PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences

Ingoo Lee[☯], **Jongsoo Keum**[☯], **Hojung Nam***

School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Buk-ku, Gwangju, Republic of Korea

☯ These authors contributed equally to this work.
* hjnam@gist.ac.kr

Citation: Lee I, Keum J, Nam H (2019) DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. PLoS Comput Biol 15(6): e1007129. https://doi.org/10.1371/journal.pcbi.1007129

Editor: James M. Briggs, University of Houston, UNITED STATES

Received: October 1, 2018

Accepted: May 24, 2019

Published: June 14, 2019

Data Availability Statement: All code we used in manuscript are available from GitHub repository (https://github.com/GIST-CSBL/DeepConv-DTI)

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (NRF-2018M3A9A7053266), the Bio-Synergy Research Project (NRF-2017M3A9C4092978) of the Ministry of Science and ICT through the National Research Foundation. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Identification of drug-target interactions (DTIs) plays a key role in drug discovery. The high cost and labor-intensive nature of *in vitro* and *in vivo* experiments have highlighted the importance of *in silico*-based DTI prediction approaches. In several computational models, conventional protein descriptors have been shown to not be sufficiently informative to predict accurate DTIs. Thus, in this study, we propose a deep learning based DTI prediction model capturing local residue patterns of proteins participating in DTIs. When we employ a convolutional neural network (CNN) on raw protein sequences, we perform convolution on various lengths of amino acids subsequences to capture local residue patterns of generalized protein classes. We train our model with large-scale DTI information and demonstrate the performance of the proposed model using an independent dataset that is not seen during the training phase. As a result, our model performs better than previous protein descriptor-based models. Also, our model performs better than the recently developed deep learning models for massive prediction of DTIs. By examining pooled convolution results, we confirmed that our model can detect binding sites of proteins for DTIs. In conclusion, our prediction model for detecting local residue patterns of target proteins successfully enriches the protein features of a raw protein sequence, yielding better prediction results than previous approaches. Our code is available at https://github.com/GIST-CSBL/DeepConv-DTI.

Author summary

Drugs work by interacting with target proteins to activate or inhibit a target's biological process. Therefore, identification of DTIs is a crucial step in drug discovery. However, identifying drug candidates via biological assays is very time and cost consuming, which introduces the need for a computational prediction approach for the identification of DTIs. In this work, we constructed a novel DTI prediction model to extract local residue patterns of target protein sequences using a CNN-based deep learning approach. As a result, the detected local features of protein sequences perform better than other protein descriptors for DTI prediction and previous models for predicting PubChem independent

Competing interests: No authors have competing interests.

test datasets. That is, our approach of capturing local residue patterns with CNN successfully enriches protein features from a raw sequence.

Introduction

The identification of drug-target interactions (DTIs) plays a key role in the early stage of drug discovery. Thus, drug developers screen for compounds that interact with specified targets with biological activities of interest. However, the identification of DTIs in large-scale chemical or biological experiments usually takes 2~3 years of experiments, with high associated costs [1]. Therefore, with the accumulation of drugs, targets, and interaction data, various computational methods have been developed for the prediction of possible DTIs to aid in drug discovery.

Among computational approaches, docking methods, which simulate the binding of a small molecule and a protein using 3D structure, were initially studied. Docking methods recruit various scoring functions and mode definitions to minimize free energy for binding. Docking methods have advanced by themselves, and recently, the Docking Approach using Ray-Casting (DARC) model identified 21 compounds by using an elaborate binding pocket topography mapping methodology, and the results were reproduced in a biochemical assay [2]. In addition, studies have examined several similarity-based methods in which it was assumed that drugs bind to proteins similar to known targets and vice versa. One of the early methods is that of *Yamanashi et al*., which utilized a kernel regression method to use the information on known drug interactions as the input to identify new DTIs, combining a chemical space and genomic spaces into a pharmacological space [3]. To overcome the requirement of the bipartite model for massive computational power, *Beakley et al*. developed the bipartite local model, which trains the interaction model locally but not globally. In addition to substantially reducing the computational complexity, this model exhibited higher performance than the previous model [4]. As another approach to DTI prediction models, matrix factorization methods have been recruited to predict DTIs, which approximate multiplying two latent matrices representing the compound and target protein to an interaction matrix and similarity score matrix [5, 6]. In this work, regularized matrix factorization methods successfully learn the manifold lying under DTIs, giving the highest performance among previous DTI prediction methods. However, similarity-based methods are not commonly used at present to predict DTIs, as researchers have found that similarity-based methods work well for DTIs within specific protein classes but not for other classes [7]. In addition, some proteins do not show strong sequence similarity with proteins sharing an identical interacting compound [8].

Thus, feature-based models that predict DTI features of drugs and targets have been studied [9–11]. For feature-based DTI prediction models, a fingerprint is the most commonly used descriptor of the substructure of a drug [12]. With a drug fingerprint, a drug is transformed into a binary vector whose index value represents the existence of the substructure of the drug. For proteins, composition, transition, and distribution (CTD) descriptors are conventionally used as computational representations [13]. Unfortunately, feature-based models that use protein descriptors and drug fingerprints showed worse performance than previous conventional quantitative structure-activity relationship (QSAR) models [9]. To improve the performance of feature-based models, many approaches have been developed, such as the use of interactome networks [14, 15] and minwise hashing [16]. Although various protein and chemical descriptors have been introduced, feature-based models do not show sufficiently good predictive performance [17]. For conventional machine learning models, features must be built to be

readable by modeling from original raw forms, such as simplified molecular-input line entry system (SMILES) and amino acid sequences. During transformation, rich information, such as local residue patterns or relationships, is lost. In addition, it is hard to recover lost information using traditional machine learning models.

In recent years, many deep learning approaches have recently been developed and recruited for omics data processing [18] as well as drug discovery [19], and these approaches seem to be able to overcome limitations. For example, DeepDTI built by *Wen et al.* used the deep belief network (DBN) [20], with features such as the composition of amino acids, dipeptides, and tripeptides for proteins and extended-connectivity fingerprint (ECFP) [21] for drugs [7]. The authors also discussed how deep-learning-based latent representations, which are nonlinear combinations of original features, can overcome the limitations of traditional descriptors by showing the performance in each layer. In another study by *Peng et al.* [22], MFDR employed sparse Auto-Encoder (SAE) to abstract original features into a latent representation with a small dimension. With latent representation, they trained a support vector machine (SVM), which performed better than previous methods, including feature- and similarity-based methods. In another study called DL-CPI by *Tian et al.* [23], domain binary vectors were employed to represent the existence of domains used to describe proteins.

One way to reduce the loss of feature information is to process raw sequences and SMILES as their forms. In a paper by *Öztürk et al.*, DeepDTA was used to represent raw sequences and SMILES as one-hot vectors or labels [24]. With a convolutional neural network (CNN), the authors extracted local residue patterns to predict the binding affinity between drugs and targets. As a result, their model exhibited better performance on a kinase family bioassay dataset [25, 26] than the previous model, kronRLS [27] and SimBoost [28]. Because their model is optimized by densely constructed kinase affinities, DeepDTA is appropriate to predict kinase affinities not to predict new DTIs with various protein classes. Furthermore, they evaluated their performances on the identical dataset, rather than on independent dataset from new sources or databases.

To overcome the aforementioned problems, here, we introduce a deep learning model that predicts massive-scale DTIs using raw protein sequences not only for various target protein classes but also for diverse protein lengths. The overall pipeline of our model is depicted in Fig 1. First, for the training model, we collected large-scale DTIs integrated from various DTI databases, such as DrugBank [29], International Union of Basic and Clinical Pharmacology (IUPHAR) [30], and Kyoto Encyclopedia of Genes and Genomes (KEGG) [31]. Second, in model construction, we adopted convolution filters on the entire sequence of a protein to capture local residue patterns, which are the main protein residues participating in DTIs. By pooling the maximum CNN results of sequences, we can determine how given protein sequences match local residue patterns participating in DTIs. Using these data as input variables for higher layers, our model constructs, abstracts and organizes protein features. After new protein features are generated, our model concatenates protein features with drug features, which come from fingerprints in the fully connected layer and predict the probability of DTIs via higher fully connected layers. Third, we optimized the model with DTIs from MATADOR [32] and negative interactions predicted from *Liu et al.* [33]. Finally, with the optimized model, we predicted DTIs from bioassays such as PubChem BioAssays [34] and KinaseSARfari [35] to estimate the performance of our model. As a result, our model exhibits better performance than previous models.

Results

Performances of the validation dataset and selected hyperparameters

As a normal step of hyperparameter setting, we first tuned the learning rate of the weight update to 0.0001. After the learning rate was fixed, we benchmarked the sizes and number of

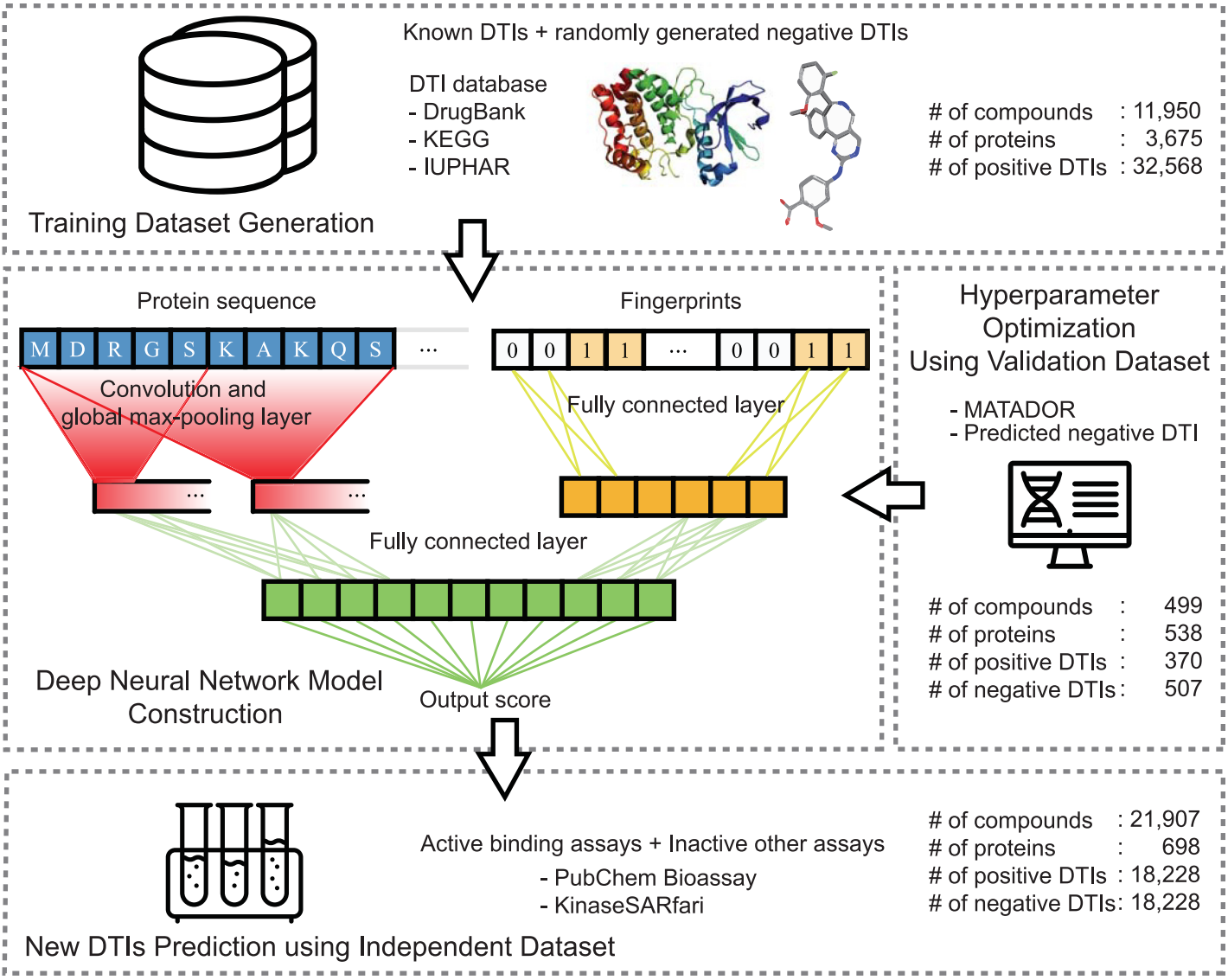


Fig 1. Overview of our model. First, we collected training DTI datasets from various databases (DrugBank, KEGG, IUPHAR). Second, we constructed the neural network model using convolution, which is able to capture local residue patterns that can help the DTIs. Third, we optimized the hyperparameters with an external validation dataset that we constructed. Finally, we predicted DTIs from bioassays (independent test dataset) and evaluated the performance of our model. The numbers (#) of compounds, proteins and DTIs are summarized in each step.

