EMPOWERING HETEROGENEOUS NETWORKS FOR DRUG-TARGET AFFINITY PREDICTION

by

Selen Parlar

B.S., Computer Engineering, Marmara University, 2018

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2022

ABSTRACT

EMPOWERING HETEROGENEOUS NETWORKS FOR DRUG-TARGET AFFINITY PREDICTION

Predicting drug-target binding affinity is a critical phase in computer-aided drug design, which can help accelerate the drug development process and reduce experimental validation costs caused by the significant false-positive rates. Hence, developing in-silico computational algorithms to predict drug-target binding affinity values has become an important research area. Machine learning approaches have been proposed for this task, including models that use readily available biomolecule sequences and heterogeneous networks enriched with drug and target-related information. We present WideDeepDTA, the first study that leverages both text-based and network-based approaches and predicts drug-target binding affinities. Given homogeneous and heterogeneous networks containing multiple types of biological entities, relationships between these entities, and pre-trained language models for biomolecular language, WideDeepDTA first learns the low-dimensional feature representation of drugs and targets using the node embedding technique Metapath2Vec. Then, it predicts affinity values based on the learned features. WideDeepDTA demonstrates its ability to create rich representations in the drug-target affinity prediction task compared to one of the state-of-the-art methods, DeepDTA, on the BDB dataset in terms of concordance index and mean squared error. Experiments indicate that integrating pre-trained language models with heterogeneous information improves model performance, especially while predicting the affinity values between proteins and unseen ligands. Moreover, the results show that the model performance improves when heterogeneous graphs are empowered with the information extracted from text-based representations.

ÖZET

İLAÇ-HEDEF BAĞLILIK İLGİSİ TAHMİNİ İÇİN HETEROJEN AĞLARI GÜÇLENDİRME

İlaç-hedef bağlılık ilgisi tahmini, bilgisayar destekli ilaç tasarımında, ilaç geliştirme sürecini hızlandırmaya ve çok sayıda bulunan yanlış pozitif oranlarının neden olduğu deneysel doğrulama maliyetlerini düşürmeye yardımcı olabilecek kritik bir aşamadır. Bu nedenle, ilaç-hedef bağlılık ilgisi değerlerini tahmin etmek için bilgisayar ortamında hesaplama algoritmaları geliştirmek ilgi çekici bir araştırma alanı haline gelmiştir. Güncel çalışmalar bu görev için, kolayca bulunabilen biyomolekül dizilerini ve ilaçlarla ve hedeflerle alakalı bilgilerle zenginleştirilmiş heterojen ağları kullanan modeller de dahil olmak üzere, makine öğrenimi yaklaşımlarını kullanır. Bu tezde, hem metin tabanlı hem de ağ tabanlı yaklaşımlardan yararlanan ve ilaç-hedef bağlılık ilgisi değerlerini tahmin eden ilk çalışma olan WideDeepDTA'yı sunuyoruz. WideDeepDTA içerisinde birden fazla biyolojik varlık türü, bu varlıklar arasındaki ilişkiler ve biyomoleküler dil için önceden eğitilmiş dil modellerini içeren homojen ve heterojen ağları barındırır. Tüm bunlar göz önüne alındığında, WideDeepDTA önce ağlarda bulunan tüm düğümler için bir vektör gösterim öğrenme yöntemi olan Metapath2Vec'i kullanarak ilaçların ve hedeflerin düşük boyutlu vektör temsillerini öğrenir. Ardından, öğrenilen temsillere dayanarak ilaç-hedef bağlılık ilgisi değerlerini tahmin eder. WideDeepDTA, BDB veri kümesindeki en başarılı yöntemlerden biri olan DeepDTA'ya kıyasla ilaç-hedef bağlılık ilgisi tahmini görevinde uyumluluk indeksi ve ortalama kare hata başarı metriklerinde iyileşme göstererek zengin temsiller oluşturmayı başarmıştır. Yapılan deneyler, ilaçlar ve proteinler için önceden eğitilmiş dil modellerini heterojen ağlarla birlikte kullanmamın model performansını geliştirdiği göstermektedir. Ayrıca sonuçlar, metin tabanlı temsillerden elde edilen bilgilerle heterojen ağlar güçlendirildiğinde model performansının arttığını göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

e	A single edge
ε	Edge set
G	Graph
$N(v)$	Neighborhood of node v
T	Relation set
M	Negative sample size
$P(u)$	Pre-defined distribution
R	Relation
v	A single node
V	Node set
X	Low-dimensional matrix
$\phi(v)$	Node mapping function
$\sigma(x)$	Sigmoid function
$\varphi(e)$	Edge mapping function

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
BPE	Byte Pair Encoding
CI	Concordance Index
CID	Compound ID Number
CNN	Convolutional Neural Network
CTD	Comparative Toxicogenomics Database
DDI	Drug-Drug Interaction
DDiPA	Drug-Disease-Protein Association
DDS	Drug-Drug Similarity
DNN	Deep Neural Network
DOI	Digital Object Identifier
DSA	Drug-Side Effect Association
DTA	Drug-Target Affinity
DTI	Drug-Target Interaction
EMBL	European Molecular Biology Laboratory
GCN	Graph Convolutional Network
GNN	Graph Neural Network
IC50	The Half Maximal Inhibitory Concentration
IMC	Inductive Matrix Completion
InChI	International Chemical Identifier
K_d	Dissociation Constant
K_i	Inhibition Constant
KronRLS	Kronecker Regularized Least Square
LM	Language Model
LMCS	Ligand Maximum Common Substructure
MedDRA	Medical Dictionary for Regulatory Activities

MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
nM	Nanometer
NPLM	Neural Probabilistic Language Model
pKd	Log Transformed Dissociation Constant
PLM	Pre-trained Language Models
PPI	Protein-Protein Interaction
PPS	Protein-Protein Similarity
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
SMILES	Simplified Molecular-Input Line-Entry System
UniProt	The Universal Protein Source

1. INTRODUCTION

Drug design is a costly and time-consuming process that can be accomplished by discovering new candidate chemicals and evaluating these chemicals against various protein targets [1]. The main goal of drug design is to provide a selective effect while minimizing the side effects by targeting only the disease-specific receptors and protecting the healthy cells [2]. Today, rational designs that save time and cost in the pharmaceutical design are applied, and possible to develop drugs with selectively practical and fewer side effects [3]. This rational discovery process often begins with developing a drug-active substance by selecting and improving ligands from a molecule library [4]. Proteins, DNA, RNA, and other small molecules can all interact with a drug. However, the size of the drug search space is enormous when we consider the existence of 100 million chemicals in the chemical database PubChem [5], the 16,526 drugs in the DrugBank [6], and over 189 million proteins in UniProt [7].

Generating novel drug molecules considers several properties. For instance, a drug molecule should be synthesizable and should have target specificity, *i.e.*, it should have binding affinity to the target protein of interest. On the other hand, it should have off-target selectivity so that its binding affinity is low to other targets. These properties make drug discovery cost over 3 billion USD, consume more time than a decade, and give a success rate of less than 10% [8]. Due to the expansive search space, high cost, and time consumption, the need for computational methods has emerged for this multi-stage, trial and error-based process [9].

Traditional computational drug design relies on simulations, heuristic search algorithms, and extensive domain knowledge. Furthermore, there is a great deal of interest in developing machine learning algorithms that can efficiently discover a large number of plausible and novel candidate drugs. Recently, deep learning approaches have gained attraction in the in silico drug design, increasing the available data and computing power of computers.

As an initial step, studies try to elaborate on a drug's interacting targets, and computational methods try to determine the interacting and non-interacting drug and target pairs and use binary classification methods [10–14] for that purpose. However, the strength of the protein-ligand interactions, *i.e.,* the binding affinity, is essential in the drug design pipeline since a strong interaction is the first step in finding a selective drug [15]. Binary classification-based approaches provide information about a possible interaction between proteins and ligands; however, these methods cannot determine binding affinity. Therefore, the prediction of the binding affinity value still remains a challenge [16].

In order for a machine learning algorithm to interpret any type of input, it requires to get some effective representation, that is, vectorization [17]. In the case of the computer-aided drug design, ligands, proteins, or any other types of biomolecule-related data need to be vectorized and represented numerically [18]. Text-based or graph-based representation approaches are commonly employed to vectorize biomolecules. These representations are then used as training data in drug discovery studies in order to learn the relations between them or make some inference about their interactions [19–22]. Chemical or protein sequences or structures are some of the crucial data for drug-target interaction or the affinity prediction task; however, in online databases, there are many other types of available information related to chemicals and targets which are proven to affect the binding process [23,24], such as the associated diseases of proteins, the side effects of the drug molecule, and the other interacting drugs or proteins. Therefore, rather than using only one type of information while learning the representation, studies show that integrating more information into the representations increases their ability to learn more features [25–28] of the data, resulting in richer representations. Inspired by the richer representations of biomolecules, this study integrates drug and protein sequences, the text-based similarities of biomolecule sequences, associated diseases, and side effects while learning the representation vectors of drugs and proteins using a heterogeneous network-based approach to the drug-target affinity prediction task. This is the first study that combines heterogeneous graphs and biomolecular language-based information for the drug-target affinity prediction task to the best of our knowledge.

2. RELATED WORK

The computational methods used in drug discovery recently focused on four strategies; ligand similarity-based [29], molecular docking/structure-based [30,31], deep learning-based [27, 32], and network-based approaches [27, 33–35]. The performance of the ligand similarity-based approaches is often low when a target has a few known binding ligands. Also, the limited availability of 3D structures of target proteins limits the molecular docking performance. Due to the limited data availability, some efforts have been devoted to developing machine learning-based approaches for drug target affinity (DTA) predictions through computational techniques. The growing amount of drug-target binding affinity data available in online databases has led to the adoption of advanced learning techniques such as deep learning architectures in predicting binding affinities [16, 36–40]. Last but not least, with networks, the ability to integrate several types of information, the affinity prediction task gained some other insights, and in the last decade, the number of studies increased [27, 32, 41].

Several types of deep learning frameworks have been adopted in the DTA prediction task. DeepDTA [16] and WideDTA [40] approaches are proposed to predict the binding affinities of protein-ligand interactions. Both methods utilize deep learning models that use only 1D representations of proteins and ligands. As the 1D representation, both studies use SMILES (Simplified Molecular Input Line Entry System) representations of the compounds rather than complex external features. DeepDTA learns high dimensional features from full-length sequences of the proteins and ligands. It uses two Convolutional Neural Networks (CNNs) to learn the representations of drugs and proteins. Then, the concatenated representations of drugs and proteins are fed into a multi-layer perceptron (MLP). Nevertheless, it fails to capture the biologically important short subsequences. WideDTA overcomes this problem by integrating different kinds of text-based information such as protein sequence, ligand SMILES, protein domains and motifs, and maximum common substructure words to provide better representation and predict binding affinity. To do that, WideDTA employs four

CNNs and learns the representations of drugs and proteins. Similarly, it uses the MLP with the concatenated representations. The DeepConv-DTI [42] also utilizes CNNs on the protein sequences. On the other hand, they use 2D structural images of chemicals to learn complex features using CNNs and produce DTA predictions.

Although the extensive experiments and enhanced performance in the DTA prediction task, representing the drugs as strings cause a loss of information since 1D representations cannot fully represent the structural information beneath the biomolecules. Graph neural networks (GNNs) are employed to address this problem, and drugs are represented as graphs. Tsubaki *et al.* [43] propose to use CNNs and GNNs together to learn the representation of compound graphs and protein sequences. They demonstrate performance improvement on the DTA task compared to the feature-based methods. GraphDTA [41] also suggests a new neural network architecture for the drug-target affinity prediction task. Rather than using the 1D representation of SMILES, they convert SMILES representation into a molecular graph and employ a graph neural network (GNN) to learn a graph representation. Moreover, they encode and embed protein amino acid sequences and use CNN to create protein representations. Then, combine CNNs and GNNs to predict the binding affinity value. Another method, DGraphDTA [44], uses graphs to represent both compounds and proteins with GNNs. Additionally, to address the interpretability, several models employ an attention mechanism [45–48].

Rather than using only the known drug-target interaction (DTI) data in deep learning models, some other diverse information from heterogeneous data sources integrated into the systems, such as protein-protein interaction (PPI), drug-disease association, drug-side effect association as in the work of MSCMF [33], HNM [35], DTINet [27], and NeoDTI [32]. They employ networks that can capture the complex relationships between different types of components, such as drugs and proteins. These methods have improved the performance in the DTI prediction task, yet they have some limitations to be addressed. For instance, in MSCMF [33], drug and protein similarity matrices are gathered from different data sources via a weighted averaging scheme in order to

use in the matrix factorization of a given DTI network. However, this data integration often causes data loss, resulting in a suboptimal solution. Moreover, DTINet [27] is developed as a computational pipeline to predict novel DTI from a heterogeneous network. First, it learns low-dimensional feature representations of drugs and targets in an unsupervised manner. Then it predicts new DTIs with inductive matrix completion (IMC) as in the work of Natarajan and Dhillon [49]. Since DTINet handles the unsupervised feature learning procedure and the prediction task separately, it may cause non-optimal solutions. NeoDTI [32] targets this problem and combines feature learning and classification into a single task, improving the accuracy.

