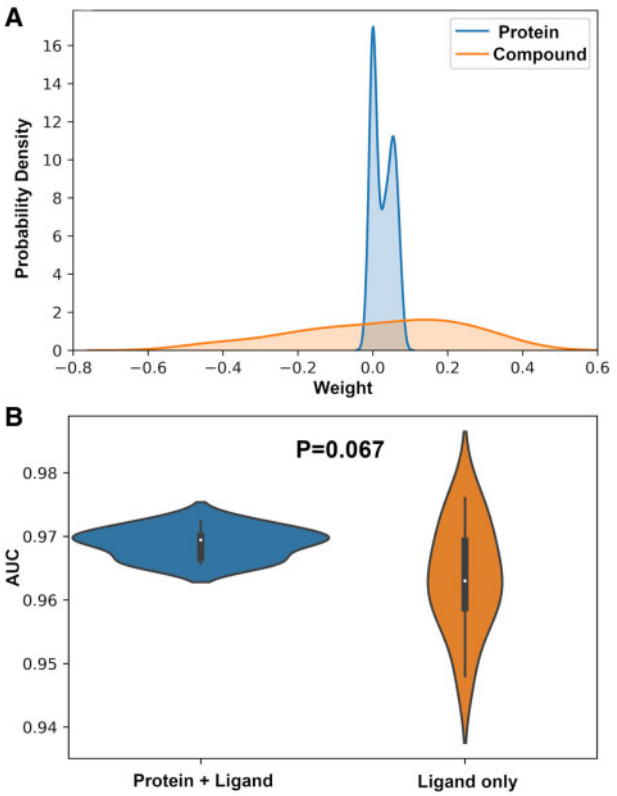

Fig. 1. Common pitfalls analysis. (**A**) Weight distribution plot of CPI–GNN model. Blue line depicts the weight distribution of CNN blocks used to extract protein features. Orange line depicts the weight distribution of GNN blocks used to extract compound features. (**B**) Violin plots of AUROC for two CPI–GNN models, one utilizing both protein and ligand information, and the other using ligand information only. The white dots are the average AUROCs. The upper and lower end points of the black segments are the first and the third quartile, respectively. The *P*-value of the *t* test is shown above the violin. Each model was evaluated on 10 different trials, and the *P*-value was 0.067

always predicted to interact or not interact with different proteins. These results highlighted the possibility that ligand patterns can mislead the model.

1.3 Splitting dataset inappropriately

The risk of hidden ligand bias is difficult to eliminate but can be reduced. Usually, machine learning researchers split data into training and test sets at random. However, using conventional classification measurements on a randomly split test set, we are not clear whether the model learns true interaction features or other unexpected hidden variables, which may produce precise models that answer the wrong questions (Riley, 2019). Thus, test sets should be designed according to the real goal of modeling and its application scenario.

To address these pitfalls, we proposed a novel transformer neural network named TransformerCPI, constructed new datasets specific for CPI modeling, and introduced a more rigorous label reversal experiments to evaluate whether a data-driven model falls into common pitfalls of AI. As a result, TransformerCPI achieved the best performance on three public datasets and two label reversal datasets. Moreover, we further studied the interpretability of TransformerCPI to uncover its underlying prediction mechanism by mapping attention weights back to protein sequences and compound molecules, and the results also confirmed that the self-attention mechanism of TransformerCPI is useful in capturing the desired interaction features. We hope that these findings may raise our attention to improve the generalization and interpretation capability of CPI modeling.

2 Materials and methods

2.1 Model architecture of TransformerCPI

The model we proposed is based on the transformer architecture (Vaswani *et al.*, 2017), which was originally devised for neural machine translation tasks. Transformer is an autoregressive encoder–decoder model using a combination of multiheaded attention layers and positional feed forward to solve sequence-to-sequence (seq2seq) tasks. Recently, transformer architecture achieves great success in language representation learning task, and many novel and powerful pre-training models have been established, such as BERT (Devlin *et al.*, 2019), GPT-2 , Transformer-XL (Dai *et al.*, 2019) and XLnet (Yang *et al.*, 2019). Transformer is also applied in chemical reaction prediction (Schwaller *et al.*, 2019), however, it is still confined in seq2seq tasks. Inspired by its great ability of capturing features between two sequences, we modified the transformer architecture to predict CPI, regarding compounds and proteins as two kinds of sequences. An overview of the proposed TransformerCPI is shown in Figure 2, where we remained the decoder of the transformer and modified its encoder and final linear layers.

To convert protein sequences into sequential representation, we first split a protein sequence into an overlapping 3-gram amino acid sequence, and then translated all words to real-valued embeddings by the pretraining approach word2vec (Mikolov *et al.*, 2013a,b). Word2vec is an unsupervised technique to learn high-quality distributed vector representations that describe sophisticated syntactic and semantic word relationships, comprising two pretraining technique called Skip-Gram and Continue Bag-of-Words (CBOW). Skip-Gram is used to predict a certain word from its context, while CBOW is used to predict context from a given word. Integrating Skip-Gram and CBOW, word2vec can finally map the words to low-dimensional real-valued vectors, where the words that have similar semantics map to the vectors that are close to each other. There have been some works to apply word2vec to represent protein sequences (Kimothi *et al.*, 2016; Kobeissy *et al.*, 2015; Mazzaferro and Carlo, 2017; Yang *et al.*, 2018), in which the amino acid sequence of constant length k (k-mers) were split as words and the whole amino acid sequence was regarded as a document. We followed these works to preprocess protein sequence, and included all human protein sequences in UniProt as corpus to pretrain the word2vec model, and set hidden dimension to 100. After training the word2vec model for 30 epochs on the large corpus we built before, protein sequences can be inferred to real-valued 100-dimensional vectors.

