DeepLPI: a novel deep learning-based model for protein-ligand interaction prediction for drug repurposing

Bomin Wei[1]

1. Princeton International School of Mathematics and Science, NJ, USA

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 DeepLPI, a novel deep learning-based model that mainly consists of ResNet-based 1-dimensional convolutional neural network (1D CNN) and bi-directional long short term memory network (biLSTM), to establish an end-to-end framework for protein-ligand interaction prediction. We first apply pre-trained embedding methods to encode the raw drug molecular sequences in the form of SMILES strings and target protein sequences in the FASTA format into dense vector representations. The embedded representations of drug melocular and target proteins go through two ResNet-based 1D CNN modules to derive features, respectively. 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 protein-ligand interaction. We downloaded the well-known BindingDB dataset for training and internal independent testing our DeepLPI model. We further applied it on the Davis dataset and a COVID-19 dataset for externally evaluating the prediction ability of DeepLPI. To benchmark our model, we compared our DeepLPI with the state-of-the-art DeepCDA method towards protein-ligand interaction prediction. We observed that our DeepLPI reaches AUROC of 0.794, Sensitivity of 0.724, specificity of 0.749, PPV of 0.661, and NPV of 0.800 on the internal independent testing set. For the external evaluation on the COVID-19 dataset, we archived AUROC of 0.610, sensitivity of 0.538, specificity of 0.576, PPV of 0.110, and NPV of 0.928, respectively. We found that our DeepLPI outperformed DeepCDA in term of these assessment metrics, suggesting the high accuracy of the DeepLPI towards protein-ligand interaction prediction. The high prediction

performance of DeepLPI on the different protein-ligand interaction datasets of BindingDB, Davis and COVID-19 displayed its high capability in generalization, demonstrating that the DeepLPI has the potential to pinpoint new drug-target interactions and to find better destinations for proven drugs.

Keywords: drug repurposing, deep learning, protein-ligand binding interaction prediction

Table of Contents

DeepLPI: a novel deep learning-based model for protein-ligand interaction prediction for drug repurposing

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) or protein-ligand interactions (PLIs) 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 PLIs 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}). Smaller values in the constants means a stronger binding affinity of a protein-ligand pair. Experimental methods dentifying the protein-ligand interactions involve complicated quantum chemical calculation of molecular/biological structures. Thereby, those conventional methods did not enable to make use of the large-scale existing protein-ligand interaction databases for fast and efficiently screening and discovering candidate drugs for new disease.

Over the last decade, a variety of machine learning-based models have been developed to identify PLIs from millions of ligands and proteins such as random forest (RF) based algorithm[8][9], SimBoost[10], and ChemBoost[11]. These methods 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 a 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. For instance, MFDR[12] obtained features with auto-encoder from chemical structures and protein sequences and then employs SVM models to predict PLI as a binary classification problem. DeepDTA[13] applied two different CNNs modules separately to represent sequences of compounds and proteins as information modules. The resulting features then entered 3 fully connected layers to predict protein-ligand binding affinity. Yet, the use of a simple label encoding method in DeepDTA and MFDR 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] utilized 3D structures of proteins and drug molecules to predict the drug-target binding affinity[16]. They may partly diminish the problem of losing information. However, those models' practicability and accuracy 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 DeepLPI, an innovative deep learning-based model to predict protein-ligand interaction using the simple formats of raw protein 1D sequences and 1D ligands (ie., drug molecular) SMILES strings as inputs, rather than manual-generated features or complex 3D protein structures. 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 drug SMILES strings and protein sequences as numeric vectors. These embedded numeric vectors are then fed into two blocks, each of them consisting of two modules termed Head convolutional neural network (CNN) module and ResNet-based CNN module, to encode proteins and drug sequences, respectively. The encoded representations are concatenated into a vector and further fed into a bi-directional long short-term memory (bi-LSTM) layer, followed by three fully connected layers. With a sigmoid function, the output of DeepLPI is transformed into a continuous value, representing the probability of binding/interaction of the input pair of protein and ligand. We download the BindingDB dataset[22] to train the DeepLPI model and internally independently evaluate its performance towards PLI prediction. We further applied the model on the Davis[27] dataset and a COVID-19 3CL Protease[30, 31] dataset for externally assessing the prediction ability of DeepLPI. To benchmark our model, we compared our DeepLPI with the start-of-the-art DeepCDA method towards protein-ligand interaction prediction on each of Binding DB, Davis and the a COVID-19 3CL Protease dataset. The prediction performance is quantativly represented in terms of area under the receiver operating characteristic curve (AUROC), sensitivity, specificity, positive (PPV), predictive value , and negative predictive value (NPV). The high performance of our DeepLPI towards protein-ligand interaction prediction suggests that our model has the potential to accurately identify protein-ligand interaction and hence, promote the new drug development.

