國立臺灣大學電機資訊學院電信工程研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

基於局部極值的圖卷積與列表損失之藥物與標靶的互動預測

Local Extrema Based Graph Convolution and Listwise Loss for Drug Target Interaction Prediction

Tanoj Ramesh Langore

指導教授: 林澤 博士

Advisor: Che Lin, Ph.D.

中華民國 112 年 2 月

February 2023

doi:10.6342/NTU202300617

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

基於局部極值的圖卷積與列表損失之藥物與標靶的互動預測

Local Extrema Based Graph Convolution and Listwise Loss for Drug Target Interaction Prediction

The undersigned, appointed by the Department / Institute of _Comunication Engineering_ on 13 (date) 02 (month) 2023 (year) have examined a Master's thesis entitled above presented by _Tanoj Ramesh Langore (R09942149) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

林澤

(指導教授 Advisor)

系主任/所長 Director:

Acknowledgement

At this moment in 2023, a phase of my life has come to an end. My journey as a student at National Taiwan University wraps up as I finish this work. Two and half years back, when I arrived in Taiwan with a heart full of hope of achieving new heights, my brain was full of fear and questions. It was the beginning of a life-changing journey that I knew would be full of learning and unstoppable growth. However, the despair of parting from my home country and family was also pouring down. But when I first met my advisor Professor Che Lin, all those doubts and fear vanished away. The warm welcome that I received in his lab, iDSSP, made me feel that I was being looked after at each moment. I am sincerely thankful to Professor Lin for his constant guidance and support. He has always been there for me whenever I was stuck on my project or in life. He taught me not only to excel in my research work but to complete every task, big or small, with perfection and dedication. Every discussion with him was knowledgeable and helped me grow a little more.

Also, the members of the iDSSP lab have been the best teammates one can ever find. The team has always helped me solve research-related problems. They have been a great support in completing this thesis. The seniors in the lab have always guided me whenever I needed them. Along with them, the juniors have been great companions too. I sincerely thank the group for maintaining a comfortable environment in the lab. A place where each

doi:10.6342/NTU202300617

one of us grew both as an individual and as a team. The place is full of people willing to teach each other and learn from each other as well.

Along with my lab mates, I would like to extend my gratitude to my friends who never stopped believing in me. They were the people who I knew I could reach whenever I needed a family in this foreign country.

Finally, I am thankful to my family, who, despite being far from me, were always supporting and looking after me. Their support and care have made tough times comparatively easier on me. I owe them everything that I have achieved today.

doi:10.6342/NTU202300617

摘要

預測藥物-靶標親和力 (DTA) 是藥物發現和設計的重要部分之一。研究人員提出了預測 DTA 的計算方法，以規避更昂貴的體內和體外測試。最新的方法採用深度網絡架構來獲取藥物分子和蛋白質一維序列的特徵。在本論文中，我們驗證了二維藥物表示比一維表示包含更多信息，有助於更好地預測 DTA。具體來說，藥物化合物可表示為圖，目標蛋白質表示為序列訊息。我們開發了一種新的基於圖神經網絡的預測模型，稱為 LE-DTA，其性能優於目前文獻所提出的方法。LE-DTA 應用局部極值卷積進行有效的特徵提取。它側重於節點嵌入圖的局部和全局極值。我們探討了所提出模型在三個不同基準數據集（即 Davis、KIBA 和 BindingDB）上的性能。我們提出的模型改進了已知模型的一致性指數 (CI) 和均方誤差 (MSE)。實驗結果顯示，我們所提出的 LE-DTA 在 Davis、KIBA 和 BindingDB 數據集上分別實現了 0.898、0.902、0.855 的 CI 與 0.210、0.120 和 0.464 的 MSE。這些結果在 Davis 資料集上得出與已知模型相當的結果，但在 KIBA 資料集中，CI 提高了 1.12%，MSE 降低了 7.7%。最後，在 BindingDB 資料集上，CI 比已知模型提高 0.35%，MSE 降低了 3.33%。我們的模型顯示出令人滿意的預測準確性，並顯著提高了藥物發現過程的效率。

關鍵字：深度網絡架構，藥物靶標親和力，局部極值，圖神經網絡

doi:10.6342/NTU202300617

Abstract

One of the essential parts of drug discovery and design is the prediction of drug-target affinity (DTA). Researchers have proposed computational approaches for predicting DTA to circumvent the more expensive in vivo and in vitro tests. More recent approaches employed deep network architectures to obtain the features from the drug molecules and protein 1D sequences. In this work, we demonstrated that 2D drug representation contains more information than 1D representation and helps predict DTA better. Specifically, the drug compounds are represented as graphs to extract this information. We developed a new graph-based prediction model, termed LE-DTA, that performed better than existing benchmark models. LE-DTA utilizes local extrema convolutions for effective feature extraction. It focuses on the local and global extrema of graphs for node embedding. We investigated the performances of the proposed model on three different benchmark datasets, i.e., Davis, KIBA, and BindingDB. Our proposed models have improved the Concordance Index (CI) and Mean Square Error (MSE) over existing benchmarks. Experiment results showed that the proposed LE-DTA achieved a CI of 0.898, 0.902, 0.855 and an MSE of 0.210, 0.120, and 0.464 on the Davis, KIBA, and BindingDB datasets, respectively. These results are in the range of the existing benchmarks for Davis, while it shows a 1.12% improvement in CI with a 7.7% reduction in MSE for KIBA. Finally, on BindingDB, the CI is 0.35% better than the baseline models, with an MSE reduction of 3.33%. Our models

doi:10.6342/NTU202300617

show satisfactory prediction accuracies and improve the efficiency of the drug discovery process.

Keywords: Deep network architecture, drug-target affinity, local extrema, graph neural network

doi:10.6342/NTU202300617

Contents

doi:10.6342/NTU202300617

doi:10.6342/NTU202300617

doi:10.6342/NTU202300617

List of Figures

doi:10.6342/NTU202300617

List of Tables

doi:10.6342/NTU202300617

Chapter 1

Introduction

Today's society is facing an unpredictable situation because of the outbreak of the COVID-19 pandemic. The disease has infected more than 650 million people, with a death toll of 6.6 million as of December 2022. While such a terrifying number locked people around the globe into their houses, the pharmacological industries worked day and night to find the cure for this disease. However, after a long struggle, an efficacy of only 36% has been achieved [4]. The present scenario suggests that the drug discovery regime still requires lots of development and innovation.

In Pharmacology, drugs are biological macro-molecules used to diagnose, treat, cure, and even prevent diseases. These drug molecules combine with the specific molecules present in our bodies known as drug targets. These drug targets are molecules in the human body, such as certain proteins and nucleic acids, which interact with the drug molecules, and have a pharmacodynamic function. The drug molecules treat or prevent disease by binding to a specific target and changing their function [5].

Drug Discovery is a complex and lengthy process that requires a combination of

doi:10.6342/NTU202300617

scientific knowledge, technological capabilities, and financial resources. Bringing a new drug to market can take several years and involve significant financial investment, making it a challenging and risky venture. The drug discovery process involves several steps [6]. It begins with identifying a target. This involves finding a molecule or biological process believed to be involved in the disease. The target can be a protein, enzyme, receptor, or any other type of molecule. This is the first step in the drug discovery process. The second step is compound screening. In this stage, researchers examine databases of prospective drug candidates to locate substances that bind to the target and produce the intended therapeutic result. High-throughput screens are frequently used in this method, which involves evaluating thousands of chemicals against the target. In the preclinical testing stage, preclinical testing is performed on a potential molecule to determine its safety and efficiency in animal models. This helps scientists decide whether to test the substance on humans in more detail. If a substance passes preclinical testing, the next step is to put it through human clinical trials and determine whether it is safe and effective for treating the disease. Clinical trials are carried out in stages, beginning with modest, early-stage trials and moving to larger, more thorough investigations. Regulatory organizations like the US Food and Drug Administration may allow the use of a substance if it is demonstrated to be safe and effective in clinical trials. A medicine may be promoted and made accessible to patients once it has received approval [7].

Finding new applications for current pharmaceuticals is a process known as drug repositioning or drug repurposing [8]. This can involve discovering new uses for a drug, such as treating a different illness or finding new patient populations that might profit from the therapy. Drug repurposing, which uses already-approved medications with known safety and efficacy profiles, can be a quicker and more affordable substitute for con-

doi:10.6342/NTU202300617

ventional drug discovery and development. This can hasten the development of novel medicines for diseases for which there are currently few effective options. There are several approaches to drug repurposing, such as rational drug repurposing, high-throughput screening, and machine learning. Rational drug repurposing is the process of finding new applications for a drug based on the biology of the target disease and the drug's known mechanism of action [9]. In high-throughput screening, a massive library of medicines is tested against a variety of targets to find those that have the desired therapeutic effect [10]. Machine learning uses algorithms to analyze data from various sources, such as clinical trials, electronic health records, and genomics data, to discover potential new uses for existing pharmaceuticals [11].

An accurate drug-target affinity (DTA) prediction is crucial in drug discovery and repurposing [12]. DTA stands for the degree of therapeutic interaction (strength of interaction), which can significantly affect the efficacy and safety of a medication candidate. Drugs with a high DTA for the disease state they are designed to treat are more likely to be successful. This is because medications with a high affinity for their intended target are more likely to exert therapeutic benefits. On the other hand, drugs with a low DTA for their intended target may not bind to the target as well and may be less successful in treating the disease. DTA can alter a drug candidate's safety profile and its effect on efficacy. The likelihood that a drug will bind specifically to its intended target is increased when the DTA for that target is high [13]. This can lower the possibility of off-target effects and adverse effects. Contrarily, medications with low DTAs for their intended targets may bind to unanticipated targets and have undesirable side effects. Researchers frequently test sizable libraries of possible drug candidates against a range of targets to find the most promising DTA in drug discovery. Laboratory tests or computational modeling methods,

doi:10.6342/NTU202300617

such as docking simulations, are frequently used in this procedure [14]. The strength of their interaction and the binding site on the target is predicted via molecular docking simulations using the known 3D structures of the medicines and targets. The affinity of a medicine for a target can be predicted via molecular docking, and any off-target effects can be noted. Molecular docking simulations are regarded as target-based approaches because they are based on the target's 3D structure and are used to forecast a drug's affinity for that target. The chemical structure and properties of the pharmaceuticals form the foundation of various computational approaches [15], such as machine learning algorithms and network-based approaches, which are used to predict the drug affinity for multiple targets. These approaches are known as drug-based methods [16]. Evaluating and handling a lot of data is one of the main advantages of utilizing AI for DTA prediction. For instance, data from high-throughput screens, which include evaluating thousands of potential drug candidates against a range of targets, can be analyzed using AI techniques. AI techniques can also be utilized to examine data from different sources, such as structural or chemical-genomic data, to increase the accuracy of DTA prediction. Making forecasts faster than with traditional approaches is another advantage of employing AI for DTA prediction. In the early stages of drug discovery, when time is of the essence and researchers must swiftly identify the most promising drug candidates, this can be crucial.

The initial implementation of AI techniques was cornered around machine-learning approaches. One such approach is Kronecker Regularized Least Square (KronRLS). It is based on the Kronecker product, a mathematical operation involving taking the outer product of two matrices and forming a larger matrix. KronRLS is an extension of the regularized least squares (RLS) algorithm, a linear regression algorithm used to predict continuous outcomes. The Kronecker product is used to incorporate additional informa-

doi:10.6342/NTU202300617

tion about drugs and targets into the model, which can improve the accuracy of the predictions [17]. However, this approach can handle only a small amount of data. DeepDTA [18] came up with the deep learning approach which can handle huge amounts of data and complex tasks. DeepDTA used 1D convolution on the one-dimensional representation of both protein and drug sequences to capture the predictive pattern in data. However, representing the drug as only a string results in a loss of structural information. GraphDTA [19] and SAG-DTA [20] used the drug as a graph and protein as a one-dimensional sequence to further leverage the structural information for drugs. GraphDTA used several techniques for a graph representation of drugs, such as graph convolution network (GCN) [21], graph isomorphism network (GIN) [22], graph attention network (GAT) [23], and a combination of these models. SAG-DTA introduced the self-attention graph pooling operation in the graph model (GCN), further improving the results. Both GraphDTA and SAG-DTA represented the protein structure as a one-dimensional sequence. Due to their superior performance in existing literature, we regard them as the baseline models and extend them to our proposed model.

This study proposes LE-DTA: local extrema convolution for drug target affinity prediction. This new neural network architecture predicts the drug target affinity and outperforms the state-of-the-art approaches. It utilizes the local extrema (LE) convolution of graphs when learning node embeddings for a graph representation of drugs but we represented proteins as a 1d representation because representing protein sequences as graphs for drug-target affinity prediction can be challenging due to the lack of a standard method, resulting in difficulties in comparing results across different studies. Furthermore, while 3D protein representation captures more information about the protein conformation, it may not account for important elements such as the dynamic nature of the protein or additional

doi:10.6342/NTU202300617

ligands, potentially limiting DTA prediction accuracy. Additionally, training a model using 3D representation can be computationally expensive and challenging for proteins with limited structural data, which may not provide enough diversity to properly train a DTA prediction model. In particular, LE-DTA focuses on TOP-K pooling. We compared LE-DTA against baseline models on three commonly used data sets for DTA: the Davis [24], KIBA [21], and BindingDB datasets [25]. In terms of concordance index (CI) and mean square error (MSE), the performance of our suggested model was the best.

The thesis is divided into a total of 6 Chapters, summarized in the following sections.

In Chapter 1, *Introduction*, The process of discovering new drugs, known as drug discovery, is briefly explained in this text along with its significance and the current state of the field. The advancements and innovations in drug discovery are then discussed. The concept of drug target affinity and its estimation using a technique called Deep-DTA are explained in detail, including the various currently available models.

In Chapter 2, *Data Types and Datasets*, firstly, the data types for protein and drug data are discussed, followed by an explanation of the affinity values and their calculations. Then the four datasets used in this study, BindingDB, BindingDB RTK, Davis, and KIBA, are introduced.

In Chapter 3, *Methods*, Deep Learning methods used in this work are described. Neural Network and the three different pooling methods used in our proposed model, namely, TOP-K, SAG, and ASAP pooling, are explained. Then, the ranking Loss and the baseline model design are elaborated. Finally, the proposed model LE-DTA and the evaluation models are described in the end.