Sequential feature vectors of proteins were then passed to the encoder to learn more abstract representations of proteins. Of note here we replaced the original self-attention layers in the encoder with a relatively simple structure. Considering that conventional transformer architecture usually requires a large training corpus and is easy to overfit on small or modestly sized datasets (Qiu *et al.*, 2020), we used a gated convolutional network (Dauphin *et al.*, 2016) with Conv1D and gated linear unit instead because it showed better performance on our designed datasets. The input to gated convolutional network is a sequence of protein feature vectors. We compute the hidden layers $h_0, \ldots, h_L$ as Equation 1

$$h_l\ (\boldsymbol{X}) = (\boldsymbol{X} * \boldsymbol{W}_1 + \boldsymbol{s}) \otimes \sigma(\boldsymbol{X} * \boldsymbol{W}_2 + \boldsymbol{t}), \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{n \times m_1}$, is the input of layer h_l, $\boldsymbol{W}_1 \in \mathbb{R}^{k \times m_1 \times m_2}$, $\boldsymbol{s} \in \mathbb{R}^{m_2}$, $\boldsymbol{W}_2 \in \mathbb{R}^{k \times m_1 \times m_2}$, $\boldsymbol{t} \in \mathbb{R}^{m_2}$, are learned parameters, L is the number of hidden layers, n is the sequence length, m_1, m_2 are the dimension of input and hidden features, respectively, and k is the patch size, σ is the sigmoid function and $\otimes$ is the element-wise product between matrices (Dauphin *et al.*, 2016). The output of the gated convolutional network is the final representation of protein sequences, as shown in Figure 2. In our implementation, L is 3, m_1 is 64, m_2 is 128 and k is 7. The output of the encoder is protein sequence $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_b$, where b is the length of protein sequence.

Each of the atom features was initially represented as a vector of size 34 using RDKit python package, and the list of atom features is summarized in Table 1. We then used GCNs to learn the representation of each atom by integrating its neighbor atom features.

The GCN is originally devised to solve the problem of semisupervised node classification, which can be transferred to solve molecular representation problem. We here denote a graph for a compound molecule as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \in \mathbb{R}^{a \times f}$ is the set of a atoms in a molecule, each represented as a f-dimensional feature vector and $\mathcal{E}$ is the set of covalent bonds in a molecule represented as an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{a \times a}$. The propagation rule is shown in Equation 2:

$$\boldsymbol{H}^{(l+1)} = f\left(\boldsymbol{H}^{(l)}, \boldsymbol{A}\right) = \sigma\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{H}^{(l)} \boldsymbol{W}_3^{(l)}\right), \tag{2}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{H}^{(l)} \in \mathbb{R}^{a \times f}$ is the output of the ℓth layer, $\boldsymbol{W}_3^{(l)} \in \mathbb{R}^{f \times f}$ is a weight matrix for the ℓth neural network layer, $\tilde{\boldsymbol{D}} \in \mathbb{R}^{a \times a}$ is the diagonal node degree matrix of $\tilde{\boldsymbol{A}} \in \mathbb{R}^{a \times a}$, and $\sigma(\cdot)$ is a nonlinear activation function. In our implementation, we chose f to be 34, and the number of GCN layers to be 1. After processed by GCN layer, the atom sequence $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_a$ is obtained, where a is the number of atoms.

When protein sequence representation and atom representation were obtained, we successfully converted proteins and compounds into two sequences, which fitted the transformer architecture. Interaction features are learned through the decoder of transformer, which consists of self-attention layers and feed forward layers. In our work, protein sequence is the input of encoder, while the atom sequence is the input of decoder, and the output of decoder is the interaction sequence which contains interaction features and has the

Table 1. List of compound atom features

Atom type	C, N, O, F, P, S, Cl, Br, I, other (one hot)
Degree of atom	0, 1, 2, 3, 4, 5, 6 (one hot)
Formal charge	0 or 1
Number of radical electrons	0 or 1
Hybridization type	sp, sp2, sp3, sp3d, sp3d2, other (one hot)
Aromatic	0 or 1
Number of hydrogen atoms attached	0, 1, 2, 3, 4 (one hot)
Chirality	0(False) or 1(True)
Configuration	R, S (one hot)

same length with atom sequence. Given that the order of atom feature vectors has no effect on CPI modeling, we removed positional embeddings in TransformerCPI.

The key technique in the decoder is multiheaded self-attention layer. A multiheaded self-attention layer consists of several scaled-dot attention layers to extract interaction information between the encoder and the decoder. The self-attention layer takes three inputs, the keys, K, the values, V and the queries, Q, and calculates the attention as follows:

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{3}$$

where d_k is a scaling factor depending on the layer size. This mechanism allows the decoder to focus on some crucial parts from the output of encoder dynamically, which directly captures the interaction features of the given two sequences. In addition, the original transformer was designed to solve sequence prediction tasks and utilize mask operation to cover the downstream context of each word in the decoder. Therefore, we modified the mask operation of the decoder to ensure that our model is accessible to whole sequence, which is one of the most crucial modification to transfer transformer architecture from autoregressive task to classification task (Fig. 2).