https://doi.org/10.1371/journal.pcbi.1007129.g001

windows, hidden layers of the drug features, and the concatenating layers with the area under precision-recall (AUPR) on the external unseen validation dataset, which was built with MATADOR and a highly credible negative dataset. Finally, we selected the hyperparameters of the model, as shown in Table A in S1 Text, with the external unseen validation dataset, yielding an AUPR of 0.832 and area under the curve (AUC) of 0.852, as shown in Fig 2. The AUPR value of our model was less than the AUPR of the similarity descriptor; however, that does not mean that our method has lower prediction performance than the similarity method because the size of the validation is too small to evaluate the general performance. In addition, we further examined the effect of fixed maximum protein length on the prediction performance. As shown in Fig A in S1 Text, we confirmed that the prediction performance of our model is not biased to the fixed maximum protein length. Finally, the fully optimized model is visualized as

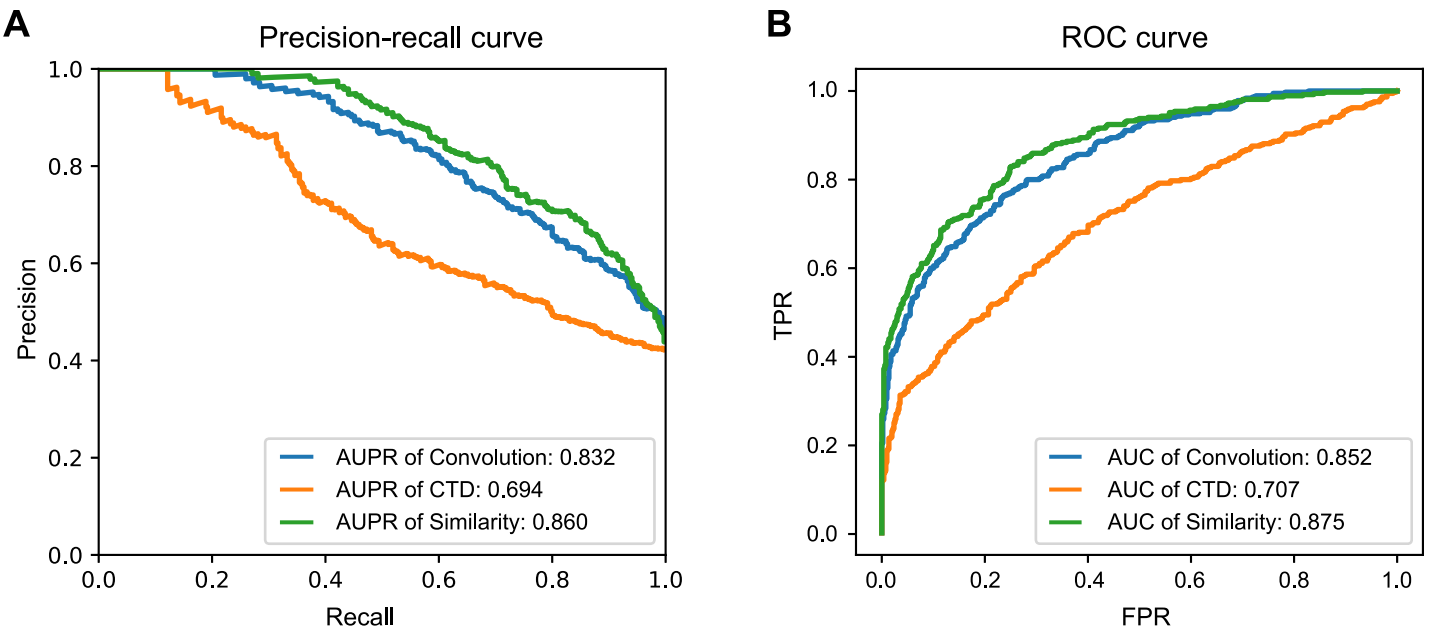


Fig 2. Performance curves for optimized models of protein descriptors. The AUPR and AUC of the convolution, CTD, and similarity descriptors are shown in panels (A) and (B), respectively.

https://doi.org/10.1371/journal.pcbi.1007129.g002

a graph, shown in S1 Fig, respective to our model, the CTD descriptor, and similarity descriptors. In the same manner, we built and optimized models that use other protein descriptors with the same activation function, learning rate, and decay rate.

Comparison of performance with other protein descriptors

After the hyperparameters were tuned, we compared the performance based on the independent test datasets with the different protein descriptors, the CTD descriptor (which is usually used in the conventional chemo-genomic model) [13], the normalized Smith-Waterman (SW) score [36], and our convolution method. The results showed that our model exhibited better performance than the other protein descriptors for all datasets, as shown in Fig 3 and Fig B in S1 Text. With the threshold selected by the equal error rate (EER) [37], our model performed equally well with both the PubChem and KinaseSARfari datasets, indicating that our model has general application power. Our convolution method gave the highest accuracy score and F1 score for the PubChem dataset (Fig 3**A**) [34] and its subsets (Fig 3**B**–3**D**) and a slightly lower F1 score for the KinaseSARfari dataset (Fig B in S1 Text) [35]. The CTD descriptor gave the lowest score for any dataset and any metric, which implies that CTD is less informative and less enriched than the other descriptors. Here, we also observed that the model performance using a similarity descriptor for the KinaseSARfari dataset was similar to that of the proposed model. We can interpret this result as the similarity descriptor acts as an informative feature as a local residue pattern at the domain level, not the whole protein complex.

Performance comparison with a previous model

In addition to the comparison between convolution in our model and other protein descriptors, in this section, we compared the performance of our model against recently developed deep-learning-based models. We selected three deep learning models for comparisons, SAE (MFDR, *Peng et al*, 2016) [22], DBN (DeepDTI, *Wen et al*, 2017) [7] and CNN (DeepDTA,

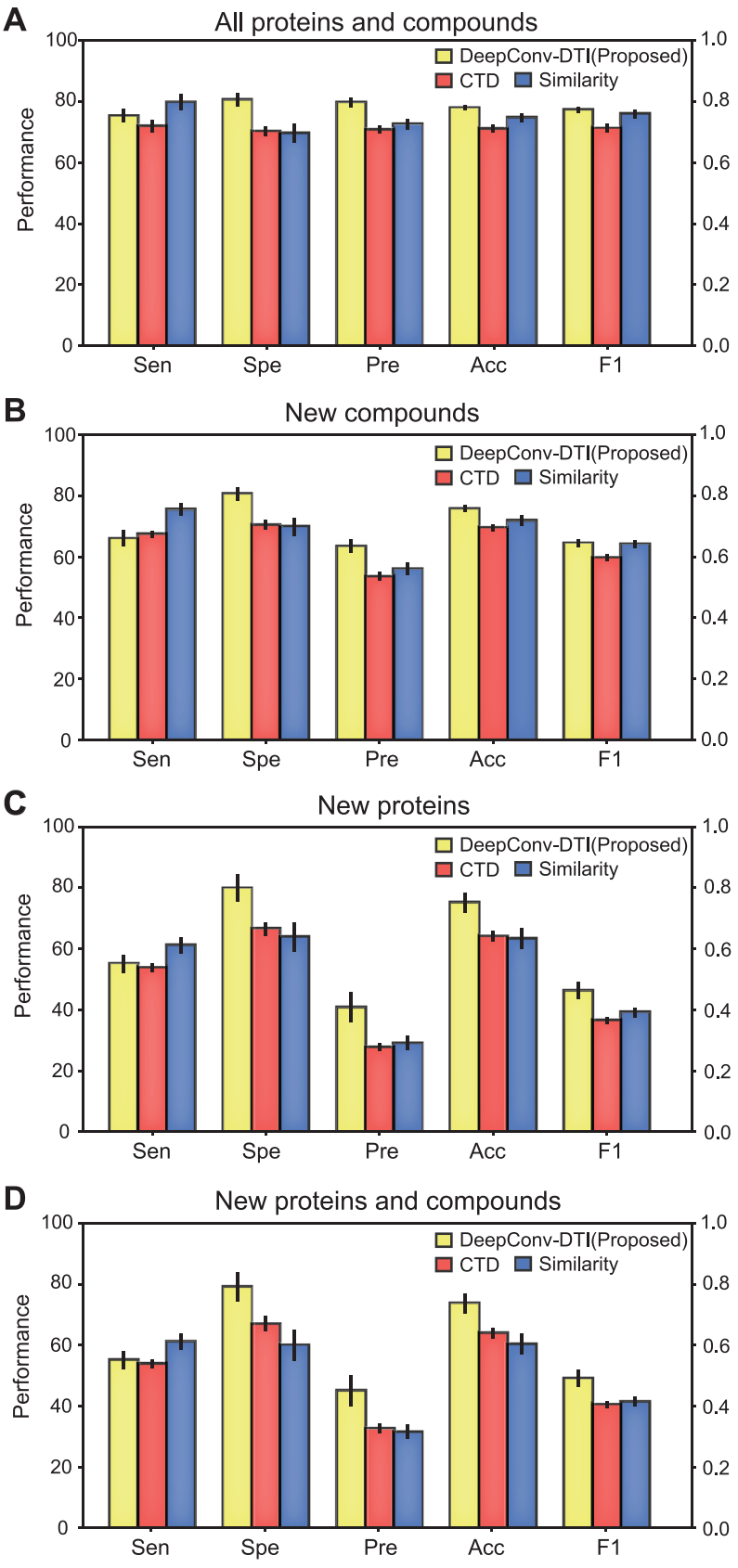

A
All proteins and compounds
DeepConv-DTI(Proposed)
CTD
Similarity
Performance
Sen
Spe
Pre
Acc
F1
B
New compounds
C
New proteins
D
New proteins and compounds

Fig 3. Performance measures for all of the independent datasets of the PubChem dataset. We measured various performances such as sensitivity (Sen.), specificity (Spe.), precision (Pre.), accuracy (Acc.), and F1 score (F1) from the prediction results given by descriptors (A-D). (A) All queried PubChem datasets. (B) PubChem dataset whose compounds are not in the training dataset. (C) PubChem dataset whose targets are not in the training dataset. (D) PubChem dataset whose compounds and targets are not in the training dataset. Our convolution model shows better performances for all datasets in terms of accuracy and F1 score.

https://doi.org/10.1371/journal.pcbi.1007129.g003

Ozturk et al, 2018). First, MFDR trains SAE in an unsupervised manner, while proteins are represented by multi-scale local descriptor feature [38] and compounds are represented by PubChem fingerprints as input and output for SAE. With trained deep representations of sparse Auto-Encoder, they performed 5-fold cross-validation by using SVM. As a result, their model gives better performances than previous bipartite local models. Because the authors do not provide the model, we implemented the MFDR model with optimized parameters the author provided in their original paper. We tested the validity of implemented MFDR and confirmed that the implemented model produces reasonably same performance compared to the results from its original work (see Fig C S1 Text). Second, DeepDTI built by *Wen et al*. is based on DBN [20], which is a stack of restricted Boltzmann machine (RBM). DeepDTI takes amino acid, dipeptide and tripeptide compositions (protein sequence composition descriptors, PSC) as the protein input and ECFP with radius 1, 2 and 3 as the compound input. We used DeepDTI with the code that the authors provided (https://github.com/Bjoux2/DeepDTIs_DBN) and optimized hyperparameters as the authors mentioned. Third, DeepDTA built by *Ozturk et al*. used stacked CNN on protein sequences and SMILES to predict affinity between target protein and compound. DeepDTA is optimized for Davis [25] and KIBA [26] dataset which contains kinases protein, their inhibitors, and dense affinity values, showing better prediction performances than previous affinity prediction models. We also used DeepDTA with the code from the original work (https://github.com/hkmztrk/DeepDTA) and optimized hyperparameters they provided. For the DTI prediction performance comparison, we activate the last layer with sigmoid function to predict interaction, not affinity, also we changed loss function as binary cross-entropy from mean squared error. It should be noticed that we compared the performance of all three models by training and testing with the same data set we used for a fair comparison.