In Chapter 4, *Experimental Settings and Results*, firstly, the experimental setting of

doi:10.6342/NTU202300617

the baseline models and the proposed models are discussed. This is followed by the results obtained for both the baseline models and our proposed models when tested on the mentioned datasets.

In Chapter 5, *Discussion*, the analysis of various pooling layers combined with LE-Convolution is discussed. Three different pooling methods mentioned in Chapter 3 are combined with LEConv, and the models are tested on three datasets to find the best pooling pair method to be paired. Then, we discuss how BindingDB improves the prediction over the BindingDB RTK dataset, along with the cross-dataset evaluation of LE-DTA. Then, the Limitations of our models are explained, followed by the current limitations of representing proteins as 2D or 3D, and finally scope for future research and improvement.

In Chapter 6, *Conclusion*, the complete summary of this work is provided. Starting from the architecture of our proposed models, the results obtained and the significance of this work are discussed.

doi:10.6342/NTU202300617

Chapter 2

Data Types and Datasets

In this work, we have used four different datasets. Our lab's previous work used a data set, and three more datasets were added to this work. We used two sets of information from the four data sets: a one-letter sequence of proteins and the Simplified Molecular Input Line Entry System (SMILES) code of drug molecules.

2.1 Data types and affinity value

2.1.1 The protein data

Complex macromolecules called proteins are necessary for the human body to operate appropriately. They are composed of chains of amino acids and compact chemical molecules with carboxyl groups (-COOH) and amine groups (-NH2). In proteins, there are 20 different amino acids, each with special chemical characteristics. Peptide bonds, covalent bonds produced between the carboxyl group of one amino acid and the amine group of another, hold the amino acids together in a protein. The function of a protein is

doi:10.6342/NTU202300617

determined by the genetic code, which also dictates the amino acid sequence in a protein. The order of amino acids and the chemical characteristics of the side chains of the individual amino acids combine to form the particular three-dimensional structure of proteins. This structure affects how the protein interacts with other molecules and can carry out its specialized functions in the body, making it crucial for the protein's ability to function. The N-terminus and C-terminus describe the ends of a protein chain. The protein chain's N-terminus, also called the amino terminus, has a free amine group, while the C-terminus, also called the carboxy terminus, has a free carboxyl group. A protein's sequence is represented left to right, from the N-terminus to the C-terminus. The backbone of a protein is made up of linked groups of carbon, nitrogen, and oxygen atoms that make up the protein's primary chain. The various amino acids' side chains are joined to this backbone. The ex-

Protein Name	one letter code	three letter code
G	Glycine	Gly
P	Proline	Pro
A	Alanine	Ala
V	Valine	Val
L	Leucine	Leu
I	Isoleucine	Ile
M	Methionine	Met
C	Cysteine	Cys
F	Phenylalanine	Phe
Y	Tyrosine	Tyr
W	Tryptophan	Trp
H	Histidine	His
K	Lysine	Lys
R	Arginine	Arg
Q	Glutamine	Gln
N	Asparagine	Asn
E	Glutamic Acid	Glu
D	Aspartic Acid	Asp
S	Serine	Ser
T	Threonine	Thr

Table 2.1: Amino acids with their one-letter and three-letter codes

doi:10.6342/NTU202300617

act arrangement and orientation of the amino acids' side chains determine the protein's overall three-dimensional structure.

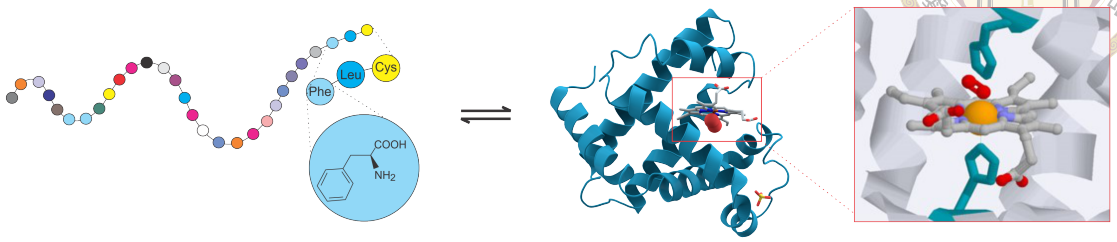


Figure 2.1: Proteins are long chains of amino acids.

2.1.2 The drug data

A standard for expressing the structure of chemical species using brief ASCII strings is the SMILES. These are generated by assigning a number to each atom in a molecule and then working through its order. As an illustration, the SMILES string for water (H_2O) would consist of "O" (for oxygen), two "H"s, and a "-" to denote the solitary bond. As another example, a "C" (for carbon) would be followed by two "H" atoms with a "-" between them, another "C" with two more "H" atoms and a "-" between them, and finally, an "O" with a "-" between it and one of the carbon atoms to indicate the bond. This makes up the SMILES string for ethanol (C_2H_6O), which is more complex. However, this may result in different SMILES for the same molecule because it can number multiple atoms. From a variety of canonicalization methods, specific SMILESs have been designated to avoid conflicts, particularly a SMILES representation for the same molecule. In our previous studies, we used SMILES strings of drug molecules. However, in this study, we use drug molecules as a graph. RD-Kit [26] is an open-source tool that accepts the SMILES string of the drug and produces a graph representation of it. It would generate two matrices, one adjacency matrix and the other the node feature matrix.

doi:10.6342/NTU202300617

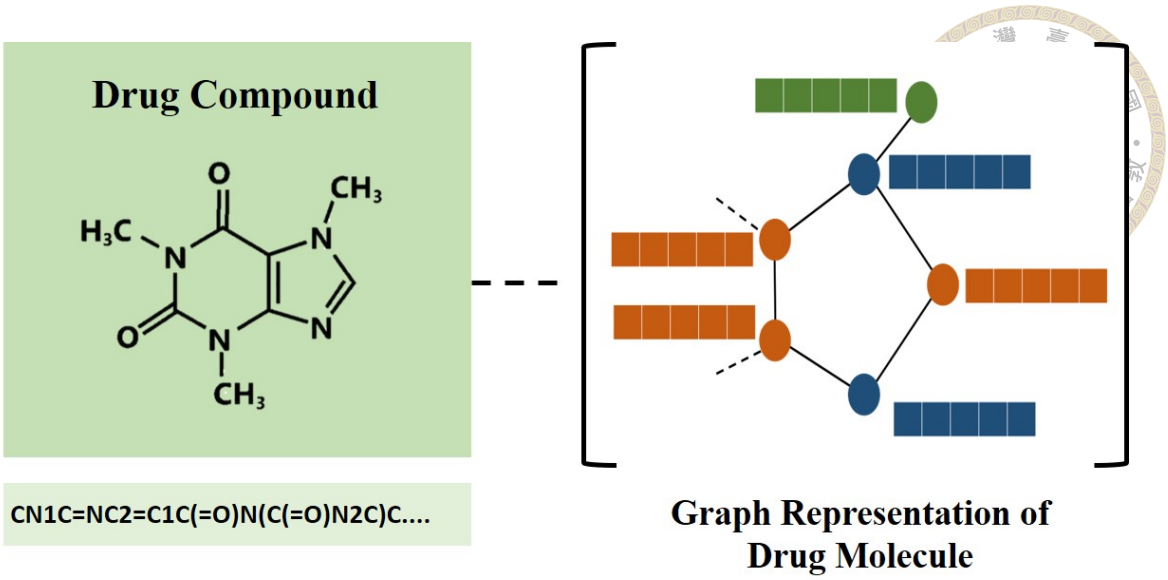


Figure 2.2: Drug molecule and its graph

2.1.3 Affinity value

The binding affinity value measures how well two molecules bind together. In our case Drug molecule binds with the protein molecule this reaction affinity is measurable. Generally, affinity is expressed as a dissociation constant (K_d). We can state the reaction between protein P and drug D as shown in Equation 2.1.

$$\mathrm{D} + \mathrm{P} \rightleftharpoons \mathrm{DP}. \tag{2.1}$$

Then, the association constant K_a can be defined as the concentration of drug-protein complex (DP complex) divided by the product of drug concentration and protein concentration, as shown in Equation 2.2, where K_d is the inverse of K_a.

$$K_a = \frac{[DP]}{[P][D]}. \tag{2.2}$$

The inhibition becomes competitive for a protein that is an enzyme and the drug acting as an enzyme inhibitor. In this case, the inhibition constant K_i, which is the same

doi:10.6342/NTU202300617

as K_d, is frequently used. Thus, K_i and K_d are constants that describe the same reaction.

The concentration of an inhibitor needed to inhibit a certain system to a specific percentage is known as the inhibitory concentration (IC). Generally, in vitro enzyme inhibition IC_{50} is preferred. This means 50% concentration is required to inhibit the reaction. The relation between IC_{50} and K_i for the case of competitive binding reaction to an enzyme following the Michaelis-Menten kinetics is given as Equation 2.3

$$IC_{50} = K_i \left(1 + \frac{[S]}{K_M}\right), \tag{2.3}$$

where [S] is the substrate concentration and $[K_M]$ is the enzyme's Michaelis constant. [27]

2.2 Datasets

Dataset	Proteins	Compounds	Binding Entities	Task Type
Davis	442	68	30,056	DTA (regression)
KIBA	229	2,111	118,254	DTA (regression)
BindingDB	4232	362,601	531,055	DTA (regression)
BindingDB RTK	41	19,990	24,337	DTA (regression)

Table 2.2: Datasets summary

The Table lists four different datasets: Davis, KIBA, BindingDB, and BindingDB RTK. We include the number of proteins and compounds and the binding entities it contains. The information in the Table can be used to compare the size and nature of the different datasets and understand how they are used for DTA prediction. All datasets are used for the regression task as shown in Table 2.2

doi:10.6342/NTU202300617

2.2.1 BindingDB and BindingDB RTK dataset

Some receptors may act like enzyme-bound receptors, even though all cell surface receptors are chemical structures that take in and relay environmental signals. The majority of them are Receptor Tyrosine Kinases (RTK). Cell development and differentiation would cease if RTKs failed. RTKs are often mutated in malignant tumors, making them targets for many chemotherapeutic treatments. These proteins have been located using BindingDB [25].

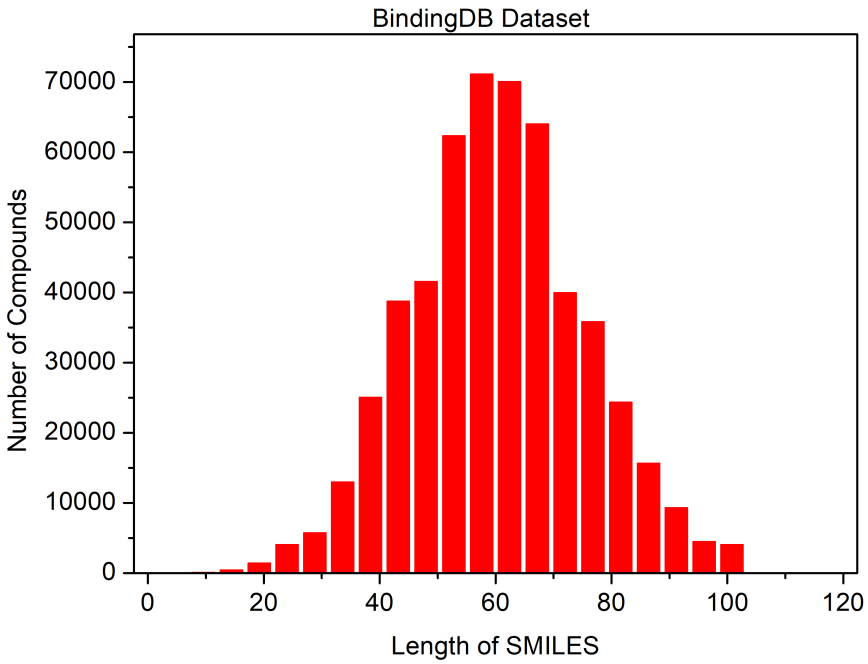


Figure 2.3: The histograms of the length of drug SMILES in the BindingDB dataset

An accessible database of binding affinities with an emphasis on protein-drug interactions is called BindingDB. There are $1,006,573$ small compounds and $2,331,208$ binding interaction data for $8,625$ protein targets. This dataset records various interaction entities, but we used IC50 as the target affinity value for this study.

doi:10.6342/NTU202300617

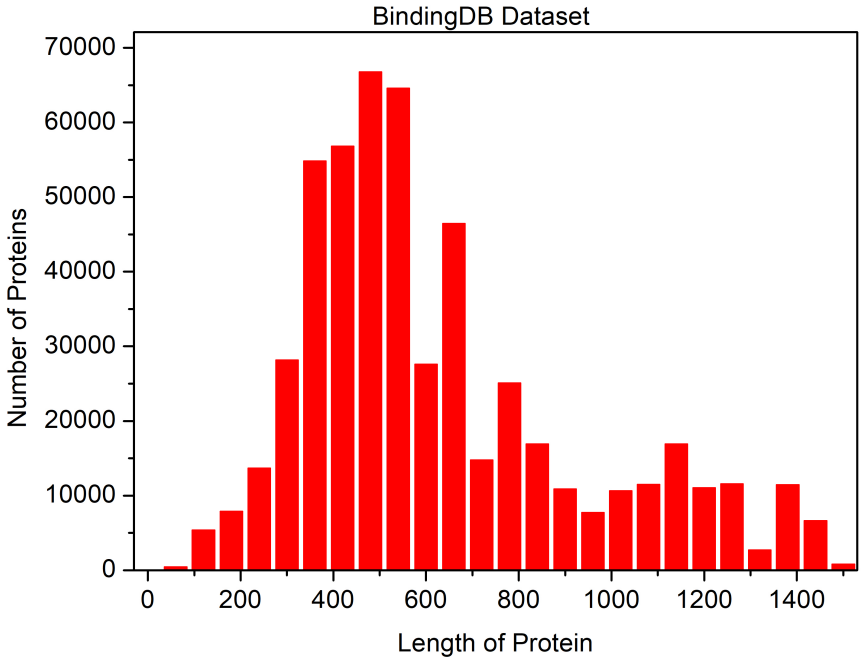


Figure 2.4: The histograms of the length of protein sequence in the BindingDB dataset

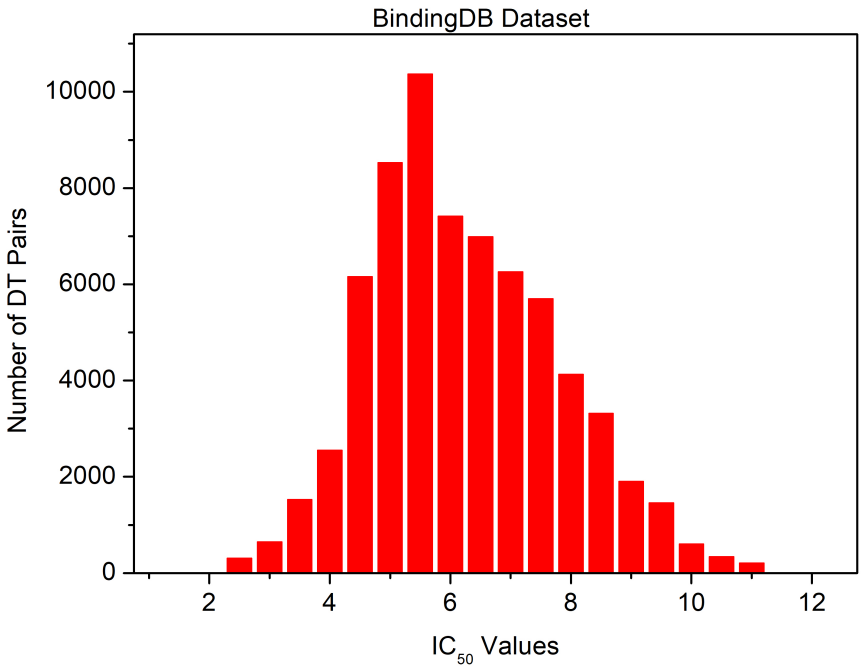


Figure 2.5: The histograms of the IC_{50} in the BindingDB dataset

doi:10.6342/NTU202300617

In this work, we used both BindingDB complete dataset and the BindingDB RTK dataset. The reason behind using the BindingDB dataset over the BindingDB RTK dataset will be discussed in Chapter 5 *Discussion*.