My research report is structured as follows. Section 2 describes the dataset and the DeepLPI model; section 3 illustrates the results of the protein-ligand interaction prediction using the DeepLPI; 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] and Davis[27] datasets to train and evaluate (both internally and externally) our DeepLPI model. We also use the COVID-19 3C-like Protease dataset from Diamond Light Source [30, 31] for further assessment. All datasets are publicly accessible. The BindingDB is a continually updating database 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 1**): (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_d value and it means that other measurements in the form of $IC50$ or K_i 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_d values and removing duplicated data entries. For example, we notice that some data used ">" and "<" in the labeled values to indicate ranges, so directly exclude them for the subsequent analysis. Additionally, there are some zeros in the values which should not appear based on the definition of binding affinity measurement of K_d. Thus, we treat them as invalid values and simply removed them; (5) As a binary classification problem in this study, we label 1 representing a pair of protein and ligand being binding/interaction if their corresponding K_d value less than 100 nM or 0 otherwise according to the work of DeepCDA [29]. As a result, a total of $36{,}111$ interactions

with 17,773 drug molecules and 1,915 protein targets are finally used in developing our model.

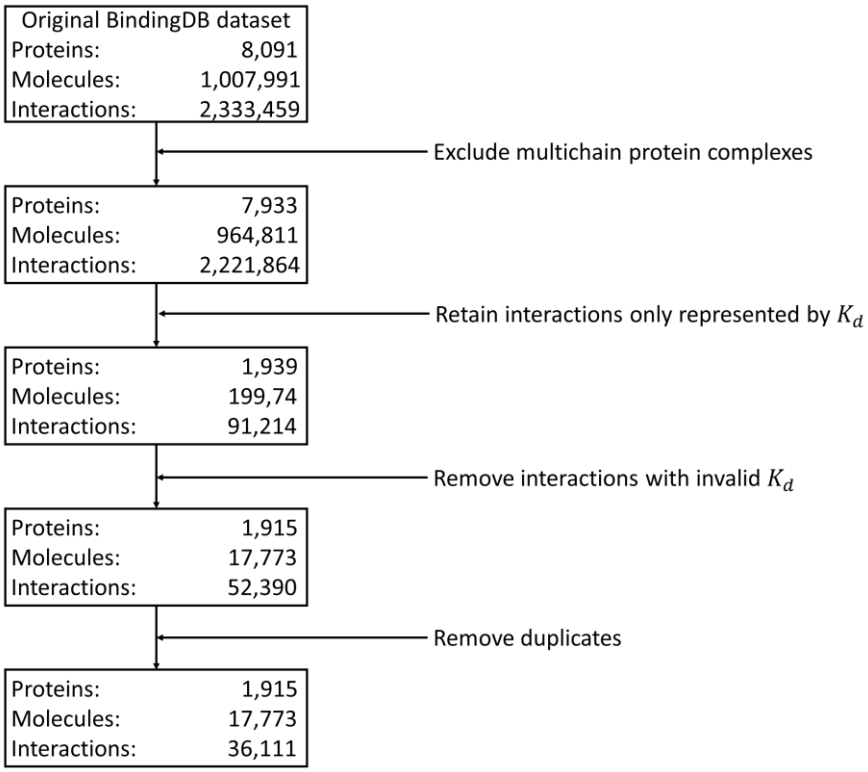

Figure 1 Preprocessing of BindingDB dataset. Data exclusion criteria to compile BindingDB dataset.

The reason for choosing K_d rather than other binding measurements is for enabling our trained model to externally evaluate on Davis testset, which contains interactions of 442 unique proteins and 68 unique compounds. The Davis dataset only reports K_d values of the kinase protein family and the relevant inhibitors. We used the same protocol to obtain the class label as we did above. The Davis dataset was referenced from the Davis work [27] and downloaded from the URL therein. All binding affinity values are only measured in K_d. The dataset contained duplicated data entries where the drug-protein pairs are the same but the binding affinity values are different, potentially due to the experiment conditions. We keep only one entry in each group of duplicates. This doesn't affect the balance of the dataset because according to our binary threshold, all data entries in the same duplicate group in fact have the same binary label. After the treatment, there are $24{,}548$ interaction data entries. We split them into training, validation, and testing sets according to the same method described above.