Results of performance comparison between our proposed model and the three related models are shown in Fig 4, showing that performances (accuracy, F1) of our model (DeepConv-DTI) are better than other models. MFDR which gave high AUC in 5-fold cross-validation shows decreased performances in the independent test dataset. We can speculate that SAE which learns deep representation of DTI in an unsupervised way is not appropriate for a case that datasets are composed of various protein classes. In the case of DeepDTI, DeepDTI takes physicochemical properties (PSC) of whole protein sequence including subsequences or domains which do not participate in the interaction with compounds, resulting in worse performance than our model which extracts local residue patterns. For DeepDTA, DeepDTA also shows worse performances than our model with having a relatively large variance. We interpret the worse performance of DeepDTA as follows. DeepDTA is optimized for a densely constructed dataset with specific protein class, while the training dataset in this comparison covers various protein classes (kinase, protease, ion channel, nuclear receptor, GPCR, etc), not only kinase class. Thus, DeepDTA which is specialized for a specific protein class could not achieve better prediction performance in the generalized protein classes.

In addition to the three models we compared, we also compared our model with DL-CPI [23] built by *Tian et al*. which used protein domain information. For proteins whose domain information is not in Pfam [39], datasets for training, validation and test are not fully available.

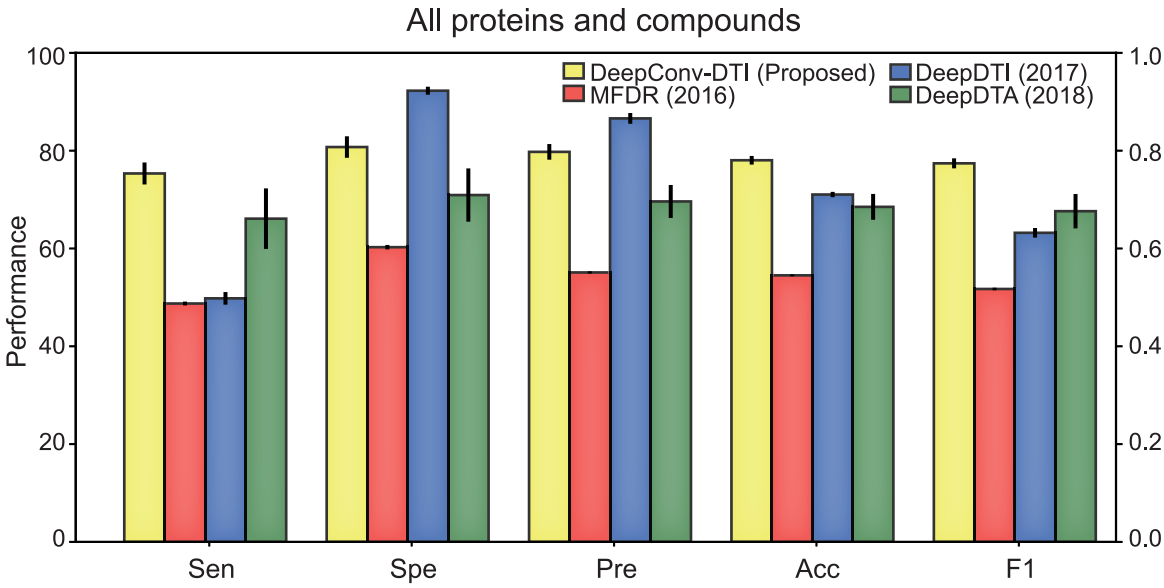


Fig 4. Comparison of performances between our model and previous models. We compared performances of our model on independent test dataset (PubChem) with previous models (MFDR, DeepDTI, and DeepDTA). Our model gives better performances than previous models for accuracy and F1 metrics.

https://doi.org/10.1371/journal.pcbi.1007129.g004

Therefore, we independently compared performances between DL-CPI and our model by additionally built the training, validation, and test datasets. Performance comparison results are described in Fig E in S1 Text. We confirmed that the proposed model shows better performance than DL-CPI. Because protein descriptor of DL-CPI is sparse, containing few values in large dimension, which may decrease performances.

In overall, our model shows better performance than previous deep learning models in an independent test dataset from a different database, which contains distinct DTIs, dealing with DTIs with various protein classes and their interacting compounds.

Analysis of convolution results

Because we pooled the maximum convolution results by each filter for each window, the pooled results could highlight regions of matches with local residue patterns. Although we cannot measure exactly how those values affect the DTI prediction results, the pooled maximum convolution result will affect the prediction performance by going through higher fully connected layers. Therefore, if our model is capable of capturing local residue patterns, it would give high values to important protein regions, such as actual binding sites.

Examining and validating the convolution results from the intermediate layer showed that our model could capture local residue patterns that participate in DTIs. The sc-PDB database provides atom-level descriptions of proteins, ligands, and binding sites from complex structures [40]. By parsing binding site annotations, we can query binding sites between protein domains and pharmacological ligands for 7,179 entries of Vertebrata. From the queried binding sites and pooled maximum convolution results, we statistically test our assumption that the pooled maximum convolution results cover the important regions, including binding sites. Each window has 128 pooled convolution results, which shows bias in covering some regions. Thus, we randomly generated 128 convolution results 10,000 times for each sc-PDB entry and counted how many of those random results covered each amino acid in the binding sites, which resulted in the construction of normal distributions. For each normal distribution

constructed by the randomly generated convolution results, considered a null hypothesis, we executed a right-tailed *t*-test with the number from the convolution results of our model for each window. Because we did not know which window detects the binding site, we took the most significant p-value (minimum p-value adjusted by the Benjamini-Hochberg procedure [41]). The sc-PDB entry information and p-values of a window for each sc-PDB entry are summarized in the S1 File. We summarize the results of binding site detection from the most significant p-value among windows by significance level cutoff in Fig 5. In addition, we examined sc-PDB entries with the most significant p-values for diverse window sizes. We visualized two high-score sc-PDB entries from two perspectives—the whole receptor-ligand complex and binding site-ligand perspectives—by using UCSF Chimera [42] as shown in Fig 6. To visualize convolution results with a simplified view, first, we selected the top 5 ranked globally max-pooled results among all filters for each window because whole protein sequences are usually covered by convolution results if we select all results. Second, we rendered residues covered by convolution results by the number of covering convolution results. We visualized two sc-PDB entries, 1a7x_1 and 1ny3_1. 1a7x_1, representing the complex of the ion channel, protein Peptidyl-prolyl cis-trans isomerase FKBP1A (FKB1A_HUMAN in UniProt), which has a short sequence length (108), and BENZYL-CARBAMIC ACID [8-DEETHYL-ASCOMYCIN-8-YL] ETHYL ESTER (FKA in PDB ligand) [43]. 1ny3_1 is the complex of the kinase protein, MAP kinase-activated protein kinase 2 (MAPK2_HUMAN in UniProt) with sequence length 400, and ADENOSINE-5'-DIPHOSPHATE (ADP in PDB ligand) [44]. Through the above evaluation, we can confirm that our proposed model is capable of capturing local residue patterns of proteins that are considered important features for DTI prediction, such as actual binding sites.

t-SNE visualization of proteins

From the results shown in Fig 6, we can confirm that our model can capture the local residue patterns of proteins that participate in DTIs. Thus, to examine further characteristics of the captured protein local residue patterns, we visualized the protein features from the fully connected layer after the global max-pooling of convolution results. We visualized 1,527 proteins used in the training dataset categorized in various protein classes. Specifically, we visualized 257 GPCRs, 44 nuclear receptors, 304 ion channel receptors, 604 kinases, and 318 proteases. For visualization, we conducted t-distributed stochastic neighbor embedding (t-SNE) for dimension reduction and visualization [45]. t-SNE can map high-dimensional features to low-dimensional ones, such as 2-dimensional features, minimizing information loss during dimension reduction. Surprisingly, although our model is not intended to identify protein classes, it can roughly discriminate protein classes from the intermediate protein layer, as shown in Fig G in S1 Text.

Discussion

In this work, we built a novel DTI prediction model to extract local residue patterns of whole target protein sequences with CNN. We trained the model with DTIs from various drug databases and optimized the model with an external validation dataset. As a result, the detected local features of protein sequences perform better than other protein descriptors, such as CTD and SW scores. Our model also performs better than a previous model built on DBN. In addition, by analyzing pooled convolution results and statistically and manually comparing them with annotations from sc-PDB entries, we showed that, for some proteins, our model is capable of detecting important regions, including binding sites. Therefore, our approach of capturing local residue patterns with CNN successfully enriches protein features for DTI prediction.

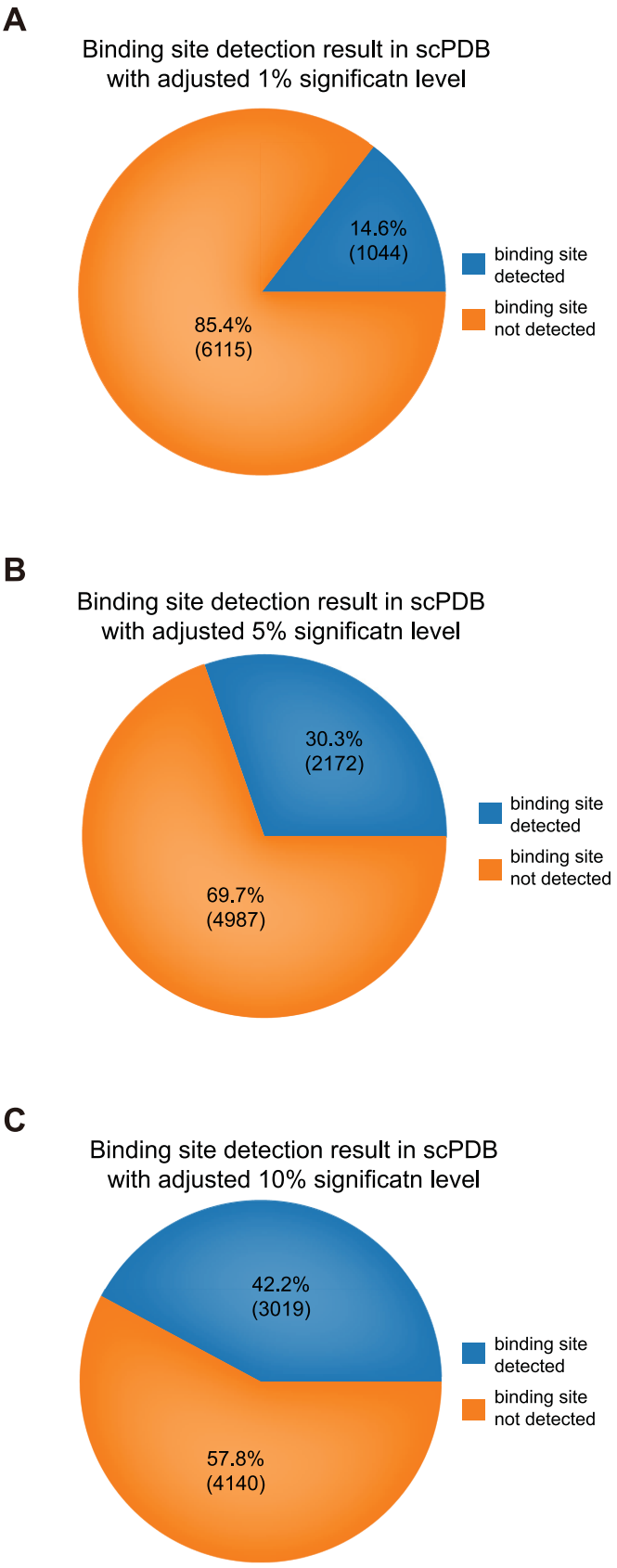

A
Binding site detection result in scPDB
with adjusted 1% significatn level
14.6%
(1044)
85.4%
(6115)
binding site
detected
binding site
not detected
B
Binding site detection result in scPDB
with adjusted 5% significatn level
30.3%
(2172)
69.7%
(4987)
binding site
detected
binding site
not detected
C
Binding site detection result in scPDB
with adjusted 10% significatn level
42.2%
(3019)
57.8%
(4140)
binding site
detected
binding site
not detected

Fig 5. Statistical test for binding region detection. We executed a right-tailed *t*-test for the number of covering binding sites from the convolution results with a null distribution, which was constructed from the randomly generated convolution results in the sc-PDB database consisting of 7,179. Because each sc-PDB test has many windows, we selected the most significant p-values adjusted by the Benjamini-Hochberg procedure and examined whether they were significant at levels of 1%, 5% and 10%. The results showed that 14.6%, 30.3% and 42.2% of sc-PDB entries were significantly enriched, respectively (A-C).

https://doi.org/10.1371/journal.pcbi.1007129.g005

The number of 3D structures in Protein Data Bank [46] is relatively smaller than the number of sequences, limiting 3D structure-based DTI prediction methods. For example, the number of PDB entries for *Homo sapiens* is 42,745, while the number of protein sequences for *Homo sapiens* is 177,661 in UniProtKB. However, our method does not depend on the 3D structure of proteins because it considers only protein sequence, rather than classical protein feature descriptors such as the CTD descriptor and normalized SW score. As a result, our method can be more generally applied to predict DTIs than methods needing 3D structures.