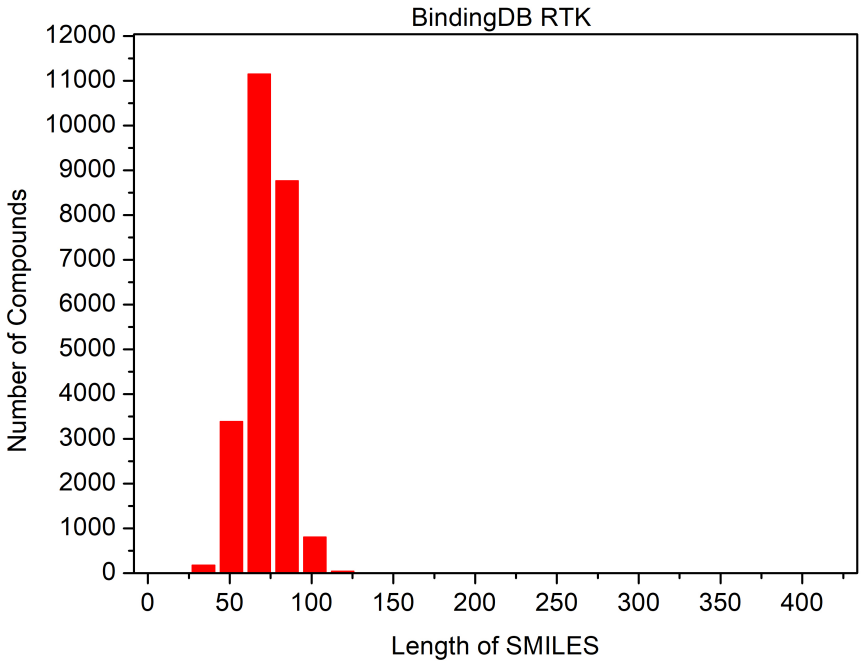


Figure 2.6: The histograms of the length of drug SMILES in the BindingDB RTK dataset

We extracted the protein sequence and the SMILES of the drug compound with their binding affinity score represented in terms of IC50 from the complete BindingDB dataset file that collected information until 2020 of version 5. We first removed invalid entries, such as numeric values as protein sequence and drug compound, and multiple entries of the same drug target pair. We took the mean of all IC50 values. We also replaced the original IC50 values with logarithmic values and considered the affinity values from 2 to 11. All this preprocessing reduced the dataset size to $531,055$ affinities with 4232 unique protein IDs and $362,601$ small compounds. We used these affinities as a complete BindingDB dataset. Since the longest SMILES for chemicals in the BindingDB dataset is 100, all drugs are less than or equal to 100 characters long, as shown in Figure 2.3. At

doi:10.6342/NTU202300617

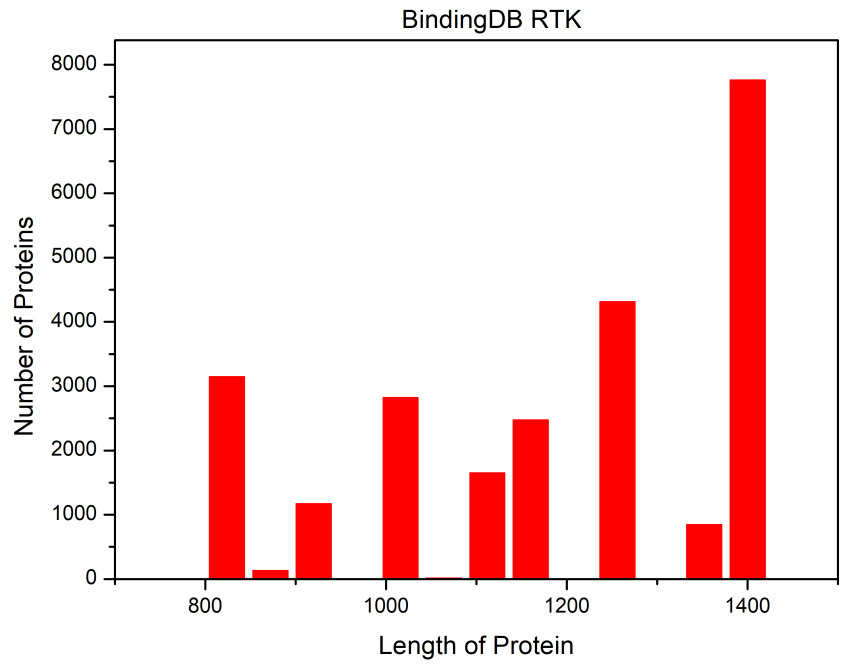


Figure 2.7: The histograms of the length of protein sequence in the BindingDB RTK dataset

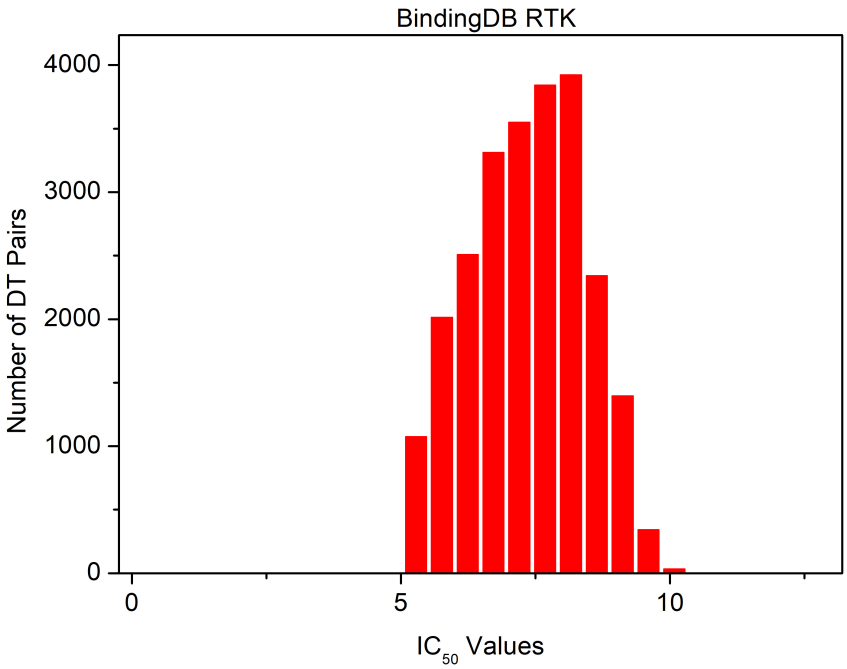


Figure 2.8: The histograms of the IC_{50} in the BindingDB RTK dataset

doi:10.6342/NTU202300617

the same time, the average length of drug SMILES sequences is 58. A protein sequence has a maximum length of 1484characters, as shown in Figure 2.4. Like the drugs, most proteins have a length of less than 1000, so we set it as a maximum input limit. They have an average length of 600 characters. Distribution of IC_{50} scores in Figure 2.5 shows that most DT pairs are in the range of 2 to 11. We chose the maximum number of input sequences so that it includes at least 86% of proteins.

For the BindingDB RTK dataset, we selected the interaction related to the RTK class proteins of Homo sapiens from the complete BindingDB dataset. The SMILES strings in the source were not in their canonical forms. Therefore, we transformed them into those forms by retrieving the canonical versions from PubChem by comparing the CIDs. By the time the preprocessing was complete, we had obtained 24337 BindingDB IC50 samples with 41 distinct proteins and 19990 unique drug molecules. Since the longest SMILES for chemicals in the RTK dataset is 140, most drugs are less than or equal to 100 characters long, as shown in Figure 2.6. A protein sequence has a maximum length of 1390 characters, as shown in Figure 2.7. Distribution of IC_{50} scores in Figure 2.8 shows that more than 99% of DT pairs are in the range of 5 to 10. For BindingDB RTK and BindingDB dataset, We divided the dataset into three pieces, with one serving as the independent test set, to create a generalized model. The partition ratio for the training, validation and test sets was 60:20:20.

2.2.2 Davis dataset

The Davis dataset [24] includes selectivity assays of the relevant inhibitors and members of the kinase protein family, together with information on their corresponding dissociation constants (Kd). It consists of 442 proteins covering more than 80% of the human

doi:10.6342/NTU202300617

catalytic protein kinome and 68 ligands. There are 30,056 interactions in the dataset.

In [17], they used K_d values directly as a binding affinity value, but in this work, we took a log of actual binding affinity values, pK_d similar to [18] as shown in Equation 2.4.

$$pK_d = -log_{10}\left(\frac{K_d}{10^9}\right). \tag{2.4}$$

These pairs' extremely low binding affinities (Kd > 10,000 nM because they are negative

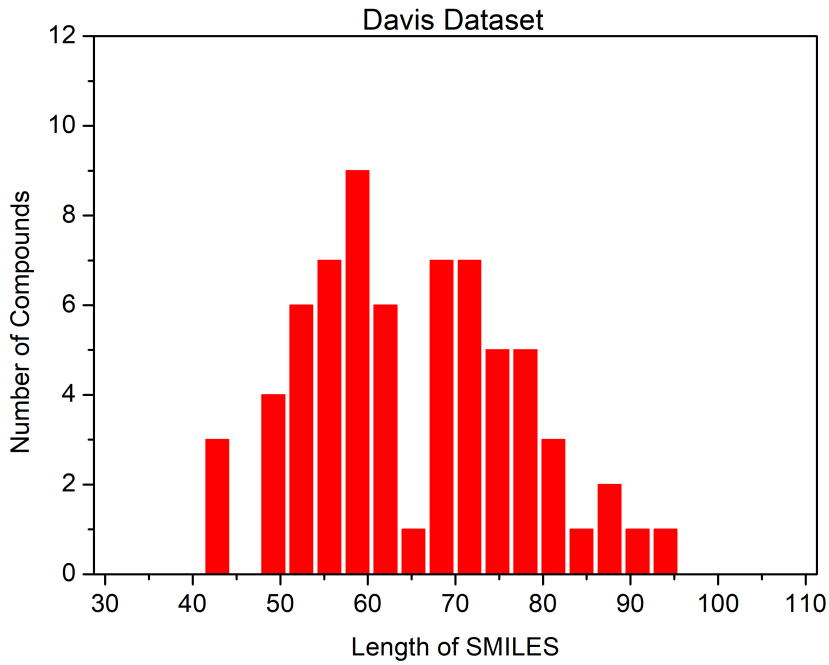


Figure 2.9: The histograms of the length of drug SMILES in the Davis dataset

values) are exempted from the dataset. The longest SMILES for chemicals in the Davis dataset is 103, while the average length is 64. A protein sequence has a maximum size of 2549 characters and an average length of 788. Since the longest SMILES for chemicals in the Davis dataset is 103, most drugs are less than or equal to 100 characters long, as shown in Figure 2.9. At the same time, the average length of drug SMILES is 64. A protein sequence has a maximum length of 2549 characters, as shown in Figure 2.10. Like the drugs, most proteins have a length of less than 1000, so we set it as a maximum input limit.

doi:10.6342/NTU202300617

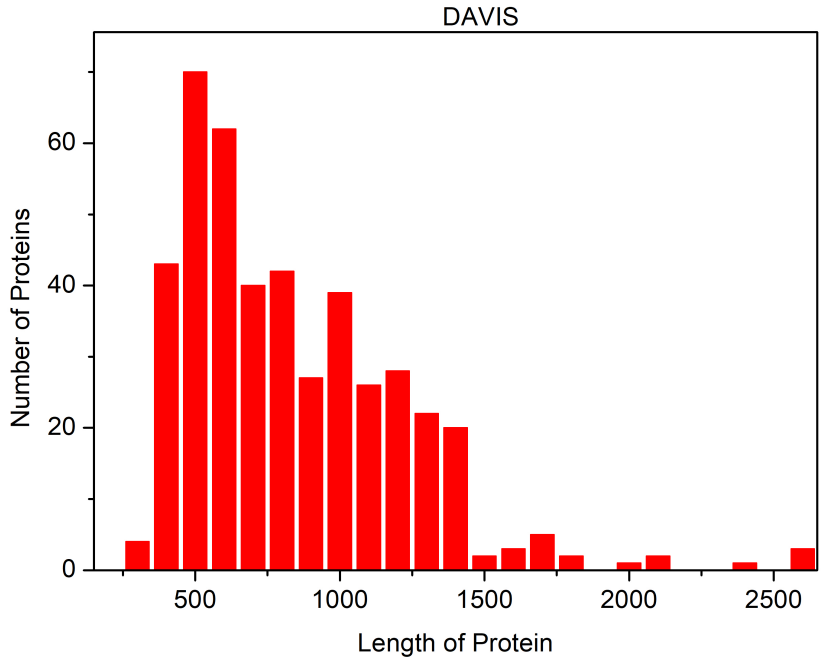


Figure 2.10: The histograms of the length of protein sequence in the Davis dataset

