DeepPLA: a novel deep learning-based model for protein-ligand binding affinity prediction

Bomin Wei[1,*], Xiang Gong[1]

[1.] Princeton International School of Mathematics and Science. 19 Lambert Dr, Princeton NJ 08540

E-mail:

Bomin Wei: david.wbm.2022@gmail.com

Xiang Gong: xiang.gong@prismsus.org

*** Corresponding Author:**

Bomin Wei: david.wbm.2022@gmail.com

Abstract

The substantial cost of new drug research and development has consistently posed a huge burden and tremendous challenge for both pharmaceutical companies and patients. In order to lower the expenditure and development failure rate, repurposing existing and approved drugs and identifying novel interactions between the drug molecules and the target proteins based on computational methods have gained growing attention. Here, we propose the DeepPLA, a novel deep learning-based model that combines ResNet-based 1D CNN and biLSTM, to establish an end-to-end network for protein-ligand binding affinity prediction. We first apply pre-trained embedding methods to encode the raw drug molecular SMILES strings and target protein sequences into dense vector representations. The dense vector representations separately go through ResNet-based 1D CNN modules to derive features. The extracted feature vectors are concatenated and further fed into the biLSTM network after average pooling operation, followed by the MLP module to finally predict binding affinity. We used BindingDB dataset for training and evaluating our DeepPLA model. The result shows that the DeepPLA model reaches a good performance for the protein-ligand binding affinity prediction in terms of R, $RMSE$, MAE, R^2 and MSE with 0.89, 0.68, 0.50, 0.79 and 0.46 on the training set; and scores 0.84, 0.80, 0.60, 0.71 and 0.64 on the independent testing set, respectively. This result suggests the high accuracy of the DeepPLA prediction performance, as well as its high capability in generalization, demonstrating that the DeepPLA can be the potential upgrade to pinpoint new drug-target interactions to find better destinations for proven drugs.

Keywords: drug repurposing, deep learning, binding affinity perdiction

1 Introduction

Introducing a new drug to the market has been characterized to be risky, time-consuming, and costly. Indeed, it may take 10 to 20 years, and more than 2 billion US dollars to develop a new drug as reported in 2018 [1][2]. Typically, the overall process of drug research and development is complicated by serval core steps [3], including (i) drug discovery, (ii) preclinical research, (iii) clinical trials, and (iv) FDA review, and (v) post-marketing safety surveillance. Particularly, drug discovery is the first phase starting with identifying targets of an unmet disease such as proteins, followed by creating and optimizing a promising compound that can interact with the targets efficiently and safely. This step usually involves hundreds and thousands of compounds, yet only about 8% of which as drug leads can enter the phase of the *in vitro* and *in vivo* preclinical research [4]. To shorten the duration and to improve success rate in the phase of drug discovery, drug repurposing has become a hotspot of new drug research and development over the past few years [1][5]. Drug repurposing, or drug repositioning, intends to find an effective cure of a disease from a large amount of existing and approved drugs that were developed for other purposes [1]. For example, prednisone was originally developed for the treatment of inflammatory diseases but it is likely to be effective against Parkinson's disease as well [6]. This method could potentially lower the R&D costs since the candidate drugs have already been proven to be safe. Therefore, the drug could quickly pass clinical trial phases [7]. In midst of all the drug repurposing methods, *in silico* computational-based methods to screen pharmaceutical compound libraries and identify drug-target interactions (DTIs) have gained increasing attention and made significant breakthroughs thanks to the development in high performance of computational architectures and advances in machine learning methods.

The identification of DTIs aims to study the binding affinity that measures the strength of protein-ligand interaction between a target protein and a drug compound. The binding affinity is usually represented in the form of the following constants that stand for inhibition (K_i), dissociation (K_d), and half-maximal inhibitory (IC_{50}). Small values in the constants means a stronger binding affinity of a protein-ligand pair. Conventional methods which intend to identify the protein-ligand interactions involve complicated quantum chemical calculation of molecular/biological structures. Thereby, those conventional methods could not be a capable way of learning in the existing database

to discover a positive drug-target pair. Over the last decade, a variety of machine learning-based models have been developed to identify DTI from millions of ligands and proteins such as random forest (RF) based algorithm[8][9], SimBoost[10], and ChemBoost[11]. These models using classical machine learning algorithms to predict DTI were mainly built on human-selected features. The problem is that the generation of these features not only requires much domain knowledge but also possibly leads to a loss of the information about raw protein-ligand interactions. The emergence of deep learning-based techniques and their successful applications have paved the promising way to discover new drugs, beyond applications such as computer vision or language processing. Deep learning-based models can automatically learn complex and highly abstract level of features from large-scale raw input datasets without extensive manual creation of features, and people have tried a few deep learning methods to predict DTI. For instance, MFDR[12] obtains features with auto-encoder from chemical structures and protein sequences and then employs SVM models to predict DTI as a 0-1 classification problem. DeepDTA[13] applies two different CNNs modules separately to represent sequences of compounds and proteins as information modules. The resulting features from the information modules then enter 3 fully connected layers to predict protein-ligand binding affinity. Yet, the use of a simple label encoding method in DeepDTA to embed raw input sequences (i.e., representing symbols in raw sequences using corresponding encoded integers) may lose much information about raw sequences. Atomnet[14] and SE-OnionNet[15] utilize 3D structures of proteins and drug molecules to predict the drug-target binding affinity[16] greatly diminishes the problem of losing information. However, the practicability and the accuracy of those models are limited due to the insufficient 3D protein structure data[17]. It is hard to obtain accurate 3D structure data for protein because it requires advanced experiment methods under harsh and extreme conditions[18], [19].