In an attempt to find effective drugs for SARS-CoV-2, we applied our model on a COVID-19 dataset where 879 small molecule drugs were tested on the SARS-COV-2 3C-like protease. The experiment measured EC50 results. For classification, we label 1 to indicate drug-protease active if EC50 is less than 30 nM [31] or 0 representing inactivity. The data is retrieved from a large XChem crystallographic fragment screen against SARS-CoV-2 main protease at high resolution from MIT AiCures. [30] Among those data, 78 are considered to be active according to the threshold.

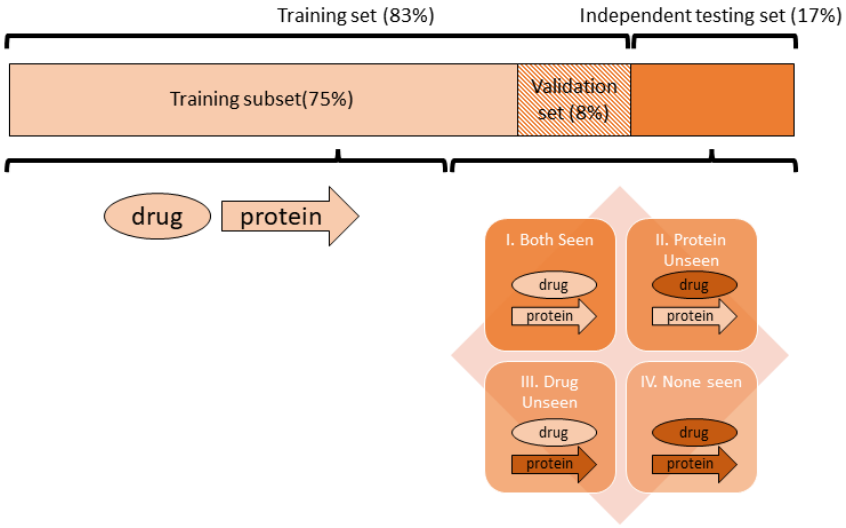

Figure 2 Split the whole dataset into a training set (83% of whole interactions) and an independent testing set (17%) for training and evaluating the model, respectively. The training set is further divided into the training subset (75%) and the validation set (8%). The independent testset was further splitted into for parts. Part I: the drug or protein information is separately included in the training set but not their pairs. Part II: the drug information included in the training set but not the protein. Part III: the protein information is included in the training set but not the drug information. Part IV: neither of drug or protein information is included in the training set.

We then randomly select 83% of the pre-processed BindingDB dataset as the training set and the remaining 17% as the internal independent testing set to train and evaluate our DeepLPI model (**Figure 2**). 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., 75% of all data). The internal independent test set was divided into four parts. Part I: the drug or protein information is separately included in the training set

but not their pairs. Part II: the drug information included in the training set but not the protein. Part III: the protein information is included in the training set but not the drug information. Part IV: neither of drug or protein information is included in the training set.

2.2 Model Design

2.2.1 Overview of DeepLPI model

The proposed DeepLPI consists of eight modules (**Figure 3**), including two embedding modules, two head modules, two ResNet-based CNN modules, one bi-directional LSTM (biLSTM) module, and one multilayer perceptron module (MLP). DeepLPI employs raw molecular SMILES strings and protein sequences as inputs, which are first represented as numeric vectors by two embedding modules, respectively. The embedded vectors for the drug 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 and the final output is passed through a sigmoid function for binary classification to predict binding/non-binding labels.