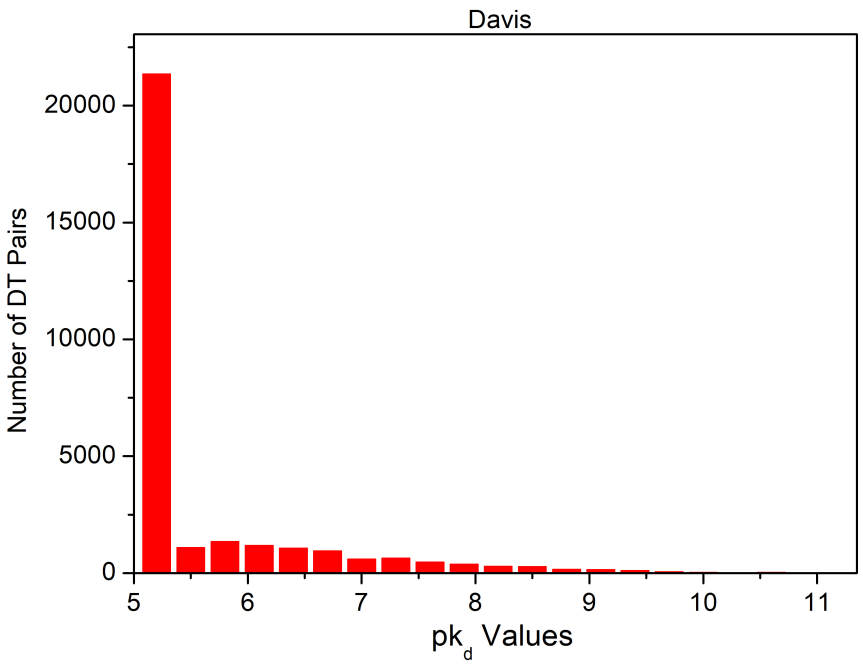
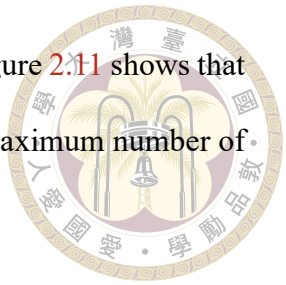


Figure 2.11: The histograms of the pK_d values in the Davis dataset

doi:10.6342/NTU202300617



An average length of 788 characters. Distribution of pK_d scores in Figure 2.11 shows that more than 75% of DT pairs are in the range of 5 to 6. we chose the maximum number of input sequences so that it includes at least 85% of proteins.

2.2.3 KIBA dataset

The KIBA dataset [28] originally consisted of 467 targets and 52498 drugs, but it is filtered to 229 proteins and 2111 drugs to guarantee there are at least ten interactions between them. This dataset uses the KIBA (kinase inhibitor bioactivity) score to represent the interaction by combining K_i, K_d, and IC_{50} to optimize the consistency between them. If K_i and IC_{50} are available, Equation (2.5) will be used; if K_d and IC_{50} are available, Equation (2.6) will be used; if all scores are available then average of Equation (2.5) and Equation (2.6) will be used.

$$\hat{K}_i = \frac{IC_{50}}{1 + L_i \left(\frac{IC_{50}}{K_i}\right)}, \tag{2.5}$$

$$\hat{K}_d = \frac{IC_{50}}{1 + L_d \left(\frac{IC_{50}}{K_d}\right)}, \tag{2.6}$$

$$KIBA = \begin{cases} \hat{K}_i, & \text{if } IC_{50} \text{ and } K_i \text{ are available} \\ \hat{K}_d, & \text{if } IC_{50} \text{ and } K_d \text{ are available} \\ \left(\frac{\hat{K}_i + \hat{K}_d}{2}\right), & \text{if } IC_{50}, K_d \text{ and } K_i \text{ are available} \end{cases} \tag{2.7}$$

where L_i and L_d are the variables used to calculate the IC_{50} weights in the K_d and K_i model-based corrections. We follow the adjusted regression target values for binary classification as suggested in the original paper [28].

The longest SMILES for chemicals in the KIBA dataset is 590, but the length of most of the drugs is 100 characters, as shown in Figure 2.12. A protein sequence has a maximum

doi:10.6342/NTU202300617

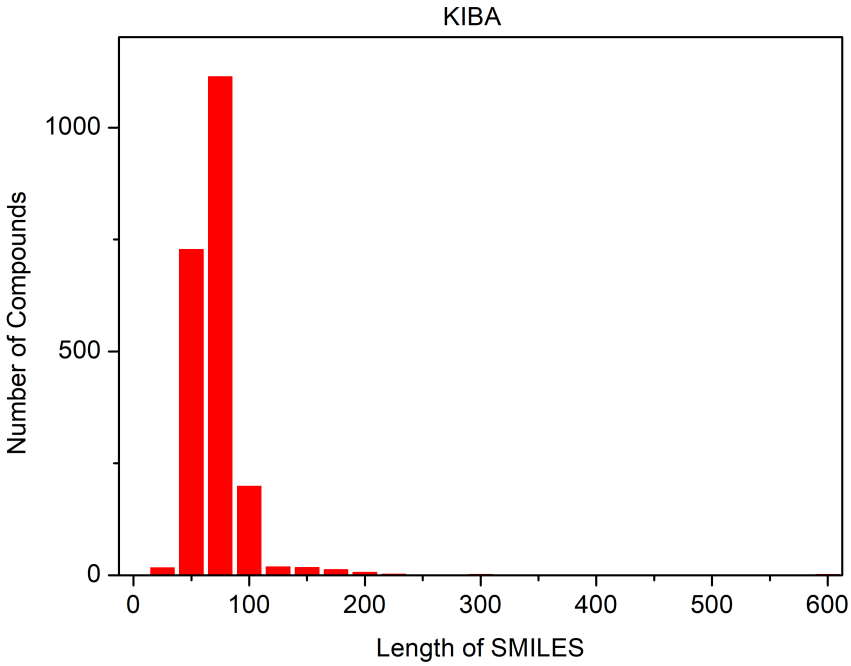


Figure 2.12: The histograms of the length of drug SMILES in the KIBA dataset

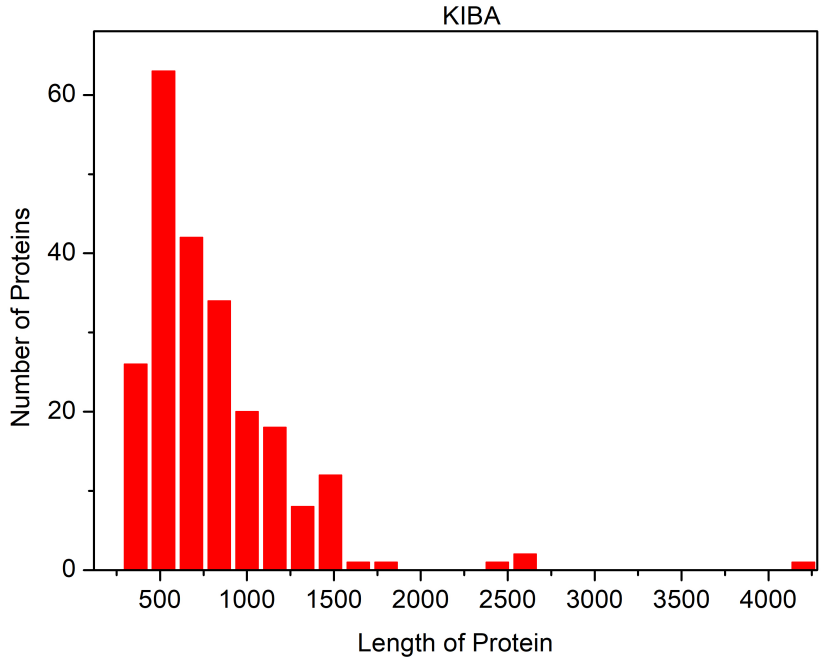


Figure 2.13: The histograms of the length of protein sequence in the KIBA dataset

doi:10.6342/NTU202300617

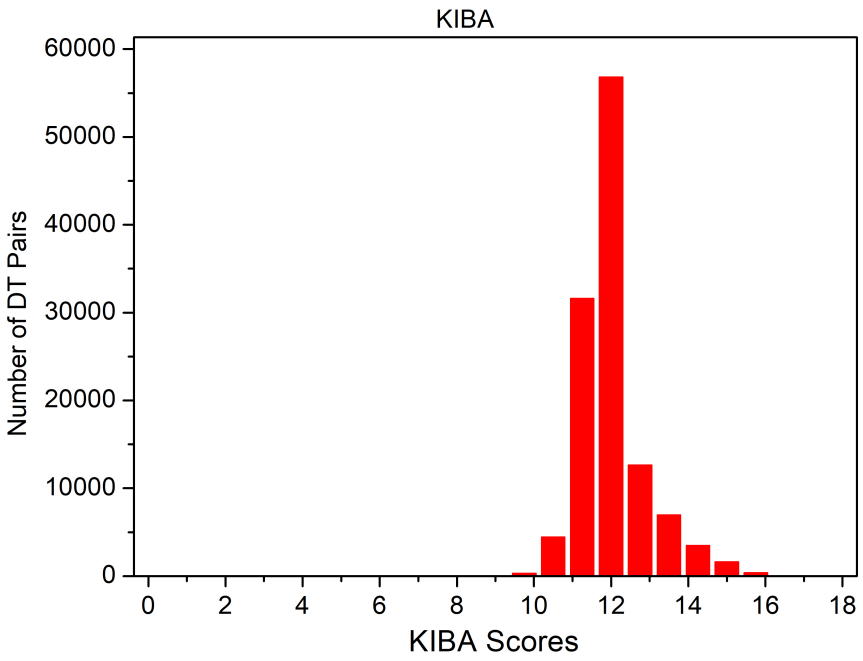


Figure 2.14: The histograms of the KIBA scores in the KIBA dataset

length of 4128 characters, as shown in Figure 2.13. Like the drugs, most proteins have a length of less than 1000, so we set it as a maximum input limit. They have an average length of 728 characters. The distribution of KIBA scores in Figure 2.14 shows that more than 85% of DT pairs are in the range of 11 to 14. We chose the maximum number of input sequences to include at least 85% of proteins.

doi:10.6342/NTU202300617

Chapter 3

Methods

3.1 Deep learning

3.1.1 Feedforward neural networks

In the year 1943, two researchers from the University of Illinois and the University of Chicago published a research paper that showed how we could reduce complex problems like how our brain produces complex problems to binary logic [29]. Later in 1958, Frank Rosenblatt developed a perceptron that tells us how a computer might use neural networks to recognize scenarios and draw conclusions; despite this remarkable contribution, perceptron could not handle nonlinear problems [30]. In 1982 John Hopfield came up with Hopfield Net, cornered around recurrent neural networks [31]. Hinton used the back-propagation algorithm to train a deeply structured artificial network in 1986, replacing the conventional simple single-layer structure perceptron [32]. There are different types of neural networks, but we will focus on simple feedforward neural networks. It is defined with three different types of layers and mathematical equations for the computation of the

doi:10.6342/NTU202300617

output at each layer:

Input layer: This layer receives the input data, x, and passes it to the hidden layer. It consists of units that represent the features of the input data.

Hidden layer: This layer processes the input data and passes it to the output layer. It consists of units that use weighted connections and activation functions to transform the input data. The output of the hidden layer, a, is computed as: $\mathbf{a} = f(W\mathbf{x} + \mathbf{b})$, where W is the weights matrix connecting the input layer to the hidden layer, $\mathbf{b}$ is the bias term for the hidden layer, and f is the activation function.

Output layer: This layer produces the model's predictions or decisions based on the input data. It consists of units that represent the possible outputs of the model. The output of the output layer, y, is computed as: $\mathbf{y} = g(V\mathbf{a} + \mathbf{c})$ Where V is the matrix of weights connecting the hidden layer to the output layer, $\mathbf{c}$ is the bias term for the output layer, and g is the activation function.

3.1.2 Convolution neural networks (CNNs)

CNNs are the most widely used deep learning architecture today. Their state-of-the-art performance in image recognition tasks like categorization and detection is where they receive the most recognition. LeNet [33], AlexNet [34], ResNet [35], and GoogleNet [36] are among the famous CNN models.

Multilayer perceptron uses every pixel, and each node in the layer is connected with each node of the previous and next layer, making it a very dense network. This network gets very complex for large images, which could be prone to overfitting. In comparison, CNN uses the sliding window and moves it on the image (convolution) from the top left to

doi:10.6342/NTU202300617

the bottom right. The size of the window and the sliding steps (stride) find the patterns in the image. This image pattern improves in each CNN layer to find very complex patterns. The pooling layers follow these CNN layers. The pooling operation uses a feature map, combines the features of the nearby region into a single feature, and gives better generalization. This operation helps to reduce the size by selecting an essential feature from the image. The dropout layer prevents overfitting and provides better generalization.

3.1.3 Graph neural networks (GNNs)

A neural network called a graph neural network (GNN) is created to process data represented as a graph. A graph is made up of a number of nodes and a number of connecting edges. In a graph, each node stands for a specific object or entity, and the edges signify connections or interactions between the nodes. When processing data that contains an innate graph structure, such as when predicting drug-target interactions or examining social networks, GNNs are especially well suited for the job. In the fields of chemistry, biology, social science, and many others, they can be used to examine data with graph structure. Similar to other kinds of neural networks, GNNs are often made up of several layers of connected nodes. Each layer receives input from the one below it, processes it, and then sends the processed data to the layer above it. A prediction or classification based on the input data is frequently the GNN's output. Due to their effectiveness in processing and analyzing huge, complicated, and heterogeneous datasets that are difficult to express as a straightforward matrix or vector, GNNs have become more and more well-liked in recent years. By making it possible to study data that was previously too complicated or challenging to evaluate using conventional approaches, they have the potential to revolutionize numerous fields.

doi:10.6342/NTU202300617

3.1.3.1 Graph convolution network (GCN)

The graph convolution operation is similar to the convolution operation in CNN. In CNN, we multiply input with the filters or kernels. Sliding kernels over the image allow CNN to learn features from neighboring cells. Similarly, in GCN inspecting neighboring nodes allows the model to learn.