After extracting interaction features in the decoder, a set of interaction sequence $x_1, x_2, \ldots, x_a$ are obtained, and the modulus of each vector is computed as follows:

$$x_i' = \| x_i \|_2^2 , \tag{4}$$

where $i = 1, 2, 3, \ldots, a$. The weight of each vector can be calculated by softmax function as follows:

$$\alpha_i = \frac{e^{x_i'}}{\sum_{i=1}^{a} e^{x_i'}} . \tag{5}$$

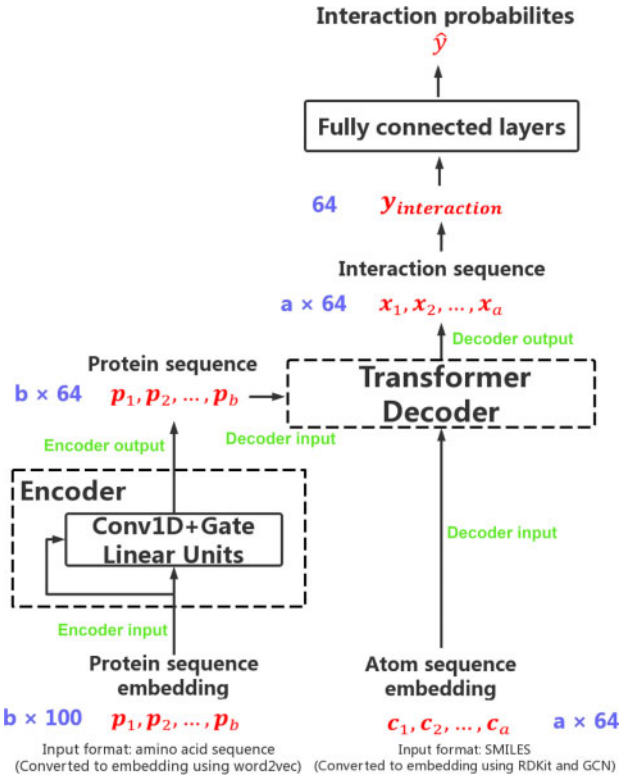

Fig. 2. Computational graph of TransformerCPI

The final interaction feature vector is calculated by weighted sum of interaction vectors with attention weights:

$$y_{\text{interaction}} = \sum_{i=1}^{a} \alpha_i x_i . \tag{6}$$

At last, the final interaction feature vector $y_{\text{interaction}}$ is fed to the following fully connected layers, and the probability $\hat{y}$ that a compound interacts with a protein is returned. As a conventional binary classification task, we used binary cross entropy loss to train TransformerCPI model:

$$\text{Loss} = -\left[y\log\hat{y} + (1-y)\log(1-\hat{y})\right] . \tag{7}$$

TransformerCPI was implemented with Pytorch 1.2.0, and the word2vec model was built and trained with Gensim 3.4.0. Whereas the base transformer model had six layers with 512 hidden dimension, we decreased the number of layers from 6 to 3 and the dimension of hidden layers from 512 to 64. The dimension of protein representation, atom representation, hidden layers and $y_{\text{interaction}}$ is 64. We kept the original eight attention heads because this configuration achieved superior generalization ability. For the training, we used the LookAhead (Zhang *et al.*, 2019) optimizer combined with RAdam (Liu *et al.*, 2019) optimizer, which solved the most serious convergence problems caused by Adam optimizer without the learning rate warmup. The learning rate was set to 0.0001, the batch size was set to 8 and the gradients were accumulated over eight batches. All the settings and hyperparameters of TransformerCPI are summarized in Table 2.

2.2 Datasets

2.2.1 Public datasets

We compared our model on three previous benchmark datasets, Human dataset, *Caenorhabditis elegans* dataset (Tsubaki *et al.*, 2019) and BindingDB dataset (Gao *et al.*, 2018). Human dataset and *C.elegans* dataset include positive CPI pairs from DrugBank 4.1 (Wishart *et al.*, 2008) and Matador (Gunther *et al.*, 2007) and highly credible negative CPI samples obtained using a systematic screening framework (Liu *et al.*, 2015). In detail, the human dataset contains 3369 positive interactions between 1052 unique compounds and 852 unique proteins; the *C.elegans* dataset contains 4000 positive interactions between 1434 unique compounds and 2504 unique proteins and the training, valid and test sets are randomly split (Tsubaki *et al.*, 2019). BindingDB dataset contains 39 747 positive examples and 31 218 negative examples from a public database (Gilson *et al.*, 2016). The training, valid and test sets of BindingDB are well-designed, and the test set includes CPI pairs where ligands or proteins are not observed in training set. Therefore, BindingDB dataset can assess models' generalization ability to unknown ligands and proteins.

2.2.2 Label reversal datasets

To construct datasets specifically for chemogenomics-based CPI modeling, we followed two rules: (i) collecting CPI data from

Table 2. Hyperparameters of TransformerCPI

Name	Value
Number of encoder layers	3
Number of decoder layers	3
Dimension of atom representation	64
Number of attention heads	8
FFN inner hidden size	512
Hidden size	64
Patch size	7
Learning rate	1e–4
Weight decay	1e–4
Batch size	8
Dropout	0.2

experimentally validated database; (ii) each ligand should exist in both two classes. Many previous studies generated negative samples by random cross combination of CPI pairs or by using similarity based approaches, which may introduce unexpected noise and unnoticed bias. Here, compiled negative data that have been experimentally validated.