Here, we propose DeepPLA, an innovative deep learning-based model for protein-ligand binding affinity prediction using protein 1D sequences and 1D ligands SMILES strings. SMILES, shorthanded for **S**implified **M**olecular **I**nput **L**ine **E**ntry **S**ystem, is a well-known compact linear notation method for representing molecular structures based upon chemical rules [20]. We first respectively employ pre-trained models of Mol2Vec[20] and ProSE[21] to embed ligand SMILES strings and protein sequences as numeric vectors, then feed the embedded numeric vectors of proteins and ligands into the Head CNN modules and ResNet-based CNN modules to encode

proteins and ligands sequences, respectively. The encoded representations are concatenated into a vector and further fed into a bi-directional LSTM (long short-term memory) layer, followed by three fully connected layers. The output of DeepPLA is a continuous value, representing the binding affinity of the input pair of ligand and protein. We use the BindingDB dataset[22] to train the DeepPLA model and evaluate the performance on binding affinity. The raw data is first preprocessed, including removal of protein-ligand pairs without K_i values, construction of a training set by randomly selecting 80% of the entire datset, and the rest 20% are saved as an independent testing set for evaluation. The experimental result shows that DeepPLA achieved high accuracy of protein-ligand binding affinity prediction in terms of R, $RMSE$, MAE, R^2, and MSE, as 0.89, 0.68, 0.50, 0.79, and 0.46 based upon the training set, while reached 0.84, 0.80, 0.60, 0.71, and 0.64 on the independent testing set, respectively. The result suggests DeepPLA has the potential to accurately identify DTI and hence, promote the new drug development.

This paper is structured as follows. Section 2 describes the dataset and the DeepPLA model; Section 3 illustrates the results of the prediction using the DeepPLA; and I discuss the current study and draw the conclusion in Section 4 and Section 5, respectively.

2 Methods

2.1 Dataset and data preprocessing

We use the BindingDB[22], a publicly accessible dataset, to train and evaluate our DeepPLA model. The BindingDB is a continually updating dataset that contains 2,278,226 experimentally identified binding affinities between 8,005 target proteins and 986,143 small drug molecules up to July 29, 2021. We first apply the following criteria to compile the dataset for the development of our model (Figure 1A): (1) excluding binding interactions with multichain protein complexes because it is not capable of identifying which chain of the protein interacts with the molecular; (2) retaining binding interactions only represented by K_i value and it means that other measurements in the form of $IC50$ or K_d values are removed; (3) keeping common drug molecules and target proteins occurring in at least three and six interactions in the entire dataset [11], respectively; (4) removing data with invalid K_i values. For example, we notice that some data used “>” and “<” in the K_i values to indicate ranges and we directly exclude them for the subsequent analysis. Additionally,

there are some zeros in the K_i values which should not appear based on the definition of K_i. Thus, we treat them as invalid values and simply removed them; (5) transferring K_i values into pK_i using the following equation (eq.1) due to the skewed distribution of original values of K_i:

$$pK_i = -\log_{10} K_i \qquad \text{eq.1}$$

(6) removing data where the pK_i values are less than -8 because these extremely small pK_i values suggest that the corresponding drug molecules and protein targets held much weak or even no interactions. As a result, a total of 249,517 interactions with 54,135 drug molecules and 1,844 protein targets are finally used in developing our model.