The regular neural network forward propagation function calculates the feature representation of the next hidden layer by analyzing weights, feature representation, and bias for the current layer. Additionally, there is a nonlinear activation function. Graph convolutional network will add an adjacency matrix to the equation. We offer our graph to the network as a node feature matrix and adjacency matrix. After the data normalization, GCN performs some aggregation amongst nearby nodes, such as taking an average. This procedure can be compared to the transmission of a message, in which each layer of our GCN transfers an aggregate of neighboring nodes, one ”hop” to the subsequent node. We can convolve each node’s ”fourth-order” neighborhood if we have a four-layer GCN. This allows us to embed the social structure of the graph by passing a message to each node’s neighbor four ”hops” away. The GCN layer is formulated as [21]:

$$\mathbf{x}^{l+1} = \sigma(D^{-1/2}AD^{-1/2}\mathbf{x}^{l}), \tag{3.1}$$

where σ is the activation function (e.g., *tanh*), $\mathbf{x}^{l}$ and $\mathbf{x}^{l+1}$ are the node feature matrices in lth and $(l+1)$th layer whaere as D is diagonal degree matrix, A$\in R^{N\times N}$ is the adjacency matrix with self-connections, D$\in R^{N\times N}$ is the degree matrix of A, and X$\in R^{N\times F}$ is the input features of the graph with N nodes and F-dimensional features.

doi:10.6342/NTU202300617

3.1.3.2 Local extrema convolution (LEConv)

LEConv is a convolution operation used in GNNs to process data represented as a graph. Instead of doing the standard convolution over the entire graph, the LEConv operation only performs it over a small local neighborhood of nodes. The operation aims to find local extrema with extreme values. Based on the input data, predictions or classifications can subsequently be made using these local extrema.

When discovering local patterns or linkages in the data, such as when predicting drug-target interactions or examining social networks, LEConv is very helpful. It can be applied to the analysis of data from a variety of fields, including chemistry, biology, social science, and many more.

Overall, the LEConv operation is a strong tool for processing and analyzing graph-based data. It has the potential to revolutionize several fields by allowing the examination of data that was previously too complex or challenging to examine using conventional techniques. The LEConv layer is formulated as below [3]:

$$\mathbf{x_i}^{l+1} = \mathbf{x_i} W_1 + \sum_{j \epsilon N(i)} e_{ij}(W_2 \mathbf{x_i}^{l} - W_3 \mathbf{x_j}^{l}), \tag{3.2}$$

where $\mathbf{x_i}^{l+1}$ is node embedding of node i in the $(l+1)$ layer, and W_1, W_2, W_3 are learnable matrices. We denote $N(i)$ as the neighborhood of the ith node in $N(i)$, and e_{ij} as the edge weight from source i to target node j.

LEConv can compute node embeddings by taking into account their global relevance by using self-loops (the first part of the equation (3.2). The second part of the equation

doi:10.6342/NTU202300617

(3.2) acts as a Laplacian regularization term. In general,

$$\nabla f = (f(i) - f(j))_{(A(i,j)>0)} \tag{3.3}$$

The local structure of a graph is modeled by the Laplacian term in graph convolutional networks by calculating the difference between the node's own feature vector and the average of its nearby nodes' feature vectors. This enables the model to understand how each node functions in relation to the nodes around it and can be used to locate local graph extrema. The node embeddings are encouraged to resemble the average of their neighbors via the Laplacian regularization term. Since these nodes will have substantially larger embeddings than their neighbors after the Laplacian regularization term has been applied, the model can detect local extrema by observing nodes with significantly different feature vectors than their neighbors. As a result, LEConv may locate local extrema and represent the local structure of the graph while accounting for the node's global relevance.

3.1.4 Graph pooling

In order to obtain topology-aware node representation, GNNs have been extensively employed in message propagation between nodes in graph data. However, in other cases (such as graph classification), we must represent the entire graph instead of just the raw nodes and edges. Thus, the graph must be continuously downscaled until it becomes a representation with a lower scale, a process known as graph pooling, similar to image pooling. Numerous ideas for graph pooling may be loosely separated into flat pooling and hierarchical pooling to provide an efficient and appropriate graph representation. The latter method of reducing the graph size gradually incorporates two approaches: node clustering pooling and node drop pooling. The former method creates a single graph-

doi:10.6342/NTU202300617

level representation in one step, which typically involves computing the average or sum of all node embeddings. Node clustering pooling groups nodes into clusters, creating new supernodes to build the smaller graph, but it is time-consuming and requires a large amount of memory. On the other hand, node drop pooling, which is more efficient and well-suited for large-scale graphs, selects a subset of nodes from the original graph to generate the reduced graph. All of the explained pooling approaches (TOP-K, SAG, and ASAP) use node drop pooling to generate a smaller graph.

3.1.4.1 TOP-K pooling

TOP-K [1] is a sort-based approach that converts nodes into corresponding scores for each pooling using a projection vector. Then, for the sake of further processing, only the nodes and connected edges with TOP-K scores are retained. It should be emphasized that the score solely considers how each node is represented separately. The scores for each node in a graph $G = (A, X)$ (where A is adjacency matrix and X is node feature matrix) are obtained using a trainable projection vector $\mathbf{p}$ as follows:

$$\begin{aligned} \mathbf{z} &= X^{(l)}\mathbf{p}^{(l)}/\|\mathbf{p}^{(l)}\|, \\ \text{idx} &= \text{top-rank}(\mathbf{z}, \lceil KN \rceil), \\ A^{(l+1)} &= A^{(l)}_{\text{idx,idx}} \end{aligned} \tag{3.4}$$

where top-rank is a function that returns the indices of the top $\lceil KN \rceil$, $_{\cdot idx}$ is an indexing operation, $A^{(l)}_{\text{idx,idx}}$ is the row-wise and col-wise indexed adjacency matrix. $X^{(l+1)}$ and $A^{(l+1)}$ are the new feature matrix and the corresponding adjacency matrix, respectively.

doi:10.6342/NTU202300617

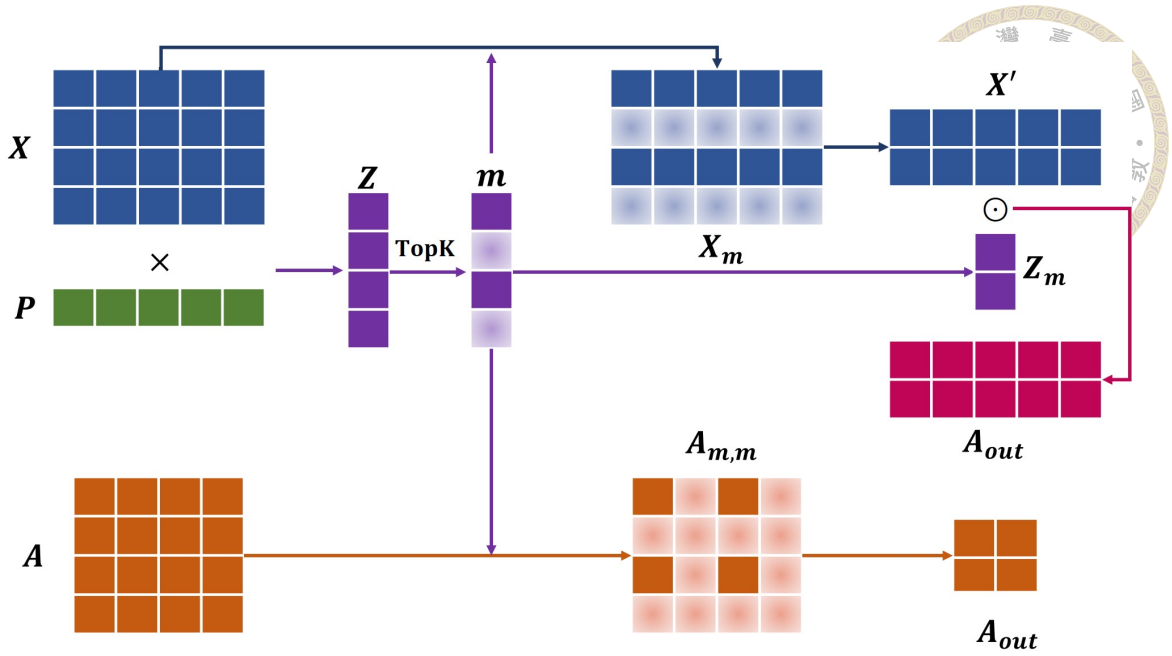


Figure 3.1: TOP-K pooling (Modified from [1])

3.1.4.2 SAG pooling

SAG pooling [2] is a sorting-based technique. It sorts the nodes and produces ranking scores based on a GNN that further incorporates the topology of the graph compared with only the independent attributes of the TOP-K Pool nodes. Each node is scored using GNN (In the case of SAG pooling, it is GCN) using the equation below [2]:

$$\begin{aligned} \mathbf{z} &= \sigma(D^{-1/2}AD^{-1/2}X\theta_{att}), \\ \text{idx} &= \text{top-rank}(\mathbf{z}, \lceil KN \rceil), \\ \mathbf{z}_{mask} &= \mathbf{z}_{idx} \\ X^{'} = X^{(l)}_{idx,:}, X^{(l+1)} &= X^{'} \odot \mathbf{z}_{mask}, A^{(l+1)} = A^{(l)}_{\text{idx,idx}} \end{aligned} \tag{3.5}$$

where **z** is node attention score, σ is the activation function (e.g., *tanh*), **x** is node feature matrix whaere as D is diagonal degree matrix, A$\in R^{N\times N}$ is the adjacency matrix with self-connections, D$\in R^{N\times N}$ is the degree matrix of A, and X$\in R^{N\times F}$ is the input features of the graph with N nodes and F-dimensional features. Note that $X^{(l)}_{idx,:}$ is a row-wise

doi:10.6342/NTU202300617

index feature matrix, $\odot$ is a broadcasted elementwise product, $A^{(l+1)}$ and $X^{(l+1)}$ are the adjacency matrix and node feature matrix of a pooled graph.

The top-performing nodes will be selected using the top-ranking nodes based on Z, similar to the TOP-K method. The node feature matrix and the adjacency matrix for the pooled graph are calculated using the masking operation, as defined in Equation 3.5.

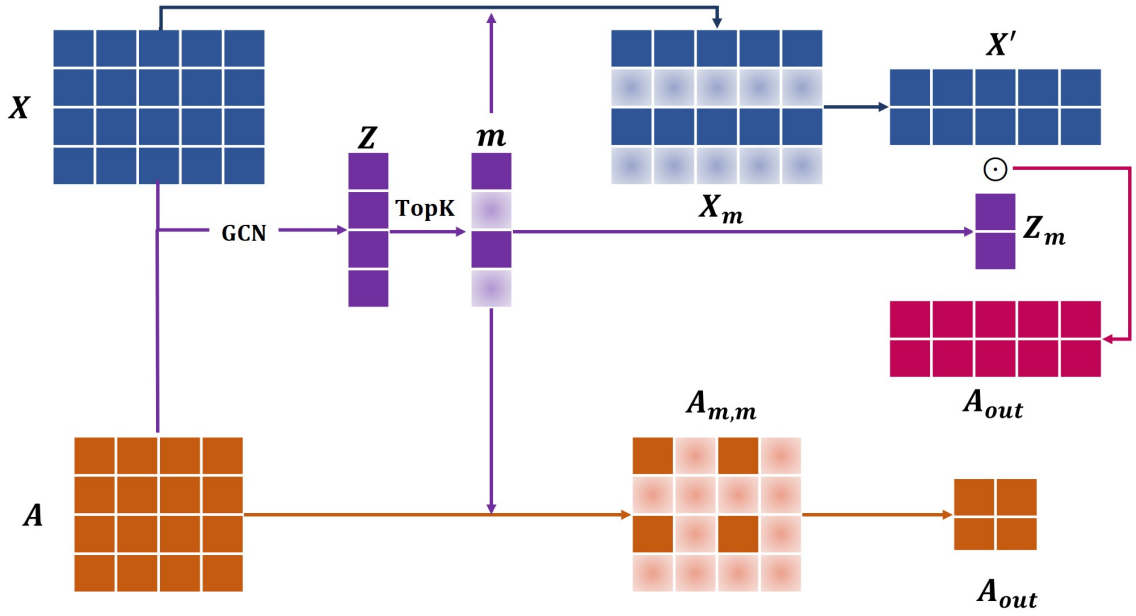


Figure 3.2: SAG pooling (Modified from [2]

3.1.4.3 ASAP pooling

TOP-K assigns node scores based on a projection vector that can be learned and samples a subset of nodes with high scores. It avoids computing soft assignment matrices and node aggregation to retain the sparsity in graph operations. The soft edge matrix in pooling is a weight matrix that represents the importance of edges between pairs of nodes in a graph and is used to aggregate information from neighboring nodes and edges. By applying a GCN to account for the graph's topology when determining a node's score, SAG outperforms TOP-K pooling. TOP-K and SAG pooling cannot adequately maintain node

doi:10.6342/NTU202300617

and edge information since they neither aggregate nodes nor compute soft edge weights. ASAP [3] solves this issue.

The process of clustering in chemical structure analysis involves multiple steps, beginning with the consideration of each node in the graph as a supernode or representative member of a cluster, which represents only the local neighbors within a fixed radius of h hops.