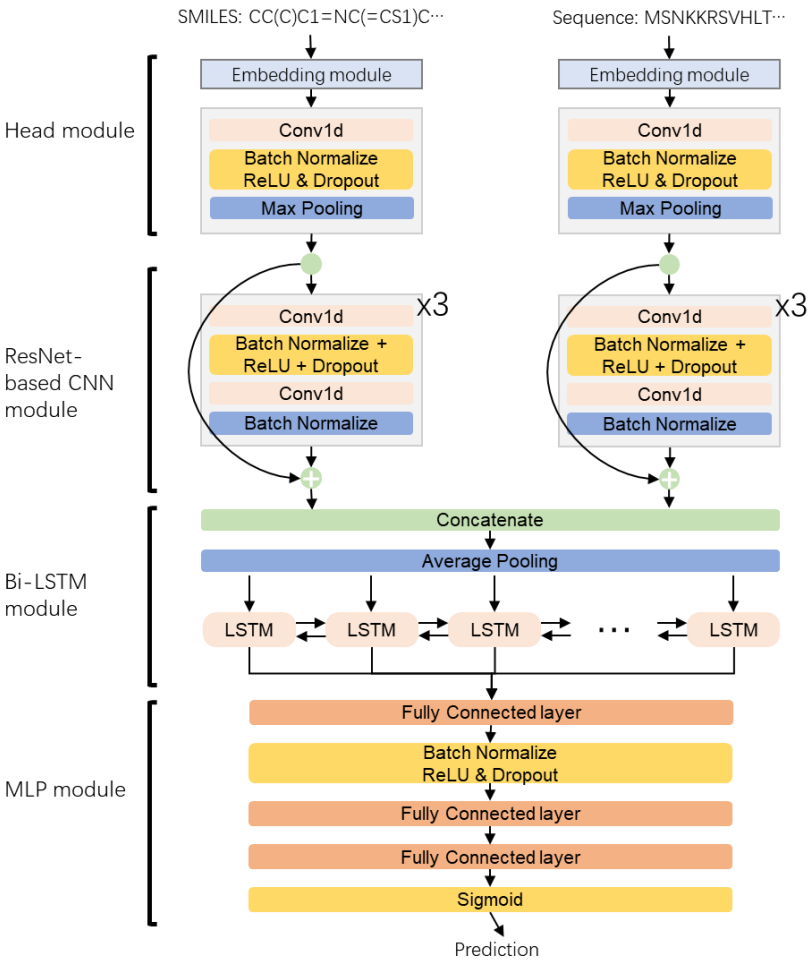

Figure 3 The overview of the DeepLPI model flow.

2.2.2 Embedding module

To utilize the raw drug molecular SMILES string and protein sequence as inputs to the DeepLPI 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. The basic workflow and embedding principle is illustrated in **Figure 4**.

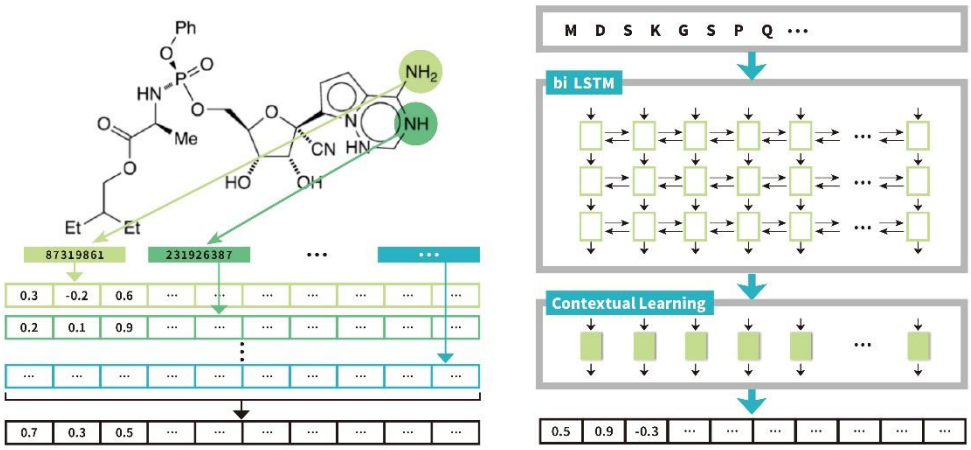

Figure 4 Embedding principle of (A) Mol2Vec for drug molecule from reference [20] and (B) ProSE for protein from reference [21]**.**

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 output is passed through a sigmoid function for binary classification to predict binding/non-binding labels.

2.3 Loss function

We treat the prediction as a classification task, predicting whether the drug and protein will bind or not, and therefore we choose the Binary Cross Entropy loss function, implement in PyTorch as BCELoss. The L2-norm regularization is added into the loss function through the

optimizer. For the n pairs of molecular SMILES strings and protein sequences the loss function of the DeepLPI model was given by:

$$\text{Loss} = \underbrace{-\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y}_i) + (1-y_i) \cdot \log(1-\hat{y}_i)]}_{\text{BCE loss}} + \underbrace{\alpha\|W\|_2^2}_{\text{L2-norm regularization}} \quad (1)$$

where $y_i \in \{0,1\}$ is class label representing whether or not binding interaction of a input pair of protein and ligand sequences i. $\hat{y}_i$ is the probability of interaction prediction for the input pair i by our model, $\hat{y}_i = sigmoid(x) = \frac{1}{1+e^{-x}}$, x is the output of the MLP module of our model. W is the trainable weight matrix in our model. α is decay rate and we set it as 0.8 in this study.

