Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

System Biology

SG-DTA: Stacked Graph Drug-Target binding Affinity prediction

Cuong Vo [1,*], Tan Thanh Tran [1,*], Son Minh Nguyen [1,*] and Duc Hau Le [1,]

[1] Computational Biomedicine Department, VinBigData Institution, HaNoi, VietNam.

[*] To whom correspondence should be addressed. Equal contribution.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: The development of new pharmaceuticals is expensive, time-consuming, and sometimes fraught with safety concerns. By identifying new uses for existing authorized pharmaceuticals, medication repurposing can circumvent the costly and time-consuming process of drug development. To efficiently repurpose pharmaceuticals, it is necessary to understand which proteins are targeted by specific medications. Computational models that predict the intensity of interactions between novel drug–target pairings have the potential to accelerate drug repurposing For this goal, several models have been presented with various types of input, from string sequence to graph. Previous papers agreed that using graphs will increase the accuracy due to the detail feature extracted from the input. In this paper, we propose the Stacked Graph DTA (SG-DTA) that not only pushes the use of graph methods to the limit when using the stacking of both graph-representation and node-representation, but also renders the higher accuracy compared with the base model. Our result confirms concretely that the using graph embedding of input will be better than any other input's representation, specifically in DTA task.
Results: (Thu nam 24/6)
Availability: Code will be available at https://github.com/CuongVoThanh/SGDTI
Contact: v.cuongvt3@vinbigdata.org
Keyword: deep learning, drug-target binding affinity, graph neural network,

1 Introduction

As all of us know, the process of developing a new drug is time-consuming, this is about more than ten years from target identification to drug approval (Rifaioglu et al., 2019), and it costs billions US Dollar (Mullard, 2014). Clearly, we need to compress the time as every problem in biomedical becomes more complex with high intensity as in the moment, for example the COVID-19 pandemic.

Fortunately, the exploding digitization of data, which can be retrieved from a variety of sources ranging from clinical pharmacology to cheminformatics-driven databases, has reshaped biomedical discovery. This “big data” approach has been accelerating the process of new research as well as improving and filling the gap that previous study could not solve. Based on these advances, lots of methods have been studied in order to quicken the progress of drug development, such as, de novo drug design and drug repurpose, etc.

At the moment, several computational techniques for drug-protein affinity prediction have been developed such as molecular docking method, non-parametric machine learning algorithms such as Random Forest (Ballester and Mitchell, 2010; Li *et al.*, 2015; Shar *et al.*, 2016), etc. In addition, collaborative filtering is another approach, the affinity similarities between medicines and targets might be attributed to other sources, for example, the Kernel-based techniques employ kernels constructed from molecular descriptors of medicines and targets inside a regularized least squares regression (RLS) framework (Cichonska *et al.*, 2017,Cichonska *et al.*, 2018). The KronRLS model computes a pairwise kernel K from the Kronecker product of the drug-by-drug and protein-by-protein kernels to accelerate model training (Cichonska *et al.*, 2017,Cichonska *et al.*, 2018). However, these methods have some disadvantages on the input or features selection. Specifically, although the molecular docking technique is instructive, it requires information and concrete knowledge of the crystalline structure of proteins, which may not be available. Or in the Random Forest algorithm, features such as atom-pair co-occurrence oversimplified the description of the protein–ligand complex and resulted in the loss of information (Öztürk *et al.*, 2018).