Although our model shows improved prediction performance, there is still room for improvement. First, we simply used Morgan/Circular fingerprints, which are binary and have large dimensions. Therefore, we will use more informative chemical descriptors, based on neural networks for DTI prediction, to achieve advanced performance. Second, as shown in a previous study [47], considering 3D structure information is an effective substitution for chemical elaboration. Therefore, in the future, we will elaborate upon our model by considering 3D structure features.

Materials and methods

Building dataset

To build the training dataset, we obtained known DTIs from three databases: DrugBank, KEGG, and IUPHAR. To remove duplicate DTIs among the three databases, we unified the identifiers of the compounds and the proteins. For the drugs, we standardized the identifiers of the compounds in the DrugBank and KEGG databases with the InChI descriptor. For the proteins, we unified the identifiers of the proteins as UniProtKB/Swiss-Prot accessions [48]. Among the collected DTIs, we selectively removed proteins of Prokaryota and single-cell Eukaryota, retaining only proteins of Vertebrata. Finally, 11,950 compounds, 3,675 proteins, and 32,568 DTIs were obtained in total. Because all collected DTIs are regarded as positive samples for training and negative DTIs are not defined in the databases above, a random negative DTI dataset is inevitably generated. To reduce bias from the random generation of negative DTIs, we built ten sets of negative DTIs exclusively from the positive dataset. The detailed statistics of the collected training dataset are shown in Table D in S1 Text.

To optimize our model with the most adequate hyperparameters, we constructed an external validation dataset that had not seen DTIs in the training phase. We collected positive DTIs from the MATADOR database [32], including 'DIRECT' protein annotations, and all DTIs observed in the training dataset were excluded. To build a credible negative dataset, we obtained negative DTIs via the method of *Liu et al.* [33]. This method selects candidate negative DTIs with low similarity to known positive DTIs. From the obtained negative dataset, we balanced the negative dataset with the positive dataset, using a negative score (>0.95). As a result, 370 positive DTIs and 507 negative DTIs were queried for the external validation set. The statistics of the external validation dataset are summarized in Table E in S1 Text.

To evaluate our model, we built two independent test datasets from the PubChem BioAssay database [34] and ChEMBL KinaseSARfari [35]; these datasets consisted of results from experimental assays. To obtain positive DTIs from PubChem, we collected 'Active' DTIs from the

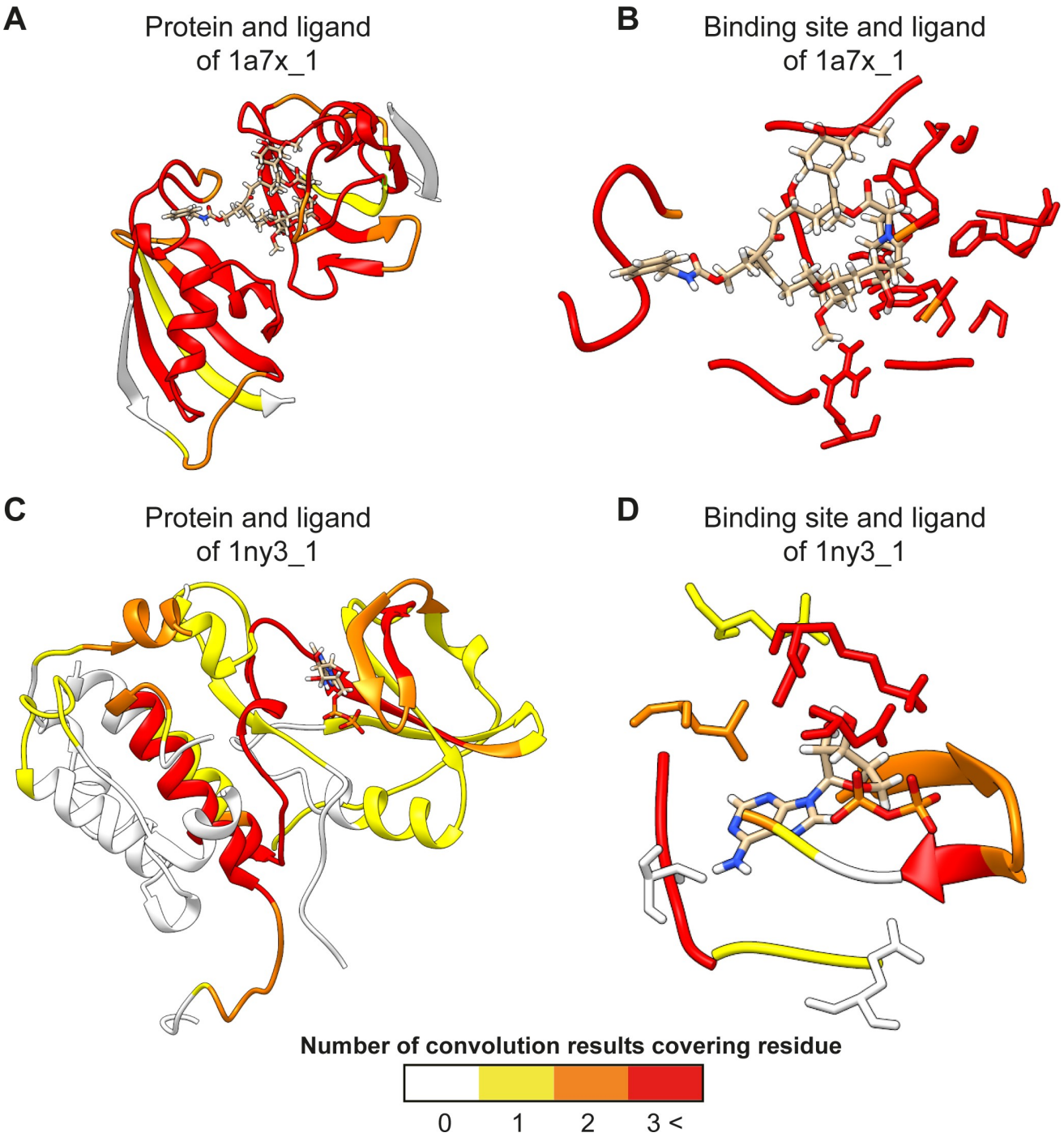


Fig 6. Visualization of convolution results. We visualized two highly scored sc-PDB entries from two perspectives—the whole receptor-ligand complex and binding site-ligand perspectives—by using UCSF Chimera. To visualize convolution results with a simplified view, first, we selected the top 5 ranked globally max-pooled results among all filters for each window because whole protein sequences are usually covered by convolution results if we select all results. Second, we rendered residues covered by the convolution results by the number of covering convolution results. (A) Complex of the ion channel protein Peptidyl-prolyl cis-trans isomerase FKBP1A (FKB1A_HUMAN in UniProt), which has a short sequence length (108), and FKA in the PDB ligand (1a7x_1 in sc-PDB). As we can see, the number of convolution results near the ligand is more than the number for the other region. (B) For the binding site and ligand of 1a7x_1, most of the binding sites are highly covered by the convolution results. (C) Complex of the kinase protein MAP kinase-activated protein kinase 2 (MAPK2_HUMAN in UniProt) with a sequence length of 400 and ADP in the PDB ligand (1ny3_1 in sc-PDB). Although half of the protein sequence is not represented as a 3D structure, our convolution results cover regions close to ligand binding sites in a biased manner. However, some residues far from binding sites are also highlighted by convolution results, potentially indicating some important structural motifs for binding. (D) For the binding site and ligand of 1ny3_1, most binding sites are covered by the convolution results, although some residues are not covered.

https://doi.org/10.1371/journal.pcbi.1007129.g006

assays with the dissociation constant ($K_d < 10\mu m$) [49]. Because we sought to predict whether a drug binds to a protein, among the many types of assays (Potency, IC_{50}, AC_{50}, EC_{50}, K_d, K_i), evaluation of the dissociation constant (K_d) was the most appropriate assay for obtaining positive samples. For the negative samples, we took the samples annotated as 'Inactive' from the other assay types. Because there were too many negative samples in the PubChem BioAssay database, we first collected only negative samples whose drug or target was included in the positive samples from the PubChem BioAssay database. Second, we selected as many random negative samples as positive DTIs from PubChem BioAssay. As a result, total 36,456 positive and negative samples were built with 21,907 drugs and 698 proteins. For the performance evaluation, we created three subsets of the PubChem bioassay independent dataset for humans, which consisted of only new compounds, new proteins, and new DTIs. Detailed summaries of the PubChem dataset and its subset are shown in Table F in S1 Text. We also collected samples from KinaseSARfari. KinaseSARfari consists of assays involving a compound that binds to a kinase domain. To obtain positive samples from KinaseSARfari, we considered each assay result with a dissociation constant of ($K_d < 10\mu m$) as positive [49]; this value is sufficiently small to be considered positive. In contrast to the PubChem BioAssay, the number of negative samples was similar to the number of positive samples in KinaseSARfari; therefore, we did not sample the negative samples. We collected 3,835 positive samples and 5,520 negative samples with 3,379 compounds and 389 proteins. Detailed statistics of the KinaseSARfari dataset are shown in Table F in S1 Text. In addition, we summarize the portion of the protein class in each dataset in Fig H in S1 Text. Here, we confirmed that the training and the validation datasets were not biased toward a specific protein class.

Drug feature representation

In our model, we used the raw protein sequence as the input for the protein but did not use the raw SMILES string as the input for the drug. For the drug, we used the Morgan/Circular drug fingerprint, which analyzes molecules as a graph and retrieves substructures of molecular structures from subgraphs of the whole molecular graph [21]. Specifically, we used RDKit [50] to yield a Morgan/Circular fingerprint with a radius of 2 from a raw SMILES string. Finally, each drug can be represented as a binary vector with a length of 2,048, whose indices indicate the existence of specific substructures.

Deep neural network model

Overall schema of the deep learning network. We extracted the local residue patterns from protein sequences via CNN and yielded a latent representation of drug fingerprints via fully connected layers. After processing both the drug and protein layers, we concatenated these layers and constructed the fully connected layer, resulting in the output. Every layer except the output layer was activated with the exponential linear unit (ELU) function [51].

$$\sigma(\alpha, x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

The output layer was activated with the sigmoid function for classification. The whole neural network model was implemented with Keras (2.16) [52].