2.4 Evaluation metrics

We calculate five metrics including area under the receiver operating characteristic curve (AUROC), sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV) to evaluate the performance of our model. The latter four metrics rely on calculation of a confusion matrix first. The definitions of latter four metrics are as follows:

- $$Sensitivity = TP / (TP + FN) \quad (2)$$
- $$Specificity = TN / (TN + FP) \quad (3)$$
- $$PPV = TP / (TP + FP) \quad (4)$$
- $$NPV = TN / (FN + TN) \quad (5)$$

where TN, FN, TP, FP refer to the number of true negative, false negative, true positive and false positive.

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 DeepLPI and the requirements are available on GitHub (https://github.com/David-BominWei/DeepLPI).

3 Results and Discussion

3.1 Distribution of data

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 5A and 5B**).

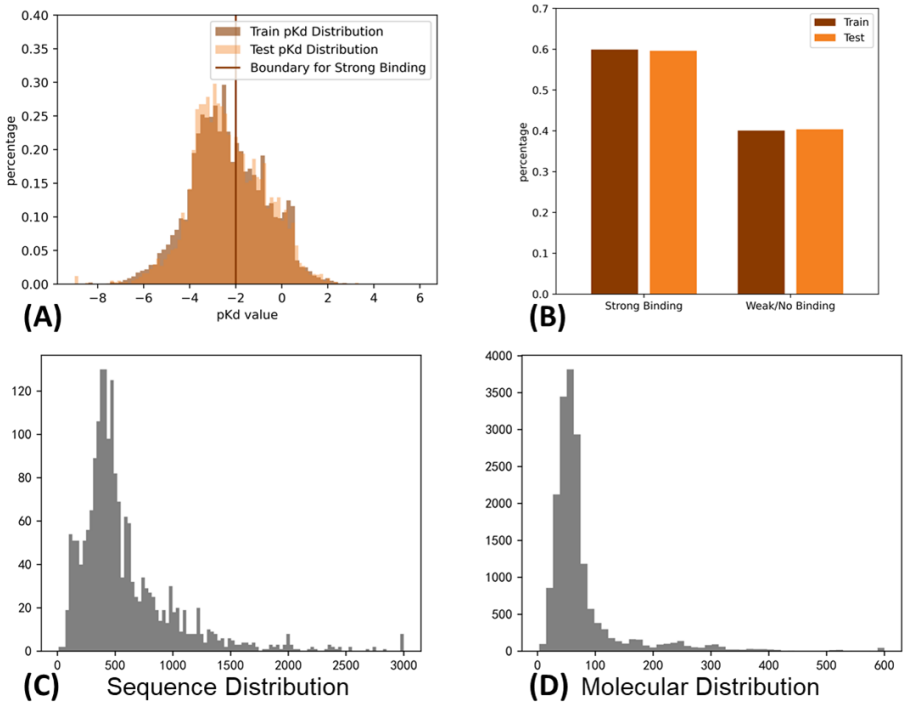

Figure 5 Distribution of BindingDB data used to develop the DeepLPI model. (A) Distribution of the pK_d values and the threshold for determining binding/non-binding. (B) distribution interaction in binary classes (C) Distribution of lengths of drug molecular SMILES strings. (D) Distribution of lengths of protein sequences.

3.2 Parameters setting for training DeepLPI

We use Kaiming Initialization to initialize DeepLPI 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 DeepLPI model are demonstrated in **Table 1**. It should be noted that we use the default parameter values

for the pre-trained Mol2Vec and ProSE, and we yield vector representations with a fixed length of 300 for the drug molecules, and two lengthes of 100 and 6,165 for the target proteins. Generally, we manually tune and optimize the hyperparameters of the DeepLPI network, and empirically chose the number of blocks in the ResNet-based module.

Table 1 The parameter settings for the DeepLPI

		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

3.3.1 BindingDB Dataset

After iteration of 90 epochs, we found that the validation loss stopped to decrease, hence we stop the model training procedure. Training beyond this point would lead to apparent overfitting marked by increase of the validation loss. We also calculate the AUROC metric values during the model training, which achieve 0.97 and 0.89 for the training and validation, respectively (**Figure 6**).