First, we constructed a GPCR dataset from GLASS database (Chan *et al.*, 2015). GLASS database provides a great amount of experimentally validated GPCR–ligand associations (Chan *et al.*, 2015), which satisfies our first rule. GLASS database used IC_{50}, Ki and EC_{50} as the binding affinity values, which were transformed into negative logarithm, pIC_{50}, pK_i and pEC_{50}. Following early works (Liu *et al.*, 2007; Wan *et al.*, 2019), a threshold of 6.0 was set to divide original dataset into a positive set and a negative set. Then, we selected protein–compound pairs that follow our second rule to construct the final GPCR dataset. Our final GPCR dataset comprises 5359 ligands, 356 proteins and 15 343 CPI among them.

Second, we constructed a Kinase dataset based on KIBA dataset (Tang *et al.*, 2014). KIBA score was developed to combine various bioactivity types, including IC_{50}, K_i and K_d, and to remove inconsistency between different bioactivity types, which greatly reduced bias in the dataset (Tang *et al.*, 2014). The KIBA dataset contains 467 targets and 52 498 ligands collected from ChEMBL and STITCH (Szklarczyk *et al.*, 2016), which ensures that data in KIBA is experimentally validated. Given that the majority of ligands only occur once, we followed SimBoost (He *et al.*, 2017) to filter original KIBA dataset to comprise only compounds and proteins with at least 10 interactions, gaining a total of 229 proteins and 2111 compounds. Then, we used the suggested threshold KIBA value of 12.1 (He *et al.*, 2017; Tang *et al.*, 2014) to divide dataset into a positive set and a negative set, and selected protein–compound pairs where compounds occur in both positive set and negative set, yielding a total of 1644 compounds, 229 proteins and 111 237 CPI. Table 3 summarizes GPCR dataset and Kinase dataset we constructed.

As mentioned before, hidden ligand bias may cause a data-driven model to learn unexpected statistical clues or patterns in data other than the desired CPI information. To confirm the model actually learn the interaction features and accurately assess the impact of hidden variables, we proposed a more rigorous label reversal experiment. The schematic illustration of label reversal experiment is shown in Figure 3A, where a ligand in the training set appears only in one class of samples (either positive or negative interaction CPI pairs), while the ligand appears only in the opposite class of samples in the test set. In this way, the model was forced to utilize protein information to understand interaction modes and make opposite predictions for those chosen ligands. If a model only memorizes the ligand patterns, it is unlikely to make correct predictions because the ligands it memorizes have the wrong (opposite) labels in test set. Therefore, this label reversal experiment is specifically designed to evaluate chemogenomics-based CPI models and is capable of indicating how much influence the hidden ligand bias has exerted.

For GPCR set and Kinase set, we randomly selected 500 and 300 ligands, respectively, and pooled together all the negative CPI samples involving these ligands in the test set. Also, we selected another 500 and 300 ligands, respectively, and pooled together all their associated positive samples in the test set. Under this experiment design, we finally established GPCR test set with 1537 interactions and Kinase test set with 19 685 interactions. The remaining datasets were used to determine the hyperparameters, and the best model was selected to evaluate on label reversal experiments.

2.2.3 Data distribution of label reversal datasets

Before training the model, we studied the data distribution of GPCR set and Kinase set. Since each ligand may occur in multiple positive and negative classes, representing either interacting or non-interacting with different proteins, the frequency of occurrence in positive and negative samples was analyzed. On account of this issue, we calculated the log ratio of two classes for each ligand as follows to describe the data distribution:

Table 3. Summary of the datasets

	Proteins	Compounds	Interactions	Positive	Negative
GPCR	356	5359	15 343	7989	7354
Kinase	229	1644	111 237	23 190	88 047

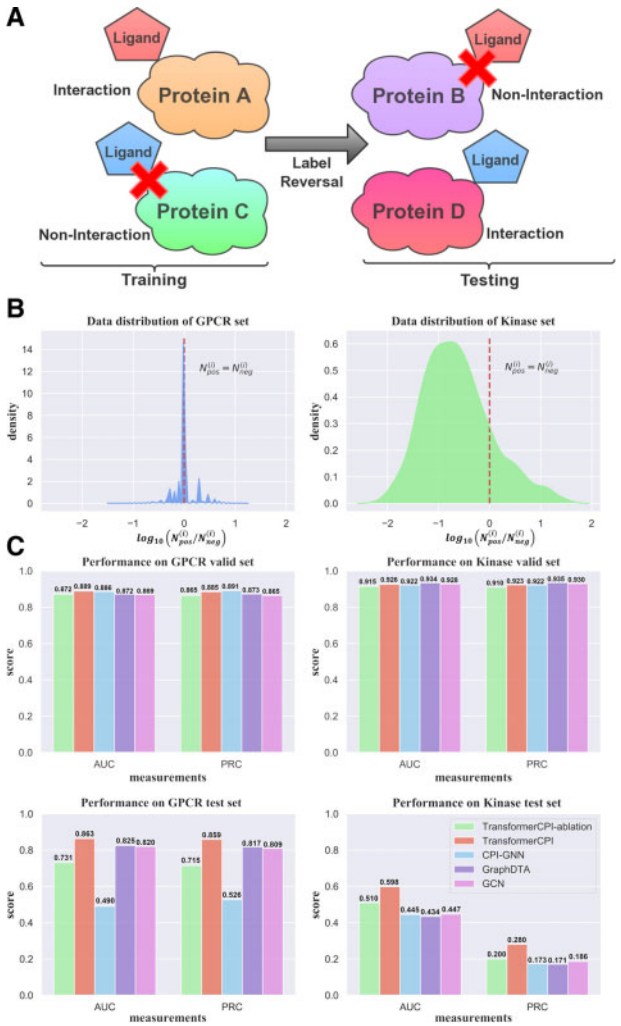

Fig. 3. (**A**) Schematic illustration of label reversal experiment, where a ligand in the training set appears only in one class of samples (either positive or negative interaction CPI pairs), and it appears only in the opposite class of samples in the test set. (**B**) Data distribution of two datasets. The red line represents where number of positive interactions equals to number of negative interactions. (**C**) Results of TransformerCPI, CPI–GNN, GraphDTA, GCN and TransformerCPI-ablation on GPCR valid set, Kinase valid set, GPCR test set and Kinase test set