Convolution layer with protein embedding vector. One of the difficulties in describing the protein features for the machine learning model and the deep learning model was that the protein lengths were all different. Another difficulty was that only certain parts of a protein, such as specific domains or motifs, are involved in DTIs, rather than the whole protein structure. As a

result, the physicochemical properties of the whole protein sequence do not seem to be appropriate features for predicting DTIs due to noise information from the portions of the sequence that are not involved in the DTIs. Thus, the extraction of local residue patterns involved in the DTIs is necessary for precise prediction, and CNN is known to capture important local patterns from the whole space. The overall schema of convolutional layers is depicted in Fig 7. The model starts with an embedding to transform each amino acid to the corresponding embedding vector. The embedding layer is a lookup table of embedding vectors. Embedding vector values are randomly initialized by the Xavier initializer (denoted 'glorot normal' in keras), which imposes normal distribution of weights and variance of output following variance of input [53]. Embedding vectors are trainable, meaning that embedding vector values are also changed to optimize loss during training. From the lookup table, the embedding matrix for the protein sequence is constructed by querying embedding vectors corresponding to amino acids from the embedding layer, as described in Fig I in S1 Text. The length of the embedding matrix for all proteins was set to the same as the maximum protein length, i.e., 2,500, and the margins were padded with null labels ($) and the corresponding embedding vectors, which would give a meaningless convolution result that is filtered out during global max-pooling as depicted in Fig J in S1 Text. As a result, an embedding layer was constructed for protein features. We executed convolution on the embedding layer of the protein along the sequence in 1D fashion with striding 1, with convolution from j^{th} the to the $(j+WS)^{th}$ amino acids in sequence, which can be defined as

$$(\mathrm{x} * \mathrm{w})_{\mathrm{j}} = \sum_{a=1}^{\mathrm{ES}} \sum_{b=0}^{\mathrm{WS}-1} w_{a,b} x_{a,j+b}$$

Convolution for the whole sequence results in a (MPL-WS+1) size convolution layer for each filter, where WS is the window size. Finally, to extract the most important local feature, we conducted global max-pooling for each filter, which is defined as

$$\mathrm{MaxPooling}_{\mathrm{global}}(E_{\mathrm{p_k}}) = \max((x * w)_j)$$

where j covers all of the convolution results of the embedding matrix from protein sequence p_k, resulting in a filter-sized vector with a max-valued convolution result for each window, which does not induce bias from the locations of local residue patterns and the maximum protein length. After pooling all convolution results, we concatenated them to represent the important local patterns for interactions as a vector-formatted feature. Finally, for the organization and abstraction of protein features, concatenated max-pooling results are fed into fully connected layers, which constructs a latent representation of protein.

Fully connected layers for drug fingerprints and concatenating layer. As mentioned in the Introduction, latent representations of the drug fingerprint descriptors made by fully connected layer are useful for predicting DTIs. After features of the protein and drug were refined by the neural network, we concatenated them and constructed fully connected layers to predict whether the drug and target interact.

Calculation of loss and weight optimization. Using the constructed deep neural model, the input flows to the output layer in a feed-forward fashion. The deep neural model calculates loss with binary cross-entropy:

$$J(W, b) = -\frac{1}{n}\sum_{i}^{n}[y_i \log\hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)]$$

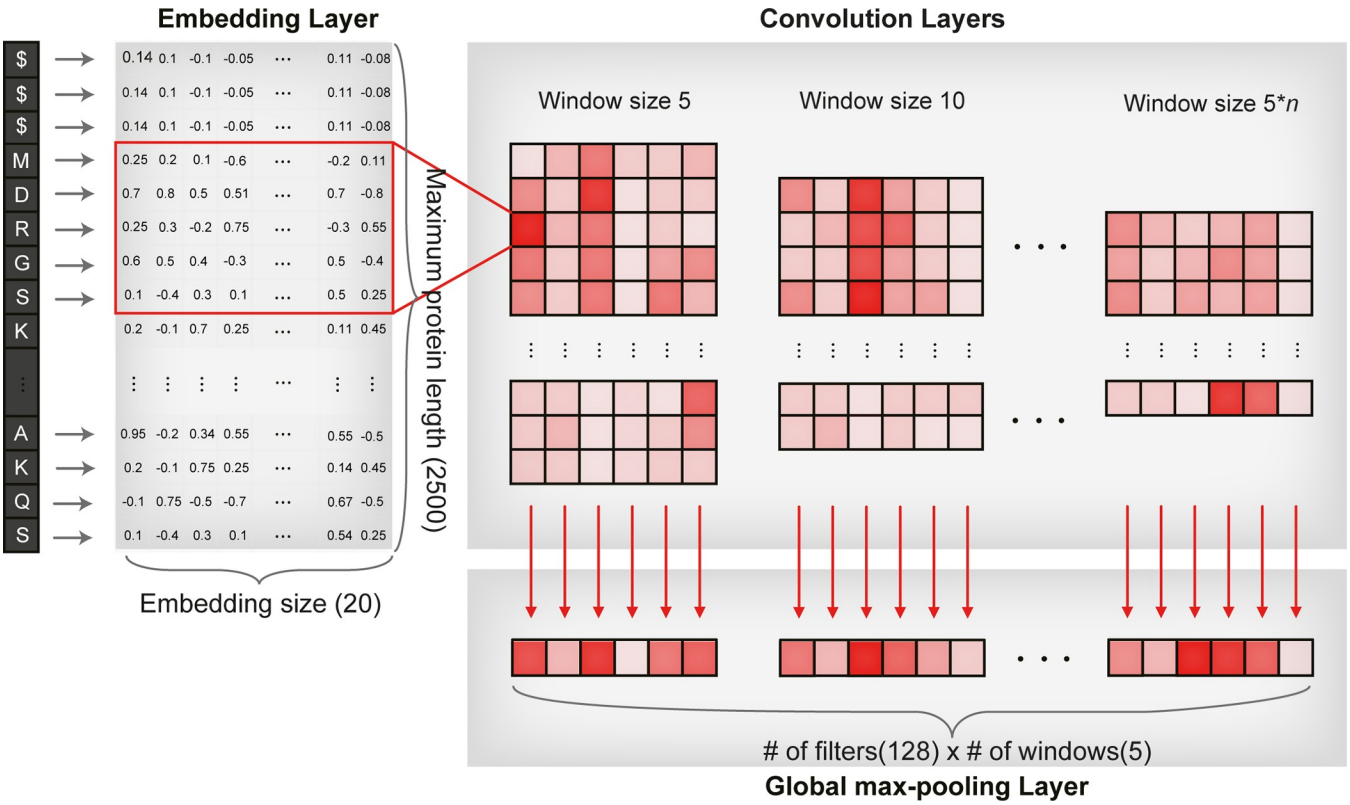


Fig 7. Overall schema for extracting the local patterns from the whole protein sequence. First, we transformed the protein sequence to an embedding vector with a fixed size, and the margins were padded, which are marked as $ and the corresponding embedding vectors. Second, we executed convolution along the sequence. Third, for each filter of window size, we pooled the max value. By concatenating all of the max-pooling values, we built a protein feature vector whose dimension is multiplying the number (#) of filters by the number (#) of windows.

https://doi.org/10.1371/journal.pcbi.1007129.g007

To prevent overfitting, we penalized the loss function with L2-norm:

$$J_{L2}(W,b) = J(W,b) + \lambda \sum_{l=1}^{L-1} \|W^l\|_2$$

Finally, we updated the weights using the Adam optimizer [54] with a penalized loss to give a generalized prediction for the model.

Regularizations of the neural network. In the artificial neural network technique, there are several ways to prevent overfitting. Currently, dropout and batch normalization are most frequently used to regularize neural networks. Dropout masks hidden nodes in the training phase, which makes a subset of hidden nodes unavailable to predict results for training labels [55]. By masking some hidden nodes in training, dropout generalizes the model, making the model independent of a specific dataset. We used 1-dimensional spatial dropout on the embedding layer [56]. In addition, we used a batch normalization technique to prevent overfitting except in the embedding layer. Batch normalization normalizes the outputs of the neural network with a mean of 0 and a standard deviation of 1 on a minibatch. However, batch normalization could induce a loss in the influence of parameters and linearity of network outputs, rather than nonlinearity. Thus, batch normalization induces a scale factor and a shift factor for normalized outputs, whose values are also introduced in the learning phase, to resolve the problem [57].

Selection of hyperparameters. In our deep learning models, hyperparameters, such as the learning rate and window sizes that affect performance, are tuned during cross-validation. However, the hyperparameters should not be determined based on the performance of the subset of the training dataset because the negative datasets are randomly sampled. With the external validation dataset, we first determined the learning rate because a model with a high learning rate is unable to learn a pattern. After the learning rate was selected, we selected activation function and regularization parameters such as the dropout ratio. Finally, we employed a grid-search method for optimization of the other hyperparameters that determine neural network shape. The search range of optimization is summarized in Table A in S1 Text. We identified hyperparameters that exhibited the best AUPR, which is an appropriate performance evaluation metric for the accuracy of classifying the positive sample. The other descriptors to compare with our methods are numerical vectors, which do not have locality. Therefore, we put fully connected layers on the protein descriptors. We also employed a grid-search strategy while sustaining hyperparameters not related to model shape. When the AUPR is measured, the optimal threshold can be given by the EER [37].

$$\mathrm{EER} = \underset{\theta}{\mathrm{argmin}}(|1 - \mathrm{recall}| - \gamma|1 - \mathrm{precision}|)$$

where θ is the classification threshold and γ is the constant determining the cost ratio for misclassification from precision and recall, which is set at 2 in our model.

Sparse Auto-Encoder (SAE) construction

SAE is Auto-Encoder whose distribution of latent representations is regularized with sparsity term [58]. In loss calculation, Kullback-Leibler divergence (KLD) loss between Bernoulli distributions each dimension in latent representation $\hat{\rho}$ and desired sparsity parameter ρ is added to reconstruction loss of Auto-Encoder and ridge loss for weights.

$$\mathrm{J}_{\mathrm{sparse}}(\mathrm{W}, \mathrm{b}) = \mathrm{J}(\mathrm{W}, \mathrm{b}) + \beta \sum_{j}^{s_2} KL(\rho||\hat{\rho}_j)$$

where

$$\hat{\rho}_{\mathrm{j}} = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})]$$

During the training of the neural network, KLD acts as a constraint for latent representation following desired sparsity parameter. As a result, for each dimension of latent representation, only a few samples are activated, giving a more reliable representation of original input. In the previous study, MFDR used SAE to build an informative latent representation of DTI, which are composed of multi-scale local descriptors [38] and PubChem fingerprints.

Deep belief network (DBN) construction

DBN is a generative graphical model proposed by Geoffrey Hinton [20]. DBN is actually a stack of an RBM. RBM consists of visible and hidden units, constructing a bipartite graph. In RBM, probabilistic distribution of visible units is learned in an unsupervised way, with a probabilistic distribution of visible and hidden units

$$\mathrm{P}(v, h|W) = \frac{1}{Z}e^{a^T v + b^T h + v^T W h}$$

and marginal distribution of visible units

$$P(v|W) = \frac{1}{Z}\sum_{h} e^{a^T v + b^T h + v^T W h}$$

to maximize the probability of visible units for V in a training set with weight matrix W

$$\underset{W}{\operatorname{argmax}} \prod_{v \in V} P(v|W)$$

In DBN, during stacking of RBMs, hidden units of the previous RBM are fed as visible layers of the next RBM. In addition, RBM adopts contrastive divergence for fast training, which uses gradient descent and Gibbs sampling. In a previous study, DeepDTI, the input concatenation of drug and target protein features, PSC descriptors and ECFP with a radius of 1, 2 and 3, was considered a first visible layer. The authors attached logistic regression to the last hidden units to predict DTIs.

Evaluation of performances

To measure the prediction performance of our deep neural model based on the independent test dataset after the classification threshold was fixed, we obtained the following performance metrics: sensitivity (Sen.), specificity (Spe.), precision (Pre.), accuracy (Acc.), and the F1 measure (F1). See the formulas below:

$$\text{Sen.} = \text{TP}/\text{P}$$

$$\text{Spe.} = \text{TN}/\text{N}$$

$$\text{Pre.} = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{Acc.} = (\text{TP} + \text{TN})/(\text{P} + \text{N})$$

$$\text{F1} = (\text{Sen} * \text{Pre})/(\text{Sen} + \text{Pre})$$

where TP is true positive, TN is true negative, FP is false positive, FN is false negative, T is positive, and N is negative.

Supporting information

S1 Text. Supporting information.
(PDF)

S1 Fig. Graph visualization of optimized models.
(PDF)

S1 File. Metadata and results of statistical test for sc-PDB entries.
(CSV)

Author Contributions

Conceptualization: Ingoo Lee, Jongsoo Keum, Hojung Nam.

Data curation: Ingoo Lee, Jongsoo Keum.

Formal analysis: Ingoo Lee.

Funding acquisition: Hojung Nam.

Investigation: Ingoo Lee, Hojung Nam.