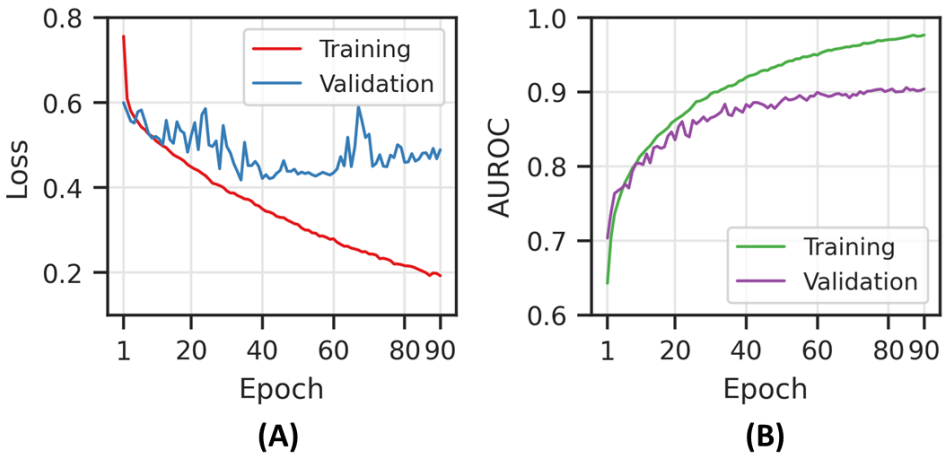

Figure 6 The loss and AUROC score during the DeepLPI training on the BindingDB dataset. (A) Loss scores for training and validation. (B) AUROC scores for training and validation

We applied the trained model on the independent testset constructed from BindingDB dataset. We used Youden's J statistic to determine the optimal classification threshold instead of using default value of 0.5 (**Figure 7A)**, which was used for later calculation of confusion matrix metrics (**Figure 7B-F**). The AUROC measured for the model performance on the independent testset is 0.794.

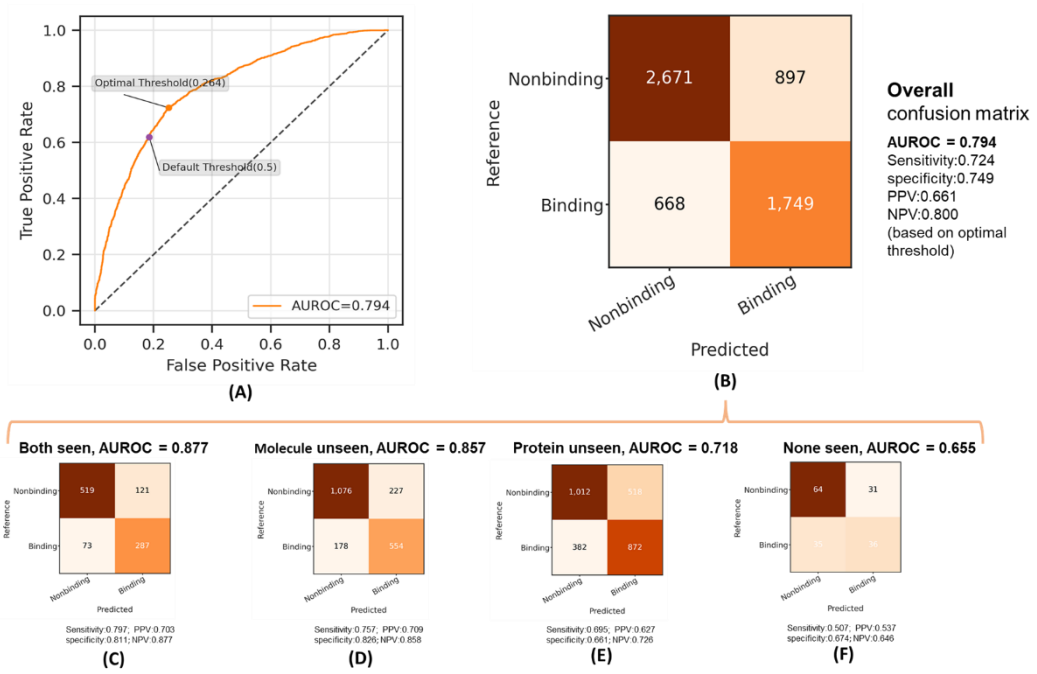

Figure 7 The prediction performance of the final DeepLPI model. (A) The ROC curve and determied optimal threshold. (B) Confusion matrix based on the optimal threshold. (C) – (F) Confusion matrix and performance metrics on the four parts of the testset.

We observe that the DeepLPI model obtain high accuracy with AUROC 0.877 on the "Both seen" testset where the drug molecule or the protein sequence but not the drug-protein pair information are included in the training set. When the training set has only partial knowledge of the testset, in testsets "Molecule unseen" where none of the drug molecules are included and in the "Protein unseen" testset where none of the protein sequences are included, the accuracy decreases to AUROC 0.857 and AUROC 0.718. When the training set has no knowledge of the testset, in testset "None seen", the accuracy reduces to 0.655.