$$\log_{\text{ratio}^{(i)}} = \log_{10}\left(\frac{N^{(i)}_{\text{pos}}}{N^{(i)}_{\text{neg}}}\right), \quad i = 1, 2, 3, \ldots, L,$$

where $N^{(i)}_{\text{pos}}$ is the number of ith ligand's positive interactions while $N^{(i)}_{\text{neg}}$ is the number of ith ligand's negative interactions, and L is the total number of ligands. The distributions of GPCR set and Kinase set are shown in Figure 3B.

3 Results and discussion

3.1 Performance on public datasets

Many machine learning methods, such as K nearest neighbors (KNN), random forest (RF), L2-logistic (L2), support vector

machines (SVM), newly reported sequence-based models CPI–GNN (Tsubaki *et al.*, 2019) and DrugVQA (Zheng *et al.*, 2020) have been evaluated on these datasets. GraphDTA (Nguyen *et al.*, 2019) was originally designed for regression task, here, we tailor its last layer to binary classification task. It should be noted that models relying on 3D structural information of protein are not compared here, due to the absence of such information for these two datasets. We followed the same training and evaluating strategies as CPI–GNN (Tsubaki *et al.*, 2019) and repeated with three different random seeds followed by DrugVQA (Zheng *et al.*, 2020) to evaluate TransformerCPI, and Area Under Receiver Operating Characteristic Curve (AUC), precision and recall of each model are shown in Tables 4 and 5. Since the implementation of KNN, RF, L2, SVM are not mentioned in the literature (Tsubaki *et al.*, 2019), these models are not compared on BindingDB dataset. Area Under Precision Recall Curve (PRC) and AUC of each model are shown in Table 6. TransformerCPI outperformed other models on three public datasets.

3.2 Performance on label reversal datasets

We chose CPI–GNN, GraphDTA and GCN as references and compared the performance of TransformerCPI with these models in terms of AUC and PRC. Figure 3C summarizes the AUC and PRC of these models. To conduct fair comparison, each model was thoroughly fine-tuned on the same valid sets. As shown in Figure 3C, all of the models achieved similar performance on both GPCR valid set and Kinase valid set, however, big performance gaps between these models were observed on test sets. Although these models have similar performance on random split valid sets, the knowledge they have learned greatly differs from each other, which is exposed by a more rigorous label reversal experiment. On GPCR set, TransformerCPI outperformed CPI–GNN, GraphDTA and GCN both in terms of AUC and PRC, showing improved power to capture interaction features between compounds and proteins. Compared with other models, CPI–GNN failed to enrich positive samples before negative samples in label reversal experiments, so we argue that ligand patterns of GPCR dataset may bring in non-negligible influence as ligand bias in CPI–GNN. On Kinase set, TransformerCPI outperforms CPI–GNN, GraphDTA and GCN in terms of AUC and PRC and AUC of reference models are all smaller than 0.5, so we argue that ligand patterns of Kinase dataset may brought in non-negligible influence in all reference models. Moreover, GraphDTA and GCN achieved good performance on GPCR dataset, which are close to TransformerCPI, but performed much worse on Kinase set. In comparison, TransformerCPI achieved the best performance on both datasets, revealing its robustness and generalization ability. Overall, these results suggested that our proposed TransformerCPI possesses capability of learning interactions between proteins and ligands, and the label reversal experiments can effectively assess the impact of hidden ligand bias on models, and, more importantly, the proposed modeling scheme is useful in reducing common risks of chemogenomics-based CPI tasks.

Table 4. Comparison results of the proposed model and baselines on human dataset

Method	AUC	Precision	Recall
KNN	0.860	0.927	0.798
RF	0.940	0.897	0.861
L2	0.911	0.913	0.867
SVM	0.910	**0.966**	**0.969**
GraphDTA	0.960 ± 0.005	0.882 ± 0.040	0.912 ± 0.040
GCN	0.956 ± 0.004	0.862 ± 0.006	0.928 ± 0.010
CPI–GNN	0.970	0.918	0.923
DrugVQA (VQA-seq)[a]	0.964 ± 0.005	0.897 ± 0.004	0.948 ± 0.003
TransformerCPI	**0.973 ± 0.002**	0.916 ± 0.006	0.925 ± 0.006

[a]It should be noted that DrugVQA uses protein structural information as input, while its alternative version VQA-seq only using protein sequence information is listed here for a fair comparison.

Table 5. Comparison results of the proposed model and baselines on *C.elegans* dataset

Method	AUC	Precision	Recall
KNN	0.858	0.801	0.827
RF	0.902	0.821	0.844
L2	0.892	0.890	0.877
SVM	0.894	0.785	0.818
GraphDTA	0.974 ± 0.004	0.927 ± 0.015	0.912 ± 0.023
GCN	0.975 ± 0.004	0.921 ± 0.008	0.927 ± 0.006
CPI–GNN	0.978	0.938	0.929
TransformerCPI	**0.988 ± 0.002**	**0.952 ± 0.006**	**0.953 ± 0.005**

3.3 The system dependency of models

When comparing the results between GPCR set and Kinase set, it is also of note that TransformerCPI, GraphDTA and GCN perform much better on GPCR set than Kinase set. We argue that there might be two potential reasons for this performance difference. The first one is that the data distribution of GPCR set and Kinase set is different, resulting in performance gap between the two datasets. The second one is that the sequence features of GPCR are relatively easier for TransformerCPI to learn.