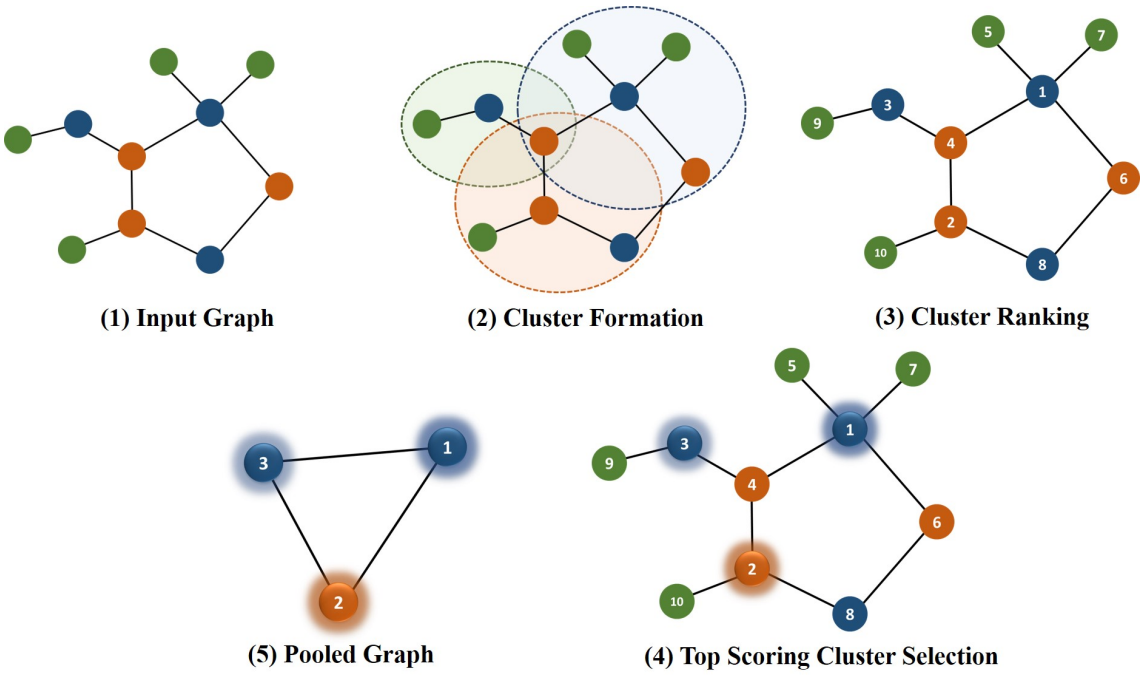


Figure 3.3: ASAP pooling (Modified from [3])

The cluster assignment matrix (S) is then used to represent the membership of each node in a cluster, with S_{ij} representing the membership of node v_i in cluster j. Using Master2Token [3] (The goal of the algorithm is to learn a set of supernodes that capture important structural information about the graph), clusters of similar molecules are formed based on their chemical structures and other properties. These clusters are used to create a new graph in which the nodes represent the clusters, and the edges represent the relationships between the clusters. During the cluster selection stage, a subset of clusters is

doi:10.6342/NTU202300617

selected from the full set of clusters formed during the cluster formation process. This is done by using the selection of TOP-K supernodes ranked or scored (Z score) by using LEConv 3.6. To create the pooled graph in the ASAP method, the cluster assignment matrix is used to represent the membership of each node in a cluster. The adjacency matrix for the pooled graph is constructed using the cluster assignment matrix and the adjacency matrix for the clusters, which is obtained by multiplying the cluster assignment matrix with the adjacency matrix for the clusters and then transposing the result. This process ensures that any two clusters in the pooled graph are connected if there is any common node in the clusters or if any of the cluster's constituent nodes are neighbors in the original graph, with the strength of the connection between clusters determined by both the membership of the constituent nodes through the cluster assignment matrix and the edge weights in the adjacency matrix for the clusters.

Note that $\mathbf{z}$ below is computed based on A and $\mathbf{x}_i$ of the supernode in each cluster.

$$\mathbf{z} = \sigma\left(W_1\mathbf{x}_i^{(l)} + \sum_{j\epsilon N(i)} A_{ij}\left(W_2\mathbf{x}_i^{(l)} - W_3\mathbf{x}_j^{(l)}\right)\right), \tag{3.6}$$

where σ is the activation function , and W_1, W_2, W_3 are learn-able matrices. We denote $N(i)$ as the neighborhood of the i^{th} node in $N(i)$, and A_{ij} as the edge weight from source j to the target node i.

3.2 Baseline model design

We consider two baseline models for our study. Our previous work proposed the first baseline model, and the second baseline model is SAG-DTA [20]. The first baseline model was tested on the BindingDB RTK dataset and performed well for the regression task. We

doi:10.6342/NTU202300617

will discuss the performance in the following sections. The second baseline model was proposed by Zang et al. [20], which is tested on two state-of-the-art datasets, Davis and KIBA.

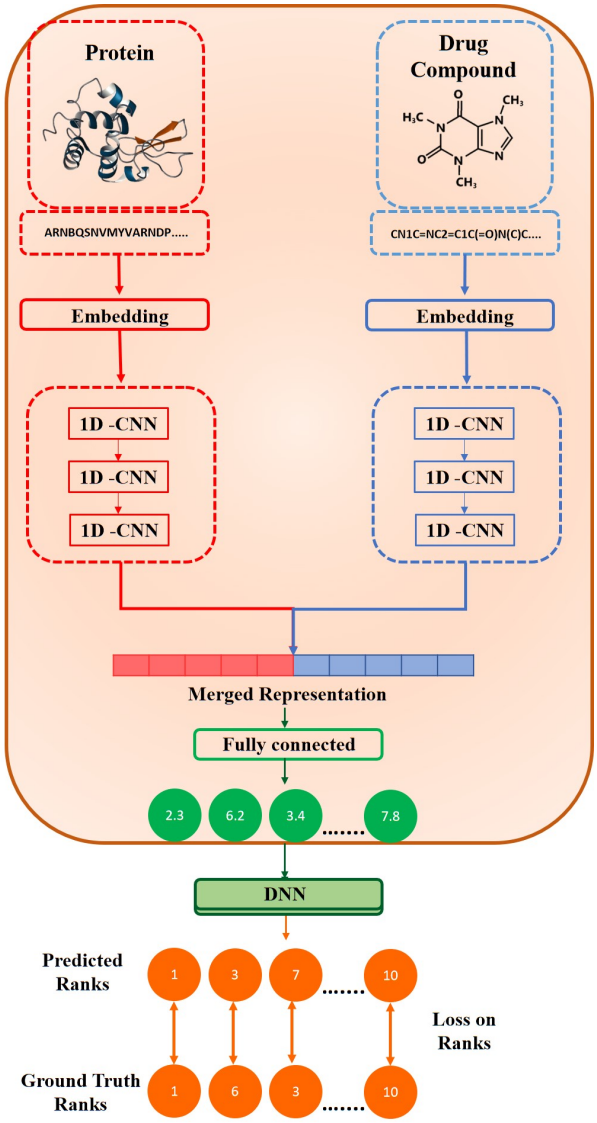


Figure 3.4: The network architecture of Sorter-DTI Freeze

The first baseline model used a 1D structure to represent protein sequences and drug SMILES. Hence, we used this baseline model to study the improvement in the results when the 2D structure was used to represent drug SMILES instead of the 1D structure. The second baseline model is a current state-of-the-art model representing the drug molecules as graphs (2 dimensional) and the protein as a one-dimensional sequence. This was used

doi:10.6342/NTU202300617

to demonstrate the enhancement in our proposed model's results.

In the first baseline model, we used two branches. The one-dimensional drug molecule and protein sequence sequences are fed to the two branches, respectively. Both the branches used the 1D-CNN, but before providing the sequence to 1D-CNN, we encoded the sequence by using trainable embedding layers. We stacked three 1D-CNN layers for both branches, as shown in Figure 3.4.

Max pooling operations were performed after the convolution to obtain the protein and SMILES sequence representations. After getting this representation, we obtained a combined representation using the concatenation operation. We added a fully connected layer after the convolution and pooling layers. According to Figure 3.4, the output layer comes after the fully connected layer. The baseline model's output layer predicts the IC50 score of the DTA. These predicted output scores are then fed to the DNN, then the output of this DNN network is ranked, i.e., predicted ranks. We computed listwise loss (ListNet [37]) on the predicted rank and the ground truth rank. This model (Sorter-DTI Freeze) was trained on ranking loss, and this training helped to improve the CI by 5% over the basic 1D-CNN representation model. [38].

In the second baseline model, the authors also used two branches. One dimensional protein sequence is first encoded by using an embedding layer, which is then fed to the one branch of the model using 1D-CNN. They stacked three 1D-CNN layers for the protein branch and max pooling operation after the convolution to obtain the protein sequence representation, as shown in Figure 3.5. On the other hand, they used graph data generated from drug SMILES in the form of an adjacency matrix and node feature matrix with the help of the RDKit toolkit. Then, GCN was performed on the adjacency matrix and

doi:10.6342/NTU202300617

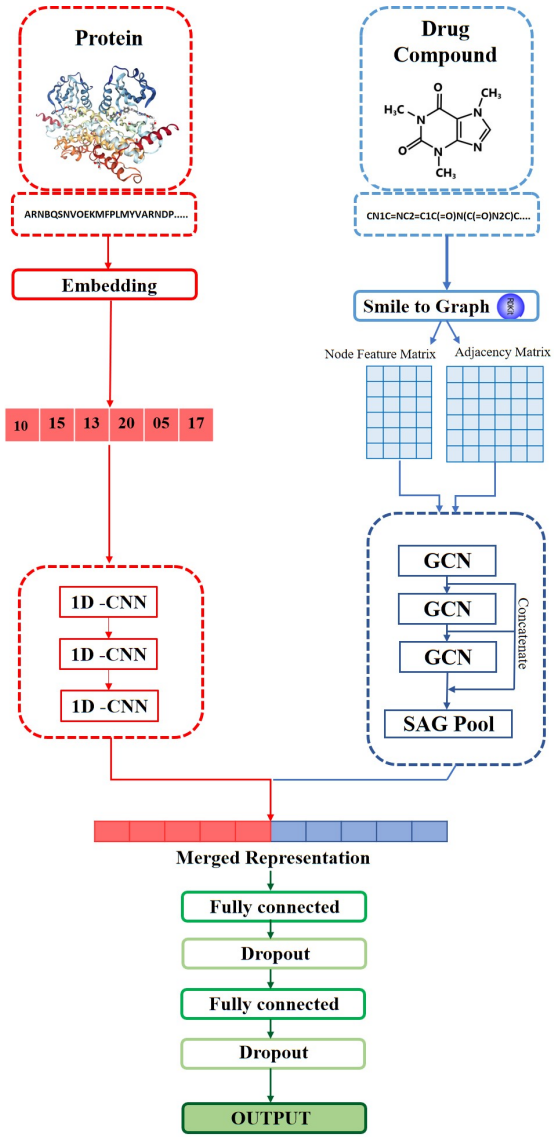


Figure 3.5: The network architecture of SAG-DTA

doi:10.6342/NTU202300617

node feature matrix; this graph convolution operation was performed three times in a row, followed by the self-attention graph (SAG) pooling operation to obtain a representation of the drug branch. They then combined both representations by using a concatenation operation. Three fully connected layers and a dropout layer were added after each fully connected layer. According to Figure 3.5, the output layer comes after the fully connected layer. The IC_{50} or KIBA score of the DTA was predicted by the baseline model's output layer. This model has two different variants in the pooling layer, i.e., global pooling and hierarchical pooling, based on the topology of the pooling operation. We compared our model against global pooling because of its best performance.

3.3 Proposed model design

In our proposed model, we also used two branches. The 1D sequence of the protein sequence was first encoded by using an embedding layer, which is then fed to the one branch of the model using 1D-CNN. We stacked three 1D-CNN layers in a row for the protein branch, followed by the max pooling operation to obtain the protein sequence representation. On the other hand, we used the graph data generated from drug SMILES in the form of an adjacency matrix and node feature matrix with the help of RDKit. Instead of using standard GCNs, we applied LEConv on the adjacency matrix and node feature matrix to update the node feature in each layer; this graph convolution operation is performed three times in a row, followed by TOP-K, SAG, and ASAP pooling operation to obtain a representation of drug branch.

We then obtained the combined representation by using the concatenation operation as shown in Figure 3.6. We added three fully connected layers and a dropout layer after

doi:10.6342/NTU202300617

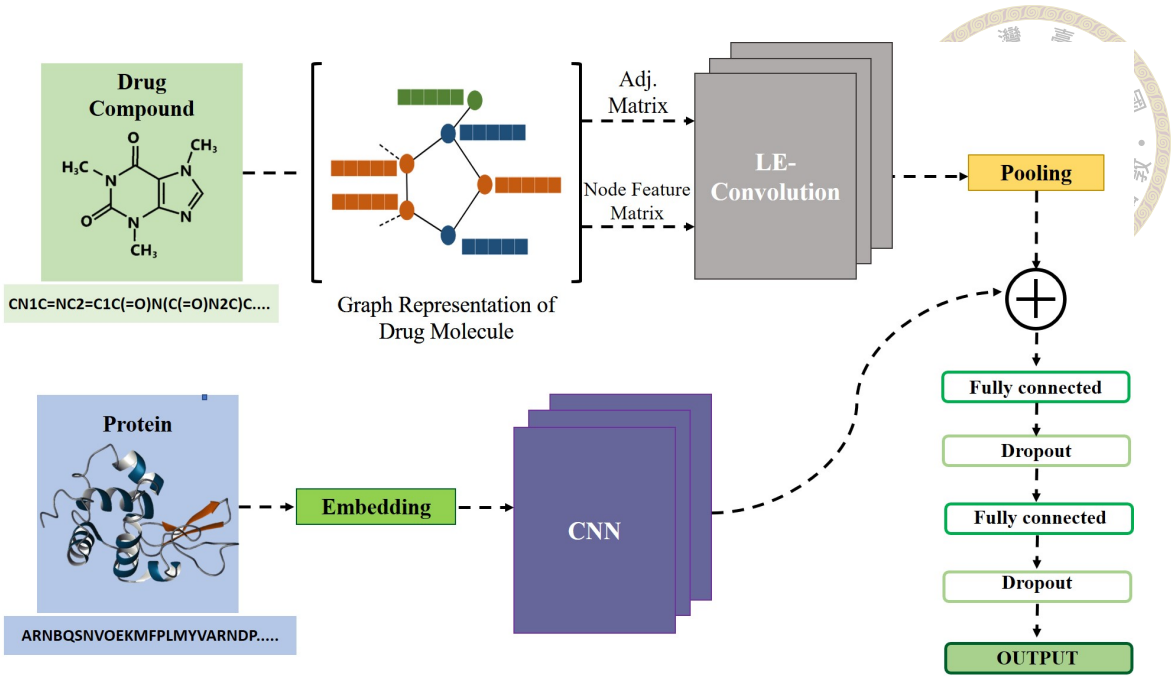


Figure 3.6: The network architecture of LE-DTA

each fully connected layer. The output layer comes after the fully connected layer. The IC_{50}, K_d, or $KIBA$ score of the DTA is predicted by the baseline model's output layer. Our model has three different variants; we tested the effect of three different pooling operations by keeping the exact configuration of LEConvs and these three pooling techniques are TOP-K, SAG, and ASA pooling.

The following Table 3.1 shows the summary of the methods used in this work.