More recently, Zhao *et al.* [50] propose a method that combines GNNs and deep neural networks (DNNs) for the DTI prediction task. They build a drug-protein network using drug-drug interaction, protein-protein interaction, and drug-protein interaction networks in which nodes represent drugs and proteins, and edges represent the link strength between them. Then, handles the DTI prediction problem as a node classification problem. Another network-based method EEG-DTI [51] proposes an end-to-end heterogeneous graph representation learning-based framework to predict the interaction between drugs and targets using graph convolutional networks (GCNs). DTiGEMS+ [52] constructs a heterogeneous graph using the DTI graph with drug-drug similarity and target-target similarity graphs. It combines feature-based and similarity-based approaches to model the identification of drug-target pairs. After performing graph augmentation, it applies node2vec [53] for feature representation learning of drugs and targets and uses them in a link prediction task. To improve the DTiGEMS+'s performance, DTi2Vec [54] is proposed in which representation learning and ensemble learning techniques are combined to identify the drug-target interactions. Unlike the previous work, it uses edge embeddings between drug-target node pairs rather than node embeddings. Given the success of heterogeneous graphs in the DTI prediction and text-based methods in DTA prediction, in this thesis, we propose a method for DTA prediction that utilizes heterogeneous graphs together with the biomolecular language-based information obtained from the text representations of chemicals and proteins.

3. BACKGROUND

This thesis combines several topics and applies multiple computer sciences and cheminformatics studies. This chapter provides insight into this study's techniques and terminology: homogeneous and heterogeneous graphs, graph representation learning, language model-based representation learning, and evaluation metrics.

3.1. Graphs

A graph $G = (V, \varepsilon)$ is defined by a set of nodes V and a set of edges ε between these nodes as going from node $u \in V$ to node $v \in V$ as $(u, v) \in \varepsilon$ [55]. In this thesis, we concern only simple graphs, i.e., there exists at most one edge between each pair of nodes and no edges between a node and itself. Moreover, all edges are undirected, so $(u, v) \in \varepsilon \longleftrightarrow (v, u) \in \varepsilon$. A graph has a single type of edge or different types of edges. In a multi-relational graph the edge notation can be extended as $(u, \tau, v) \in \varepsilon$ to include the relation type τ [56]. Throughout the thesis, we consider two important subsets of graphs; graphs with single and multiple relation types, *i.e.*, homogeneous and heterogeneous, respectively.

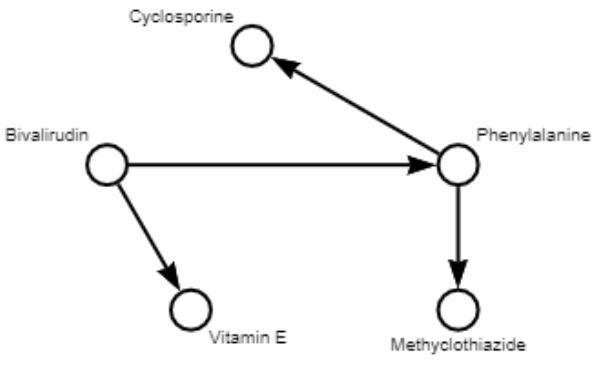

Figure 3.1. Homogeneous drug-drug interaction graph.

A graph is homogeneous when all the nodes represent the same type of instances, and all the edges represent the same type of relations [57,58]. For instance, a drug-drug interaction network is a homogeneous graph consisting of drugs and the connections between these drugs, representing the same type of entity.

Figure 3.1 shows a homogeneous drug-drug interaction graph with five drugs; Bivaluridin, Cyclosporine, Methyclothoazide, Vitamin E, and Phenylalanine; and the edges represent the interaction between drug pairs. For example, Cyclosporine interacts with Phenylalanine, and this interaction is extracted from the DrugBank database, depicting the increased risk of bleeding when Cyclosporine is consumed with Phenylalanine [59].

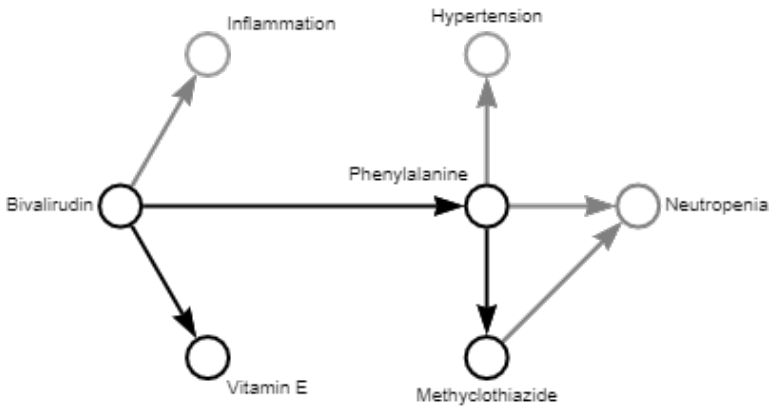

Figure 3.2. Heterogeneous drug-disease association graph.

A graph is heterogeneous if the set of nodes can be partitioned into disjoint sets $V = V_1 \cup V_2 \cup ... \cup V_k$ where $V_i \cap V_j = \emptyset, \forall i \neq j$ [60]. For instance, the drug-disease network is a heterogeneous graph consisting of two types of nodes as drugs and diseases, and two types of edges represent the treatment relationship between drug nodes and disease nodes, and similarly, the polypharmacy side effect that occurs only between two drug nodes.

Figure 3.2 shows a heterogeneous drug-disease association graph with four drugs; Bivaluridin, Phenylalanine, Vitamin E, and Methyclothoazide, three diseases; inflammation, hypertension, and Neutropenia, and two types of edges; drug interacts with a drug and drug associates with a disease. For instance, Methyclothiazide interacts with Phenylalanine, and both drugs associate with Neutropenia. This relation depicts that the risk or severity of Neutropenia can be increased when Methyclothiazide is combined with Phenylalanine [61].

3.2. Graph Representation Learning

The rapid development of molecular biology, bioinformatics, and cheminformatics and the increase in the available data has led to the modeling of the biological components as nodes and the interactions between nodes as edges of graphs [62–64]. In the case of drug discovery and disease treatment, it is crucial to examine the interactions between drug-drug, drug-disease, drug-protein, and protein-disease [65, 66]. These interactions can be formed as heterogeneous graphs and used in knowledge extraction. Traditional machine learning algorithms use the data represented in the Euclidean domain, such as 1D sequences of proteins, 2D biomedical images, or 3D protein structures. However, graphs form a non-Euclidean domain and create a challenge due to their complex topological structure [67, 68], diverse node connections, and arbitrary neighbor size. To address these challenges, graph representation learning is employed. Graph representation learning or graph embedding learns the low-dimensional representations of nodes or edges used in downstream graph analytical tasks or machine learning tasks such as node classification, link prediction, and graph classification.

The graph-based distributed representation learning method represents the chemicals and proteins. In general, this set of methods represents data that cannot be expressed in Euclidean space as a graph, and it aims to learn distributed vectors that reflect the semantic connections in the graph for nodes and edges [69].

One of the essential advantages of this approach is that it can express the relationships between different types of nodes. The relationship between each node corresponds to different relationships. Graph-based representation vectors are learned for proteins and chemicals using the heterogeneous graph structure. In this heterogeneous graph, when distributed representations for chemicals and proteins are learned, the relationships of nodes with different concepts such as disease and side effects are also included in the representations. In this way, the received vectors become richer in terms of information.

To learn the distributed representation vectors, Metapath2Vec [70, 71] is employed, which is a framework to learn representations of heterogeneous graphs. It is a neural network model that is designed to capture the rich semantics embedded in heterogeneous graphs by exploiting different types of relationships and meta-paths among nodes. To generate meaningful representations, it considers different semantics of relations, i.e., different meta-paths, the sequence of node/edge types that denote relationships between node pairs.

The word2vec model is proposed to learn the distributed representations of words within a corpus [72,73]. After that, DeepWalk [74], and node2vec [53] models were proposed, aiming to map the word-context concept of the word2vec model into a network. DeepWalk and node2vec models use random walks to map the word-context concept and utilize the skip-gram model to learn the node representation in a homogeneous network. Their objective is to maximize the network probability [53, 73, 74] as

$$arg\max_{\theta} \prod_{v \in V} \prod_{c \in N(V)} p(c|v;\theta), \tag{3.1}$$

where $N(v)$ denotes the node v's neighborhood, in which v's one-hop neighbors, and $p(c|v;\theta)$ defines the conditional probability of a context node c given node v.

Metapath2Vec formalizes the representation learning problem in heterogeneous networks by leveraging the definitions in [70, 75] as follows:

A heterogeneous network is a graph $G = (V, E, T)$ in which node v is associated with edge e with mapping functions $\phi(v) : V \rightarrow T_V$ and $\varphi(e) : E \rightarrow T_E$, respectively. Heterogeneous network representation learning aims to learn the d-dimensional representation $X \in R^{|V| \times d}, d \ll |V|$, given a heterogeneous network G, that is able to capture the topological and semantic relations among them. Therefore, the resulting representation is a low-dimensional matrix X, with the v^{th} row corresponding to the representation of node v. Regardless of the node types in V, representations of each node are mapped into the same latent space.

Similar to word2vec, Metapath2Vec introduces the heterogeneous skip-gram model for heterogeneous networks to model the heterogeneous neighborhood of a node. Therefore, Metapath2Vec aims to maximize the probability of having the heterogeneous context $N_t(v), t \in T_V$ given a node v as

$$arg\max_{\theta} \sum_{v \in V} \sum_{t \in T_V} \sum_{c_t \in N_t(V)} p(c_t|v;\theta), \tag{3.2}$$

where $N_t(v)$ denotes the node v's neighborhood with the t^{th} type of nodes. The $p(c_t|v;\theta)$ is a softmax function [73, 76] formulated as

$$p(c_t|v;\theta) = \frac{e^{X_{c_t}}.X_v}{\sum_{u \in V} e^{X_u}.X_v}, \tag{3.3}$$

where X_v is the v^{th} row of X, corresponding to the embedding vector for node v. The word2vec also introduces negative sampling [73] for optimization. A small set of words are sampled from the corpus with negative sampling to compute the softmax. Same technique is also applied for Metapath2Vec, and Equation (3.2) is updated as

$$log\sigma(X_{c_t}.X_v) + \sum_{m=1}^{M} \mathbb{E}_{u^m \sim P(u)}[log\sigma(-X_{u^m}.X_v)], \tag{3.4}$$

where M is the negative sample size, $\sigma(x) = \frac{1}{1+e^{-x}}$ and $P(u)$ is the pre-defined distribution in which node u^m is drew from M times.

In order to transform heterogeneous network structures into metapath2vec's skip-gram, the model designs a meta-path-based random walks, and generate paths. A meta-path schema is a path, denoted as $V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \cdots V_t \xrightarrow{R_t} V_{t+1} \cdots \xrightarrow{R_{l-1}} V_l$, where $R = R_1 \circ R_2 \circ \cdots \circ R_{l-1}$ defines the composite relations between node types V_1 and V_l [77].

As shown above, Metapath2Vec uses random walks guided by meta-paths to generate heterogeneous node sequences rich in semantics and structural information,

and then it designs a heterogeneous skip-gram model to preserve the node v's proximity to its neighborhood nodes. It uses Equation 3.4 to calculate the similarity between a node and its neighbors.

Based on Metapath2Vec, several variants have been proposed. For instance, BHIN2vec [78] proposes an extension to the skip-gram technique in order to balance the influence of different relation types on node embeddings. HHNE [79] performs random walks in hyperbolic spaces. Another method, Hin2Vec [80], combines first-order and high-order relations to capture the heterogeneity of graphs and carries out multiple relation prediction tasks to learn the node embeddings jointly. This thesis employs Metapath2Vec since its code is freely available to use and easy to adapt to homogeneous and heterogeneous graphs.

3.3. Language Model-Based Representation Learning

Conventional Natural Language Processing (NLP) requires feature engineering, thus considerable expertise. Likewise, representation learning aims to automatically learn representations of row data to be fed as input to classification or prediction tasks as useful information. A typical example of representation learning approaches is deep learning [81] since the output of each intermediary layer can be considered as a representation of the input data. Deep learning algorithms represent each object with a low-dimensional, real-valued dense vector called distributed representation or embedding. When two objects are projected into a unified low-dimensional semantic space, the geometric distance between these objects in the semantic space indicates their semantic relatedness. Therefore, the semantic meaning of an object is related to its close neighbors [82].

One of the exciting approaches to distributed representation in NLP is Neural Probabilistic Language Model (NPLM) [83]. A language model is created to predict the joint probability of word sequences. In NPLM, a distributed vector is assigned for each word at first, then using a neural network, the next word is predicted. In the

end, it learns how to model the joint probability of sentences and outputs the word embeddings as learned parameters. Some of the famous methods inspired by NPLM are word2vec [73], GloVe [84], and fastText [85], all of which are very efficient to train.

With the growing number of data in the corpus and the parameters, ELMo [86], and BERT [87] models have become more popular. Rather than assigning a fixed distributed vector to each word as in word2vec, ELMo and BERT use multilayer neural networks to calculate dynamic representations of words. Most importantly, they consider a word's context while learning the representation.

BERT-like models are also called Pre-trained Language Models (PLM) because they are pre-trained through text modeling objectives on large corpora and fine-tuned the model on downstream tasks. BERT is a bi-directional transformer that can learn a language representation from a large amount of unlabeled textual data and then fine-tune it for certain machine learning applications. ProtBERT [88, 89] is a protein sequence-pre-trained model with a masked language modeling objective. It is based on the BERT model, which is self-supervised and pre-trained on a vast corpus of protein sequences. This implies it was pre-trained on raw protein sequences only, with no human labeling, and used an automatic mechanism to generate inputs and labels from those sequences. To initialize the distributional representations of proteins, we leverage the ProtBERT model.