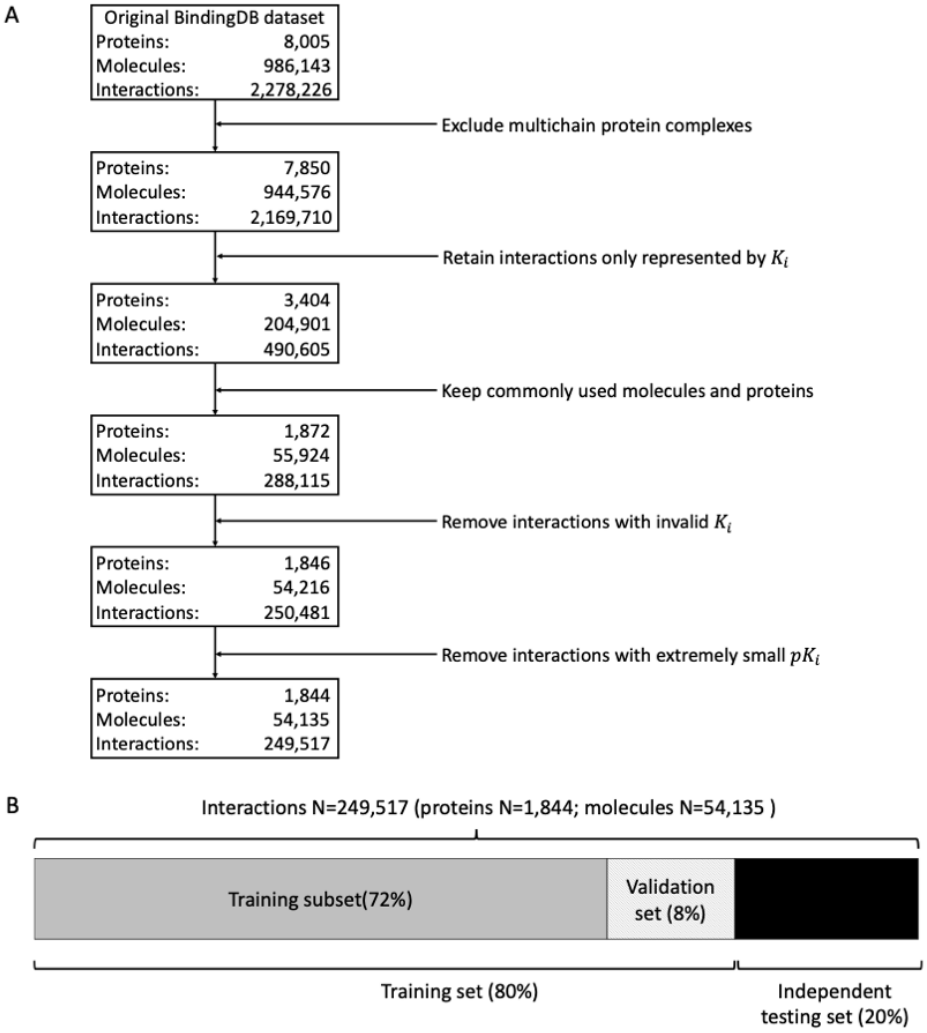


Figure 1 Preprocessing of BindingDB dataset. (A) Data exclusion criteria to compile BindingDB dataset. After performing the exclusion procedure, we finally obtain 249,517 interactions with 1,844 protein targets and 54,135 drug molecules. (B) Splitting the whole dataset into a training set (80% of whole interactions) and an independent testing set (20%) for training and evaluating the

DeepPLA model, respectively. The training set is further divided into the training subset (72%) and the validation set (8%).

We then randomly select 80% of whole interactions as the training set and the remaining 20% as the independent testing set to train and evaluate our DeepPLA model. In order to optimize hyperparameters, we further allocate 10% of the training set for validation during the training phase (i.e., 8% of all data), and the rest are used as a training subset (i.e., 72% of all data) (Figure 1B). After identification of the optimal hyperparameters, the whole training set is used to re-train the final DeepPLA model.

2.2 Model Design

2.2.1 Overview of DeepPLA model

The proposed DeepPLA consists of eight modules (Figure 2), including two embedding modules, two head modules, two ResNet-based CNN modules, one bi-directional LSTM (biLSTM) module, and one multilayer perceptron module. DeepPLA employs raw molecular SMILES strings and protein sequences as inputs, which are first represented as numeric vectors by two embedding modules separately. The embedded vectors for the ligand SMILES and the protein sequences are then fed into the respective head module and ResNet-based CNN module to extract features. The feature vectors for the inputs of drug molecules and protein targets are concatenated, pooled (max-pooling operation), and then encoded by a bi-LSTM layer. Subsequently, the encoded vectors are finally fed into an MLP module to predict binding affinity in terms of pK_i value.

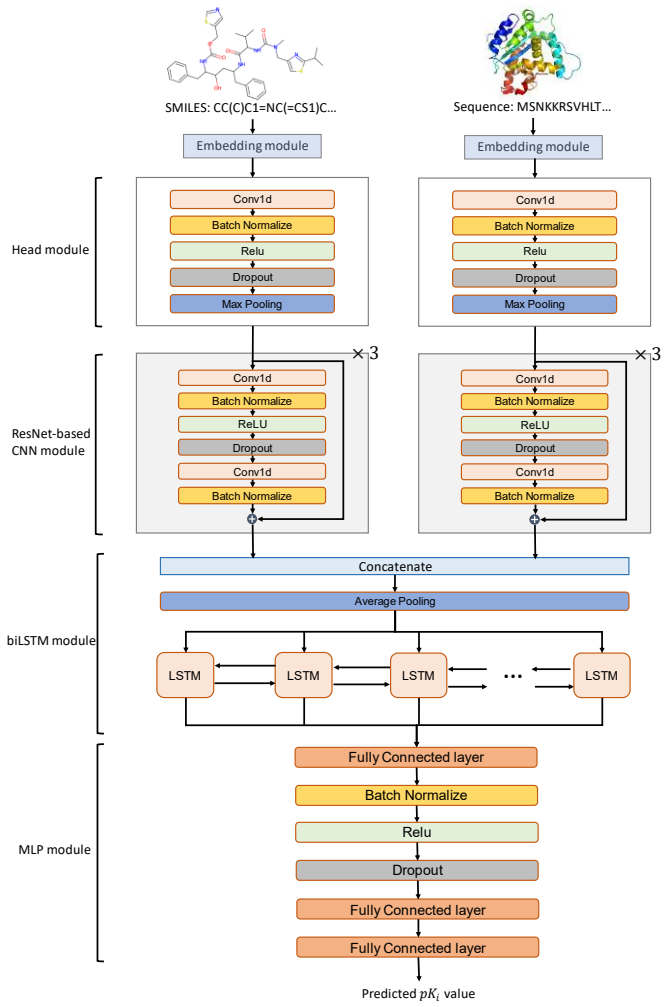


Figure 2 The overview of the DeepPLA model flow.