As shown in Figure 3B, the peak of log_ratio distribution of GPCR set is located at zero, which means most ligands in GPCR set have an equal quantity of interaction pairs and noninteraction pairs. In contrast, the peak of log_ratio distribution of Kinase set is significantly shifted to -1, indicating that most ligands in Kinase set possess almost ten times more noninteraction pairs than interaction pairs. Therefore, the highly unbalanced distribution of positive pairs and negative pairs may introduce severe ligand bias to the dataset, which might increase the risk that a data-driven model memorizes ligand patterns, causing inferior prediction performance on Kinase set.

Another potential reason is that the CPI-associated sequence features of GPCR are easier to learn than those of Kinase. Although GPCR family shares massive alpha-helix regions and seven transmembrane structures, the binding location and binding pockets are more diverse across the family, which is relatively easy for models to learn CPI-associated sequence features to distinguish interaction pairs and noninteraction pairs. However, compared with GPCR family, Kinase family shares a more conservative ATP binding pocket with fewer different residues. It is challenging for models to distinguish interaction and non-interaction pairs since the model has to learn to detect and understand the minor changes on protein sequences. Furthermore, the system dependency of TransformerCPI also informs us that there is still room for improvement in chemogenomics-based CPI prediction, especially the representation of protein sequences.

3.4 Model ablation study

Previous chemogenomics-based CPI models extract ligand and protein features separately and independently, and then concatenate these two feature vectors as input features. To validate the role of the transformer encoder–decoder architecture, we next evaluated the TransformerCPI-ablation model which replaces transformer

Table 6. Comparison results of the proposed model and baselines on BindingDB dataset

Method	AUC	PRC
GraphDTA	0.929	0.917
GCN	0.927	0.913
CPI–GNN	0.603	0.543
TransformerCPI	**0.951**	**0.949**

decoder by conventional vector concatenation on the same label reversal experiment. As shown in Figure 3C, this ablation procedure significantly compromised the performance of TransformerCPI on both GPCR set and Kinase set, demonstrating that self-attention mechanism together with encoder–decoder architecture indeed plays a key role in extracting CPI features between two types of sequences.

3.5 Model interpretation

Although deep learning is known as a black box algorithm, it is essential to understand how the model makes a prediction, and whether the model can provide suggestions or guidance for optimization. Due to the transformer architecture and self-attention mechanism, it provides easy access for understanding the mechanism behind the model through attention weights of protein sequences and compound atoms.

As illustrated in Figure 4A, the attention weights were mapped to compound atoms to reveal the knowledge TransformerCPI has learned. TransformerCPI pays attention to different atoms when facing different compound protein pairs and correctly classify the compound protein pairs into two classes, interaction and noninteraction.

Here, TransformerCPI generates different ligand features corresponding to different proteins, which is consistent with the fact that binding modes of the ligand are different when interacting with different proteins. Therefore, it is difficult for TransformerCPI to memorize ligand patterns because the ligand features are changing with regard to different proteins. This result also explains why TransformerCPI shows better performance on label reversal experiments. Dynamic feature extraction of TransformerCPI based on a specific protein context helps the model extract the key information of the interaction, while also reducing the probability of hidden ligand bias. Moreover, the decoder of TransformerCPI integrates the features of protein sequences and compound atoms dynamically to form direct interaction features, which is similar to language translation task and agrees well with the binding process of ligands to proteins.

To further verify the meaning of the attention weights of atoms, we selected the compound phenothiazine to show the interpretation of TransformerCPI. Phenothiazine is a classic antipsychotic drug targeted on dopamine receptor, and its structure–activity relationship (SAR) has been thoroughly explored. As illustrated in Figure 4B, the atoms of phenothiazine highlighted by attention weights agree well with the SAR of phenothiazine, which confirms that our model is capable of catching true interaction features and finding out key atoms interacting with proteins. This information of great value assists medicinal chemists to speculate the potential SAR of the target molecule and may offer useful guidance to further structural optimization.

After interpreting atom-level attention mechanism, we also studied the attention weights of the protein sequences to see which parts of the protein sequences became the focus of attention. As a result, TransformerCPI may roughly speculate whether the binding site of the ligand to the GPCR family is in the extracellular region or in the transmembrane region, and detect the ATP-binding pocket of Kinase family. We selected histamine H1 receptor, 5-HT_{1B} receptor and mitogen-activated protein kinase 8 (MAPK8) with their corresponding actives as examples.

As shown in Figure 5, TransformerCPI successfully localized the binding site of the ligand to histamine H1 receptor in the transmembrane region, the binding site of the ligand to 5-HT_{1B} receptor in the extracellular region, and detected ATP-binding pocket of MAPK8, which further verifies that TransformerCPI has learned biological knowledge and gained structural insights. These results suggested that TransformerCPI can speculate whether a new compound binds to the transmembrane region or the extracellular region of a GPCR target, which is useful in drug design especially when 3D structure of the GPCR target is unknown. Meanwhile, we may also notice that the highlighted region involves more extensive regions, and does not correspond to the exact binding site residues. To address this issue, more high-quality data with precise annotation need to be incorporated, and new sequence-based deep representation learning may be also of help for better encoding and decoding the structural information. For example, a new representation scheme recently proposed by Alley *et al.* (2019) has demonstrated efficiency improvement for studying protein sequences.