Methodology: Ingoo Lee, Jongsoo Keum, Hojung Nam.

Project administration: Hojung Nam.

Resources: Hojung Nam.

Software: Ingoo Lee, Jongsoo Keum.

Supervision: Hojung Nam.

Validation: Ingoo Lee, Jongsoo Keum, Hojung Nam.

Visualization: Ingoo Lee.

Writing – original draft: Ingoo Lee.

Writing – review & editing: Ingoo Lee, Hojung Nam.

References

1. Kapetanovic IM. Computer-aided drug discovery and development (CADDD): in silico-chemico-biological approach. Chem Biol Interact. 2008; 171(2):165–76. https://doi.org/10.1016/j.cbi.2006.12.006 PMID: 17229415; PubMed Central PMCID: PMC2253724.
2. Gowthaman R, Miller SA, Rogers S, Khowsathit J, Lan L, Bai N, et al. DARC: Mapping Surface Topography by Ray-Casting for Effective Virtual Screening at Protein Interaction Sites. J Med Chem. 2016; 59 (9):4152–70. https://doi.org/10.1021/acs.jmedchem.5b00150 PMID: 26126123; PubMed Central PMCID: PMC4707132.
3. Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M. Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. Bioinformatics. 2008; 24(13):i232–40. https://doi.org/10.1093/bioinformatics/btn162 PMID: 18586719; PubMed Central PMCID: PMC2718640.
4. Bleakley K, Yamanishi Y. Supervised prediction of drug-target interactions using bipartite local models. Bioinformatics. 2009; 25(18):2397–403. https://doi.org/10.1093/bioinformatics/btp433 PMID: 19605421; PubMed Central PMCID: PMC2735674.
5. Zheng X, Ding H, Mamitsuka H, Zhu S. Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining; Chicago, Illinois, USA. 2487670: ACM; 2013. p. 1025–33.
6. Ezzat A, Zhao P, Wu M, Li XL, Kwoh CK. Drug-Target Interaction Prediction with Graph Regularized Matrix Factorization. IEEE/ACM Trans Comput Biol Bioinform. 2017; 14(3):646–56. https://doi.org/10.1109/TCBB.2016.2530062 PMID: 26890921.
7. Wen M, Zhang Z, Niu S, Sha H, Yang R, Yun Y, et al. Deep-Learning-Based Drug-Target Interaction Prediction. J Proteome Res. 2017; 16(4):1401–9. https://doi.org/10.1021/acs.jproteome.6b00618 PMID: 28264154.
8. Kimothi Dhananjay SA, Biyani Pravesh,Anand Saket, Hogan James M. Metric learning on biological sequence embeddings. 2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). 2017;(1–5).
9. Cheng F, Zhou Y, Li J, Li W, Liu G, Tang Y. Prediction of chemical-protein interactions: multitarget-QSAR versus computational chemogenomic methods. Mol Biosyst. 2012; 8(9):2373–84. https://doi.org/10.1039/c2mb25110h PMID: 22751809.
10. He Z, Zhang J, Shi XH, Hu LL, Kong X, Cai YD, et al. Predicting drug-target interaction networks based on functional groups and biological features. PLoS One. 2010; 5(3):e9603. https://doi.org/10.1371/journal.pone.0009603 PMID: 20300175; PubMed Central PMCID: PMC2836373.
11. Wang F, Liu D, Wang H, Luo C, Zheng M, Liu H, et al. Computational screening for active compounds targeting protein sequences: methodology and experimental validation. J Chem Inf Model. 2011; 51 (11):2821–8. https://doi.org/10.1021/ci200264h PMID: 21955088.

12. Cereto-Massague A, Ojeda MJ, Valls C, Mulero M, Garcia-Vallve S, Pujadas G. Molecular fingerprint similarity search in virtual screening. Methods. 2015; 71:58–63. https://doi.org/10.1016/j.ymeth.2014.08.005 PMID: 25132639.

13. Dubchak I, Muchnik I, Holbrook SR, Kim SH. Prediction of protein folding class using global description of amino acid sequence. Proc Natl Acad Sci U S A. 1995; 92(19):8700–4. https://doi.org/10.1073/pnas.92.19.8700 PMID: 7568000; PubMed Central PMCID: PMC41034.

14. Li ZC, Huang MH, Zhong WQ, Liu ZQ, Xie Y, Dai Z, et al. Identification of drug-target interaction from interactome network with 'guilt-by-association' principle and topology features. Bioinformatics. 2016; 32(7):1057–64. https://doi.org/10.1093/bioinformatics/btv695 PMID: 26614126.

15. Lee I, Nam H. Identification of drug-target interaction by a random walk with restart method on an interactome network. BMC Bioinformatics. 2018; 19(Suppl 8):208. https://doi.org/10.1186/s12859-018-2199-x PMID: 29897326; PubMed Central PMCID: PMC5998759.

16. Tabei Y, Yamanishi Y. Scalable prediction of compound-protein interactions using minwise hashing. BMC Syst Biol. 2013; 7 Suppl 6:S3. https://doi.org/10.1186/1752-0509-7-S6-S3 PMID: 24564870; PubMed Central PMCID: PMC4029277.

17. Sawada R, Kotera M, Yamanishi Y. Benchmarking a Wide Range of Chemical Descriptors for Drug-Target Interaction Prediction Using a Chemogenomic Approach. Molecular Informatics. 2014; 33(11–12):719–31. https://doi.org/10.1002/minf.201400066 PMID: 27485418

18. Min S, Lee B, Yoon S. Deep learning in bioinformatics. Brief Bioinform. 2017; 18(5):851–69. https://doi.org/10.1093/bib/bbw068 PMID: 27473064.

19. Gawehn E, Hiss JA, Schneider G. Deep Learning in Drug Discovery. Mol Inform. 2016; 35(1):3–14. https://doi.org/10.1002/minf.201501008 PMID: 27491648.

20. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Comput. 2006; 18(7):1527–54. WOS:000237698100002. https://doi.org/10.1162/neco.2006.18.7.1527 PMID: 16764513

21. Rogers D, Hahn M. Extended-Connectivity Fingerprints. Journal of Chemical Information and Modeling. 2010; 50(5):742–54. WOS:000277911600004. https://doi.org/10.1021/ci100050t PMID: 20426451

22. Peng W, Chan KCC, You ZH, editors. Large-scale prediction of drug-target interactions from deep representations. 2016 International Joint Conference on Neural Networks (IJCNN); 2016 24–29 July 2016.

23. Tian K, Shao M, Wang Y, Guan J, Zhou S. Boosting compound-protein interaction prediction by deep learning. Methods. 2016; 110:64–72. https://doi.org/10.1016/j.ymeth.2016.06.024 PMID: 27378654.

24. Ozturk H, Ozgur A, Ozkirimli E. DeepDTA: deep drug-target binding affinity prediction. Bioinformatics. 2018; 34(17):i821–i9. https://doi.org/10.1093/bioinformatics/bty593 PMID: 30423097; PubMed Central PMCID: PMC6129291.

25. Davis MI, Hunt JP, Herrgard S, Ciceri P, Wodicka LM, Pallares G, et al. Comprehensive analysis of kinase inhibitor selectivity. Nat Biotechnol. 2011; 29(11):1046–51. https://doi.org/10.1038/nbt.1990 PMID: 22037378.

26. Tang J, Szwajda A, Shakyawar S, Xu T, Hintsanen P, Wennerberg K, et al. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. J Chem Inf Model. 2014; 54(3):735–43. https://doi.org/10.1021/ci400709d PMID: 24521231.

27. Nascimento AC, Prudencio RB, Costa IG. A multiple kernel learning algorithm for drug-target interaction prediction. BMC Bioinformatics. 2016; 17:46. https://doi.org/10.1186/s12859-016-0890-3 PMID: 26801218; PubMed Central PMCID: PMC4722636.

28. He T, Heidemeyer M, Ban F, Cherkasov A, Ester M. SimBoost: a read-across approach for predicting drug-target binding affinities using gradient boosting machines. J Cheminform. 2017; 9(1):24. https://doi.org/10.1186/s13321-017-0209-z PMID: 29086119; PubMed Central PMCID: PMC5395521.

29. Law V, Knox C, Djoumbou Y, Jewison T, Guo AC, Liu Y, et al. DrugBank 4.0: shedding new light on drug metabolism. Nucleic Acids Res. 2014; 42(Database issue):D1091–7. https://doi.org/10.1093/nar/gkt1068 PMID: 24203711; PubMed Central PMCID: PMC3965102.

30. Southan C, Sharman JL, Benson HE, Faccenda E, Pawson AJ, Alexander SP, et al. The IUPHAR/BPS Guide to PHARMACOLOGY in 2016: towards curated quantitative interactions between 1300 protein targets and 6000 ligands. Nucleic Acids Res. 2016; 44(D1):D1054–68. https://doi.org/10.1093/nar/gkv1037 PMID: 26464438; PubMed Central PMCID: PMC4702778.

31. Kanehisa M, Furumichi M, Tanabe M, Sato Y, Morishima K. KEGG: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. 2017; 45(D1):D353–D61. https://doi.org/10.1093/nar/gkw1092 PMID: 27899662; PubMed Central PMCID: PMC5210567.

32. Gunther S, Kuhn M, Dunkel M, Campillos M, Senger C, Petsalaki E, et al. SuperTarget and Matador: resources for exploring drug-target relationships. Nucleic Acids Res. 2008; 36(Database issue):D919–22. https://doi.org/10.1093/nar/gkm862 PMID: 17942422; PubMed Central PMCID: PMC2238858.

33. Liu H, Sun J, Guan J, Zheng J, Zhou S. Improving compound-protein interaction prediction by building up highly credible negative samples. Bioinformatics. 2015; 31(12):i221–9. https://doi.org/10.1093/bioinformatics/btv256 PMID: 26072486; PubMed Central PMCID: PMC4765858.

34. Wang Y, Bryant SH, Cheng T, Wang J, Gindulyte A, Shoemaker BA, et al. PubChem BioAssay: 2017 update. Nucleic Acids Res. 2017; 45(D1):D955–D63. https://doi.org/10.1093/nar/gkw1118 PMID: 27899599; PubMed Central PMCID: PMC5210581.

35. Bento AP, Gaulton A, Hersey A, Bellis LJ, Chambers J, Davies M, et al. The ChEMBL bioactivity database: an update. Nucleic Acids Res. 2014; 42(Database issue):D1083–90. https://doi.org/10.1093/nar/gkt1031 PMID: 24214965; PubMed Central PMCID: PMC3965067.

36. Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol. 1981; 147 (1):195–7. PMID: 7265238.

37. Efron B. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. Journal of the American Statistical Association. 1983; 78(382):316–31. https://doi.org/10.1080/01621459.1983.10477973

38. You ZH, Chan KCC, Hu PW. Predicting Protein-Protein Interactions from Primary Protein Sequences Using a Novel Multi-Scale Local Feature Representation Scheme and the Random Forest. Plos One. 2015; 10(5). WOS:000354049700088.

39. Finn RD, Bateman A, Clements J, Coggill P, Eberhardt RY, Eddy SR, et al. Pfam: the protein families database. Nucleic Acids Res. 2014; 42(Database issue):D222–30. https://doi.org/10.1093/nar/gkt1223 PMID: 24288371; PubMed Central PMCID: PMC3965110.

40. Desaphy J, Bret G, Rognan D, Kellenberger E. sc-PDB: a 3D-database of ligandable binding sites—10 years on. Nucleic Acids Res. 2015; 43(Database issue):D399–404. https://doi.org/10.1093/nar/gku928 PMID: 25300483; PubMed Central PMCID: PMC4384012.

41. Benjamini Y, Hochberg Y. Controlling the False Discovery Rate—a Practical and Powerful Approach to Multiple Testing. J Roy Stat Soc B Met. 1995; 57(1):289–300. WOS:A1995QE45300017.

42. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, et al. UCSF Chimera—a visualization system for exploratory research and analysis. J Comput Chem. 2004; 25(13):1605–12. https://doi.org/10.1002/jcc.20084 PMID: 15264254.

43. Schultz LW, Clardy J. Chemical inducers of dimerization: the atomic structure of FKBP12-FK1012A-FKBP12. Bioorg Med Chem Lett. 1998; 8(1):1–6. PMID: 9871618.