Table 2. Comparing Performance of DeepLPI model and DeepCDA model on internal independent testing set from the BindingDB data.

BindingDB	**AUROC**	**Sensitivity**	**Specificity**	**PPV**	**NPV**	**Remark**
Our 6165	**0.790**	**0.684**	0.773	**0.671**	**0.783**	
Our 100	0.751	0.541	0.818	0.668	0.725	
DeepCDA	0.448	0.000	**1.0**	Nan	0.596	All nonbinding

In **Table 2**, we list the performance metrics of two versions of our models and the state-of-the-art DeepCDA model on internal independent testing set from the Binding DB data. DeepLPI-6165 and DeepLPI-100 use a 6165-dimensional and 100-dimensional vector for protein embedding, respectively. The longer vector embedding is slightly better than the shorter embedding method in terms of AUROC value metric. Both models are significantly better than the DeepCDA model. On the AUROC metric, DeepLPI performance is 76% better than DeepCDA, but in fact, DeepCDA is even worse than the number suggested. Its predictions on the independent test set yield all nonbinding results.

3.3.2 Evaluation on Davis dataset

After iteration of 40 epochs, we found that the validation loss stopped to decrease, hence we stop the model training procedure. Training beyond this point would lead to apparent overfitting marked by increase of the validation loss. We also calculate the AUROC metric values during the model training, which achieve 0.98 and 0.91 for the training and validation, respectively (**Figure 8**).

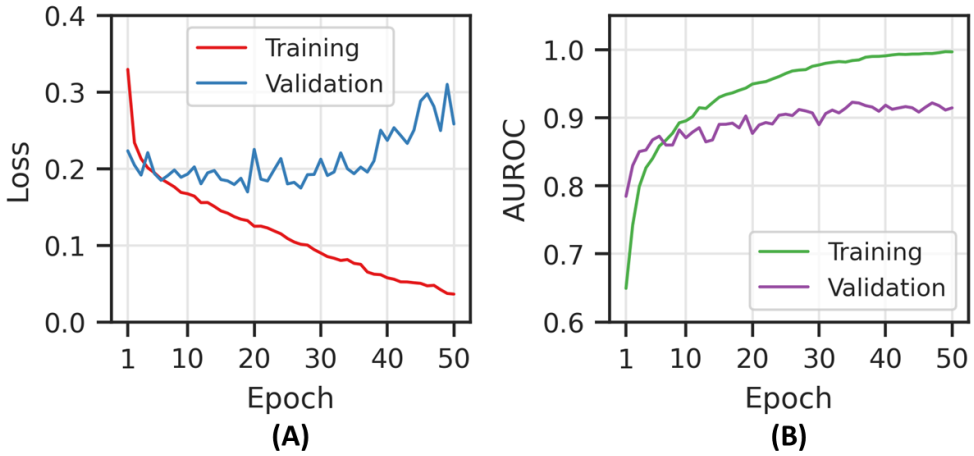

Figure 8 The loss and AUROC score during the DeepLPI training on Davis dataset (A) Loss scores for training and validation. (B) AUROC scores for training and validation

We applied the trained model on the independent testset constructed from Davis dataset. We used Youden's J statistic to determine the optimal classification threshold instead of using default value of 0.5 (**Figure 9A)**, which was used for later calculation of confusion matrix metrics (**Figure 9B-F**). The AUROC measured for the model performance on the independent testset is 0.791.