2.2.2 Embedding module

To utilize the raw drug molecular SMILES string and protein sequence as inputs to the DeepPLA model, we firstly encode them into numeric vector representations using the pre-trained embedding models called Mol2Vec[20] and ProSE[21], respectively. Mol2Vec is an unsupervised deep learning-based approach to convert a molecule into a numeric vector representation. Inspired by natural language processing (NLP) techniques, Mol2Vec regards the molecular substructures obtained by the Morgan identifier [23] as "words" and the compound as "sentences", and then encodes them into dense vector representations based on a so-called corpus of compounds. On the other hand, the ProSE is a deep learning-based method developed to represent protein sequences into numeric vectors that encode protein structural information. It first translates a protein sequence into a list of specific alphabets (as a “sentence”) which map similar amino acids (as “words”) into close numbers. Then, the ProSE model

encodes the words into numeric vectors. We utilize the pre-trained Mol2Vec (download link: https://github.com/samoturk/mol2vec) and ProSE (download link, https://github.com/tbepler/prose) to obtain vector representations with a fixed length for the drug molecular compound and protein, respectively.

2.2.3 Head module and ResNet-based CNN module

Head modules extract features from inputs. After the embedding, we separately feed the drug molecular SMILES string vector and protein sequence vector each into the head modules with the same network architecture. The head module contained the following layers: 1D convolutional, batch normalization, nonlinear transformation (with the rectified linear unit, i.e., ReLU activation), dropout, and max-pooling. Subsequently, two ResNet-based CNN modules are connected to the corresponding head module to further encode the information of input. Similar to the head module, the two ResNet-based CNN modules had the same network architecture. Specifically, each ResNet-based CNN module consists of three consecutive ResNet-based blocks, and each block comprises two branches, where the right branch is known as “shortcut connection”; and the left branch is known as a residual network that contains several stacked layers, including a 1D convolutional layer, a batch normalization layer, a ReLU layer, a dropout layer, another 1D convolutional layer, and one more batch normalization layer in sequence. Suppose x is the input into a ResNet-based block, the output of stacked layers is called residual, denoted as $F(x)$, we then calculated ResNet-based block output with equation $H(x) = F(x) + x$ [24].

2.2.4 biLSTM module and MLP module

In the bi-LSTM module, we first concatenate the outputs of features extracted by the aforementioned two ResNet-based CNN modules, following with a average-pooling layer. The bi-LSTM, which stands for **bi**directional **l**ong **s**hort-**t**erm memory, can learn long-term dependency from inputs. The LSTM network is a widely used RNN (**r**ecursive **n**eural **n**etwork) model, typically consisting of a list of memory blocks called Cells. Each Cell sends the cell state and the hidden state to next neighbor as memory. A Cell in an LSTM has three "gates", including (i) the forget gate to remove useless information from the cell state by performing a sigmoid function; (ii) the input gate which adds information to the cell state; and (iii) the output gate to further filter and select information

from the current cell state, and finalize the hidden state as an output of current Cell. Bi-LSTM network processes the input twice, once from starting to the end and once the reverse way. With this process, the network can keep information from the past and future. To this end, it can both keep the information of molecular when extracting from protein and the protein information when extracting the molecular once. Finally, the output from each Cell on each side of bi-LSTM will be combined as the output vector.

In the MLP module, we flatten the output vector of bi-LSTM and fed it into a stack of consecutive layers for processing. Finally, the DeepPLA output a scalar value that represents the predicted binding affinity in the form of pK_i.

2.3 Loss function

We use the mean squared error (MSE) as a loss function. The L2-norm regularization is added into the loss function. For the n pairs of molecular SMILES strings and protein sequences corresponding to the binding affinities in the form of pK_i, the loss function of the DeepPLA model was given by:

$$L = MSE + L_2 = \frac{1}{n}\sum_{k=1}^{n}\left(f(x_k^{mol}, x_k^{prot}) - y_k\right)^2 + \alpha\|w\|_2^2 \qquad \text{eq.2}$$

where $f(x_k^{mol}, x_k^{prot})$ is the predicted output of the DeepPLA model with inputs of the kth molecular SMILES string x_k^{mol} and the *k*th protein sequence x_k^{prot}. y_k is the true value of pK_i. w represents all the weight parameters of the DeepPLA model. L_2 stands for the L2-norm regularization. α is a scalar hyperparameter weight for balancing the losses between the MSE function and the L2-norm regularization.

2.4 Evaluation metrics

We calculate five metrics including the coefficient of determination (also known as R-squared, R^2), square root of R-squared (R), mean squared error (MSE), root mean squared error ($RMSE$), and mean absolute error (MAE) to evaluate the performance of our DeepPLA using the independent testing set, which is defined as follows. (eq.3,4,5,6,7)

$$R^2 = \frac{\sum_{k=1}^{n}(\hat{y}_k - \bar{y})^2}{\sum_{k=1}^{n}(y_k - \bar{y})^2} \qquad \text{eq.3}$$

$$R = \sqrt{R^2} = \sqrt{\frac{\sum_{k=1}^{n}(\hat{y}_k - \bar{y})^2}{\sum_{k=1}^{n}(y_k - \bar{y})^2}} \quad \text{eq.4}$$

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(\hat{y}_k - y_k)^2 \quad \text{eq.5}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(\hat{y}_k - y_k)^2} \quad \text{eq.6}$$

$$MAE = \frac{1}{n}\sum_{k=1}^{n}|\hat{y}_k - y_k| \quad \text{eq.7}$$

where n is the total number of interactions in the dataset; $\hat{y}_k$ and y_k are the predicted value of the DeepPLA model and true pK_i value, respectively; and $\bar{y} = \frac{1}{n}\sum_{k=1}^{n} y_k$ is the mean of true pK_i values in the dataset.