44. Underwood KW, Parris KD, Federico E, Mosyak L, Czerwinski RM, Shane T, et al. Catalytically active MAP KAP kinase 2 structures in complex with staurosporine and ADP reveal differences with the autoinhibited enzyme. Structure. 2003; 11(6):627–36. PMID: 12791252.

45. van der Maaten L, Hinton G. Visualizing Data using t-SNE. J Mach Learn Res. 2008; 9:2579–605. WOS:000262637600007.

46. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Res. 2000; 28(1):235–42. https://doi.org/10.1093/nar/28.1.235 PMID: 10592235; PubMed Central PMCID: PMC102472.

47. Malhotra S, Karanicolas J. When Does Chemical Elaboration Induce a Ligand To Change Its Binding Mode? (vol 60, pg 128, 2017). Journal of Medicinal Chemistry. 2017; 60(13):5940–. WOS:000405764900046. https://doi.org/10.1021/acs.jmedchem.7b00868 PMID: 28653841

48. Boutet E, Lieberherr D, Tognolli M, Schneider M, Bansal P, Bridge AJ, et al. UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View. Methods Mol Biol. 2016; 1374:23–54. https://doi.org/10.1007/978-1-4939-3167-5_2 PMID: 26519399.

49. Niijima S, Shiraishi A, Okuno Y. Dissecting kinase profiling data to predict activity and understand cross-reactivity of kinase inhibitors. J Chem Inf Model. 2012; 52(4):901–12. https://doi.org/10.1021/ci200607f PMID: 22414491.

50. Landrum G, Kelley B, Tosco P, sriniker, gedeck, NadineSchneider, et al. rdkit/rdkit: 2018_03_1 (Q1 2018) Release. 2018. https://doi.org/10.5281/zenodo.1222070

51. Clevert D-A, Unterthiner T, Hochreiter S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). ArXiv e-prints [Internet]. 2015 November 01, 2015. Available from: https://ui.adsabs.harvard.edu/#abs/2015arXiv151107289C.

52. Chollet F. Keras. GitHub repository. 2015.

53. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Yee Whye T, Mike T, editors. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics; Proceedings of Machine Learning Research: PMLR; 2010. p. 249–56.

54. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. ArXiv e-prints [Internet]. 2014 December 1, 2014; 1412. Available from: http://adsabs.harvard.edu/abs/2014arXiv1412.6980K.

55. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J Mach Learn Res. 2014; 15:1929–58. WOS:000344638300002.

56. Gal Y, Ghahramani Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. ArXiv e-prints [Internet]. 2015 December 01, 2015. Available from: https://ui.adsabs.harvard.edu/#abs/2015arXiv151205287G.

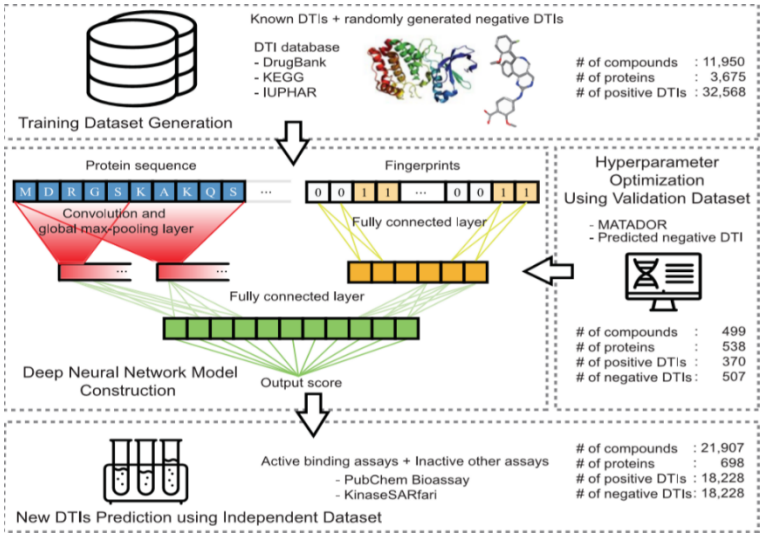
57. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv e-prints [Internet]. 2015 February 01, 2015. Available from: https://ui.adsabs.harvard.edu/#abs/2015arXiv150203167I.

58. Ng A. Sparse autoencoder. CS294A Lecture notes 2011; 72.

我选读的 paper 是《DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences》，它是由 Ingoo LeeID、Jongsoo Keum、Hojung Nam 三位教授于 2019 年 6 月 14 日在“PLoS Computational Biology”期刊上联合发表的。众所周知，“PLoS Computational Biology”是生物界著名的期刊，偏重的研究方向有生命科学、遗传学与生物信息学、神经生物学、生物信息学、计算生物学等，而这三位教授是来自韩国的 Gwangju 科技大学的电气工程与计算机科学学院，有着丰厚的研究实力。

首先介绍一下文章的摘要，文章主要讨论药物-靶标相互作用(DTIs)的识别。传统地，做相关的识别实验需要花费大量的人力物力，所以诞生了 silico-based DTI 的预测方法，这种方法节省了大量的资源。然而，在一些计算模型中，靠蛋白质描述符来进行预测的方法并不是很准确，所以这篇文章提出了一种基于深度学习的预测模型，这个模型可以捕捉参与 DTIs 的蛋白质的局部残余图像，从而提高预测的准确率。为了有效地训练模型，论文的作者使用了大量的数据，并在检测阶段使用了独立的数据集，因此，这个模型表现得比以前的模型都要好。同时，通过池化卷积的结果，这个模型也可为 DTIs 提供蛋白质结合位点的检测。总之，这个模型丰富了蛋白质序列的特征，在 DTIs 方面也有着较高的准确度。

接着，先对模型进行阐述，如图：



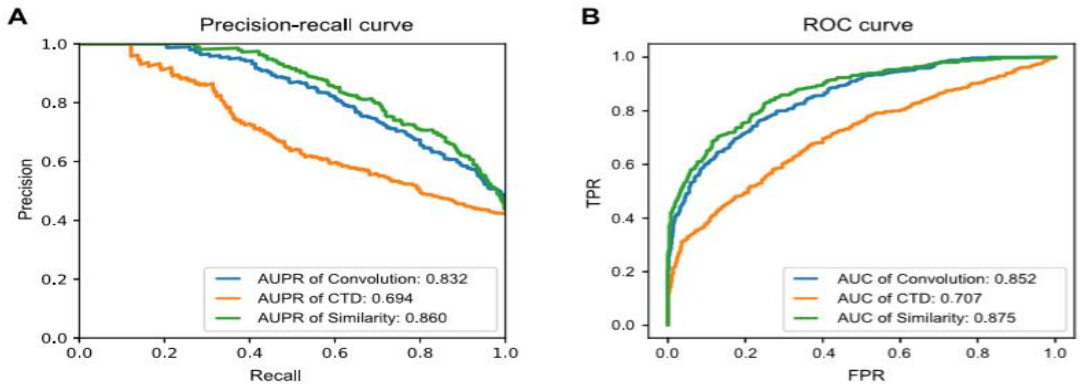
(1) 训练数据集的产生：这一块包括已知真实的 DTIs 和随机产生的负样本 DTIs（包含一定的噪声），真实样本来自 DrugBank 等数据库。在总的样本中，有 11950 个复合物，3675 个蛋白质，32568 个正样本。

(2) 训练过程：将蛋白质序列进行卷积和最大池化后的结果与指纹信息进行全连接后的结果，总的做一个全连接，输出一个分数。

(3) 优化：使用构建的外部验证数据集来优化超参数，其中有 499 个化合物，538 个的蛋白质，370 个正样本 DTIs，507 个负样本 DTIs。

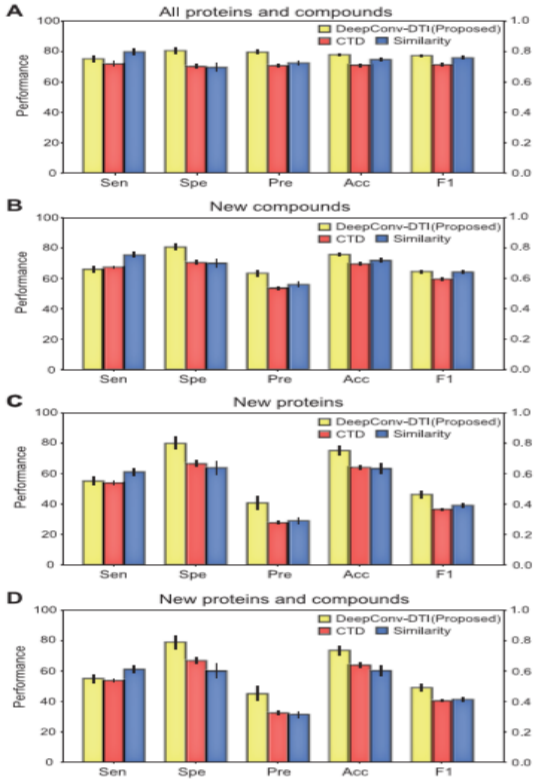
(4) 得出结论：使用独立的数据集来来进行预测，在生物测试中预测 DTIs，评估模型的性能。总结化合物、蛋白质和 DTIs 的数量。

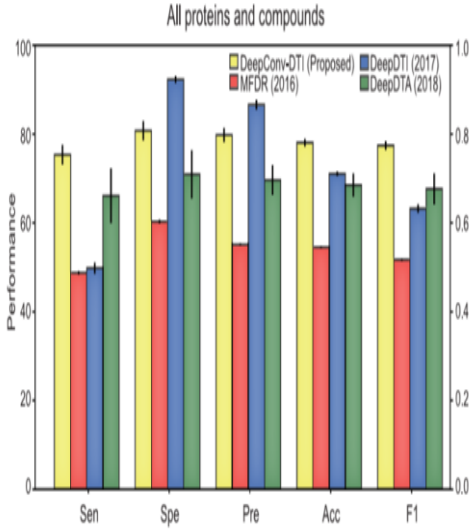
再者，对比论文中的模型的效果和其他模型方法的效果，如图：



从中可以看出，论文中给出了两个比较的方面，一个是 Precision-Recall 曲线图，一个是 ROC 特性曲线图。先来解释一下 A 曲线图，给出定义如下：TP(真正例)将正样本预测为正样本、FN(假反例)将正样本预测为负样本、FP(假正例)将负样本预测为正样本、TN(真反例)将负样本预测为负样本，那么 Precision 表征分类器的分类效果(查准效果)，它是在预测为正样本的实例中预测正确的频率值，其计算方法为 P= TP/ (TP+FP)；而 Recall 表征某个类的召回(查全)效果，它是在标签为正样本的实例中预测正确的频率，其计算方法为 R= TP/ (TP+FN)，图中的三条曲线分别是 Convolution(论文的方法)、CTD、Similarity 的 AUPR，而 AUPR 是该曲线下的面积。B 曲线图中，ROC 是 TPR 为 y 轴，FPR 为 x 轴做的曲线图,TPR 代表 True Positive Rate，计算方法为 TP/ (TP+FN)；FPR 为 False Positive Rate，计算方法为 FP/(FP+TN)，图中的三条曲线分别是 Convolution(论文的方法)、CTD、Similarity 的 AUC，而 AUC 是该曲线下的面积。可以说，在一定的前提下，论文的方法并不是最好的，然而，作者给出的解释是验证的规模太小，无法评估整体的性能，所以作者又做了后续的实验，评估了固定最大蛋白质长度对预测性能的影响，得出了他们的模型不会随着最大蛋白质长度而偏移的结论，在此基础上，作者又使用了其他蛋白质描述符来进一步训练、优化他们的模型。