Along with the explosion of data, deep learning step by step became a popular architecture recently, supported by high-capacity computing devices that challenged existing machine learning approaches. Deep learning methods are now being used extensively in many other research fields, including bioinformatics such as genomics studies (K.K.Leung *et al.*, 2014; Y.Xiong *et al.*, 2015) and quantitative-structure activity relationship (QSAR) studies in drug discovery, inspired by the remarkable success rate in image processing and speech recognition (Ma *et al.*, 2015). The main advantage of deep learning architectures is that they offer improved representations of raw data through non-linear modifications in each layer (LeCun *et al.*, 2015), making it easier to understand the underlying patterns in the data. A few research have previously been conducted utilizing Deep Neural Networks (DNN) for DTI binary class prediction utilizing multiple input models for proteins and medicines, in addition to some research that use stacked auto-encoders (Wang *et al.*, 2018) and deep-belief networks (Wen *et al.*, 2017). Similarly, stacked auto-encoder models incorporating Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) were used to describe chemical and genomic structures in real-valued vector forms (Gómez-Bombarelli *et al.*, 2018; Jastrzębski *et al.*, 2016). Deep learning algorithms have also been applied to protein–ligand interaction scores, with CNNs learning from the 3D structures of protein–ligand complexes being a frequent use (Gomes *et al.*, 2017; Ragoza *et al.*, 2017; Wallach *et al.*, 2015). This technique, however, is confined to known protein–ligand complex structures, with only 25 000 ligands described in PDB (W.Rose *et al.*, 2016).

Another method is to train neural networks on 1D representations of drug and protein sequences. For example, the DeepDTA model captures predictive patterns in data using 1D representations and layers of 1D convolutions (with pooling) (Öztürk *et al.*, 2018). The final convolution layers are then concatenated, processed through many hidden layers, and regressed with drug–target affinity ratings. The WideDTA model is an expansion of the DeepDTA model in which the drug and protein sequences are first summarized as higher-order characteristics (Öztürk *et al.*, 2019). Drugs, for example, are represented by the most common sub-structures (the Ligand Maximum Common Substructures (LMCS) (Woźniak *et al.*, 2018), whereas proteins are represented by the most conserved sub-sequences (the Protein Domain profiles or Motifs (PDM) from PROSITE (J.A.Sigrist *et al.*, 2009)). While WideDTA (Öztürk *et al.*, 2019) and DeepDTA (Öztürk *et al.*, 2018) develop a latent feature vector for each protein, the PADME model (Feng et al., 2018) represents proteins using fixed-rule descriptors and outperforms DeepDTA (Öztürk *et al.*, 2018).

However, the 1D representation has its own disadvantage when it might lead to the loss of information about ligand and protein structure. Besides, the traditional alignment between these two objects would not depict in the right way how molecules and protein are combined. Previously, Thin et al. (Nguyen *et al.*, 2020a) improved the 1D input of drug by embedding the SMILES structure into the graph, then learning the molecule's features using graph-embedding method. By using this way, the characteristic of the drug will not be lost when aligning the drug structure matrix with the learned feature of the embedded protein. The model might not be stable for further use because the 1D protein is kept for later fusion between them. Some other approaches such as the early fusion of the GEFA architecture (Nguyen *et al.*, 2020b), tried to fill the gaps of target representation by using the 2D protein contact maps then pushing the combination of embedding features and the maps to the layers of GCN for extracting the protein characteristic before aligning to the given molecules. Compared to the GraphDTA, it seems to have better intuition and the way proteins are represented, but it cannot maintain the accuracy of the task (on CI measure metric)

In this paper, we proposed the upgrade version by exploiting the better use of graph representation for increasing the accuracy of the previous model GraphDTA by Thin et al. (Nguyen *et al.*, 2020a), meanwhile maintaining the dynamic of any combination drug-protein. As compared with the base model, our architecture has already outperformed due to the combination method. Instead of the traditional alignment, we propose the extension of node-embedding to capture as much as possible the characteristics of the connection between molecule and protein target. The next section of this paper is ....