We adopted the ChemBERTa [90] model, which was pre-trained on 10M PubChem substances using BPE tokenization [91], to initialize the distributional representations of chemicals. ChemBERTa is based on the transformer and pre-trained with masked language modeling, similar to ProtBERT.

To initialize the distributional representations of diseases and side effects, we employed the BioBERT [92] model. It is a domain-specific language representation model pre-trained on large-scale biomedical corpora, the same as the preceding models.

The DeepDTA model revealed the difficulties of modeling proteins using their sequences, which was one of the study's most intriguing findings [16]. In the affinity prediction task, the CNN module was not as good at describing proteins when modeled separately by CNN-based modules as it was with SMILES.

The WideDTA model utilizes the protein sequence and ligand SMILES string to address this issue by representing them as a set of words. Moreover, to better represent the interaction, the WideDTA model incorporated many kinds of text-based information. Although the addition of text-based information such as protein domain and motif information, the protein representation and thus prediction performance were unable to create a statistically significant gain in predictive power. Inspired by this, we expand the DeepDTA model by integrating more information and creating better and semantically meaningful representations for chemicals and proteins.

3.4. Evaluation Metrics

To evaluate the performance of the graph-based and language-based models, we utilize four metrics: Concordance Index (CI) [93], R-squared (R^2), Mean Square Error (MSE), Root Mean Square Error (RMSE), and Cosine Similarity. These five metrics will be introduced in the following.

3.4.1. Concordance Index

In the DTA prediction task, the Concordance Index (CI) can be utilized as an evaluation metric for prediction correctness, as described in KronRLS [94]. For continuous values, CI is a ranking metric. The CI is used to determine whether two random drug-target combinations' projected binding affinity values were predicted in the same order as their actual values or not. CI is calculated as

$$CI = \frac{1}{Z} \sum_{s_i > s_j} h(b_i - b_j), \tag{3.5}$$

where b_i is the prediction value for the larger affinity s_i, b_j is the prediction value for the smaller affinity s_j, Z is a normalization constant equal to the number of data pairs with different label values. The CI spans from 0 to 1.0, with 1.0 indicating perfect prediction accuracy and 0.5 indicating a random predictor. The Heaviside function, $h(x)$, is [93] step function, which is discontinued and it is defined as

$$h(x) = \begin{cases} 1, & X > 0 \\ 0.5, & X = 0 \\ 0, & X < 0. \end{cases} \tag{3.6}$$

3.4.2. R-squared

R-squared R^2 is a statistical measure that quantifies the proportion of variation explained by an independent variable or variables in a regression model for a dependent variable. R^2 reveals how much the variation of one variable explains the variance of the second variable, whereas correlation explains the strength of the relationship between an independent and dependent variable. So, if a model's R^2 is 0.5, the model's inputs can explain nearly half of the observed variation. Unlike MSE and RMSE, R^2 is a scale-invariant prediction quality metric. R^2 is computed as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - p_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{3.7}$$

where n corresponds to the number of samples, $\bar{y}$ is the mean of the actual values, y_i is the actual data, and the p_i is the prediction.

3.4.3. Mean Square Error

The MSE is a widely used statistic for continuous prediction error. It is used in regression tasks to see how near the fitted line is to the actual data points, which is shown by connecting the estimated values. We use the MSE as a metric because

drug-target binding affinity prediction is also a regression task. MSE is formulated as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2, \tag{3.8}$$

where n corresponds to the number of samples, y_i is the actual data, and p_i is the prediction.

3.4.4. Root Mean Square Error (RMSE)

RMSE is the average distance between data points and the fitted line, calculated as the square root of MSE

$$RMSE = \sqrt{MSE}. \tag{3.9}$$

3.4.5. Cosine Similarity

The cosine similarity is a measure that can be used to compare vectors. It measures the similarity using the cosine of the angle between two vectors in a multidimensional space. It is formulated as

$$similarity(x, y) = \frac{x \cdot y}{|x||y|}, \tag{3.10}$$

where $|x|$ is the Euclidean norm of a vector $x = (x_1, x_2, ..., x_3)$ defined as $\sqrt{x_1^2 + x_2^2 + ... + x_p^2}$. Similarly, $|y|$ is the Euclidean norm of vector y. The closer the cosine value to 1, the smaller the angle and the greater the match between vectors. Therefore, one can expect to see an increase in cosine similarity value between two vectors if their embeddings are getting closer, *i.e.,* they are being similar as it is shown in Word2Vec [73].

4. MATERIALS AND METHODS

This thesis predicts the binding affinity score of drug-target pairs by using heterogeneous graphs generated with the existing information and the language-based information extracted from chemical and protein sequences. We divided the study into five stages to do so, and this chapter summarizes these stages. In Stage 1, we compiled data from several online databases; in Stage 2, we assembled the compiled data and extracted useful information from them. In Stage 3, we created homogeneous and heterogeneous graphs using assembled data. Then in Stage 4, we learned the distributed vector representations of proteins and ligands using homogeneous and heterogeneous graphs with and without several language models. Finally, in Stage 5, we predict the affinity scores of drug-target pairs and evaluate the performance of our model using the evaluation metrics explained in Section 3.4.

4.1. Dataset Compilation

For the chemicals, we employ six databases and extract drug-related information; unique IDs (CID and DrugBank ID), SMILES strings, interacting drugs, interacting targets, side effects, and diseases. For the proteins, we use four databases and extracted protein-related information; unique IDs (Entrez Gene ID and UniProt ID), amino acid sequences, interacting proteins, interacting drugs, and diseases.

Figure 4.1 shows these eight databases with the corresponding extracted information. This section provides details and up-to-date statistical data about eight databases, namely BindingDB, ChEMBL, CTD, DrugBank, PubChem, SIDER, STRING, and UniProt.

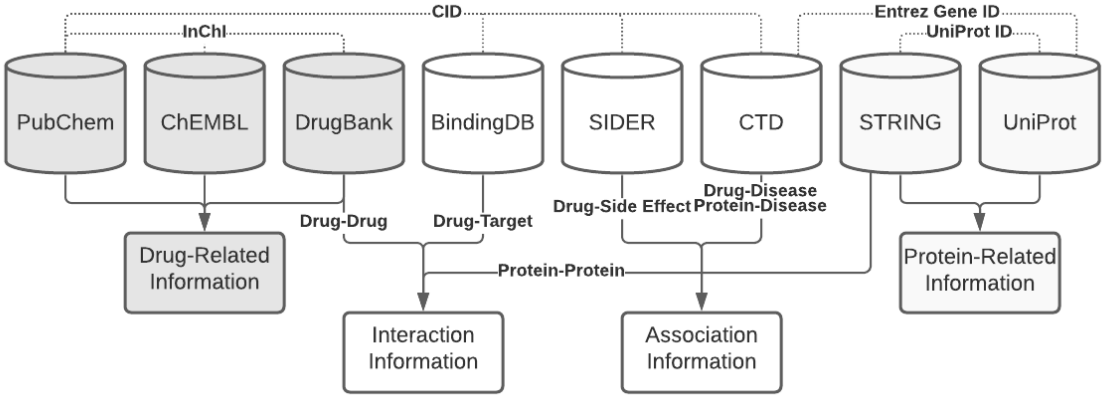

Figure 4.1. Compiled databases.

4.1.1. BindingDB

BindingDB [95] is an online database of drug-target interactions and measured binding affinity values. As of February 2021, BindingDB contains 41,328 entries with DOI, 2,114,159 binding affinity data for 928,022 small molecules, and 8,202 protein targets. Binding affinity is usually expressed in measures such as inhibition constant (K_i), dissociation constant (K_d), and the half-maximal inhibitory concentration (IC50). For that purpose, 2,077,458 K_i (nM), K_d (nM), and IC50 (nM) values were compiled from the database within the scope of the thesis.

To benchmark the performance of graph-based representational learning, we use BDB dataset [39] that is filtered from the BindingDB database. 24,404 binding affinities were observed for all pairs of 924 ligand and 480 proteins, measured by the pK_d value (log-transformed kinase dissociation constant) [39]. pK_d correlates positively with the binding strength, and the value varies between 1.6 and 13.3. The number of ligands with strong binding affinity values is 3,428 (*i.e.*, $pK_d \geq 7$) according to literature [96]. Figure 4.2 illustrates the distribution of the binding affinity values of proteins - ligand pairs in the BDB dataset.

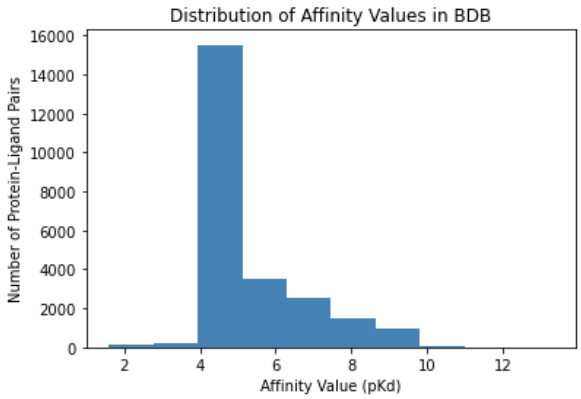

Figure 4.2. Distribution of binding affinity values in BDB.

4.1.2. ChEMBL

ChEMBL [97,98] is a manually curated chemical database of molecules with drug-like properties and biological activity, which is maintained by the European Molecular Biology Laboratory (EMBL). The ChEMBL database contains bioactivity data of active pharmaceutical ingredients, which are reported with K_i, K_d, and IC50 values. ChEMBL examines how small molecules interact with target proteins and how these compounds affect cells and whole organisms. Moreover, ChEMBL includes information about the 2D structure, calculated molecular properties, and the ADMET properties such as in vivo absorption, distribution, metabolism, excretion, and toxicity of small molecules. As of May 2020, there are 1,941,412 chemicals in the ChEMBL database and we extract 10,935 drugs' 1D text representations, *i.e.,* SMILES strings from ChEMBL. Using SMILES strings of drugs, we trained the CNN-based model that encodes each SMILES string character as numbers and the language-based model that leverages the similarity between drugs' SMILES strings.

4.1.3. Comparative Toxicogenomics Database

The Comparative Toxicogenomics Database [99] (CTD), is a database that provides information on manually curated chemical–gene/protein interactions, chemical-disease, and gene-disease relationships. CTD has several categories of data. These are chemicals, diseases, chemical-disease relationships, and gene-disease relationships.

Table 4.1. CTD statistics (02.2021).

Data	Number
Chemicals	16,572
Diseases	7,246
Chemical-Disease Relation	2,958,797
Gene-Disease Relation	28,253,189

Table 4.1 shows the available number of chemicals, diseases, and the association between diseases and chemicals/genes, as of February 2021. Since some diseases are common for both chemicals and genes, we extracted 2,958,797 chemical-disease relationships and 28,253,189 gene-disease relationships within the scope of this thesis in order to generate heterogeneous graphs.

4.1.4. DrugBank

DrugBank [100–102] is an online database that contains drugs, drug-related data (chemical, pharmacological, and pharmaceutical), and target-related data (sequence, structure, and pathway) as both bioinformatics and cheminformatics sources. As of January 2021, DrugBank contains 14,350 drugs. In the scope of this thesis, we extract 2,682,158 drug-drug interaction information of 14,350 drugs. We create homogeneous and heterogeneous graphs and generate representations for ligands using drugs and the relation between drugs.

4.1.5. PubChem

PubChem [103] is an online chemistry database that contains small molecules, nucleotides, as well as information on chemical structures, identifiers, chemical and physical properties. As of February 2021, the current statistics in the database are shown in Table 4.2.

Table 4.2. PubChem statistics (02.2021)

Data	Number
Compounds	109,487,163
Substances	270,034,522
Proteins	96,280
Genes	89,655

Table 4.3. SIDER statistics (10.2015).

Side Effects	Drugs	Drug-Side Effect Pairs
5,868	1,430	139,756

Since PubChem contains a considerable amount of data, other chemical databases map their entries with PubChem's Compound ID number (CID). In the context of this thesis, we leverage the PubChem CID of each chemical and map with the entries of BindingDB, SIDER, and CTD. Moreover, PubChem contains the International Chemical Identifier (InChI) of chemicals that textually identifies chemical substances. Similar to CID, We used chemicals' InChIs to relate the same chemicals across other databases, ChEMBL and DrugBank, that do not share any common IDs.

4.1.6. SIDER

SIDER [104, 105] is a database of drugs that have entered the market and their recorded adverse drug reactions extracted from public documents and prospectuses. Side effect frequency, drug classification, side effect classification, and drug-target relationships are presented in a computer-readable format. SIDER uses the Anatomical Therapeutic Chemical (ATC) Classification System, a drug classification system that classifies the active substances of drugs according to the organ or system they act on and their therapeutic, pharmacological, and chemical properties. The Medical Dictionary codes side effects for Regulatory Activities (MedDRA) terminology, a clinically validated medical terminology thesaurus. The statistics as of October 2015 are shown in Table 4.3. We used 5,868 side effects and extracted 139,756 drug-side effect relations.

4.1.7. STRING

Search Tool for the Retrieval of Interacting Genes/Proteins, STRING [106], is a biological database of known and predicted protein-protein interactions. The interactions include direct (physical) and indirect (functional) associations; they stem from computational prediction, from knowledge transfer between organisms, and interactions aggregated from other (primary) databases.