3.4 Evaluation metrics

3.4.1 Concordance index

We use the Concordance Index (CI) as an evaluation metric because the affinity values are continuous. It is obtained by taking the ratio of the number of concordance pairs

doi:10.6342/NTU202300617

Graph Neural Network	Graph Convolution Network (GCN)	[21]
	Local Extrema Convolution (LEConv)	[3]
pooling	TOP-K pooling	[1]
	Self-Attention Graph pooling (SAG)	[2]
	Adaptive Structure-Aware pooling (ASAP)	[3]
Models	Baseline model (Sorter-DTI Freeze)	[38]
	Baseline model (SAG-DTA)	[2]
	Proposed model (LE-DTA)	-

Table 3.1: Methods summary.

and the total number of possible pairs.

$$CI = \frac{\text{no. of concordance pairs}}{\text{no. of possible pairs}} \tag{3.7}$$

The order of the predictions is what the CI is most interested in, not the forecast itself. We can compare the actual and anticipated affinities for each pair of affinities, i and j. It can be formulated using the following formula.

$$CI = \frac{1}{B} \sum_{y_i > y_j} h(f_i - f_j), \tag{3.8}$$

where f_i is the prediction value for the bigger affinity y_i (ground truth), f_j is the prediction value for the smaller affinity y_j, B denotes the number of possible non-equal pairs, and

$$h(u) = \begin{cases} 0, & u > 0 \\ 0.5, & u = 0 \\ 1, & u < 0 \end{cases} \tag{3.9}$$

CI values vary from 0 to 1 with 0 to 0.6 denoting a random or subpar prediction, 0.6 to 0.8 denoting a reasonable prediction, and 0.8 to 1.0 denoting a nearly flawless model

doi:10.6342/NTU202300617

prediction in the test data. CI can be at its maximum value of 1 if the predictions are in the same order as the input sequence, irrespective of the predicted values. If the order of the predicted pairs is opposite to the ground truth, then the CI would be at its lowest value of 0. As a common performance metric for DTA-based regression models, CI aids in assessing the rankings' quality. When ranking DTAs, it is more crucial to forecast the relative order of the labels than their exact values.

3.4.2 Mean square error

Since our target values are numeric, we use the regression task. Models can be evaluated by the most common loss function, i.e., the mean squared error. It measures the closeness of the target value to the predicted value and is formulated by the following equation.

$$MSE(y, \hat{y}) = \frac{1}{T}\sum_{i=1}(y - \hat{y})^2 \tag{3.10}$$

where vector T predictions are generated from a sample of T data points on all variables. This metric only considers one interaction and calculates the error between the actual affinity score and the predicted affinity score of the model. This loss function is interested in the loss of one pair of predicted values and a ground-truth value of the drug-target affinity. It might not be beneficial for drug screening as we are more interested in the ranking rather than the exact prediction of the affinity value. Hence, we consider it as the secondary metric for DTA prediction.

doi:10.6342/NTU202300617

Chapter 4

Experiment Settings and Results

4.1 Experiment settings

4.1.1 Baseline models

Sorter-DTI Freeze: According to the length distribution of the RTK dataset, we chose the maximum length of protein sequence and drug SMILES to be 1500 and 100, respectively. To avoid overfitting, we used dropout, a regularization technique that sets the activation of some of the neurons to 0 randomly. The following were the best hyper-parameters observed for the model in the validation set. The first FC layer consists of 1024 neurons with a dropout rate of 0.1. The regression task was done in 30 epochs, and a batch size of 256 was used to update the weights of the network. After hyperparameter tuning, *Nadam* was used as the optimization algorithm to train the network with a default learning rate of 0.001.

SAG-DTA: We set the learning rate to 0.001. The number of epochs was set to 2000 for both the KIBA and Davis datasets and 1000 for BindingDB because of its costly

doi:10.6342/NTU202300617

computation. The batch size was 512, and the *Adam* optimizer was used as in the original paper. SAG-DTA has two variants differentiated based on the pooling operation. One is global, and another is hierarchical pooling. In global pooling, three layers of GCN are followed by the SAGpooling layer. Similarly, in a hierarchical variant, each layer of GCN is followed by the SAGpooling layer. The dropout rate was set to 0.1.

4.1.2 Proposed model

We set the learning rate to 0.001. The number of epochs was set to 2000 for the KIBA and Davis datasets and 1000 for BindingDB because of its costly computation. The batch size was 512, and the *Adam* optimizer was used with three layers of LEConv, followed by the pooling layer. The dropout rate was set to 0.1. The input length for the protein sequence was set to 1000 characters.

4.2 Results

4.2.1 2D data contains more information than 1D data

Our team's previous work proposed Sorter-DTI Freeze that focused on the 1D representation of drug SMILES and protein sequences to predict the DTA [38]. The baseline model was tested on the BindingDB RTK dataset. The baseline model has many variants based on their training style. To improve the CI of the DTA prediction, our team built FC layers on the top of the regressor model to rank them. Since CI mainly focuses on the ranking and not its exact value. To improve the ranking, our team proposed several sorter variants. Sorter-DTI Freeze performs the best in CI. Another variant unfroze the layers

doi:10.6342/NTU202300617

above the 1D-CNN model and was termed Sorter-DTI. We also tried to optimize the pre-trained baseline model's dense layer for the ranking task, termed dSorter-DTI, along with its frozen version, dSorter-DTI Freeze. We tested LE-DTA (SAG) with SAG pooling on the same BindingDB RTK dataset to show a direct comparison with the baseline model.

In our proposed model, we represented the drug as a Graph (2D information) and the protein as a sequence. Note that we did not train our proposed model with list-wise ranking loss. The primary motivation for this decision was to enhance the CI score. However, our experimentation revealed that a better result in CI was achieved without utilizing it. Thus, its ranking loss is high compared to the previous models. We obtained a CI of 0.835 on LE-DTA (SAG) on the BindingDB RTK dataset, whereas the Sorter-DTI Freeze model showed a CI of 0.820 on the same. Results are summarised in Table 4.1.

Models	Drug Representation	BindingDB	
		CI	Ranking Loss
dSorter-DTI	sequence	0.802	0.207
dSorter-DTI Freeze	sequence	0.801	0.207
Sorter-DTI	sequence	0.803	2.078
Sorter-DTI Freeze	sequence	0.820	2.060
LE-DTA (SAG)	graph	**0.835**	**7.006**

Table 4.1: Performance summary: the performance of LE-SAG is compared with the baseline models

doi:10.6342/NTU202300617

4.2.2 LE convolution improves DTA prediction

The table shows that the LE-DTA model outperforms the other models on two out of three datasets (KIBA and BindingDB) in terms of both CI and MSE. On the Davis dataset, the LE-DTA model performs similarly to the SAG-DTA (GlobPool) model. The table also includes a row that shows the improvement in the performance of the LE-DTA model compared to the best-performing model on each dataset.

Models	Davis		KIBA		BindingDB	
	CI	MSE	CI	MSE	CI	MSE
GraphDTA (GAT) [19]	0.892	0.232	0.889	0.139	0.825	0.641
GraphDTA (GIN) [19]	0.893	0.229	0.891	0.139	0.823	0.556
SAG-DTA (HierPool) [20]	0.901	0.212	0.893	0.131	0.847	0.511
SAG-DTA (GlobPool) [20]	**0.903**	**0.209**	0.892	0.130	0.852	0.480
LE-DTA (ASAP)	**0.902**	**0.210**	0.901	0.121	0.854	0.470
LE-DTA (TOP-K)	0.898	**0.210**	**0.902**	**0.120**	**0.855**	**0.464**
Improvement	-	-	1.12%	7.7%	0.35%	3.33%

Table 4.2: Performance summary: the performance of LE-DTA is compared with the state-of-the-art model, such as GraphDTA and SAG-DTA, on three datasets (Davis, KIBA, BindingDB). The best performance is marked in boldface.

The CI and MSE values were obtained by testing our model on the three datasets, Davis, KIBA, and BindingDB, with the existing baseline models. The final results are summarized in Table 4.2. In this context, we decided not to utilize the Listwise Loss function. The primary motivation for this decision was to enhance the CI score, and our experimentation revealed that better results were achieved without utilizing it. We used TOP-K pooling for LE-DTA since it performs the best (will be discussed later). We di-

doi:10.6342/NTU202300617

rectly cite the results from baseline models for the Davis and KIBA datasets. However, there has been less research with BindingDB as a regression task; hence, we ran the baseline models under the same setting. LE-DTA achieved CI and MSE values in the range similar to the existing baseline models for Davis; LE-DTA has a CI of 0.898 and an MSE of 0.210. For the KIBA and BindingDB datasets, LE-DTA performed better than the baseline models. LE-DTA resulted in a CI of 0.902 and MSE of 0.120 on KIBA. When tested on the BindingDB datasets, the CI values obtained for LE-DTA is 0.855 with an MSE value of 0.464. These results were in the range for Davis while it showed a 1.12% improvement in CI with a 7.7% reduction in MSE for KIBA. Finally, on BindingDB, the CI was 0.35% better than the baseline models with an MSE reduction of 3.33%. Our model performed similarly to the baseline on Davis. Sightly improved on BindingDB and showed better performance on KIBA.

doi:10.6342/NTU202300617

Chapter 5

Discussion

5.1 Analysis of various pooling layers

In this study, the proposed model LE-DTA discussed in the previous chapter has the LEConv technique paired with the TOP-K pooling method. Here, we study the effect of how the variation in the pooling method can change the measured CI and MSE. Along with TOP-K pooling, SAG and ASAP, pooling has been combined with LEConv. The network architecture was kept exactly the same each time except for the change in the pooling method. The resulting models were then again tested on three datasets: Davis, KIBA, and BindindDB. Table 5.1 below summarizes the result obtained from testing the three models on all three datasets.

As inferred from the table, the difference in the measured value of CI and MSE was insignificant. For all three models, we see that when LEConv was combined with SAG pooling, we obtained the measured CI value of 0.900 for both Davis and KIBA, while the value for BindingDB is 0.853. The model gives an MSE value of 0.212, 0.120, and 0.477 for Davis, KIBA, and BindingDB, respectively. LEConv with ASAP pooling provided

doi:10.6342/NTU202300617

Models	Davis		KIBA		BindingDB	
	CI	MSE	CI	MSE	CI	MSE
SAG	0.900	0.212	0.900	0.120	0.853	0.477
ASAP	**0.902**	**0.210**	0.901	0.121	0.854	0.470
TOP-K	0.898	**0.210**	**0.902**	**0.120**	**0.855**	**0.464**

Table 5.1: Comparison of various pooling strategies: we compared various pooling layers, i.e., TOP-K, SAG, and ASAP pooling. The best-performing metric is bold-faced.

the CI value of 0.902, 0.901, and 0.854 and MSE values of 0.210, 0.121, and 0.470 for the Davis, KIBA, and BindingDB datasets, respectively. The measured value of CI and MSE for the TOP-K Pooling method was the same as in Table 5.1 for LE-DTA.

These value obtained by testing various pooling methods showed that the SAG pooling paired with LEConv has the minimum value of CI and highest value of MSE for all three datasets. Whereas for the Davis dataset, we obtained a higher value of CI with ASAP pooling, and the importance of MSE is the same as that of the TOP-K pooling method. Apart from this, for KIBA and Davis, the measured CI and MSE values are best for the TOP-K pooling method. Hence, it can be seen that the overall performance of the TOP-K pooling method with LEConv was better than the other two pooling method. It has an advantage over other pooling methods because of its simplicity and efficiency, as it does not require any additional learnable parameters or complex operations. Thus, for LE-DTA, we have combined LEConv with TOP-K pooling.

It is also noticed that for all the pooling methods, the results were improved when compared with the baseline models. This shows that LEConv can improve the results irrespective of the pooling method.

doi:10.6342/NTU202300617

5.2 Using BindingDB to improve prediction over the BindingDB RTK dataset

The BindingDB RTK dataset was developed to provide researchers interested in small molecule binding to RTKs a better-targeted source of information. A dedicated collection of binding affinity data for RTKs, a significant class of proteins that are frequently targeted by small molecule medicines, can be helpful for drug development and design efforts.

In our previous work, we suggested that the transferable property of our proposed model could be increased by training it on a bigger model. Obviously, the BindingDB dataset contains more information because it has $531,055$ affinities with 4232 unique protein IDs and $362,601$ small compounds compared to $24,337$ affinities with 41 unique RTK proteins and $19,990$ drug compounds in BindingDB RTK. On the BindingDB RTK dataset, LE-SAG achieved a CI of 0.835, while the same model achieved a CI of 0.853 on the BindingDB dataset. Training our model on this more diverse and bigger dataset can improve the model's performance. This model can then be tested on separate protein families.

5.3 Cross-dataset evaluation of LE-DTA (ASAP)

In our research, we conducted an experiment to evaluate the performance of the LE-DTA (ASAP) model on different datasets. Specifically, we trained the model on the BindingDB dataset and then tested it on the BindingDB RTK dataset. The results of this experiment showed that the model had a CI of 0.763 and an MSE of 0.572. However, we found that these results were not as impressive as those obtained from training and testing

doi:10.6342/NTU202300617

the model on the BindingDB dataset. In this case, the model showed a CI of 0.854 and an MSE of 0.470.

The fact that the model's performance was worse when tested on the BindingDB RTK dataset suggests that the model may not be as effective at predicting the affinities between proteins and ligands in this particular dataset. This could be due to differences in the characteristics of the proteins and ligands in the BindingDB RTK dataset compared to the BindingDB dataset. It is also possible that the model may not have been optimized for the BindingDB RTK dataset specifically.

Despite these results, our experiment provides valuable insights into the performance of the LE-DTA (ASAP) model and its ability to predict protein-ligand affinities. This knowledge could be used to improve the performance of the model on different datasets in the future. Additionally, our findings highlight the importance of testing machine learning models on different datasets to gain a comprehensive understanding of their capabilities and limitations.

5.4 Limitation of LE-DTA

A potential disadvantage of employing a 1D sequence for drug-target affinity prediction is that it does not give the model enough context to estimate the affinity effectively. The sequence might only provide the protein's main structure, leaving out other vital details like its three-dimensional structure, function, or other ligands that might impact the affinity.

The 1D sequence might not entirely reflect the characteristics of the protein that are important to its affinity for the medication, which is another possible problem. For in-

doi:10.6342/NTU202300617

stance, the conformational changes that the protein goes through when it binds to the drug or the existence of other binding sites on the protein that can affect the affinity might not be captured by the sequence. Using a 2D or 3D representation for drug-target affinity prediction may be helpful for future studies. However, there are still some current limitations, as discussed below.