(Thu 7 19/6)

2 Methods

Here we propose the general setting of SG-DTA model to predict the binding affinity between the target protein P and drug compound D. Let y be the affinity score showing how the strength of the binding force, f maps the input vector P and D to y with parameter θ, which is formulated as the equation below:

$$y = f_\theta(D, P) \tag{1}$$

In section 3.1 and 3.2, we briefly present the mechanism to represent drug and protein features into the vector space. And the rest, we explain details the SG-DTA architecture model. Figure 1 depicts the entire the SG-DTA architecture as the function mapping f. (sua lai y doan nay)

2.1 Graph Neural Networks

Graph Neural Networks(GNNs) are one of the most powerful framework for representation learning of graphs. Graphs could both learn the elements of a system and their relationship through a set of nodes and edges. The features output of GNNs methods are aggregated by the neighbours of this element. It turns out to be dependencies across instances and beyond the independent and identically distributed assumption. Let say $G = (V, E)$ is represented as a graph in common where V is a set of N nodes with respect to node feature vectors $X_v \in \mathrm{I\!R}^{\mathrm{N x D}}$ (D is the dimension of features each node). Learning a representation vector of a node v or the entire graph G can be utilized the graph structure and node features X_v in GNNs. Following a neighborhood aggregation strategy, the information of a node is iteratively updated in several time. Assume that we choose l iterations to aggregate information, a node's representation captures the l-hop network neighborhood structural information. There are several ways to exploit and aggregate the relationship between each nodes. In this work, we briefly introduce four regular mechanisms in graph research as follows.

2.1.1 Graph Convolution Neural Network

Graph Convolution Neural Network (GCN) was first proposed in (Kipf and Welling, 2017), which endeavor to mimic the convolution network scheme in Computer Vision area. All of neighbors and itself are aggregated and combined as the formula below.

$$h_v^{(l+1)} = \sigma(W.Mean(h_u^{(l)}, \forall u \in N(v) \cup \{v\})). \tag{2}$$

where σ is any non-linear activation function, such as the $ReLU$ while W is the trainable weight matrix. Concretely, in the first step we initialize $h_v^{(0)} = X_v$. Then, k-hop layer-wise convolution operation is established as follows

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}). \tag{3}$$

The term $\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$ is completely equivalent to $Mean$ aggregation method (Eq. 3) between neighbours and itself where $\widetilde{A} = A + I_N$ is the adjacency matrix A of bi-directed graph G with self-connection identity matrix I_N. $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$ is the degree matrix of A. Finally, the encoding