The STRING database compiles information from several sources such as computational prediction methods, public text collections, laboratory experiments, and other databases. According to the statistics provided as of August 2021, the STRING database contains 67,592,464 proteins and 296,567,750 interactions at the highest security (score $\geq$ 0.900), 834,790,438 interactions with high security or better (score $\geq$ 0.700), medium security or better 3,112,520,562 interactions (score $\geq$ 0.400), and a total of 20,052,394,041 interactions. We leverage the protein-protein interaction data and create a homogeneous database to learn the representations for proteins.

4.1.8. UniProt

The Universal Protein Source (UniProt) [7] is an essential resource for available protein information, including protein sequence and functional information. According to current statistics, as of September 2021, the total number of sequence entries is 565,928. Using amino acid sequences of proteins, we trained the CNN-based model that encodes each amino acid sequence character as numbers and the language-based model that leverages the similarity between proteins' amino acid sequences.

4.2. Data Assembling

This thesis aims to represent chemicals and proteins better. Therefore, chemical and protein-related data from several online databases are compiled. However, this task is not straightforward because of the vast amount of available data and the dif-

ficulty of standard mapping information across the databases. Therefore, we analyze the available data in the databases mentioned in Section 4.1 and map the related information using common identifiers or unique sequences. Figure 4.1 shows used databases as well as the IDs used to map corresponding databases.

4.2.1. Chemical Related Information

DrugBank, PubChem, and ChEMBL databases are the primary chemical resources, and we mainly focus on them. We compile 14,350 drugs from DrugBank and retrieve their DrugBank IDs and International Chemical Identifiers (InChI) as an initial step. Using InChIs, we connect the data in DrugBank with PubChem and ChEMBL databases and retrieve information about 10,935 different drugs. With 10,935 drugs, we extract 2,196,820 drug-drug relation information from the DrugBank database. Using the PubChem Compound ID number (CID) information available in the PubChem database, we map the PubChem to SIDER and CTD databases. We extract 5,452 distinct side effects from the SIDER database and 115,871 drug-side effect association information for 1,003 drugs. We extract 7,086 distinct diseases from the CTD database and 995,654 drug-disease association information for 3,387 drugs. Finally, we map DrugBank to ChEMBL and compile SMILES representations of 10,935 drugs using the InChI keys.

Apart from the already existing information, we create a new relation named drug-drug similarity (DDS). Using the compiled SMILES representations of drugs from the ChEMBL database, DDS data is obtained by calculating the similarity of these representations to each other according to the Jaccard Similarity. In order to find similar SMILES sequences, we use the Byte Pair Encoding (BPE) algorithm [91]. The BPE approach is utilized to identify the language unit vocabulary of chemicals. This method is commonly employed for discovering the tokens of a language in the field of NLP. The BPE algorithm divides SMILES sequences into language units [107]. Then,

the similarity of the drugs is calculated in pairs according to the Jaccard Criterion as

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \times 100. \tag{4.1}$$

By calculating the Jaccard Similarities of all drug-drug pairs, we obtain the values shown in Figure 4.3. Accordingly, drug pairs in the dataset with a similarity value greater than 58 were determined to be similar, with a threshold value determined to cover at least 10% of the whole data, resulting in 2,924,270 drug-drug similarity values for 6,963 distinct drugs.

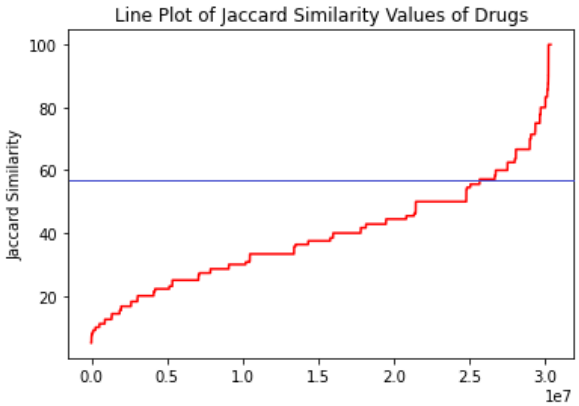

Figure 4.3. Pairwise drug-drug Jaccard similarities.

4.2.2. Protein Related Information

UniProt and STRING databases are the primary protein resources used in this thesis. We compile 505,250 proteins from the UniProt database, and more specifically, we compile 202,160 proteins belonging to the Homo sapiens and their amino acid sequences. We map 18,876 proteins to the STRING database and extract 183,746 protein-protein interaction information using the UniProt ID from the UniProt database. Finally, using the UniProt ID and Entrez Gene ID, we map the UniProt database to CTD and extract 32,495 protein-disease association information for 32,169 proteins and 126 distinct diseases.

Like the chemicals, in addition to protein-related information, we create a new relation named protein-protein similarity (PPS). Using the compiled amino acid se-

quences of proteins from the UniProt database, PPS data is obtained by calculating the similarity of these representations to each other according to the Jaccard Similarity of language units found by the BPE algorithm [107]. Then, the similarity of the proteins was calculated in pairs according to the Jaccard Criterion using the formula given in Equation 4.1. By calculating the Jaccard Similarities of all protein-protein pairs, we obtained the values shown in Figure 4.4. Accordingly, protein pairs in the dataset with a similarity value greater than 9 were determined to be similar to each other, with a threshold value was determined to cover at least 11% of the whole data. In the end, we created 528 protein-protein similarity values for 465 distinct proteins.

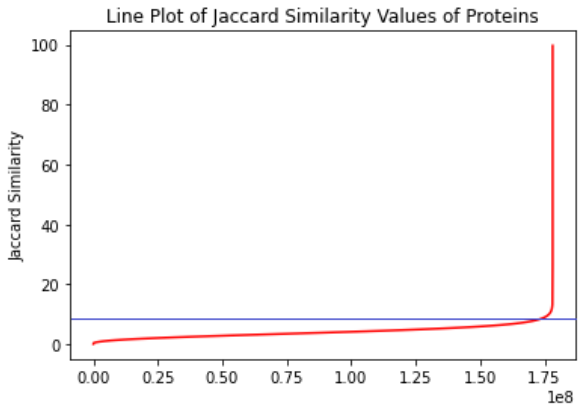

Figure 4.4. Pairwise protein-protein Jaccard similarities.

4.3. Graph Creation

Graphs are used to model complex relations between entities and preserve structural information. In this study, we combine different types of data sources in order to predict binding affinity values between drug-target pairs by using the relevant drug and target-related information. For that purpose, we employ homogeneous and heterogeneous graphs which contain one type of node and relation or many types of nodes and relations, respectively. As discussed in the previous section, we compiled various data, extracted useful information, and then preprocessed them to use with the graph structure. Using PyTorch-Geometric [108], we create graph structures that define node and edge types. Then, we load preprocessed data into graphs and generate positive and negative links between nodes. Observing the relations in the graphs, we sampled the metapaths as mentioned in Section 3.2.

Table 4.4. Homogeneous graph details.

Model Name	Number of Nodes	Metapaths	Number of Edges
Model (1), (2)	10935	Drug Interacts with Drug	2196820
Model (3), (4)	6963	Drug Similar to Drug	5848540
Model (5), (6)	12675	Protein Interacts with Protein	124536
Model (7), (8)	465	Protein Similar to Protein	1056

Table 4.4 shows details about the Drug-Drug Interaction (DDI), Drug-Drug Similarity (DDS), Protein-Protein Interaction (PPI), and Protein-Protein Similarity (PPS) homogeneous graphs. For DDI relation, we created Model (1) and Model (2) graphs with 10,935 drug nodes and 2,196,820 edges between these nodes. While learning the representation of the graph, we follow the paths from drugs to drugs, so the name of the metapath is "Drug interacts with the drug." In the case of DDS relation, we created Model (3) and Model (4) graphs with 6,963 drug nodes and 5,848,540 edges between these nodes. While learning the representation of the graph, similarly, we follow the paths from drugs to drugs, so the name of the metapath is "Drug similar to the drug."

Similarly, we created Model (5) and Model (6) graphs for the PPI relation with 12,675 protein nodes and 124,536 edges between these nodes and the paths from proteins to proteins, with the metapath "Protein interacts with the protein." Finally, we created Model (7) and Model (8) graphs for the PPS relation with 465 protein nodes and 1056 edges between these nodes, and the paths from proteins to proteins, with the metapath, "Protein similar to the protein."

Creating heterogeneous graphs needs more effort since the relations between them can be complicated, and the number of information drastically increases the run time. After creating homogeneous graphs, we create heterogeneous graphs using available disease association information.

Table 4.5. Heterogeneous graphs with disease information.

Model Name	Num. of Drug Nodes	Num. of Protein Nodes	Num. of Disease Nodes	Metapaths	Num. of Edges
Model (9) Model (10)	3387	-	7086	Drug Assoc. with Disease Disease Assoc. with Drug	1991308
Model (11) Model (12)	-	32169	126	Protein Assoc.with Disease Disease Assoc.with Protein	64990
Model (13) Model (14)	3387	32169	7087	Drug Assoc. with Disease Disease Assoc.with Protein Protein Assoc.with Disease Disease Assoc.with Drug	2056298

Table 4.5 shows details about the Drug-Disease Association (DDiA), Protein-Disease Association (PDiA), and Drug-Disease-Protein Association (DDiPA) heterogeneous graphs. For DDiA relation, we created Model (9) and Model (10) graphs with 3,387 drug nodes, 7,086 disease nodes, and 21,991,308 edges between these nodes. While learning the representation of the graph, we follow the paths from drugs to diseases and vice versa, so the metapaths are "Drug associates with a disease," and "Disease reversely associates with a drug." The distinction between the two models is the integration of language models into Model (10) in order to enrich the representation of drugs. Like the DDiA, for the PDiA relation, we created Model (11) and Model (12) graphs with 32,169 protein nodes, 126 disease nodes, and 64,990 edges between these nodes. We follow the paths from proteins to diseases and the reverse paths, so the metapaths are "Protein associates with a disease," and "Disease reversely associates with a protein." Similar to Model (10), Model (12) also employs language models. Finally, we combine these two heterogeneous graphs and create one extensive heterogeneous network. Model (13) and Model (14) graphs for the DDiPA relation with 3,387 drug nodes, 6,963 protein nodes, 7087 disease nodes, and 2,056,298 edges between these three nodes. We follow the paths as drug-disease-protein-disease-drug, with the metapaths "Drug associates with a disease," "Disease reversely associates with

a protein," "Protein associates with a disease," and "Disease reversely associates with a drug." To sum up, Model (13) combines the relations between drugs-diseases and proteins-diseases, and Model (14) uses the same types of nodes and relations; however, it integrates language model approaches for biomolecules and diseases to enrich the drug-target representations.

As an additional experiment, we create another set of models and test the effectiveness of side effect information with these models. Table 4.6 gives details about the Drug-Side Effect Association (DSA) and DDI with DSA heterogeneous graphs. For the DSA graph, we created Model (15) and Model (16) graphs with 1.003 drug nodes, 5,451 side effect nodes, and 231,742 edges between these nodes. While learning the representation of the graph, we follow the paths from drugs to side effects and vice versa, so the metapaths are "Drug associates with a side effect" and "Side effect reversely associates with a drug." DDI is created as in the case of Model (1) and Model (2). For the combination of DDI and DSA graphs, we created Model (17) and Model (18) graphs with 10,932 drug nodes, 5,451 side effect nodes, and 2,427,902 edges between these nodes. We follow the paths from drugs to side effects, side effects to drugs, and drugs to drugs, so the metapaths are "Drug associates with a side effect," "Side effect reversely associates with a drug," and "Drug interacts with a drug." Both Model (16) and Model (18) integrate language models for drugs and side effects.

In order to incorporate language models into the graphs, we used three language model-based representation learning algorithms, namely ProtBERT, ChemBERTa, and BioBERT. ProtBERT is a transformer-based model with a masked language modeling objective with 30 layers and 16 attention heads. For each protein sequence, it generates a 1024-length vector as

$$PB^t = \left[\begin{array}{ccccc} pb_1^t & pb_2^t & pb_3^t & \dots & pb_1014^t \end{array} \right]. \tag{4.2}$$

Table 4.6. Heterogeneous graphs with side effect information.

Model Name	Number of Drug Nodes	Number of Side Effect Nodes	Metapaths	Number of Edges
Model (15), Model (16)	1003	5451	Drug Associates with Side Effect Side Effect Associates with Drug	231742
Model (17) Model (18)	10935	5451	Drug Associates with Side Effect Side Effect Associates with Drug Drug Interacts with Drug	2427902

ProtBERT is based on the BERT model, which is self-supervised and has been pre-trained on a large number of protein sequences. The ProtBERT model is used to initialize the distributional representations of proteins.

To initialize the distributed representations of chemicals, we used the ChemBERTa model pre-trained on 10M PubChem substances using BPE tokenization. ChemBERTa is built on the transformer, has 12 attention heads and six layers, and is pre-trained with masked language modeling.

The BioBERT model was used to initialize the distributed representations of diseases and side effects. Like the previous models, it is a domain-specific language representation model that's been pre-trained on large-scale biological corpora.

4.4. Learning Distributed Vector Representations

Representation learning has provided a novel learning paradigm for AI domains. The subject of representation learning is examined and demonstrated in this study, focusing on homogeneous and heterogeneous networks, which contain one type of node and relations or many types of nodes and relations, respectively. The objective of this problem is to automatically project nodes in networks into latent embedding space so that the network's structural and relational properties can be encoded and preserved.

Machine learning algorithms can then employ embeddings as features to address relevant downstream machine learning tasks.

Machine learning on graph-structured data is a ubiquitous task, and one of the challenges of this task is to find a way to represent the structure itself and the information it holds so that mainly used machine learning models can easily interpret it. In this thesis, we employ Metapath2Vec [70] model and learn the graph-based distributed representation vectors that reflect the semantic connections in the graph for nodes and edges and finally represent data that cannot be expressed in Euclidean space as a graph.