Overall, there is still a long way for chemogenomics-based CPI to go, and we hope that this work can raise our attention to problems of chemogenomics-based CPI modeling and provide useful guidance for further study. In addition, experiment design plays an import role in deep learning, and more efforts should be directed toward evaluating what a deep learning model has really learned. In this way, not only the new deep learning approaches but also new validation strategies and experiment designs should be emphasized in future development of deep learning.

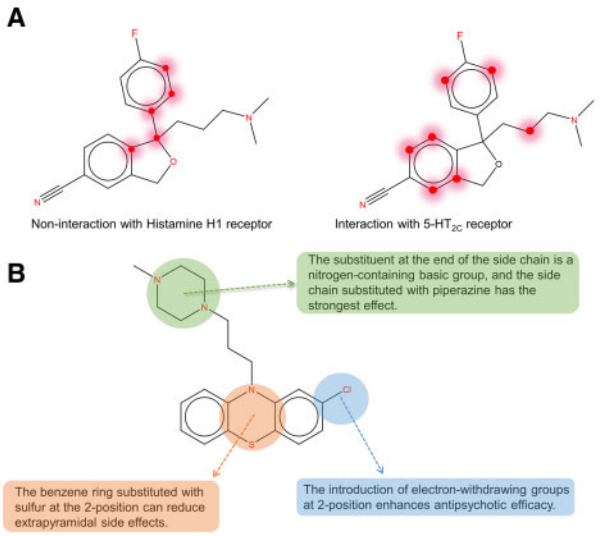

Fig. 4. Attention weights of atoms in different compounds. The atoms, which have high attention scores extracted from TransformerCPI, are highlighted in red. (**A**) A certain ligand shows different attention score distributions when interacting with Histamine H1 receptor and 5-HT_{2C} receptor, respectively. (**B**) The SAR of phenothiazine and its attention scores

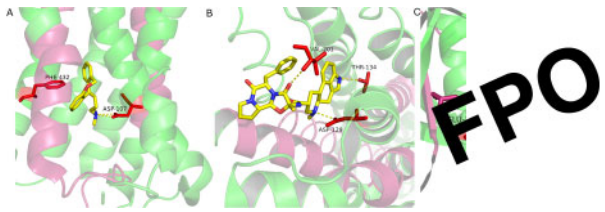

Fig. 5. Attention weights of protein sequences. The regions in proteins, which have high attention weights extracted from TransformerCPI, are highlighted in purple. (**A**) Attention weight of histamine H1 receptor (PDB: 3RZE). (**B**) Attention weight of 5-HT_{1B} receptor (PDB: 4IAQ). (**C**)Attention weight of MAPK8 (PDB: 1UKI)

4 Conclusion

In this work, a transformer architecture with self-attention mechanism was modified to address sequence-based CPI classification task, resulting in a model named TransformerCPI showing high performance on three benchmark datasets. Intriguingly, we compared it with previous reported CPI models and conventional machine learning-based control models, and noticed that most of these models yielded impressive results on those benchmark tests. Given the challenging nature of CPI prediction, we argue that these models might face potential pitfalls of deep learning. To address these potential risks, we constructed new datasets specific for chemogenomics-based CPI task, and designed more rigorous label reversal experiments as new measurements for chemogenomics-based CPI modeling. Compared with other models, TransformerCPI achieved significantly improved performance on the new experiments, suggesting it can learn desired interaction features and decrease the risk of hidden ligand bias. Finally, model interpretation capability was studied through mapping attention weights to protein sequences and compound atoms, which could help us determine whether a prediction is reliable and having physical significance. Overall, TransformerCPI provides access to model interpretation and contributes chemical biology studies with useful guidance for further ligand structural optimization.

Funding

This work was supported by the National Natural Science Foundation of China (81773634 to M.Z.), National Science & Technology Major Project 'Key New Drug Creation and Manufacturing Program', China (Number: 2018ZX09711002 to H.J.) and 'Personalized Medicines—Molecular Signature-based Drug Discovery and Development', Strategic Priority Research Program of the Chinese Academy of Sciences (XDA12050201 to M.Z.).

Conflict of Interest: none declared.

References

Alley,E.C. *et al.* (2019) Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, **16**, 1315–1322.

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

Bredel,M. and Jacoby,E. (2004) Chemogenomics: an emerging strategy for rapid target and drug discovery. *Nat. Rev. Genet.*, **5**, 262–275.

Chan,W.K.B. *et al.* (2015) GLASS: a comprehensive database for experimentally validated GPCR–ligand associations. *Bioinformatics (Oxford, England)*, **31**, 3035–3042.

Chen,L. *et al.* (2019) Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PLoS One*, **14**, e0220113.

Cheng,F. *et al.* (2012) Prediction of chemical–protein interactions: multitarget-QSAR versus computational chemogenomic methods. *Mol. Biosyst.*, **8**, 2373–2384.

Dai,Z. *et al.* (2019) Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2978–2988. Association for Computational Linguistics*. Florence, Italy.

Dauphin,Y. *et al.* (2016) Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, JMLR.org, Sydney, NSW, Australia, pp. 933–941.

Devlin,J. *et al.* (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota.

Gao,K. *et al.* (2018) Interpretable drug target prediction using deep neural representation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, AAAI Press, Stockholm, Sweden. pp. 3371–3377.