论文的深度学习模型的优势不言而喻，作者给出了一些横向、纵向的对比结果：

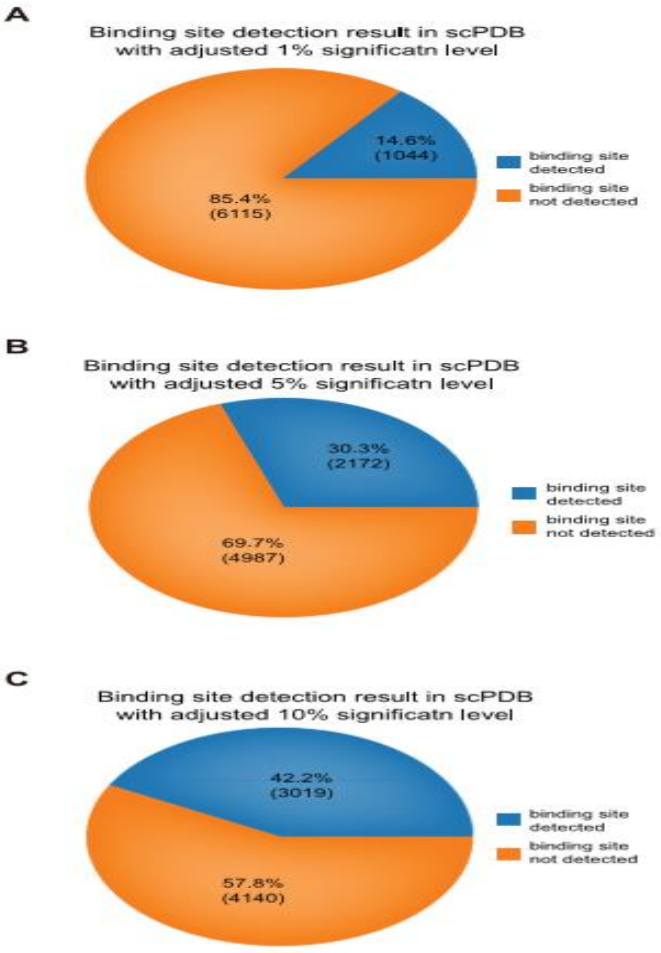




(1) 左图给出了该深度学习模型方法和其他方法在不同数据集上的测试结果。不难发现，作者从各种指标评估了测试结果，其中灵敏度 Sen=TP/(TP+FN)、特效度 Spe=TN/N=TN/(TN+FP)、精确度 Pre=TP/(TP+FP)、准确率 Acc=(TP+TN)/(TP+TN+FP+FN)、F1 等，可以直观地看出，论文的深度学习模型在不同数据集、不同评估指标上都超过了其他的预测方法。

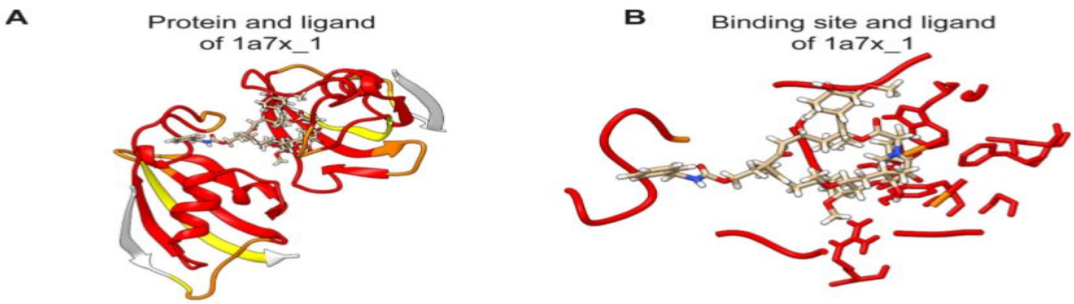
(2) 右图给出了该深度学习模型方法和以前的模型在同一数据集上的测试结果。也可以直观地看出，论文的深度学习模型在同一数据集上的不同评估指标上都超过了其他的预测方法。

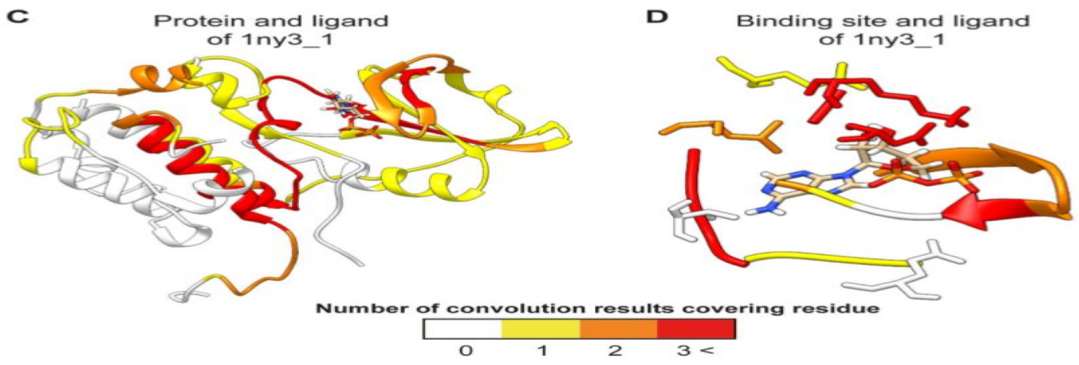
在搭建模型前，作者做了深入的分析，他们想对每一个过滤器，在做卷积之后进行了最大池化，而池化后的结果可以突出那些和局部残留图像匹配的区域。作者指出，尽管他们不能精确测出这些值是怎样影响 DTI 预测结果的，卷积、池化后的结果会通过更高的全连接层来影响预测性能，因此，如果他们的模型能够捕捉局部残留图像，那么模型会赋予重要的蛋白质区域很大的数值，比如一些真实的结合位点。基于以上思想，作者在设计模型时充分考虑了以上因素，并给出了结合位点检测的结果，如图：



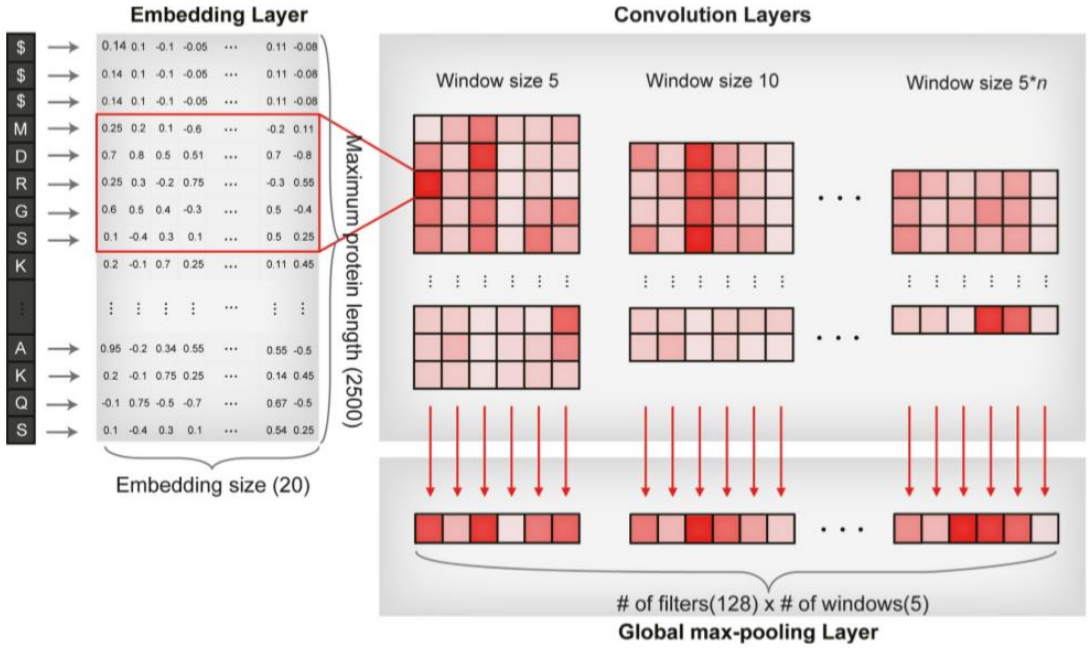
图中 significance level 一般指显著性水平，显著性水平是假设检验中的一个概念，是指当原假设为正确时人们却把它拒绝了的概率或风险。它是公认的小概率事件的概率值，必须在每一次统计检验之前确定，图中的 1%、5%、10%是指当作出接受原假设的决定时，其正确的可能性（概率）为 99%或 95%或 90%。

为了使结果更加直观，作者可视化了训练集中不同类别的共 1527 个蛋白质，其中含有 257 个 GPCRs，44 个核受体，304 个离子通道受体，604 个激酶和 318 个蛋白酶。他们用了 t-SNE 方法进行了降维和可视化，将高维特征映射到低维特征，如二维特征，从而使降维过程中的信息损失最小化。SNE 方法的思想是将欧几里得距离转换为条件概率来表达点与点之间的相似度，而 t-SNE 方法有如下的优点：使用对称版的 SNE，简化梯度公式；低维空间下，使用 t 分布替代高斯分布表达两点之间的相似度。从而避免拥挤问题和优化问题。可视化的结果如下：





接下来的工作，再对论文的模型进行一个更为细致的描述，总的可以分为下面几个部分：
(1)整体架构：通过 CNN 从蛋白质序列中提取局部残留图片，并通过全连接层获得药物指纹的表示。对药物层和蛋白质层进行处理后，将这些层连接起来，构建全连接层，从而产生输出。除输出层外的每一层都使用 ELU 函数来激活，输出层使用 sigmoid 函数来激活从而能够进行分类，整个网络都是通过 Keras 搭建的。
(2)蛋白质特征向量的构建：先上图，



第一步，将蛋白质序列通过隐藏层转换成固定大小的向量，过程中，序列的边界被填充，记为$进入隐藏层；第二步，对从隐藏层传过来的向量矩阵进行卷积操作；第三步，选用不同尺寸的窗口来进行最大池化；第四步，将池化后得到的向量进行拼接，组成蛋白质特征向量。
(3)蛋白质特征向量与药物指纹描述符的拼接：将药物指纹描述符进行全连接，可以得到一个代表它的向量，把这个向量与之前得到的蛋白质特征向量拼接后，再进行全连接操作，得到一个输出。
(4)计算损失和权重优化：损失函数采用交叉熵损失函数

$$J(W,b) = -\frac{1}{n}\sum_{i}^{n}[y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)]$$

同时为了防止过拟合，加入 L2-norm 正则项，最后损失函数变为如下：

$$J_{L2}(W,b) = J(W,b) + \lambda\sum_{l=1}^{L-1}\|W^l\|_2$$

使用的优化器是 Adam，也是常见的一种优化器。
(5)参数的选择：该深度学习模型中，在交叉验证期间调整超参数，例如影响性能的学习率和窗口大小等。超参数的选择要使模型表现出最佳的 AUPR，根据这一性能评估指标，可以得出正样本分类的准确性。当测量 AUPR 时，EER 可以给出最优阈值，公式为：

$$\mathrm{EER} = \underset{\theta}{\operatorname{argmin}}(|1 - \mathrm{recall}| - \gamma|1 - \mathrm{precision}|)$$

其中θ是分类阈值，γ是决定由精确度和召回率得出错误分类占比的常数，该模型中设置为 2。

当然，在论文的模型提出之前就有一些模型构造，分别是 SAE 构造和 DBN 构造，它们选取的损失函数和训练效果都和本文的模型不一样，这里不再赘述前人的工作。

我的评价：

（1） 在我看来，本文的文章结构特别诡异。在开头部分引出一些内容和结论，在最后部分对之前引出的内容进一步扩展并且对结论进行分析来保证合理性，导致我看的时候，对结论中的一些评估指标不理解，以及对结论不那么认同，缺少可信的指标，而好多帮助理解的内容是后续才有的。同时，文章特别强调他们的工作与别人的不同，而这种强调似乎在本文中很浓重，文中对比了好多其他的方法和模型，都是穿插在文章对某一方面的论述之间，颇有丢了西瓜捡芝麻的味道，给读者一种跑偏的感觉。总之，该论文的行文结构，我觉得并不是很好，它论述的方法对我来说并不适应。

（2） 再来评价一下论文的贡献。不可否认，该文章构建了一个简单、实用的训练模型，虽然模型中只有一些普通层，也不是很复杂，但它的效果却是意想不到的，作者在文中也提到，训练好的模型还能对蛋白质进行分类，这也是一个意外的收获。作者没有局限于生物知识，将计算机领域的深度学习知识与生物相结合，开创交叉学科的思维值得借鉴。同时，本文解决的问题也十分有意义，提高了 DTIs 的预测准确率，对生物技术的发展是有促进的，如今看来，深度学习这一旧的瓶子，仍能装入新的好酒，散发科学的魅力。