Metapath2Vec uses priori paths as its basic operating principle, so the paths should be defined in advance, as described in Section 3.2. Metapath2Vec evaluates different types of edge relations while finding meta paths, that is, paths going from one node to another node, provided that they do not repeat it, and makes semantic inferences using these edges and uses them in vector representations.

ProtBERT and ChemBERTa pre-trained language models are also employed to enrich the distributed vector representations of drugs and proteins learned from the Metapath2Vec algorithm. To do that, we initialize each biomolecule's embeddings with the corresponding vector from pre-trained language models, then start Metapath2Vec's training.

4.5. WideDeepDTA

WideDeepDTA aims to predict binding affinity values of drug-target pairs using information-rich representation vectors of corresponding drugs and targets. In order to generate rich representations, it combines text-based features with network and language model-based representation learning approaches. WideDeepDTA comprises three components, (i) CNN-based DeepDTA model, (ii) graph representation learning, and (iii) affinity prediction.

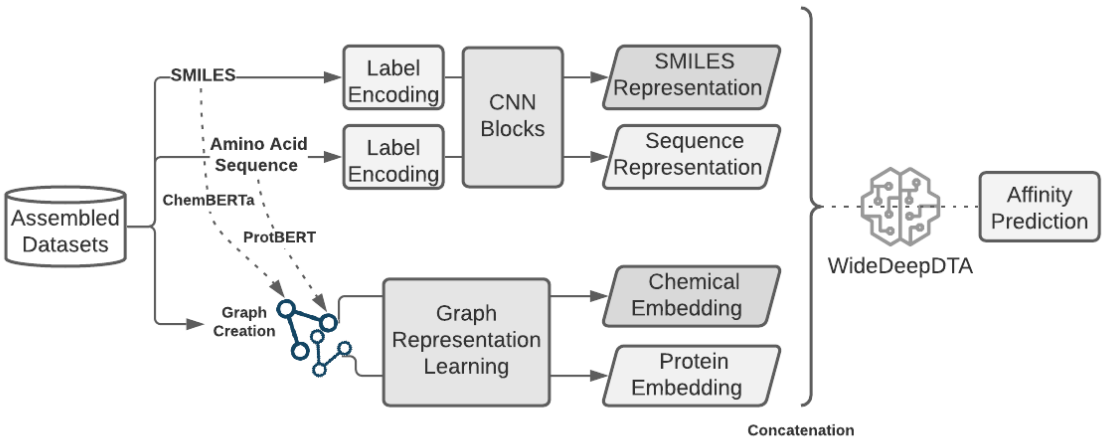

Figure 4.5. WideDeepDTA summarized.

4.5.1. DeepDTA Model

DeepDTA is an affinity prediction model that uses chemicals' SMILES strings and proteins' amino-acid sequences to represent biomolecules. To represent each character of SMILES and protein sequences, it leverages integer/label encoding with 64 and 25 unique letters, respectively. For instance, the SMILES string "$[C \;\; N \;\; = \;\; C \;\; = \;\; O]$" is encoded as $[1 \;\; 3 \;\; 63 \;\; 1 \;\; 63 \;\; 5]$. Once label encodings are generated, Keras' Embedding Layer is applied to represent characters as 128-dimensional dense vectors. Embeddings are fed into CNN blocks with three 1D-convolutional layers, followed by max-pooling layers. The final feature vectors of SMILES strings and proteins sequences are concatenated and fed into three fully connected neural networks.

4.6. Graph Representation Learning

WideDeepDTA modifies the DeepDTA model and makes it wider. To do that, it combines four input types, namely SMILES representation and protein sequence representation generated by CNN Blocks; chemical embeddings and protein embeddings generated by graph representation learning. Figure 4.5 illustrates the overall process. In this section, we explain the details of the WideDeepDTA model.

After compiling and assembling the datasets, we create our graphs. We define several homogeneous and heterogeneous graphs augmented with the drug and target-related relations explained in Chapter 5. An example heterogeneous graph is a Drug-Disease-Protein Association (DDiPA) graph. The DDiPA graph $DDiPA(V, \varepsilon))$ consists of a set of drugs $D = D_1, D_2, \ldots, D_n$ of n drug nodes, set of targets $T = P_1, P_2, \ldots, P_m$ of m protein nodes, and $Di = Di_1, Di_2, \ldots, Di_t$ of t disease nodes. DDiPA graph contains two types of edges. The first type of edge represents the association between drug and disease nodes, and the edge from this type is named "Drug Associates with a Disease." The second type of edge represents the association between protein and disease nodes, and the edge from this type is named "Protein Associates with a Disease."

Once the DDiPA graph is created, we load the edges (*i.e.,* positive edges) using drug-disease and protein disease association data extracted from the SIDER database. We split the graph into training and validation sets, in which the validation set contains at least 5% of the data. Then, we construct the negative samples by generating all possible pairs between drug-disease and protein-disease pairs, then selecting a sample from pairs as many as the number of positive edges in a validation set. Generating negative samples means introducing unknown interactions to the graph to prevent overfitting.

4.7. Experimental Setup

For the whole study, we used Python programming language and performed experiments on Google Colaboratory [109], and a machine with NVIDIA Tesla V100 GPU and Intel Xeon Scalable 6148 CPU.

As the training and test folds, we use the same setup in the DeepDTA model. We train each model 5 times with the training set folds, measure the performance on each test set, and report the average results on the BDB dataset. The BDB dataset contains five different training sets and corresponding four different test sets: warm, cold ligand, cold protein, and cold both. The cold ligand test set is used to identify the interactions

between unknown ligands and known proteins, *i.e.,* biomolecules that are not used during the training, biomolecules that are used during the training, respectively. The cold protein test set identifies the interactions between known ligands and unknown proteins. Identifying the interactions between known biomolecules form the warm test set, and finally, the interactions between unknown biomolecules form the cold both test set. However, some information related to biomolecules listed as unknown in the test set can be present in the graph creation part. We compute each model's CI, MSE, RMSE, and R^2 scores and report the standard deviation in parentheses.

4.8. Hyper-parameter Search

In order to learn distributed representation vectors of chemicals and proteins using the created graphs, we employed the Metapath2Vec model. The Metapath2Vec model uses the train set to train several hyper-parameters such as the embedding size of each embedding vector, walk length throughout the metapath, context size, which is considered for positive samples, and the number of walks to sample for each node in order to find the best model that generated the most convenient representations for the corresponding dataset. The model uses two initialization approaches; (i) random initialization and (ii) initialization using pre-trained language models. In the former case, each node is represented with 32-dimensional vectors initialized as samples from a uniform distribution over $[0, 1)$, and in the latter case, each node is represented with 32-dimensional vectors initialized by the pre-trained language models. For that purpose, we used ChemBERTa and ProtBERT pre-trained language models and loaded the embeddings of chemicals and proteins used in the creation of graphs. We also leveraged BioBERT, which is used by the disease names and side effect names associated with the drugs and proteins. In both cases, training goes for 100 epochs, and the model tests the best set of parameters over the validation set. We calculate the training and the validation loss to see the model's performance.

Since we aim to create better representation for drugs and proteins, we test the success of representations using the cosine similarity metric. Therefore, we first

calculate the cosine similarity of positive edges and the cosine similarity of negative edges. Then observe the difference between these two calculations to see whether the model is good at learning the representations or not. In the end, the model sticks with the graph that gives the highest total cosine similarity value.

Finally, we obtained the low-dimensional representation vectors for each chemical and protein node in the graph. Later on, we concatenate low-dimensional representation vectors of chemicals and proteins generated by the Metapath2Vec algorithm with the representation vectors generated by the CNN blocks. Like the DeepDTA model, combined representation is fed into three fully connected layers. We used 1024 nodes in the first two fully connected layers, followed by a dropout layer of rate 0.1. The last fully-connected layer contains 512 nodes, followed by the output layer. The proposed model that combines CNN and network-based methods is illustrated in Figure 4.6.

4.9. Affinity Prediction

Similar to the DeepDTA, the WideDeepDTA model handles the drug-target binding affinity prediction task as a regression problem, using Rectified Linear Unit (ReLU) [110] as the activation function and MSE as the loss function. With MSE, the model aims to maximize the difference between the actual and the predicted value during training. In the case of the CNN-based model, the Adam optimization algorithm [111] is used. In the case of the network-based model SparseAdam is employed.

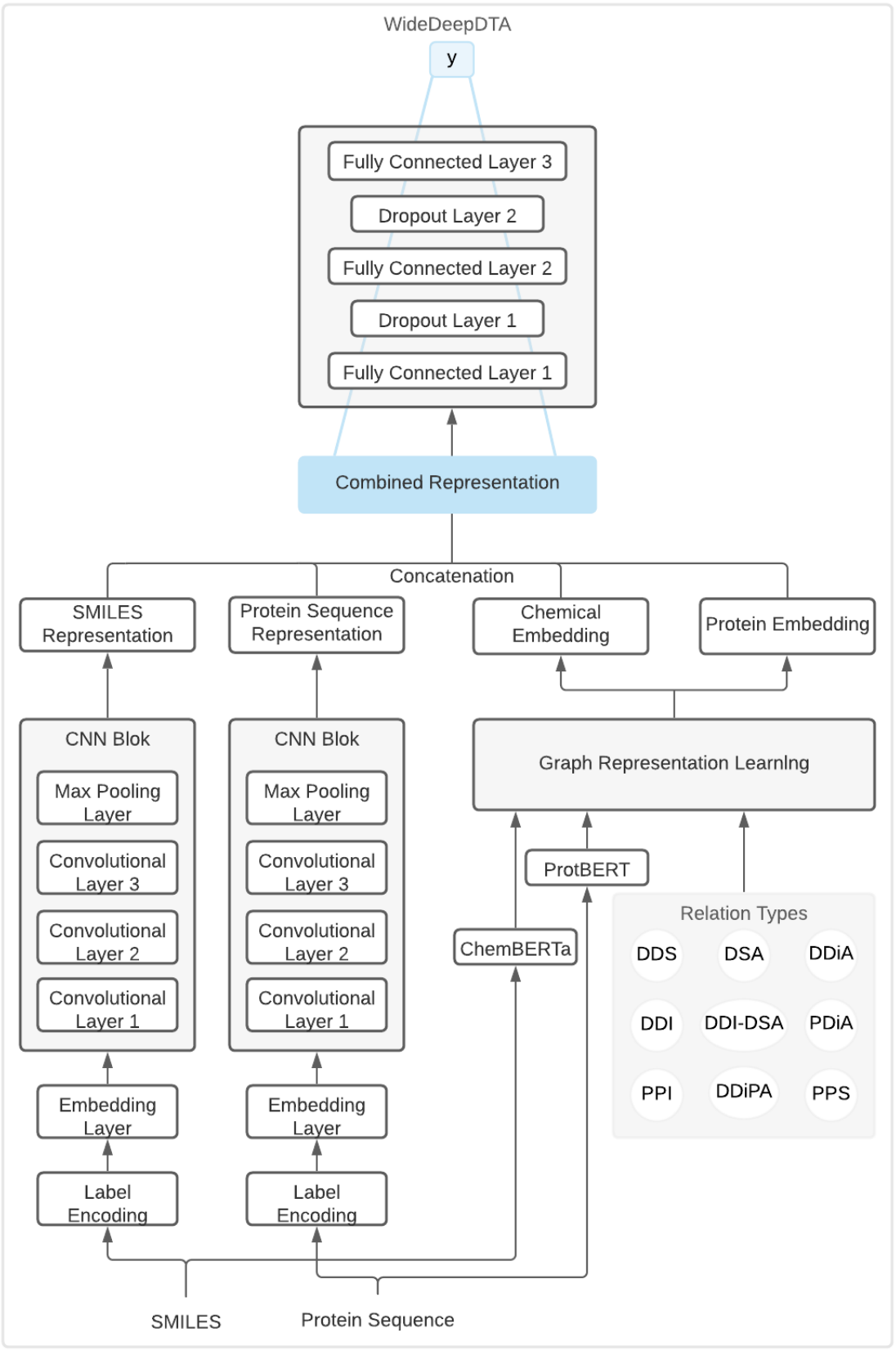

Figure 4.6. WideDeepDTA in details.

5. RESULTS

5.1. Evaluation

Representation vectors for chemicals and proteins were obtained using eighteen different models, and then these vectors were evaluated in the drug-target affinity task with the WideDeepDTA model using the BDB dataset. The DeepDTA [16] model represents proteins using amino acid sequences and chemicals using characters of the SMILES notations. In this study, the DeepDTA model has been updated, as shown in Figure 4.6, to take as input the representation vectors containing additional information in the generated graphs. The model's performance is measured by the CI, MSE, RMSE, and R^2 metrics.

5.2. Model Comparisons

This section lists the details of experiments and the trained models. Then, compare the results with the DeepDTA model.

Ligand representation through homogeneous graphs. First, we generate homogeneous graphs with only one node and edge type, *i.e.*, drugs, and interaction between drugs, respectively. Then test the WideDeepDTA performance for the ligand representation with and without empowered homogeneous graphs.