2.5 Experiment setup

Model training was done in Aliyun Cloud Computing. The node CPU used Intel(R) Xeon(R) Platinum 8163 (2.50GHz). An Nvidia Tesla T4 GPU is supplied. The sources code is available on Github. The model is implemented using the PyTorch library (version 1.8.1). The source code of training and evaluating DeepPLA and the requirements are available on GitHub (https://github.com/David-BominWei/DeepPLA).

3 Results

3.1 Distribution of data

After data preprocessing, we finally obtain a total of 249,517 interactions with 54,135 drug molecules and 1,844 protein targets. The median (standard deviation, [minimum, maximum]) lengths of drug molecular SMILES strings and protein sequences are 52 (45.81, [1, 760]) and 445 (456.1, [9, 7096]) (Figure 3A and 3B). The median (standard deviation) value of pK_i is -1.91 (1.60) (Figure 3C).

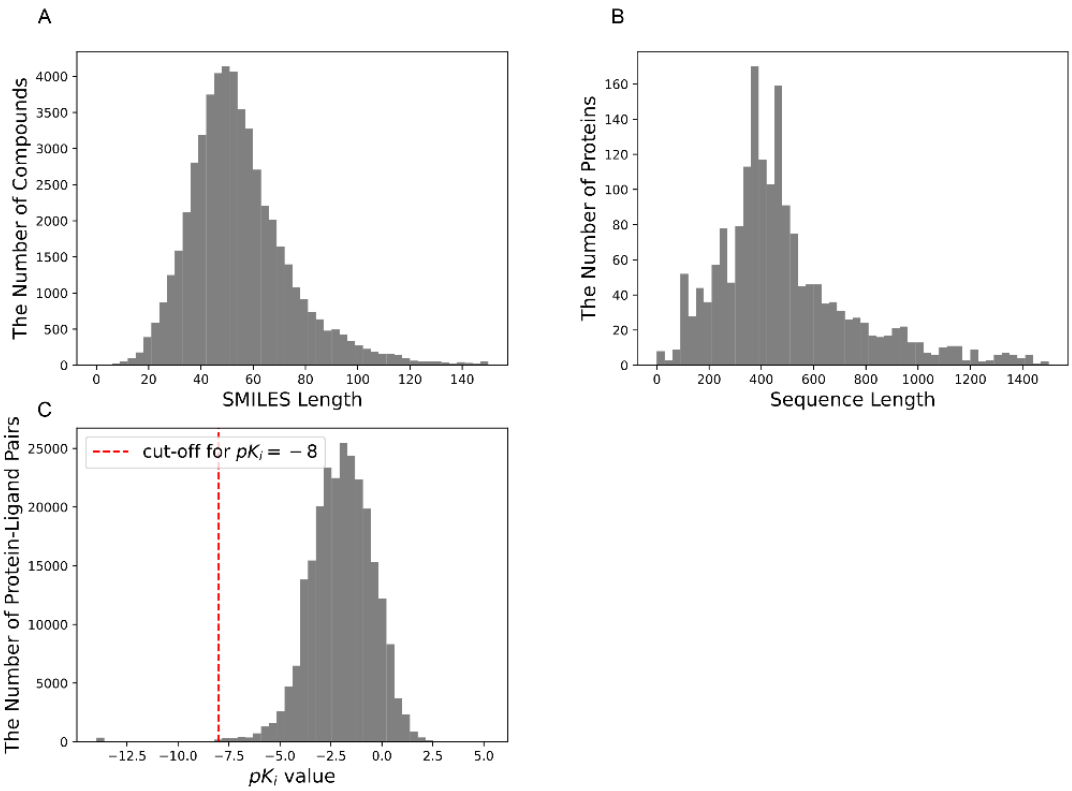


Figure 3 Distribution of data used to develop the DeepPLA model. (A) Distribution of lengths of drug molecular SMILES strings. SMILES strings longer than 150 were not depicted. (B) Distribution of lengths of protein sequences. (C) Distribution of the pK_i value.

3.2 Parameters setting for training DeepPLA

We use Kaiming Initialization to initialize DeepPLA network weights [25]. The Adam optimizer[26] is also employed with default parameters of $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as an optimization algorithm to train our model. Furthermore, we use a batch size of 256 and initialize the learning rate at 0.001 with a decay rate of 0.8 for every 10 epochs. The maximum number of epochs is 1000 epochs. All settings for the parameters implemented in our DeepPLA model are demonstrated in Table 1. It should be noted that we use the default parameter values for the pre-trained Mol2Vec and ProSE, hence we yield vector representations with a fixed length of 300 for the drug molecules and a fixed length of 6,165 for the target proteins. Generally, we manually tune and optimize the hyperparameters of the DeepPLA network, and empirically chose the number of blocks in the ResNet-based module.