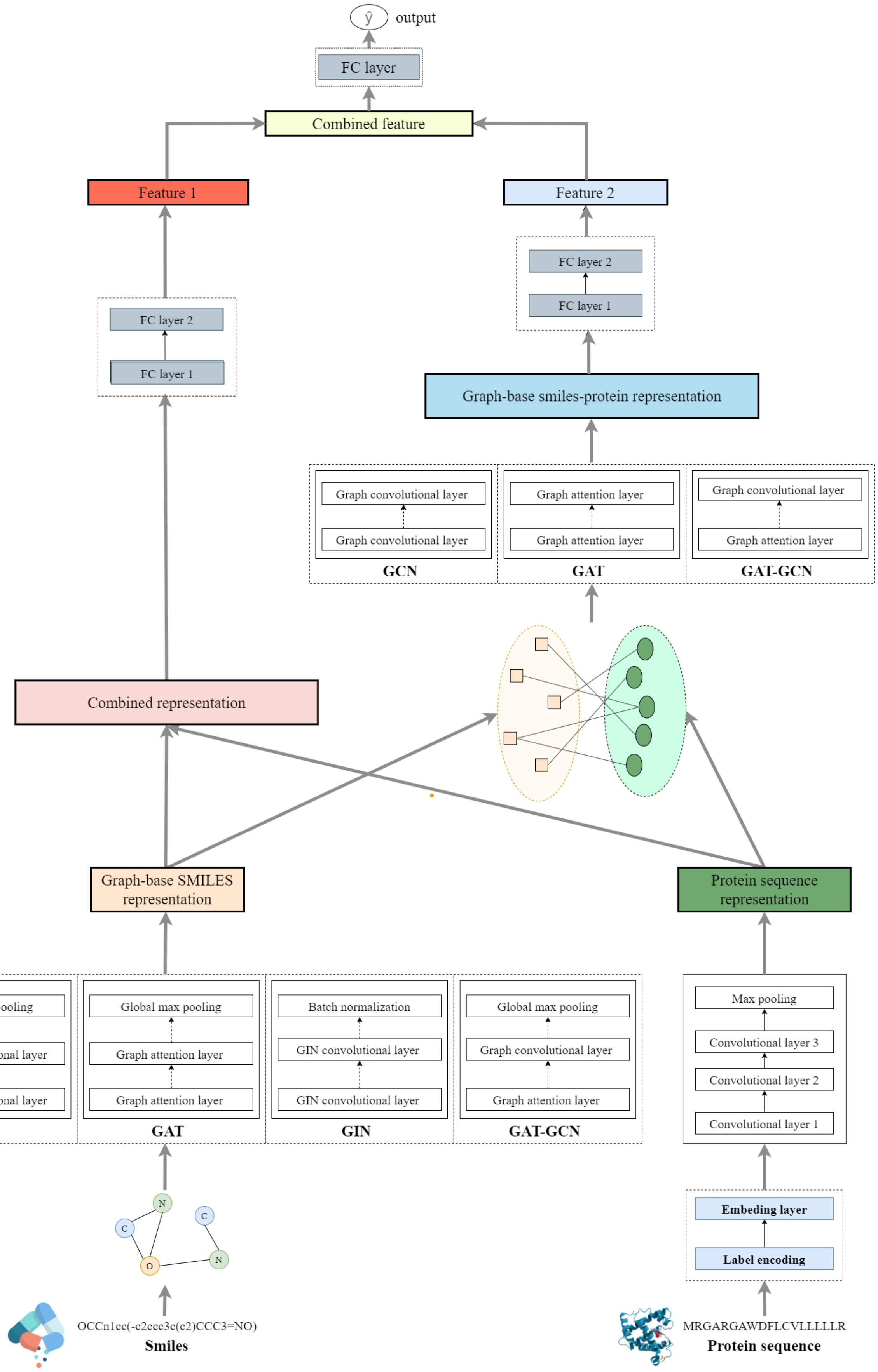

Fig. 1. A caption.

node embedding feature is the last layer-wise: $Z = H^L$ where L is the number of layer of model. However, the limitation of this approach is based on the assumption that equal importance of self-connections vs edges to neighboring nodes while it is not true for graph when there are relationships that are more important than others.

2.1.2 Graph Attention Network

Attention mechanisms (neural machine translation) have become the most successful standard in both Computer Vision and Nature Language Processing fields. When an attention mechanism is utilized to compute a representation of data (such as a image or a sequence), it is usually known as self-attention or intra-attention. Self-attention is not only useful for learning sentence representations (Lin *et al.*, 2017, Vaswani *et al.*, 2017) or learning image representations (Dosovitskiy *et al.*, 2020) but also for learning graph representation (Veličković *et al.*, 2018). The main factor that makes GAT more outstanding than previous approaches is that not all node's neighbors are equally important. This leads to a change of Eq. 2 a new formula.

$$h_v^{(l+1)} = \sigma\Big(\sum_{u \in N(v)} \alpha_{vu} W^{(l+1)} h_u^{(l)}\Big). \tag{4}$$

In term of attention α_{vu}, it concentrates on the important parts of the input data and fades out the rest. When explicitly defining $\alpha_{vu} = \frac{1}{|N(v)|}$, all neighbors $u \in N(v)$ are equally important to node v and we need a new implicitly definition for α_{uv} (such as a neural network) in order to specify arbitrary importance to different neighbors. A weight matrix $W \in \mathbb{R}^{\mathrm{D'xD}}$ is applied to every node in order to extract high-level features of node input features. Then a attention mechanism function a (such as a neural network) map from $\mathbb{R}^{\mathrm{D'xD'}}$ to $\mathbb{R}$ which shows the importance of node u to node v.