Table 5.1. Scores of DDI and DDS models on warm test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.888 (0.009)	0.781 (0.028)	0.288 (0.021)	0.536 (0.012)
Model (1)	**0.896 (0.009)**	0.777 (0.026)	0.295 (0.036)	0.542 (0.033)
Model (2)	0.890 (0.014)	0.782 (0.017)	0.287 (0.014)	0.535 (0.014)
Model (3)	0.893 (0.005)	0.787 (0.020)	0.280 (0.017)	0.529 (0.017)
Model (4)	0.890 (0.006)	**0.789 (0.008)**	**0.278 (0.011)**	**0.527 (0.010)**

- **Model (1)**: A Drug-Drug Interaction (DDI) graph is created, the nodes of which are formed by all drugs (D) and the edges by interactions (D-D) between these drugs.
- **Model (2)**: A second DDI graph consists of drug nodes, and D-D edges is created similarly. Moreover, it is empowered with the knowledge of initial embeddings of the ChemBERTa model.
- **Model (3)**: A Drug-Drug Similarity (DDS) graph is created, the nodes of which are formed by all drugs (D) and the edges by Jaccard similarity between these drugs.
- **Model (4)**: A second DDS graph is created similar to Model (3). The model is initialized with the ChemBERTa embeddings.

We first test the impact of ligand representation using homogeneous graphs by creating two different models with two different versions. Model (1) represents each drug with a 32-dimensional vector in which they are initialized as samples from a uniform distribution over $[0, 1)$ and trained by the Metapath2Vec model on DDI relation, and Model (2) represents each drug with the same dimensional size vector; however, Metapath2Vec model's embeddings are initialized as ChemBERTa embeddings of corresponding ligands. On the other hand, Model (3) and Model (4) are trained on DDS relation with the same setup, *i.e.,* Model (3) is initialized randomly, and Model (4) is initialized with ChemBERTa embeddings.

Table 5.2. Scores of DDI and DDS models on cold protein test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.759 (0.006)	0.315 (0.049)	1.085 (0.146)	1.040 (0.146)
Model (1)	**0.779 (0.030)**	**0.375 (0.104)**	**0.992 (0.207)**	**0.991 (0.101)**
Model (2)	0.768 (0.009)	0.327 (0.072)	1.064 (0.154)	1.029 (0.077)
Model (3)	0.775 (0.018)	0.340 (0.082)	1.045 (0.174)	1.019 (0.086)
Model (4)	0.774 (0.015)	0.327 (0.075)	1.065 (0.162)	1.029 (0.079)

The scores regarding the comparison of these four models are shown in Table 5.1, Table 5.2, Table 5.3, and Table 5.4. For the relation-oriented tables please refer to the Appendix A. Considering the results shown in Table 5.1;

- DDS (Model 3-Model 4) models outperform the DeepDTA model; thus, using text-based information in the graph improves the overall performance.
- Using PLMs for drugs improves the performance compared to the random initialization since it increases the performance for three out of four metrics for both the DDI and DDS model (Model 2-Model 4).
- DDI models and DDS models perform similarly on the warm test set of BDB, *i.e.*, their trends are the same for the same performance metrics. For instance, the R^2 score for models with PLMs (Model 2-Model 4) is higher than the randomly initialized models (Model 1-Model 3); likewise, MSE and RMSE values are lower for the same case. Therefore, if the drug-drug interaction data is scarce, we recommend employing the text-based similarity measures between two drugs while using empowered homogeneous graphs on the warm test set of BDB.

Table 5.3. Scores of DDI and DDS models on cold ligand test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	**0.687 (0.096)**	**0.039 (0.243)**	**1.448 (0.939)**	**1.152 (0.348)**
Model (1)	0.664 (0.057)	-0.053 (0.210)	1.472 (0.650)	1.187 (0.249)
Model (2)	0.640 (0.066)	-0.125 (0.180)	1.548 (0.602)	1.225 (0.220)
Model (3)	0.651 (0.085)	-0.139 (0.132)	1.603 (0.645)	1.242 (0.246)
Model (4)	0.666 (0.064)	-0.102 (0.330)	1.502 (0.615)	1.202 (0.239)

While learning the graph-based representations with DDI and DDS models, we leverage only the drug-related information. So, we do not integrate any protein-related information and set all the protein representations as zeros. To predict the affinity values model employs the protein embeddings learned from the CNN-based part using the protein sequence and the ligand embeddings learned from CNN-based and graph-based parts. Considering the results shown in Table 5.2, all the models outperform the DeepDTA model for the interaction prediction task with unknown proteins. Therefore, integrating drug information into the graph for the known ligands increases the performance. However, compared to the improvement on the warm test set, using PLMs lowered the evaluation scores on the cold protein test set.

Taking into account the results shown in Table 5.3 and Table 5.4, adding drug-related information using the homogeneous or empowered homogeneous graph did not improve the performance for the cold ligand and cold both test cases. This result is expected since cold both test sets are challenging and show low scores for all the metrics.

On the other hand, the cold both test set's results are slightly better than the cold ligand test set's results due to the performance improvement of cold proteins. As opposed to the previous observation, PLMs improved the performance of the DDS models on both of the test sets.

Table 5.4. Scores of DDI and DDS models on cold both test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	**0.554 (0.047)**	**-0.154 (0.164)**	**2.007 (1.223)**	**1.356 (0.410)**
Model (1)	0.554 (0.044)	-0.287 (0.184)	2.013 (0.767)	1.395 (0.260)
Model (2)	0.495 (0.037)	-0.496 (0.211)	2.348 (0.978)	1.504 (0.294)
Model (3)	0.519 (0.036)	-0.390 (0.235)	2.278 (1.069)	1.467 (0.356)
Model (4)	0.536 (0.068)	-0.274 (0.269)	2.021 (0.950)	1.388 (0.308)

To sum up, 7 out of 16 models outperformed the DeepDTA model, and 4 out of 8 models improved homogeneous graphs with language models. To predict the affinity values between known biomolecules, we recommend using Model (4), which integrates language models and text-based features into homogeneous graphs since it shows the best performance. In the case of cold ligand and cold both test sets, the usage of PLMs with homogeneous graphs is promising; however, it still has room for improvement. Moreover, if the interaction data is not available for drugs, we suggest using sequence similarity between drugs in the homogeneous graphs since both models perform similarly and provide more information to the representation than DDI in the warm test set.

Homogeneous protein representation. Second, we generate homogeneous graphs with only one node and edge type; proteins, and interaction between proteins, respectively. Then test the WideDeepDTA performance for the protein representation with and without empowered homogeneous graphs.

- **Model (5)**: A Protein-Protein Interaction (PPI) graph is created, the nodes of which are formed by all proteins (P) and the edges by interactions (P-P) between these proteins.
- **Model (6)**: A second PPI graph consists of protein nodes, and P-P edges are created similar to Model (5). Moreover, it is empowered with the knowledge of initial embeddings of the ProtBERT model.
- **Model (7)**: A Protein-Protein Similarity (PPS) graph is created, the nodes of

which are formed by all proteins belonging to the human species (P) and the edges formed by the Jaccard similarities (P-P) between the amino acid sequences of these proteins.

- **Model (8)**: A second PPS graph is created similar to Model (7). The model is initialized with the ProtBERT model.

First, we test the impact of protein representation using homogeneous graphs by creating two different models with two different versions. Model (5) represents each protein with a 32-dimensional vector in which they are initialized as samples from a uniform distribution over $[0, 1)$ and trained by the Metapath2Vec model on PPI relation, and Model (6) represents each protein with the same dimensional size vector. However, the Metapath2Vec model's embeddings are initialized as ProtBERT embeddings of corresponding proteins. (For the detailed results please refer to the Appendix B and see Table B.1 and Table B.2.) On the other hand, Model (7) and Model (8) are trained on PPS relation with the same setup. Model (7) is initialized randomly, and Model (8) is initialized with ProtBERT embeddings.(For the detailed results please refer to the Appendix B and see Table B.3 and Table B.4.)

The results regarding the comparison of these four models are shown in Table 5.5, Table 5.6, Table 5.7, and Table 5.8. Considering the results shown in Table 5.5;

- Homogeneous graphs empowered by language models and text-based features generated (Model 8) adds more information to the representation of proteins compared to the homogeneous graphs empowered by only language models (Model 6) on warm test set.
- Homogeneous and language model-empowered homogeneous graphs generated by PPI relation perform similarly on the warm test set. So, language models do not improve the performance of PPI models on the warm test set.

Table 5.5. Scores of PPI and PPS models on warm test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.888 (0.009)	0.781 (0.028)	0.288 (0.021)	0.536 (0.012)
Model (5)	0.888 (0.009)	0.773 (0.025)	0.299 (0.020)	0.546 (0.018)
Model (6)	0.888 (0.009)	0.777 (0.012)	0.295 (0.017)	0.543 (0.015)
Model (7)	**0.893 (0.006)**	0.775 (0.018)	0.296 (0.018)	0.544 (0.016)
Model (8)	0.892 (0.008)	**0.785 (0.019)**	**0.283 (0.015)**	**0.532 (0.015)**

Table 5.6. Scores of PPI and PPS models on cold protein test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.759 (0.006)	0.315 (0.049)	1.085 (0.146)	1.040 (0.146)
Model (5)	0.765 (0.017)	0.323 (0.073)	1.069 (0.144)	1.031 (0.071)
Model (6)	0.759 (0.019)	0.299 (0.084)	1.109 (0.176)	1.050 (0.083)
Model (7)	**0.783 (0.010)**	**0.362 (0.047)**	**1.008 (0.115)**	**1.002 (0.056)**
Model (8)	0.773 (0.022)	0.339 (0.086)	1.046 (0.171)	1.019 (0.081)

Considering the results shown in Table 5.6, and comparing with the previous observations, language model-empowered graphs generated by PPS relation did not work well in the case of cold proteins. It may be caused by the long sequence of proteins and ProtBERT's inability to generate good representations for proteins as initial embedding. So, adding language-based information to the graph does not improve the performance of the unseen proteins, and we recommend continuing with the simpler version (Model 7) since adding PLM information for proteins adds memory and execution time overheads due to the longer amino acid sequences. However, homogeneous graphs and homogeneous graphs empowered by language models and text-based features still outperform the DeepDTA model.

Table 5.7. Scores of PPI and PPS models on cold ligand test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.687 (0.096)	0.039 (0.243)	1.448 (0.939)	1.152 (0.348)
Model (5)	0.674 (0.095)	-0.031 (0.290)	1.561 (1.059)	1.192 (0.373)
Model (6)	0.698 (0.083)	-0.020 (0.347)	1.542 (1.146)	1.179 (0.390)
Model (7)	0.675 (0.083)	-0.067 (0.211)	1.538 (0.816)	1.204 (0.298)
Model (8)	**0.728 (0.046)**	**0.155 (0.162)**	**1.162 (0.451)**	**1.060 (0.193)**

Table 5.8. Scores of PPI and PPS models on cold both test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.554 (0.047)	-0.154 (0.164)	2.007 (1.223)	1.356 (0.410)
Model (5)	0.551 (0.049)	-0.269 (0.165)	2.186 (1.310)	1.419 (0.415)
Model (6)	0.561 (0.031)	-0.363 (0.279)	2.288 (1.303)	1.457 (0.407)
Model (7)	0.557 (0.036)	-0.260 (0.090)	2.033 (0.908)	1.393 (0.305)
Model (8)	**0.598 (0.061)**	**-0.065 (0.297)**	**1.590 (0.510)**	**1.246 (0.191)**

Taking into consideration the results shown in Table 5.7 and Table 5.8, adding protein-related information using the language model and text-based feature-empowered homogeneous graph (Model 8) improve the performance and outperforms DeepDTA for the cold ligand and cold both test cases. To sum up, 6 out of 16 models outperformed the DeepDTA model, and 5 out of 8 models improved homogeneous graphs with language models. To predict the affinity values in all kinds of test sets, we recommend using Model (8), which integrates language models and text-based features into homogeneous graphs since it shows the best performance. Moreover, the usage of PLMs with homogeneous graphs is promising since it shows a performance improvement. However, PLMs do not affect the model performance when the test set contains unknown proteins since they hardly represent the long sequences.

Heterogeneous representations with disease information. After completing experiments with homogeneous graphs, we continue with the heterogeneous graph experiments. We start with generating heterogeneous graphs with several nodes and edge

types. We represent drugs, diseases, and proteins as nodes; drug-disease associations and protein-disease associations as edges. Then test WideDeepDTA performance for both ligand and protein representations.

- **Model (9)** A Drug-Disease Association (DDiA) graph is created, with drug (D) and disease (Di) nodes, and the edges by association (D-Di) between them.
- **Model (10)** A DDiA graph is created similar to Model (9), this time with the knowledge of initial embeddings of ChemBERTa and BioBERT models for SMILES sequences and disease names, respectively.
- **Model (11)** A Protein-Disease Association (PDiA) graph is created, the nodes of which are formed by all proteins (P) and diseases (Di) and the edges by association (P-Di) between these proteins and diseases.
- **Model (12)** A PDiA graph is created similar to Model (11), this time with the knowledge of initial embeddings of ProtBERT and BioBERT models for protein amino acid sequences and disease names, respectively.
- **Model (13)** A Drug-Disease-Protein Association (DDiPA) graph is created, the nodes of which are formed by all drugs (D), diseases (Di), proteins (P), and the edges by association (D-Di-P) between these drugs, proteins, and diseases.
- **Model (14)** A DDiPA graph is created similar to Model (13), with the initial embeddings of ChemBERTa, BioBERT, and ProtBERT models for SMILES sequences, protein amino acid sequences, and disease names, respectively.