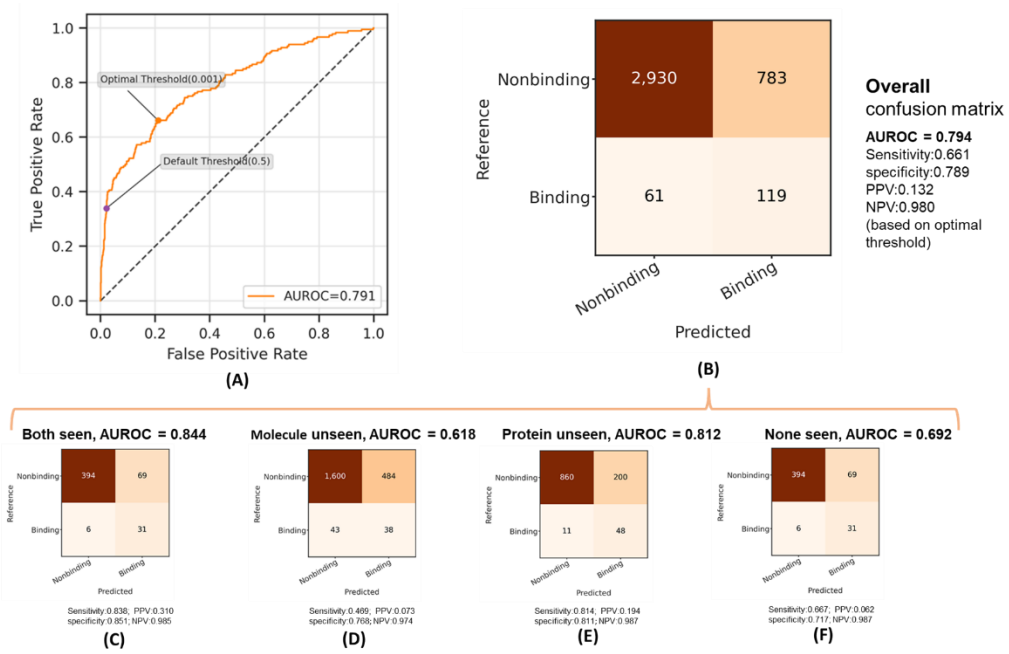

Figure 9 The prediction performance of the final DeepLPI model. (A) The ROC curve and determied optimal threshold. (B) Confusion matrix based on the optimal threshold. (C) – (F) Confusion matrix and performance metrics on the four parts of the testset.

We observe that the DeepLPI model obtain high accuracy with AUROC 0.844 on the "Both seen" testset where the drug molecule or the protein sequence but not the drug-protein pair information is included in the training set. When the training set has only partial knowledge of the testset, in testsets "Protein unseen" testset where none of the protein sequences are included but all molecules are included, the accuracy decreases to AUROC 0.812. When the training set has no knowledge of the testset, in testset "None seen", the accuracy reduces to 0.692. The testset "Molecule unseen" where none of the drug molecules are included does not follow the above trend. The training set has partial knowledge about the proteins in this testset, but its accuracy AUROC is the lowest at 0.692. The reason could be the specific drug molecules in the random choice and the accuracy might increase if more randomly selected testsets are tested and the calculate the statistics of the accuracy.

Table 3. Comparing Performance of DeepLPI model and DeepCDA model on Davis dataset

	AUROC	**Sensitivity**	**Specificity**	**PPV**	**NPV**	**Remark**
Our 6165	**0.791**	**0.661**	0.789	0.132	**0.980**	
Our 100	0.731	0.534	0.794	0.480	0.826	
DeepCDA	0.741	0.511	**0.813**	**0.495**	0.823	

On Davis dataset, our DeepLPI-6165 model is 6.7% better than DeepCDA model, scored a 0.05 increase in AUROC metric value (**Tables 3**).

3.3.4 Evaluation on COVID-19

The as-trained models on BindingDB dataset are directly applied without any fine-tuning to predict the COVID-19 dataset for transferability study. DeepLPI outperformes DeepCDA by 50% better AUROC metric value, and DeepCDA can not produce meaningful predictions on the COVID-19 external dataset since it produces on nonbinding predictions.

Table 4. Comparison of DeepLPI and DeepCDA on transferring BindingDB trained model to COVID-19.

	AUROC	Sensitivity	Specificity	PPV	NPV	Remark
Our 6165	**0.610**	0.538	0.576	**0.110**	**0.928**	
Our 100	0.473	**0.692**	0.332	0.092	0.912	
DeepCDA	0.400	0.000	**1.000**	nan	0.911	All nonbinding

3.4 Discussion and Future Work

In our work, we successfully build a model called DeepLPI based on deep learning to predict DTI in classification tasks 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 and eventually through a Sigmoid function to finally predict binary binding or non-binding based on K_d affinity labeled data. We used the BindingDB dataset to train and evaluate our DeepLPI 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, DeepLPI, 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 DeepLPI due to past experience. We use 1D CNN in our DeepLPI 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 DeepLPI. 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 DeepLPI 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 DeepLPI 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 DeepLPI 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/non-binding interaction with high accuracy, the model still gets some limitations. There is still room for improvement regarding the prediction accuracy, especially when the model is applied on external datasets. 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.

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.

[30] Diamond Light Source, "Main protease structure and XChem fragment screen," *Online data source.* https://www.diamond.ac.uk/covid-19/for-scientists/Main-protease-structure-and-XChem.html.

[31] Huang, K. et al, , "Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development", *arXiv e-prints*, 2021..