$$e_{vu} = a(W^{(l)} h_u^{(l-1)}, W^{(l)} h_v^{(l-1)}) \tag{5}$$

The attention coefficients e_{vu} gives exactly information about the first-order neighbors of v (including v). Then a soft-max normalization is applied to e_{vu} to convert the value to the probability of the importance between u and v.

$$\alpha_{vu} = softmax_u(e_{vu}) = \frac{exp(e_{vu})}{\sum_{k \in N(v)} exp(e_{vk})} \tag{6}$$

Finally, the final representation vector is the output vector after stacked k-hop layers $Z = H^L$ (Eq. 4).

Furthermore, multi-head attention (Vaswani *et al.*, 2017) could also be applied in order to stabilize the learning process of self-attention. To put it simply, multi-head attention computes K independent attention mechanisms (Eq. 4) and aggregates AGG all the features by concatenation or summation methods, as the following formula:

$$h_v^{(l+1)} = AGG_{k=1}^{K}\Big(\sigma\Big(\sum_{u \in N(v)} \alpha_{vu} W^{(l+1)} h_u^{(l)}\Big)\Big). \tag{7}$$

2.1.3 Graph Isomorphism Network

However, the above techniques has a drawback that relates to isomorphism property of graph. For example, suppose two graphs have distinct structure like a rectangle and a triangle. The disadvantage is shown clearly that the aggregation information of a node are indistinguishable because there are only two connections to that node. Though, they should be embedded to the different vectors in space. The earlier tactic to tackle this issue is hashing (Shervashidze *et al.*, 2011). This strategy names Weisfeiler-Lehman (WL) test which is the effective and computationally efficient to distinguish two topologically identical graphs. Recently, a noticeably more powerful algorithm is Graph Isomorphism Network (GIN) (Xu *et al.*, 2019). The advantage of GIN are low-dimensional node embedding and the parameters of the update function which can be learned for the downstream tasks. In order to maximize the discrimination among non-isomorphism graph, GIN utilize a multi-layer perceptron (MLP) to learn the difference and update the node features.

$$h_v^{(l+1)} = MLP^{(l+1)}\Big(\Big(1 + \epsilon^{(l+1)}.h_v^{(l)} + \sum_{u \in N(v)} h_u^{(l)}\Big)\Big). \tag{8}$$

where ϵ can be a learnable parameter or fixed scalar. GIN can model injective multi-set function. Until now, GIN is the most expressive GNN model.

2.2 Representation of Drug

The drug input of our model is in the Simplified Input Line Entry System format (SMILES) (Weininger, 1988) as a standard format of 1D drug's structure in much recent research (Öztürk *et al.*, 2018, Öztürk *et al.* 2019, Nguyen *et al.*, 2020a, Zhao *et al.*, 2019). We encode the molecular structure of the drug into vector representation of the drug via **RDKit** tool. To make it more concrete, the drug SMILES D is a sequence of atoms, atoms degrees, and ligands between each atom (e.g: CC(C)SC1=NN=C(C=C1)NN). The drug SMILES string is converted into a molecular graph by **RDKit**. In a single molecular graph, vertices are the atoms of a compound while edges are the bond connecting between atoms. The node feature after encoding comprises of five properties: atom elements, the degree of the atom in the molecule, which is the number of directly-bonded neighbors (atoms), the total number of H bound to the atom, the number of implicit H bound to the atom and whether the atom is aromatic. The adjacency list of edges is the undirected graph describing the bonds between two atoms. After pre-processing, the drug compound data can be referred as $G_D = (V_D, A_D)$, where $V_D \in \mathbb{R}^{\mathrm{s_d x n}}$ denotes matrix of node features, s_d is the size of D, n is the dimensions of compound properties and $A_D \in \mathbb{R}^{\mathrm{s_d x s_d}}$ denotes the bond adjacency list.