Table 1 The parameter settings for the DeepPLA

		Drug compounds	Target proteins

Modules	Parameters	Value	Value
Head Module	Number of kernels	32	32
	Kernel size	7	7
	Stride	2	2
	Padding	3	3
ResNet-based CNN module	Number of kernels	[32,32], [16,16], [16, 16]	[32,32], [16,16], [16, 16]
	Kernel size	[3,3], [3,3], [3,3]	[3,3], [3,3], [3,3]
	Stride	1	1
	Padding	1	1
Max Pooling 1D	Kernel size	2	2
	Stride	2	2
Average Pooling 1D	Kernel size	5	
	Stride	3	
biLSTM module	Input size	538	
	Hidden size	64	
	Number of layers	2	
	Bidirectional	True	
MLP module	Number of neurons	[2048,512,32]	
Common parameter setting for all modules	Dropout	0.3	
	Weight initialization	Kaiming	
	Optimizer	Adam	
	Batch size	256	
	Learning rate (LR)	0.001	

	Weight for L2-norm (α)	0.0001
	LR decay rate	0.8

3.3 Training and evaluation results

As mentioned in the Methods section, we used 72% and 8% of the training set as depicted in Figure 2B to train and validate the DeepPLA during the training phase, respectively. After iteration of 900 epochs, we found that the validation loss stopped to decrease, hence we stop the model training procedure. At epoch 699, we observe that the DeepPLA losses for the training and validation reached 0.27 and 0.63, respectively (Figure 4A). We also calculate the R^2 values during the model training, which achieve 0.88 and 0.71 for the training and validation, respectively (Figure 4B). Therefore, we decide to use the trained DeepPLA resulting from the 699th epoch as the final model.

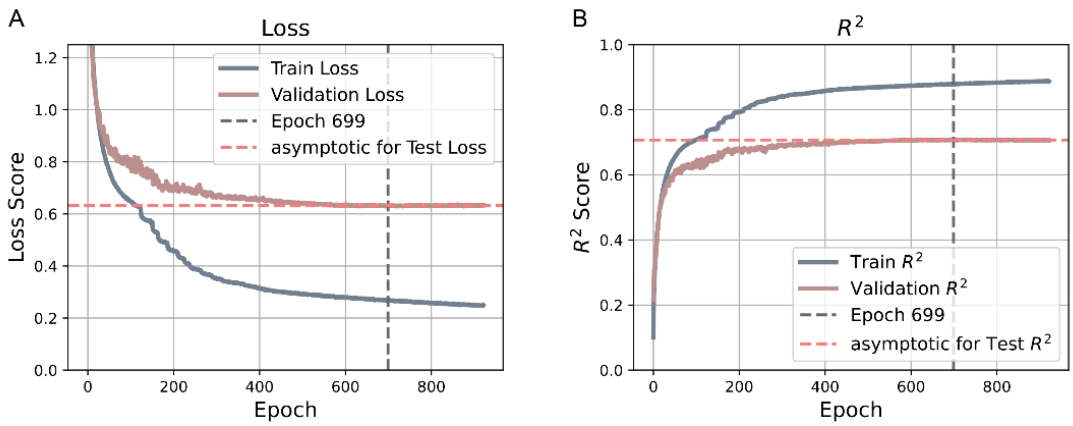
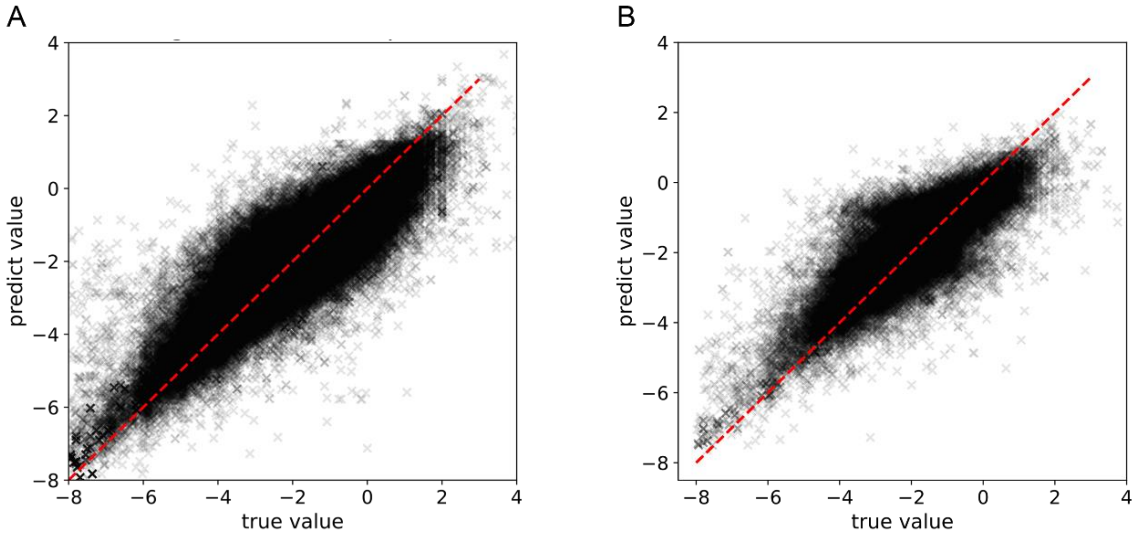


Figure 4 The loss score and R^2 score during the DeepPLA training. (A) Loss scores for training and validation. (B) R^2 scores for training and validation

The final DeepPLA model is retrained on the whole training set (i.e. 80% of the entire data) using optimal hyperparameters and then validated on the independent testing set. We observe that the DeepPLA model obtain high accuracy for protein-ligand binding affinity prediction in terms of resulting values of R, $RMSE$, MAE, R^2 and MSE, which reach 0.89, 0.68, 0.50, 0.79 and 0.46 based upon the training set (Figure 5A, 5C), while reached 0.84, 0.80, 0.60, 0.71 and 0.64 on the basis of the independent testing set (Figure 5B, 5C), respectively. Comparing the results from the training set and the independent testing set suggest that the generalization of the DeepPLA

performed quite well. It also shows that many protein-ligand pairs with pK_i values ranging from -2 to -4 have high efficiency in binding affinity prediction compared to the true values.