First, we test the impact of disease information on the protein and ligand representations using heterogeneous graphs by creating two different models with two different versions. Model (9) represents each ligand with a 32-dimensional vector in which they are sampled from a uniform distribution over $[0, 1)$ and trained by the Metapath2Vec model on DDiA relation, and Model (10) represents each ligand with the same dimensional size vector; however, Metapath2Vec model's embeddings are initialized as ChemBERTa embeddings of corresponding ligands, and BioBERT embeddings of related diseases' name. (For the detailed results please refer to the Appendix C and see Table C.1 and Table C.2).

Table 5.9. Scores of DDiA, PDiA, DDiPA models on warm test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.888 (0.009)	0.781 (0.028)	0.288 (0.021)	0.536 (0.012)
Model (9)	0.895 (0.013)	0.786 (0.023)	0.282 (0.023)	0.530 (0.022)
Model (10)	**0.896 (0.006)**	0.784 (0.019)	0.284 (0.015)	0.533 (0.014)
Model (11)	0.881 (0.007)	0.759 (0.028)	0.317 (0.034)	0.563 (0.030)
Model (12)	0.895 (0.007)	**0.794 (0.015)**	**0.271 (0.014)**	**0.520 (0.014)**
Model (13)	0.878 (0.009)	0.753 (0.020)	0.325 (0.025)	0.570 (0.022)
Model (14)	0.882 (0.008)	0.755 (0.022)	0.323 (0.029)	0.568 (0.026)

Model (11) represents each protein with a 32-dimensional vector in which they are initialized as samples from a uniform distribution over $[0, 1)$ and trained by Metapath2Vecmodel on PDiA relation, and Model (12) represents each protein with the same dimensional size vector; however, Metapath2Vec model's embeddings are initialized as ProtBERT embeddings of corresponding proteins, and BioBERT embeddings of related diseases' name. (For the detailed results please refer to the Appendix C and see Table C.3 and Table C.4). Finally, we combine these two separate heterogeneous graphs and create one large heterogeneous graph, DDiPA, to test the effect of heterogeneous and empowered heterogeneous graphs on the ligand and protein representations together. Similarly, DDiPA has two models, Model (13) and Model (14). On Model (13), Metapath2Vec is trained on randomly initialized embeddings of each drug, protein, and disease, whereas, on Model (14), Metapath2Vec is trained on embeddings loaded using PLMs of ChemBERTa, ProtBERT, and BioBERT for each drug, protein, and disease, respectively.

Table 5.10. Scores of DDiA, PDiA, DDiPA models on cold protein test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.759 (0.006)	0.315 (0.049)	1.085 (0.146)	1.040 (0.146)
Model (9)	0.784 (0.010)	0.349 (0.072)	1.032 (0.161)	1.013 (0.080)
Model (10)	0.762 (0.017)	0.238 (0.233)	1.211 (0.411)	1.087 (0.172)
Model (11)	**0.785 (0.011)**	**0.377 (0.078)**	**0.987 (0.166)**	**0.990 (0.082)**
Model (12)	0.702 (0.046)	0.030 (0.169)	1.349 (0.546)	1.140 (0.222)
Model (13)	0.749 (0.018)	0.268 (0.061)	1.156 (0.141)	1.073 (0.065)
Model (14)	0.752 (0.013)	0.282 (0.073)	1.138 (0.173)	1.064 (0.081)

The results regarding the comparison of these six models are shown in Table 5.9, Table 5.10, Table 5.11, and Table 5.12. Considering the results shown in Table 5.9;

- Considering the DDiA alone, PLMs did not improve the drug representations according to all the metrics except the CI value since it gives the highest score for the warm test set. On the other hand, PLMs increase the performance score since language model-empowered heterogeneous graphs of PDiA perform better than simple heterogeneous graphs.
- Both DDiA and PDiA models perform better than the DeepDTA model so that we can observe the importance of disease-related information on the heterogeneous graphs. However, using the protein-disease and drug-disease information in the same heterogeneous graph lowers the overall performance compared to PDiA and DDiA models.

Considering the results shown in Table 5.10, adding PLMs to the heterogeneous graphs lowers the performance of DDiA and PDiA models; however, it increases the performance of DDiPA model on the cold protein test set. Still, DDiPA models underperform the other two models and the DeepDTA model.

Table 5.11. Scores of DDiA, PDiA, DDiPA models on cold ligand test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.687 (0.096)	0.039 (0.243)	1.448 (0.939)	1.152 (0.348)
Model (9)	0.690 (0.050)	-0.025 (0.202)	1.381 (0.443)	1.162 (0.177)
Model (10)	0.695 (0.056)	**0.066 (0.122)**	**1.294 (0.476)**	**1.120 (0.202)**
Model (11)	0.695 (0.036)	-0.010 (0.201)	1.340 (0.353)	1.149 (0.143)
Model (12)	0.702 (0.046)	0.030 (0.169)	1.349 (0.546)	1.140 (0.222)
Model (13)	0.664 (0.087)	-0.040 (0.097)	1.479 (0.637)	1.190 (0.250)
Model (14)	**0.703 (0.033)**	0.030 (0.082)	1.319 (0.385)	1.137 (0.161)

Table 5.12. Scores of DDiA, PDiA, DDiPA models on cold both test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.554 (0.047)	-0.154 (0.164)	2.007 (1.223)	1.356 (0.410)
Model (9)	0.567 (0.047)	-0.187 (0.177)	1.873 (0.792)	1.341 (0.271)
Model (10)	0.564 (0.031)	-0.201 (0.095)	1.939 (0.854)	1.360 (0.298)
Model (11)	0.543 (0.068)	-0.475 (0.506)	2.140 (0.576)	1.449 (0.201)
Model (12)	0.568 (0.027)	**-0.060 (0.146)**	**1.689 (0.680)**	**1.272 (0.265)**
Model (13)	0.533 (0.063)	-0.300 (0.161)	2.096 (0.890)	1.414 (0.310)
Model (14)	**0.578 (0.050)**	-0.275 (0.123)	2.005 (0.743)	1.392 (0.260)

Regarding the results shown in Table 5.11, as opposed to above mentioned observations, adding PLM to the heterogeneous graphs increased the performance of all of the three models on the cold ligand test set. Similarly, DDiPA performance is the lowest one, except for the CI metric.

Taking into account the results shown in Table 5.12, adding disease-related information using the empowered heterogeneous graph generated by PDiA and DDiPA improve the performance for the cold both test cases. However, the empowered heterogeneous graph generated by DDiA did not show any improvement. We can conclude that combining two different heterogeneous data sources does not improve the overall performance as much as using these sources separately. Moreover, disease information is essential; thus, it can be employed to enrich the ligand and protein representations.

To sum up, 11 out of 24 models outperformed the DeepDTA model, and 9 out of 12 models improved heterogeneous graphs with language models. To predict the affinity values in warm and cold both test sets, we recommend using Model (9) and Model (12). Since combining two different heterogeneous data sources does not improve the overall performance as much as using these sources separately. For the cold test sets, integrating disease information into heterogeneous graphs increase the ability to predict affinity values between unknown biomolecules. Moreover, the usage of PLMs with heterogeneous graphs of PDiA and DDiPA is promising since we observe performance improvement.

Heterogeneous representations with side effect information. As the second heterogeneous graph experiment, we generate heterogeneous graphs with drugs and side effects as nodes, the interaction between drugs, and the association between drugs and side effects as edges. Then test WideDeepDTA performance for ligand representations.

- **Model (15)** A DSA (Drug-Side Effect Association) graph is created, the nodes of which are formed by all drugs (D), and side effects (S) and the edges by association (D-S) between these drugs and side effects.
- **Model (16)** A DSA graph is created similar to Model (15), this time initialized with the embeddings of ChemBERTa and BioBERT models for SMILES sequences and side effect names, respectively.
- **Model (17)** A DDI-DSA (Drug-Drug Interaction & Drug-Side Effect Association) graph is created, the nodes of which are formed by all drugs (D), and side effects (S) and the edges by association (D-D and D-S) between these drugs and side effects.
- **Model (18)** A DDI-DSA graph is created similar to Model (17), initialized with the embeddings of ChemBERTa and BioBERT models for SMILES sequences and side effect names.

Table 5.13. Scores of DSA and DDI-DSA models on warm test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.888 (0.009)	0.781 (0.028)	0.288 (0.021)	0.536 (0.012)
Model (15)	0.898 (0.009)	**0.791 (0.016)**	**0.275 (0.022)**	**0.524 (0.021)**
Model (16)	0.889 (0.011)	0.776 (0.021)	0.295 (0.021)	0.543 (0.019)
Model (17)	0.892 (0.013)	0.781 (0.021)	0.289 (0.037)	0.537 (0.034)
Model (18)	**0.899 (0.008)**	0.785 (0.017)	0.285 (0.026)	0.533 (0.025)

First, we test the impact of side effect information on the ligand representations using heterogeneous graphs by creating two different models with two different versions. Model (15) represents each ligand with a 32-dimensional vector in which they are initialized as samples from a uniform distribution over $[0, 1)$ and trained by the Metapath2Vec model on DSA relation, and Model (16) represents each ligand with the same dimensional size vector; however, Metapath2Vec model's embeddings are initialized as ChemBERTa embeddings of corresponding ligands, and BioBERT embeddings of corresponding side effects' name. (For the detailed results please refer to the Appendix D and see Table D.1 and Table D.2). Model (17) represents each drug with a 32-dimensional vector in which they are initialized as samples from a heterogeneous graph that includes the combination of DDI and DSA relations. Furthermore, Model (18) represents each drug with the same dimensional size vector; however, the Metapath2Vec model's embeddings are initialized as ChemBERTa embeddings of corresponding ligands and BioBERT embeddings of corresponding side effects' names. (For the detailed results please refer to the Appendix D and see Table D.3 and Table D.4). With one large heterogeneous graph, we test the effect of heterogeneous and empowered heterogeneous graphs on the ligand representations.

Table 5.14. Scores of DSA and DDI-DSA models on cold protein test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.759 (0.006)	0.315 (0.049)	1.085 (0.146)	1.040 (0.146)
Model (15)	0.774 (0.015)	0.358 (0.059)	1.015 (0.131)	1.005 (0.064)
Model (16)	0.766 (0.022)	0.323 (0.097)	1.076 (0.209)	1.032 (0.102)
Model (17)	0.769 (0.019)	0.352 (0.084)	1.026 (0.177)	1.009 (0.086)
Model (18)	**0.778 (0.014)**	**0.365 (0.069)**	**1.006 (0.162)**	**1.000 (0.081)**

Table 5.15. Scores of DSA and DDI-DSA models on cold ligand test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.687 (0.096)	0.039 (0.243)	1.448 (0.939)	1.152 (0.348)
Model (15)	0.683 (0.040)	-0.074 (0.265)	1.443 (0.461)	1.186 (0.190)
Model (16)	0.664 (0.068)	-0.196 (0.196)	1.621 (0.555)	1.258 (0.199)
Model (17)	0.688 (0.080)	-0.034 (0.245)	1.435 (0.640)	1.171 (0.250)
Model (18)	**0.711 (0.057)**	**0.217 (0.153)**	**1.082 (0.397)**	**1.022 (0.194)**

The results regarding the comparison of these four models are shown in Table 5.13, Table 5.14, Table 5.15, and Table 5.16. Considering the results shown in Table 5.13;

- In the case of DSA relation, the addition of PLM information to the heterogeneous graph did not improve the performance. However, when we combine DSA relation with DDI relation, it increases the performance for all four metrics.
- Side effect information contributes more to the ligand representation when it is the only relation type. So, using it with DDI in the heterogeneous graph did not improve the performance on the warm test set.

Considering the results shown in Table 5.14, Table 5.15, and Table 5.16, empowered heterogeneous graphs with DDI and DSA relations increased the performance. However, similar to the previous observation, PLMs did not increase the performance of the DDI Models.

Table 5.16. Scores of DSA and DDI-DSA models on cold both test set of BDB.

Model	CI	R^2	MSE	RMSE
DeepDTA	0.554 (0.047)	-0.154 (0.164)	2.007 (1.223)	1.356 (0.410)
Model (15)	0.561 (0.045)	-0.278 (0.293)	1.943 (0.647)	1.375 (0.228)
Model (16)	0.580 (0.052)	-0.445 (0.284)	2.241 (0.905)	1.470 (0.281)
Model (17)	0.569 (0.041)	-0.303 (0.244)	2.071 (0.927)	1.406 (0.306)
Model (18)	**0.607 (0.061)**	**-0.018 (0.108)**	**1.608 (0.617)**	**1.245 (0.241)**

To sum up, 9 out of 16 models outperformed the DeepDTA model, and 4 out of 8 models improved heterogeneous graphs with language models. Results of using DSA and DDI-DSA models with heterogeneous and empowered heterogeneous graphs suggest integrating side effect-related information while using all the test sets except the warm test set. For all the test sets, the usage of PLMs with heterogeneous graphs generated by DDI-DSA shows remarkable performance improvement compared to the heterogeneous graphs generated by DSA relation only.

6. CONCLUSION

The drug design pipeline is a labor-intensive, time-consuming, and expensive process that relies heavily on discovering novel drug-target interactions. An important step in this pipeline is to find the high-affinity chemical-protein pairs in pre-clinical studies. Computer-aided drug design is a promising research area that uses high-performance computers to simulate this drug design process. This simulation predicts binding affinity values for chemical-protein pairs with successful in silico experiments, speeding up the drug development process and reducing resource consumption.

Recently, deep learning approaches have been utilized to predict binding affinities due to the increased availability of publicly available data in drug-target-related databases. Most studies concentrate on using the primary information of biomolecules, such as text representation, while some concentrate on integrating several data sources using heterogeneous graph structures, in which both models show performance improvement.