Gilson,M.K. *et al.* (2016) BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.*, **44**, D1045–1053.

Gonen,M. (2012) Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization. *Bioinformatics*, **28**, 2304–2310.

Gunther,S. *et al.* (2007) SuperTarget and Matador: resources for exploring drug–target relationships. *Nucleic Acids Res.*, **36**, D919–D922. (Database issue).

Hamanaka,M. *et al.* (2017) CGBVS-DNN: prediction of compound–protein interactions based on deep learning. *Mol. Inform.*, **36**, 1–2.

He,T. *et al.* (2017) SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *J. Cheminform.*, **9**, 24.

Jacob,L. and Vert,J.P. (2008) Protein–ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, **24**, 2149–2156.

Karimi,M. *et al.* (2019) DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, **35**, 3329–3338.

Kimothi,D. *et al.* (2016) Distributed Representations for Biological Sequence Analysis. In, arXiv e-prints. 2016. p. arXiv:1608.05949.

Kipf,T. and Welling,M. (2016) Semi-supervised classification with graph convolutional networks. In, *arXiv e-prints*. 2016. p. arXiv:1609.02907.

Kobeissy,F.H. *et al.* (2015) Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS One*, **10**, e0141287.

Lee,I. *et al.* (2019) DeepConv-DTI: prediction of drug–target interactions via deep learning with convolution on protein sequences. *PLoS Comput. Biol.*, **15**, e1007129.

Liu,H. *et al.* (2015) Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics*, **31**, i221–i229.

Liu,L. *et al.* (2019) On the variance of the adaptive learning rate and beyond. In, arXiv e-prints. 2019. p. arXiv:1908.03265.

Liu,T. *et al.* (2007) BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Res.*, **35**, D198–D201.

Mazzaferro,C. (2017) Predicting protein binding affinity with word embeddings and recurrent neural networks. http://dx.doi.org/10.1101/128223.

Mikolov,T. *et al.* (2013a) Efficient estimation of word representations in vector space. In, arXiv e-prints. 2013. p. arXiv:1301.3781.

Mikolov,T. *et al.* (2013b) Distributed representations of words and phrases and their compositionality. *Adv. Neural Inform. Process. Syst.*, **26**, 3111–3119.

Mysinger,M.M. *et al.* (2012) Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *J. Med. Chem.*, **55**, 6582–6594.

Nguyen,T. *et al.* (2019) GraphDTA: prediction of drug–target binding affinity using graph convolutional networks. *bioRxiv* : doi: http://dx.doi.org/10.1101/684662.

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

Öztürk,H. *et al.* (2019) WideDTA: prediction of drug–target binding affinity. In, arXiv e-prints. 2019. p. arXiv:1902.04166.

Qiu,X. *et al.* (2020) Pre-trained Models for Natural Language Processing: A Survey. In, arXiv e-prints. 2020. p. arXiv:2003.08271.

Riley,P. (2019) Three pitfalls to avoid in machine learning. *Nature*, **572**, 27–29.

Scarselli,F. *et al.* (2009) The graph neural network model. *IEEE Trans. Neural Netw.*, **20**, 61–80.

Schwaller,P. *et al.* (2019) Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Central Science.*, **5**, 1572–1583.

Sieg,J. *et al.* (2019) In need of bias control: evaluating chemical data for machine learning in structure-based virtual screening. *J. Chem. Inf. Model.*, **59**, 947–961.

Szklarczyk,D. *et al.* (2016) STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data. *Nucleic Acids Res.*, **44**, D380–D384.

Tang,J. *et al.* (2014) Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *J. Chem. Inf. Model.*, **54**, 735–743.

Tian,K. *et al.* (2016) Boosting compound–protein interaction prediction by deep learning. *Methods*, **110**, 64–72.

Tsubaki,M. *et al*. (2019) Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, **35**, 309–318.

Vamathevan,J. *et al*. (2019) Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov*., **18**, 463–477.

van Laarhoven,T. *et al*. (2011) Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics*, **27**, 3036–3043.

Vaswani,A. *et al*. (2017) Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pp. 6000-6010. Long Beach, California, USA, Curran Associates Inc.

Wan,F. and Zeng,J. (2016) Deep learning with feature embedding for compound–protein interaction prediction. *bioRxiv* doi:10.1101/086033.

Wan,F. *et al*. (2019) DeepCPI: a deep learning-based framework for large-scale in silico drug screening. *Genomics Proteomics Bioinf*., **17**, 478–495.

Wang,F. *et al*. (2011) Computational screening for active compounds targeting protein sequences: methodology and experimental validation. *J. Chem. Inf. Model*., **51**, 2821–2828.

Wang,Y. and Zeng,J. (2013) Predicting drug–target interactions using restricted Boltzmann machines. *Bioinformatics*, **29**, i126–i134.

Wishart,D.S. *et al*. (2008) DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res*., **36**, D901–D906.

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

Yang,K.K. *et al*. (2018) Learned protein embeddings for machine learning. *Bioinformatics*, **34**, 2642–2648.

Yang,Z. *et al*. (2019) XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pp. 2978–2988. Association for Computational Linguistics. Florence, Italy.

Zhang,M. *et al*. (2019) Lookahead optimizer: k steps forward, 1 step back. In, arXiv e-prints. 2019. p. arXiv:1907.08610.

Zheng,S. *et al*. (2020) Predicting drug–protein interaction using quasi-visual question answering system. *Nat. Mach. Intell*., **2**, 134–140.