C	R	$RMSE$	MAE	R^2	MSE
Training Set	0.89	0.68	0.50	0.79	0.46
Test Set	0.84	0.80	0.60	0.71	0.64

Figure 5 The prediction performance of the final DeepPLA model on the (A) training set and (B) the independent test set. Redline: prediction equals the true value. (C) the table of the value metrics for the training set and the test set.

4 Discussion

In our work, we successfully build a model called DeepPLA based on deep learning to predict DTI with 1D information from protein and drug molecules. We first utilize the pre-trained embedding methods called Mol2Vec and ProSE to encode the raw drug molecular SMILES strings and target protein sequences respectively into dense vector representations. Then, we feed the encoded dense vector representations separately into Head modules and ResNet-based modules to extract features, where these modules are based on 1D CNN. The extracted feature vectors are concatenated and fed into the bi-LSTM network, further followed by the MLP module to finally predict binding affinity

in the form of pK_i value. We used the BindingDB dataset to train and evaluate our DeepPLA model, and the evaluation results can demonstrate that our model has a high performance on the prediction. Unlike methods to pre-define features that are heavily relied on domain knowledge such as SimBoost [10] or to represent sequences simply using sparse encoding approach such as DeepDTA [13], our new method, DeepPLA, applied pre-trained embedding models to encode the raw drug SMILES string and target protein sequences. These embedding models are trained using a huge dataset with consideration of structure information of molecule and target proteins to ensure that they are highly informative and efficient for feature encodings, which lead to dense vector representations. Such a representation is a semantic context embedding. It ensures similar sequences are not far apart in the representation space. It is admired that there exists a variety of embedding methods to encode drug compounds and protein sequences, we picked Mol2Vec and ProSE to be used in the DeepPLA due to past experience.

We use 1D CNN in our DeepPLA model. In the 1D convolutional (Conv1D) operator, the kernel slides along a one-dimensional axis and extracts key features from the local region that was overlapped by the kernel. The 1D CNN can retain the sequential correlation. Therefore, it is widely applied in the information layer of sequence data. We adopt a ResNet-based module in the DeepPLA. Traditional feed-forward CNN may lose useful information as the design grows deeper. Nevertheless, ResNet-based CNN can mitigate this drawback by developing a “shortcut connection” for the network. As a consequence, data inputted into the ResNet-based CNN module can be added with the residual of the network to alleviate the loss of information. The bi-LSTM is employed in the DeepPLA model, which can capture long-term dependencies of the sequence, equally encode input sequence once from beginning to end and once from end to beginning. Compared to the classical LSTM, the biLSTM enables the use of the two hidden states in each LSTM memory block to preserve information from both past and future.

During the computer experiments, we notice that the performance of DeepPLA is not uniform on different proteins: There might exist some common biological features of those proteins, such as the sequences or the spatial structures. Detailed analysis of the shared features of the proteins requires a deeper understanding of the protein-drug interaction and can potentially explain why the

model behaves well on some of the proteins. Such analysis would be useful to improve the model when we generalize the results later.

The DeepPLA model may help in speeding up the COVID-19 drug research. As of today, the pandemic is not showing any sign of slowing down and people are still searching for an effective and safe cure for COVID-19 patients. The current widely-used combination treatment with hydroxychloroquine and azithromycin has not been proven to be satisfactory, and there are some research efforts in using computational, especially deep neural network, techniques for searching the effective repurposed drugs. Our model can be useful in speeding up the drug search and potentially increase the success rate because the training data fed into the model is not limited to the protein structural information.

Even though we have successfully built a model that can predict binding affinity with high accuracy, the model still gets some limitations. A common practice in evaluating a model is to test the model performance on an external dataset, which comes from another database on independent experiments. For the next step, we plan to test the model build with the BindingDB dataset on other accessible databases such as the DAVIS dataset[27] or the KIBA dataset[28]. In addition, we also plan to directly compare the performance of our DeepPLA with other state-of-the-art deep learning methods such as DeepDTA[12] and DeepCDA[29]. From a broader perspective, the study of repurposing drugs should not be limited only to the binding affinities. Researchers should also pay attention to the possibility of potential adverse effects of using the repurposed drug. This can be a result of new interactions between the drug and the proposed disease target, or because the drug is administered to a new group of population. Sometimes the repurposed drug could have interactions with traditional drugs on the new disease, and adverse effects might also arise from such unexpected interactions. Deep learning methods could also be used in studying on these aspects for better safety.

Acknowledgment

The authors thank Dr. Yong Bai from BGI for insights and advice on the methods of embedding of molecular drug and protein, and also on the neural network structures.

Conflict of Interest

Tha authors declare no conflict of interests.

Funding

The authors declare no funding sources for this research work.

References

[1] S. Pushpakom *et al.*, “Drug repurposing: progress, challenges and recommendations,” *Nature Reviews Drug Discovery*, vol. 18, no. 1, Jan. 2019, doi: 10.1038/nrd.2018.168.

[2] O. J. Wouters, M. McKee, and J. Luyten, “Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018,” *JAMA*, vol. 323, no. 9, Mar. 2020, doi: 10.1001/jama.2020.1166.

[3] “The Drug Development Process,” Jan. 04, 2018. https://www.fda.gov/patients/learn-about-drug-and-device-approvals/drug-development-process (accessed Sep. 07, 2021).