5.5 Current limitations of representing proteins as 2D or 3D

Protein sequences are lengthy and complex, and representing them as a graph can increase the complexity of the data even further. Building and training a model using graph representations of protein sequences may be difficult as a result. There is no commonly used standard method for representing a protein sequence as a graph; instead, there are several distinct approaches. This can make it difficult to compare the results of different studies. Graph representations of protein sequences may be harder to understand than other representations, such as the 1D sequence. As a result, it could be more challenging to comprehend the outcomes of a model that uses a graph representation. Graph representation data for protein sequences may be limited, particularly for those challenging to make or purify. A protein's overall structure and conformation are even better represented by a 3D representation of the protein than by a 1D or 2D one. However, building and training a model utilizing 3D representations can be challenging due to the complexity and computational expense of processing a 3D representation of a protein. High-quality 3D structural data for proteins, particularly those that are challenging to express or purify, may be limited. Building a large enough dataset to train and evaluate a model can be more

doi:10.6342/NTU202300617

difficult as a result. Even if there are enough 3D structural data, it might not be enough to properly train a model to predict DTA. This can be a result of the problem's inherent complexity or the insufficient diversity of the data.

5.6 Future work

The BindingDB dataset is known for its high diversity and a large count of drug-protein affinities, which makes it a valuable resource for drug-target affinity prediction. While representing protein as 1D sequences, we need to set a maximum sequence length. The problem with this approach is that for protein sequences shorter than the maximum length, we must pad them with zeros, and for sequences longer than the maximum length, we must remove some characters. This can lead to a suboptimal representation of the protein sequence, causing a higher MSE and lower CI in the DTA prediction models. To overcome this issue, it may be necessary to explore alternative representations of protein sequences or to experiment with different sequence length cutoffs to find the optimal balance between model complexity and data representation.

A 3D representation of a drug captures more information about the chemical structure and conformation of the drug than a 1D or 2D representation. This can help improve the accuracy of DTA prediction models. A 3D representation of a drug can provide information about the specific way in which the drug binds to the target protein. This can help to understand the mechanisms of drug action and identify potential off-target effects. A 3D representation of a drug can capture information about the spatial arrangement of atoms and functional groups, which can be useful for identifying new chemical scaffolds or designing novel drugs with improved affinity. A 3D representation of a drug can more

doi:10.6342/NTU202300617

accurately reflect the properties of the drug in the context of a biological system, such as its hydration state and solubility.

The 3D structure of a protein can be represented using a variety of formats, such as a PDB file, for DTA prediction. This representation captures detailed information about the conformation of the protein and can be used to predict a wide range of properties, including binding affinity. One approach to overcome the limitations of using 3D structure representations for DTA prediction is to use machine learning techniques to extract features from the 3D structure and use these features as input to a DTA prediction model. This can help capture information about the 3D structure relevant for binding affinity while ignoring noise and variability that are not relevant.

doi:10.6342/NTU202300617

Chapter 6

Conclusions

In this study, we have proposed a novel framework, LE-DTA, to enhance the prediction of DTA. Since it is known that there is a loss of information when a drug SMILES is represented by using the 1-D sequence, in the proposed models, the drug SMILES were expressed in terms of a 2-D graph structure instead of a 1-D sequence. The drug SMILES were transformed into the 2-D structure graph structure using the RDKit program. Along with this, the protein sequence in our method of DTA prediction was represented as a 1-D structure.

Specifically, we successfully combined LEConv with different pooling methods to create our proposed models. We have combined LEConv with three pooling methods: the SAG, ASAP, and TOP-K pooling methods. The proposed model, LE-DTA, has been tested on three different datasets: the Davis, KIBA, and BindingDB datasets. To investigate the enhancement produced by our model, the CI and MSE were obtained by testing our models on different datasets and compared with that of the existing baseline models.

The results obtained showed that the proposed model outperforms the baseline mod-

doi:10.6342/NTU202300617

els. For the Davis dataset, LE-DTA achieves a CI of 0.898 and an MSE of 0.210. When tested on KIBA, a CI of 0.902 with an MSE of 0.120 was achieved. Finally, for BindingDB, CI obtained was 0.855 with an MSE of 0.464. The obtained value of CI and MSE for Davis was in range with that reported for the baseline models. However, an improvement of 1.12% for CI and a 7.7% reduction in the MSE value was achieved by LE-DTA on the KIBA dataset. For BindingDB, the model has successfully improved the value of CI by 0.35% with a reduced value of MSE by 3.33%. By attaining significant enhancement in the prediction of binding affinity, the proposed model can be used for drug repurposing, which in turn can benefit the pharmaceutical industry.

doi:10.6342/NTU202300617

Bibliography

[1] H. Gao and S. Ji, "Graph u-nets," in international conference on machine learning. PMLR, 2019, pp. 2083–2092.

[2] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in International conference on machine learning. PMLR, 2019, pp. 3734–3743.

[3] E. Ranjan, S. Sanyal, and P. Talukdar, "Asap: Adaptive structure aware pooling for learning hierarchical graph representations," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 5470–5477.

[4] A. Britton, "Effectiveness of covid-19 mrna vaccines against covid-19–associated hospitalizations among immunocompromised adults during sars-cov-2 omicron predominance—vision network, 10 states, december 2021—august 2022," MMWR. Morbidity and Mortality Weekly Report, vol. 71, 2022.

[5] H. Wang, J. Wang, C. Dong, Y. Lian, D. Liu, and Z. Yan, "A novel approach for drug-target interactions prediction based on multimodal deep autoencoder," Frontiers in pharmacology, vol. 10, p. 1592, 2020.

[6] J. P. Hughes, S. Rees, S. B. Kalindjian, and K. L. Philpott, "Principles of early drug discovery," British journal of pharmacology, vol. 162, no. 6, pp. 1239–1249, 2011.

doi:10.6342/NTU202300617

[7] P. J. Bousquet, M. A. Calderón, P. Demoly, D. Larenas, G. Passalacqua, C. Bachert, J. Brozek, G. W. Canonica, T. Casale, J. Fonseca et al., "The consolidated standards of reporting trials (consort) statement applied to allergen-specific immunotherapy with inhalant allergens: a global allergy and asthma european network (ga2len) article," Journal of Allergy and Clinical Immunology, vol. 127, no. 1, pp. 49–56, 2011.

[8] T. T. Ashburn and K. B. Thor, "Drug repositioning: identifying and developing new uses for existing drugs," Nature reviews Drug discovery, vol. 3, no. 8, pp. 673–683, 2004.

[9] J. Wang, "Fast identification of possible drug treatment of coronavirus disease-19 (covid-19) through computational drug repurposing study," Journal of chemical information and modeling, vol. 60, no. 6, pp. 3277–3286, 2020.

[10] R. P. Hertzberg and A. J. Pope, "High-throughput screening: new technology for the 21st century," Current opinion in chemical biology, vol. 4, no. 4, pp. 445–451, 2000.

[11] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer et al., "Applications of machine learning in drug discovery and development," Nature reviews Drug discovery, vol. 18, no. 6, pp. 463–477, 2019.

[12] A. W. Jones, "Early drug discovery and the rise of pharmaceutical chemistry," Drug testing and analysis, vol. 3, no. 6, pp. 337–344, 2011.

[13] B. E. Eaton, L. Gold, and D. A. Zichi, "Let's get specific: the relationship between specificity and affinity," Chemistry & biology, vol. 2, no. 10, pp. 633–638, 1995.

doi:10.6342/NTU202300617

[14] A. Rudnitskaya, B. Török, and M. Török, "Molecular docking of enzyme inhibitors: A computational tool for structure-based drug design," Biochemistry and Molecular Biology Education, vol. 38, no. 4, pp. 261–265, 2010.

[15] X. Chen, C. C. Yan, X. Zhang, X. Zhang, F. Dai, J. Yin, and Y. Zhang, "Drug–target interaction prediction: databases, web servers and computational models," Briefings in bioinformatics, vol. 17, no. 4, pp. 696–712, 2016.

[16] N. Fleming, "How artificial intelligence is changing drug discovery," Nature, vol. 557, no. 7706, pp. S55–S55, 2018.

[17] T. Pahikkala, A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang, and T. Aittokallio, "Toward more realistic drug-target interaction predictions," Briefings in bioinformatics, vol. 16, no. 2, pp. 325–337, 2015.

[18] H. Öztürk, A. Özgür, and E. Ozkirimli, "Deepdta: deep drug-target binding affinity prediction," Bioinformatics, vol. 34, no. 17, pp. i821–i829, 2018.

[19] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "Graphdta: Predicting drug–target binding affinity with graph neural networks," Bioinformatics, vol. 37, no. 8, pp. 1140–1147, 2021.

[20] S. Zhang, M. Jiang, S. Wang, X. Wang, Z. Wei, and Z. Li, "Sag-dta: prediction of drug–target affinity using self-attention graph network," International Journal of Molecular Sciences, vol. 22, no. 16, p. 8993, 2021.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.

doi:10.6342/NTU202300617

[22] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" arXiv preprint arXiv:1810.00826, 2018.

[23] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," arXiv preprint arXiv:1710.10903, 2017.

[24] M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber, and P. P. Zarrinkar, "Comprehensive analysis of kinase inhibitor selectivity," Nature biotechnology, vol. 29, no. 11, pp. 1046–1051, 2011.

[25] K. Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, P. Zhang et al., "Interpretable drug target prediction using deep neural representation." in IJCAI, vol. 2018, 2018, pp. 3371–3377.

[26] G. Landrum, "Rdkit documentation," Release, vol. 1, no. 1-79, p. 4, 2013.

[27] B. T. Burlingham and T. S. Widlanski, "An intuitive look at the relationship of ki and ic50: a more general use for the dixon plot," Journal of chemical education, vol. 80, no. 2, p. 214, 2003.

[28] J. Tang, A. Szwajda, S. Shakyawar, T. Xu, P. Hintsanen, K. Wennerberg, and T. Aittokallio, "Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis," Journal of Chemical Information and Modeling, vol. 54, no. 3, pp. 735–743, 2014.

[29] H. A. Simon and W. G. Chase, "American scientist," Scientist, vol. 61, no. 4, pp. 394–403, 1973.

[30] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.

doi:10.6342/NTU202300617

[31] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities." Proceedings of the national academy of sciences, vol. 79, no. 8, pp. 2554–2558, 1982.

[32] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[37] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in Proceedings of the 24th international conference on Machine learning, 2007, pp. 129–136.

[38] Gracie, "Top-k ranking of drug target interactions based on listwise loss and transfer learning," National Tsing Hua University, p. 68, 2021.

doi:10.6342/NTU202300617

[39] D. Grattarola, D. Zambon, F. M. Bianchi, and C. Alippi, "Understanding pooling in graph neural networks," arXiv preprint arXiv:2110.05292, 2021.

[40] K.-K. Mak and M. R. Pichika, "Artificial intelligence in drug development: present status and future prospects," Drug discovery today, vol. 24, no. 3, pp. 773–780, 2019.

[41] V. Mishra, "Artificial intelligence: the beginning of a new era in pharmacy profession," Asian Journal of Pharmaceutics (AJP), vol. 12, no. 02, 2018.

[42] M. Wen, Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun, and H. Lu, "Deep-learning-based drug-target interaction prediction," Journal of proteome research, vol. 16, no. 4, pp. 1401–1409, 2017.

[43] M. Gönen and G. Heller, "Concordance probability and discriminatory power in proportional hazards regression," Biometrika, vol. 92, no. 4, pp. 965–970, 2005.

[44] S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg, and A. L. Schacht, "How to improve r&d productivity: the pharmaceutical industry's grand challenge," Nature reviews Drug discovery, vol. 9, no. 3, pp. 203–214, 2010.

[45] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," Journal of chemical information and computer sciences, vol. 28, no. 1, pp. 31–36, 1988.

[46] B. Ramsundar, P. Eastman, P. Walters, and V. Pande, "Deep learning for the life sciences: Applying deep learning to genomics, microscopy," Drug Discovery, and More, 2019.

doi:10.6342/NTU202300617

[47] H. Öztürk, E. Ozkirimli, and A. Özgür, "Widedta: prediction of drug-target binding affinity," arXiv preprint arXiv:1902.04166, 2019.

[48] T. Oprea and J. Mestres, "Drug repurposing: far beyond new targets for old drugs," The AAPS journal, vol. 14, no. 4, pp. 759–763, 2012.

[49] M. J. O' Meara, J. Z. Guo, D. L. Swaney, T. A. Tummino, and R. Hüttenhain, "A sars-cov-2-human protein-protein interaction map reveals drug targets and potential drug-repurposing," BioRxiv, 2020.

[50] A. Mullard, "New drugs cost us 2.6 billion dollars to develop," Nature reviews. Drug discovery, vol. 13, no. 12, p. 877, 2014.

[51] J. Shim, Z.-Y. Hong, I. Sohn, and C. Hwang, "Prediction of drug–target binding affinity using similarity-based convolutional neural network," Scientific Reports, vol. 11, no. 1, pp. 1–9, 2021.

[52] J. Li, A. Fu, and L. Zhang, "An overview of scoring functions used for protein–ligand interactions in molecular docking," Interdisciplinary Sciences: Computational Life Sciences, vol. 11, no. 2, pp. 320–328, 2019.

[53] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, and M. Ester, "Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines," Journal of cheminformatics, vol. 9, no. 1, pp. 1–14, 2017.

[54] C. Liu, Y. Zhan, C. Li, B. Du, J. Wu, W. Hu, T. Liu, and D. Tao, "Graph pooling for graph neural networks: Progress, challenges, and opportunities," arXiv preprint arXiv:2204.07321, 2022.

doi:10.6342/NTU202300617

[55] D. E. Gordon, G. M. Jang, M. Bouhaddou, J. Xu, K. Obernier, K. M. White, M. J. O'Meara, V. V. Rezelj, J. Z. Guo, D. L. Swaney et al., "A sars-cov-2 protein interaction map reveals targets for drug repurposing," Nature, vol. 583, no. 7816, pp. 459–468, 2020.

[56] T. Pahikkala, A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang, and T. Aittokallio, "Toward more realistic drug-target interaction predictions," Briefings in bioinformatics, vol. 16, no. 2, pp. 325–337, 2015.

doi:10.6342/NTU202300617

Appendix A — Introduction

附錄通常拿來寫冗長證明用的。沒有需要的話去 main.tex 把 inputback/appendix01 inputback/appendix02 註解

A.1 Introduction

A.2 Further Introduction

doi:10.6342/NTU202300617