(viet doan global max pool)

2.3 Representation of Protein

On the other hand, the protein P is a sequence of amino acid. The sequence contains many ASCII characters and alphabet symbol associated with each amino acid type. We consider the primary structure of the protein as previous approaches applying deep learning to DTA (Öztürk *et al.*, 2018). One basic way to digitize the protein sequence is one-hot encoding like many recent Nature Language Processing field. There are many pros and cons to vectorize the protein sequence by this way. The advantage is the low time complexity when embedding residues into a vector space. Vector encoded $P \in \mathbb{R}^{\mathrm{s_p x m}}$ where s_p is the sequence length after cutting or padding and m is then embedded via an embedding layer as the formula

$$X_P = e_\gamma(P) \tag{9}$$

The output if the embedding operation is $X_P \in \mathbb{R}^{\mathrm{s_p x l_e}}$ where l_e is the embedding size of protein sequences. However, the drawback of this approach is that the output representation just covers the primary structure but not the tertiary structure. In DraphDTA (Jiang *et al.*, 2020), the tertiary structure gives the critical information in term of drug-target interaction. By constructing 2D structure of protein via contact map, we could gain more information about protein in the real world. Due to the lack of resource, we keep the constructing the graph representation of protein in the future work.

2.4 SG-DTA Architecture

(Chu nhat 20/6)

Table 1. Summary of the datasets

	Proteins	Compounds	Interactions
KIBA	229	2 111	118 254
Davis (K_d)	442	68	30 056

2.4.1 Drug-Target Bipartite Graph

(Chu nhat 20/6)

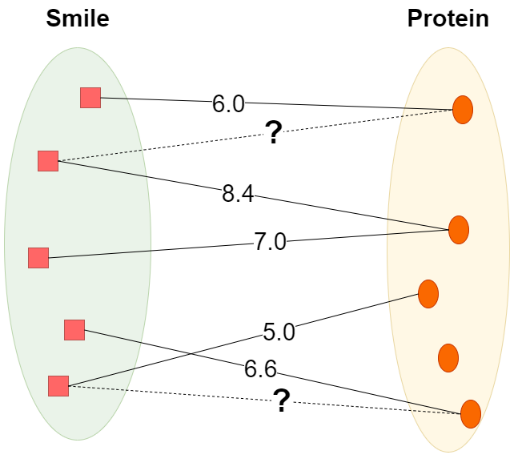

Fig. 2. An image of a leuleu

2.4.2 Anchor Model

(Chu nhat 20/6)

3 Experiment

3.1 Data Description

This paper reused the 2 popular datasets that have been leveraged by lots of previous working papers, specifically the Davis dataset (Mindy I Davis *et al.*, 2011) and KIBA dataset (Tang *et al.*, 2014). These two datasets are also known as the benchmark dataset for binding affinity prediction and evaluation. Besides, for the main purpose of taking the advantages from graph representation, SMILES and Protein Sequence will be used to build up the relevant graphs and used as the input for the training process. The Davis dataset includes selectivity assays for the kinase protein family and their inhibitors, as well as their affinities observed between them which are measured by the dissociation constant (Kd) values. It consists of 442 protein interactions and 68 ligand interactions. The KIBA dataset, on the other hand, was created using a method known as KIBA, which included kinase inhibitor bioactivities from several sources such as Ki, Kd, and IC50. By employing the statistical information included in KIBA scores, the consistency between Ki, Kd, and IC50 was optimized. Originally, the KIBA dataset had 467 targets and 52 498 medicines. The information of datasets that we employed in our study are summarized in Table 1.

Besides, instead of using the original dissociation constant (Kd) values in the Davis dataset, we use the log transformed version as in the study of He et al. (2017)

$$pK_d = \log_{10}\left(\frac{k_d}{1e9}\right) \qquad (10)$$

3.2 Parameter Setting

(Thu 7 19/6)

3.3 Evaluation Metrics

(Thu 7 19/6)

4 Results and Discussion

(Thu 5 24/6)

4.1 Results

(Thu 5 24/6)

4.2 Ablation study

(Thu 6 25/6)

5 Conclusion

(Thu 6 25/6)

Acknowledgements

(Thu 7 20/6)

References

Ballester, P. J. and Mitchell, J. B. O. (2010). A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics*, **26**, 1169–1175.