[4] R. Mahajan and K. Gupta, “Food and drug administration′s critical path initiative and innovations in drug development paradigm: Challenges, progress, and controversies,” *Journal of Pharmacy and Bioallied Sciences*, vol. 2, no. 4, 2010, doi: 10.4103/0975-7406.72130.

[5] F. Huang *et al.*, “Identification of amitriptyline HCl, flavin adenine dinucleotide, azacitidine and calcitriol as repurposing drugs for influenza A H5N1 virus-induced lung injury,” *PLOS Pathogens*, vol. 16, no. 3, Mar. 2020, doi: 10.1371/journal.ppat.1008341.

[6] P. Sun, J. Guo, R. Winnenburg, and J. Baumbach, “Drug repurposing by integrated literature mining and drug–gene–disease triangulation,” *Drug Discovery Today*, vol. 22, no. 4, Apr. 2017, doi: 10.1016/j.drudis.2016.10.008.

[7] O. J. Wouters, M. McKee, and J. Luyten, “Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018,” *JAMA*, vol. 323, no. 9, Mar. 2020, doi: 10.1001/jama.2020.1166.

[8] P. J. Ballester and J. B. O. Mitchell, “A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking,” *Bioinformatics*, vol. 26, no. 9, May 2010, doi: 10.1093/bioinformatics/btq112.

[9] H. Li, K.-S. Leung, M.-H. Wong, and P. Ballester, “Low-Quality Structural and Interaction Data Improves Binding Affinity Prediction via Random Forest,” *Molecules*, vol. 20, no. 6, Jun. 2015, doi: 10.3390/molecules200610947.

[10] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, and M. Ester, "SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines," *Journal of Cheminformatics*, vol. 9, no. 1, Dec. 2017, doi: 10.1186/s13321-017-0209-z.

[11] R. Özçelik, H. Öztürk, A. Özgür, and E. Ozkirimli, "ChemBoost: A Chemical Language Based Approach for Protein – Ligand Binding Affinity Prediction," *Molecular Informatics*, vol. 40, no. 5, May 2021, doi: 10.1002/minf.202000212.

[12] P.-W. Hu, K. C. C. Chan, and Z.-H. You, "Large-scale prediction of drug-target interactions from deep representations," Jul. 2016. doi: 10.1109/IJCNN.2016.7727339.

[13] H. Öztürk, A. Özgür, and E. Ozkirimli, "DeepDTA: Deep drug-target binding affinity prediction," in *Bioinformatics*, Sep. 2018, vol. 34, no. 17, pp. i821–i829. doi: 10.1093/bioinformatics/bty593.

[14] I. Wallach, M. Dzamba, and A. Heifets, "AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery," Oct. 2015.

[15] S. Wang *et al.*, "SE-OnionNet: A Convolution Neural Network for Protein–Ligand Binding Affinity Prediction," *Frontiers in Genetics*, vol. 11, Feb. 2021, doi: 10.3389/fgene.2020.607824.

[16] T. B. Kimber, Y. Chen, and A. Volkamer, "Deep Learning in Virtual Screening: Recent Applications and Developments," *International Journal of Molecular Sciences*, vol. 22, no. 9, Apr. 2021, doi: 10.3390/ijms22094435.

[17] Z. Liu *et al.*, "PDB-wide collection of binding data: current status of the PDBbind database," *Bioinformatics*, vol. 31, no. 3, Feb. 2015, doi: 10.1093/bioinformatics/btu626.

[18] The AlphaFold team, "AlphaFold: a solution to a 50-year-old grand challenge in biology," Nov. 30, 2020. https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology (accessed Sep. 07, 2021).

[19] A. Bayat, "Science, medicine, and the future: Bioinformatics," *BMJ*, vol. 324, no. 7344, Apr. 2002, doi: 10.1136/bmj.324.7344.1018.

[20] S. Jaeger, S. Fulle, and S. Turk, "Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition."

[21] T. Bepler and B. Berger, "Learning the protein language: Evolution, structure, and function," *Cell Systems*, vol. 12, no. 6, Jun. 2021, doi: 10.1016/j.cels.2021.05.017.

[22] T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson, “BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities,” *Nucleic Acids Research*, vol. 35, no. Database, Jan. 2007, doi: 10.1093/nar/gkl999.

[23] D. Rogers and M. Hahn, “Extended-Connectivity Fingerprints,” *Journal of Chemical Information and Modeling*, vol. 50, no. 5, May 2010, doi: 10.1021/ci100050t.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” Feb. 2015.

[26] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Dec. 2014.

[27] M. I. Davis *et al.*, “Comprehensive analysis of kinase inhibitor selectivity,” *Nature Biotechnology*, vol. 29, no. 11, Nov. 2011, doi: 10.1038/nbt.1990.

[28] J. Tang *et al.*, “Making Sense of Large-Scale Kinase Inhibitor Bioactivity Data Sets: A Comparative and Integrative Analysis,” *Journal of Chemical Information and Modeling*, vol. 54, no. 3, Mar. 2014, doi: 10.1021/ci400709d.

[29] K. Abbasi, P. Razzaghi, A. Poso, M. Amanlou, J. B. Ghasemi, and A. Masoudi-Nejad, “DeepCDA: deep cross-domain compound–protein affinity prediction through LSTM and convolutional neural networks,” *Bioinformatics*, vol. 36, no. 17, Nov. 2020, doi: 10.1093/bioinformatics/btaa544.