Cichonska, A. *et al.* (2017). Computational-experimental approach to drug-target interaction mapping: A case study on kinase inhibitors. *Plos computational biology*.

Cichonska, A. *et al.* (2018). Learning with multiple pairwise kernels for drug bioactivity prediction. *Bioinformatics*, **34**, i509–i518.

Dosovitskiy, A. *et al.* (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *Cornell University*.

Gomes, J. *et al.* (2017). Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity. *Cornell University*.

Gómez-Bombarelli, R. *et al.* (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.*, page 268–276.

J.A.Sigrist, C. *et al.* (2009). PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Research*, **38**, D161–D166.

Jastrzębski, S. *et al.* (2016). Learning to SMILE(S). *ICLR*.

Jiang, M. *et al.* (2020). Drug–target affinity prediction using graph neural network and contact maps. *Royal Society of Chemistry*.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *ICLR*.

K.K.Leung, M. *et al.* (2014). Deep learning of the tissue-regulated splicing code. *Bioinformatics*, **30**, i121–i129.

LeCun, Y. *et al.* (2015). Deep learning. *Nature*, **521**, 436–444.

Li, H. *et al.* (2015). Low-Quality Structural and Interaction Data Improves Binding Affinity Prediction via Random Forest. *Molecules*, **20**, 10947–10962.

Lin, Z. *et al.* (2017). A Structured Self-attentive Sentence Embedding. *Cornell University*.

Ma, J. *et al.* (2015). Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.*

Mindy I Davis, Jeremy P Hunt, S. H. *et al.* (2011). Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, **29**, 1046–1051.

Nguyen, T. *et al.* (2020a). GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, **37**, 1140–1147.

Nguyen, T. M. *et al.* (2020b). GEFA: Early Fusion Approach in Drug-Target Affinity Prediction. *Cornell University*.

Ragoza, M. *et al.* (2017). Protein-Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.*, page 942–957.

Shar, P. A. *et al.* (2016). Pred-binding: large-scale protein–ligand binding affinity prediction. *Journal of Enzyme Inhibition and Medicinal Chemistry*, pages 1443–1450.

Shervashidze, N. *et al.* (2011). Weisfeiler-lehman graph kernels. *Machine Learning Research 12*.

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

Vaswani, A. *et al.* (2017). Attention Is All You Need. *Cornell University*.

Veličković, P. *et al.* (2018). Graph Attention Networks. *ICLR*.

Wallach, I. *et al.* (2015). Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *Cornell University*.

Wang, L. *et al.* (2018). A Computational-Based Method for Predicting Drug–Target Interactions by Using Stacked Autoencoder Deep Neural Network. *Journal of Computational Biology*, **25**.

Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci. 1988, 28, 1, 31–36*.

Wen, M. *et al.* (2017). Deep-Learning-Based Drug–Target Interaction Prediction. *J. Proteome Res*, page 1401–1409.

Woźniak, M. *et al.* (2018). Linguistic measures of chemical diversity and the 'keywords' of molecular collections. *Scientific Reports*.

W.Rose, P. *et al.* (2016). The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Research*, **45**, D271–D281.

Xu, K. *et al.* (2019). How Powerful are Graph Neural Networks? *ICLR*.

Y.Xiong, H. *et al.* (2015). The human splicing code reveals new insights into the genetic determinants of disease. *Science*, **347**.

Zhao, Q. *et al.* (2019). AttentionDTA: prediction of drug–target binding affinity using attention model. *2019 IEEE International Conference on Bioinformatics and Biomedicine*, pages 64–69.

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

Öztürk, H. *et al.* (2019). WideDTA: prediction of drug-target binding affinity. *A PREPRINT*.