This thesis proposed WideDeepDTA, a drug-target affinity prediction framework that leverages heterogeneous networks empowered with text-based biomolecule representations. Given homogeneous or heterogeneous networks containing multiple types of biological entities, relationships between these entities, and pre-trained biomolecular language models, WideDeepDTA learns low-dimensional biomolecule representations and predicts chemical-protein affinities.

We constructed heterogeneous networks that contain drugs, proteins, diseases, and side effects in WideDeepDTA and enriched these networks with language models and 1D biomolecule sequence similarity information in the experiments. We evaluated learned feature representations on BDB dataset using warm, cold ligand, cold protein, and cold both test tests. The experiments highlight that;

(i) A novel DTA prediction framework in which homogeneous and heterogeneous networks are empowered with biomolecule sequence similarity and language models is proposed. We use 1D representations of biomolecules since they are information-rich and, unlike 2D molecular graphs or 3D structures, easily acquired and processed [112].

(ii) Employing disease and side effect relations in the graphs and empowering these relations with language models yields the largest improvement over baseline, especially for unseen biomolecules.

(iii) Using 1D similarity of biomolecules outperforms biomolecule interaction information in homogeneous graphs, indicating that 1D similarity of chemicals, which is easy to obtain, can compensate for the need for experimental drug-drug interaction data for affinity prediction. This would be useful, especially when predicting the affinities of a novel chemical with other proteins.

(iv) Using ligand-based relations in the graphs increases the WideDeepDTA model's ability to generate better representations for unseen proteins.

(v) Using protein-based relations in the graphs increases the WideDeepDTA model's ability to generate better representations for unseen ligands.

(vi) Experiments performed with the language model-empowered heterogeneous graph of drug-drug interaction with drug-side effect association relations gives the best score for the cold ligand and cold both test sets. Thus, increasing the heterogeneity of the graph with ligand-oriented information increases the WideDeepDTA model's ability to generate better representations for ligands.

(vii) Experiments demonstrate that model with the language model-empowered heterogeneous graph of protein-disease association gives the best scores on the warm and cold protein test sets. Thus, increasing the heterogeneity of the graph with protein-oriented information increases the WideDeepDTA model's ability to generate better representations for proteins.

(viii) Heterogeneous networks are empowered with pre-trained language models and improved performance for 28 out of 36 models. In general, WideDeepDTA outperforms the DeepDTA model for 33 out of 72 models. This shows the WideDeepDTA's promising ability to represent chemicals and proteins better.

6.1. Future Directions

Integrating sequence similarity-based information to graphs improved the models on the chemical-protein affinity prediction task. However, finding text-based similar protein pairs is challenging and time-consuming due to long amino acid sequences. The limited number of protein-protein similarity data limits WideDeepDTA's overall performance.

We showed that using additional information in the heterogeneous graph increases the WideDeepDTA's representation performance. However, we also found out that adding unrelated information to a graph decreases the performance. Considering the ligands, adding protein-related information to a simple graph decreases the model's ability to represent ligands compared to the simple graph.

Overall, heterogeneous networks empowered with language models give the best results compared to baseline. Also, the experiments revealed limitations of heterogeneous networks' with pre-trained language models to represent the long protein sequences [113] compared to short SMILES strings of drugs.

Moreover, integrating text-based features into graphs is promising in the chemical-protein prediction task. We encourage further studies to integrate more text-based features into the graphs. One further improvement would be handling the 1D sequences of biomolecules as documents, representing the words of these documents as entities in the graph. Then, applying natural language processing techniques with nodes of graphs would improve performance considering the simple Jaccard similarity method's demonstrated success on the homogeneous graph.

WideDeepDTA enables the generation of better representations of unknown proteins through ligand-based relations and vice versa. Introducing diverse biomolecule-based information to the graphs would enable more informative representations of novel chemicals and proteins, thus supporting the drug discovery pipeline.

REFERENCES

1. Csermely, P., T. KorcsmÁRos, H. J. Kiss, G. London and R. Nussinov, "Structure and Dynamics of Molecular Networks: A Novel Paradigm of Drug Discovery: A Comprehensive Review", *Pharmacology & Therapeutics*, Vol. 138, No. 3, pp. 333–408, 2013.

2. Hughes, J. P., 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.

3. Huggins, D. J., W. Sherman and B. Tidor, "Rational Approaches to Improving Selectivity in Drug Design", *Journal of Medicinal Chemistry*, Vol. 55, No. 4, pp. 1424–1444, 2012.

4. Wen, H., H. Jung and X. Li, "Drug Delivery Approaches in Addressing Clinical Pharmacology-Related Issues: Opportunities and Challenges", *The Aaps Journal*, Vol. 17, No. 6, pp. 1327–1340, 2015.

5. Bolton, E. E., Y. Wang, P. A. Thiessen and S. H. Bryant, "PubChem: Integrated Platform of Small Molecules and Biological Activities", *Annual Reports in Computational Chemistry*, Vol. 4, pp. 217–241, Elsevier, 2008.

6. Law, V., C. Knox, Y. Djoumbou, T. Jewison, a. C. Guo, Y. Liu, A. Maciejewski, D. Arndt, M. Wilson, V. Neveu, A. Tang, G. Gabriel, C. Ly, S. Adamjee, Z. T. Dame, B. Han, Y. Zhou and D. S. Wishart, "Drugbank 4.0: Shedding New Light on Drug Metabolism", *Nucleic Acids Research*, Vol. 42, No. D1, pp. D1091–D1097, 2013.

7. Poux, S., C. N. Arighi, M. Magrane, A. Bateman, C.-H. Wei, Z. Lu, E. Boutet, H. Bye-A-Jee, M. L. Famiglietti, B. Roechert and T. UniProt Consortium, "On

Expert Curation and Scalability: UniProtKB/Swiss-Prot as a Case Study", *Bioinformatics*, Vol. 33, No. 21, pp. 3454–3460, 07 2017.

8. Matthews, H., J. Hanison and N. Nirmalan, ""Omics"-Informed Drug and Biomarker Discovery: Opportunities, Challenges and Future Perspectives", *Proteomes*, Vol. 4, No. 3, p. 28, 2016.

9. Yu, W. and A. D. Mackerell, "Computer-Aided Drug Design Methods", *Antibiotics*, pp. 85–106, Springer, 2017.

10. Yamanishi, Y., M. Kotera, M. Kanehisa and S. Goto, "Drug-Target Interaction Prediction from Chemical, Genomic and Pharmacological Data in an Integrated Framework", *Bioinformatics*, Vol. 26, No. 12, pp. I246–I254, 2010.

11. Liu, Y., M. Wu, C. Miao, P. Zhao and X.-L. Li, "Neighborhood Regularized Logistic Matrix Factorization for Drug-Target Interaction Prediction", *PLoS Computational Biology*, Vol. 12, No. 2, p. E1004760, 2016.

12. Nascimento, A. C., R. B. PrudÊNcio and I. G. Costa, "A Multiple Kernel Learning Algorithm for Drug-Target Interaction Prediction", *BMC Bioinformatics*, Vol. 17, No. 1, p. 46, 2016.

13. Keum, J. and H. Nam, "SELF-BLM: Prediction of Drug-Target Interactions via Self-Training SVM", *PloS one*, Vol. 12, No. 2, p. E0171839, 2017.

14. Greenside, P., M. Hillenmeyer and A. Kundaje, "Prediction of Protein-Ligand Interactions from Paired Protein Sequence Motifs and Ligand Substructures", *Pacific Symposium on Biocomputing*, Vol. 23, World Scientific, 2017.

15. Kawasaki, Y. and E. Freire, "Finding a Better Path to Drug Selectivity", *Drug Discovery Today*, Vol. 16, No. 21-22, pp. 985–990, 2011.

16. Öztürk, H., A. Özgür and E. Ozkirimli, "DeepDTA: Deep Drug-Target Binding

Affinity Prediction", *Bioinformatics*, Vol. 34, No. 17, pp. i821–i829, 2018.

17. Mcculloch, W. S. and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity", *The Bulletin of Mathematical Biophysics*, Vol. 5, No. 4, pp. 115–133, 1943.

18. Lecun, Y., Y. Bengio and G. Hinton, "Deep Learning", *Nature*, Vol. 521, No. 7553, pp. 436–444, 2015.

19. Osman, A. H. and O. M. Barukub, "Graph-Based Text Representation and Matching: A Review of the State of the Art and Future Challenges", *IEEE Access*, Vol. 8, pp. 87562–87583, 2020.

20. Jiang, D., Z. Wu, C.-Y. Hsieh, G. Chen, B. Liao, Z. Wang, C. Shen, D. Cao, J. Wu and T. Hou, "Could Graph Neural Networks Learn Better Molecular Representation for Drug Discovery? A Comparison Study of Descriptor-Based and Graph-Based Models", *Journal of Cheminformatics*, Vol. 13, No. 1, pp. 1–23, 2021.

21. Jin, Y., J. Lu, R. Shi and Y. Yang, "EmbedDTI: Enhancing the Molecular Representations via Sequence Embedding and Graph Convolutional Network for the Prediction of Drug-Target Interaction", *Biomolecules*, Vol. 11, No. 12, p. 1783, 2021.

22. Shatkay, H., S. Brady and A. Wong, "Text as Data: Using Text-Based Features for Proteins Representation and for Computational Prediction of Their Characteristics", *Methods*, Vol. 74, pp. 54–64, 2015.

23. Kastritis, P. L. and A. M. Bonvin, "On the Binding Affinity of Macromolecular Interactions: Daring to Ask Why Proteins Interact", *Journal of the Royal Society Interface*, Vol. 10, No. 79, p. 20120835, 2013.

24. Wang, D. D., L. Ou-Yang, H. Xie, M. Zhu and H. Yan, "Predicting the Impacts

of Mutations on Protein-Ligand Binding Affinity Based on Molecular Dynamics Simulations and Machine Learning Methods", *Computational and Structural Biotechnology Journal*, Vol. 18, pp. 439–454, 2020.

25. Ling, Y., Y. an, M. Liu, S. A. Hasan, Y. Fan and X. Hu, "Integrating Extra Knowledge into Word Embedding Models for Biomedical NLP Tasks", *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 968–975, 2017.

26. Wang, H.-Y. and W.-Y. Ma, "Integrating Semantic Knowledge into Lexical Embeddings Based on Information Content Measurement", *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 509–515, 2017.

27. Luo, Y., X. Zhao, J. Zhou, J. Yang, Y. Zhang, W. Kuang, J. Peng, L. Chen and J. Zeng, "A Network Integration Approach for Drug-Target Interaction Prediction and Computational Drug Repositioning from Heterogeneous Information", *Nature Communications*, Vol. 8, No. 1, p. 573, 2017.

28. Moon, C., C. Jin, X. Dong, S. Abrar, W. Zheng, R. Y. Chirkova and A. Tropsha, "Learning Drug-Disease-Target Embedding (DDTE) from Knowledge Graphs to Inform Drug Repurposing Hypotheses", *Journal of Biomedical Informatics*, Vol. 119, p. 103838, 2021.

29. Keiser, M. J., B. L. Roth, B. N. Armbruster, P. Ernsberger, J. J. Irwin and B. K. Shoichet, "Relating Protein Pharmacology by Ligand Chemistry", *Nature Biotechnology*, Vol. 25, No. 2, p. 197, 2007.

30. Morris, G. M., R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell and A. J. Olson, "Autodock4 and Autodocktools4: Automated Docking with Selective Receptor Flexibility", *Journal of Computational Chemistry*, Vol. 30, No. 16, pp. 2785–2791, 2009.

31. Donald, B. R., *Algorithms in Structural Molecular Biology*, MIT Press, 2011.

32. Wan, F., L. Hong, a. Xiao, T. Jiang and J. Zeng, "NeoDTI: Neural Integration of Neighbor Information from a Heterogeneous Network for Discovering New Drug-Target Interactions", *Bioinformatics*, Vol. 35, No. 1, pp. 104–111, 2018.

33. Zheng, X., H. Ding, H. Mamitsuka and S. Zhu, "Collaborative Matrix Factorization with Multiple Similarities for Predicting Drug-Target Interactions", *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1025–1033, ACM, 2013.

34. Chen, X., M.-X. Liu and G.-Y. Yan, "Drug-Target Interaction Prediction by Random Walk on the Heterogeneous Network", *Molecular Biosystems*, Vol. 8, No. 7, pp. 1970–1978, 2012.

35. Wang, W., S. Yang, X. Zhang and J. Li, "Drug Repositioning by Integrating Target Information through a Heterogeneous Network Model", *Bioinformatics*, Vol. 30, No. 20, pp. 2923–2930, 2014.

36. Hu, P.-W., K. C. Chan and Z.-H. You, "Large-Scale Prediction of Drug-Target Interactions from Deep Representations", *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 1236–1243, 2016.

37. Tian, K., M. Shao, Y. Wang, J. Guan and S. Zhou, "Boosting Compound-Protein Interaction Prediction by Deep Learning", *Methods*, Vol. 110, pp. 64–72, 2016.

38. Hamanaka, M., K. Taneishi, H. Iwata, J. Ye, J. Pei, J. Hou and Y. Okuno, "CGBCS-DNN: Prediction of Compound-Protein Interactions Based on Deep Learning", *Molecular Informatics*, Vol. 36, No. 1-2, p. 1600045, 2017.

39. Özçelik, R., H. Öztürk, A. Özgür and E. Ozkirimli, "ChemBoost: A Chemical Language Based Approach for Protein-Ligand Binding Affinity Prediction", *Molecular Informatics*, Vol. 40, No. 5, p. 2000212, 2021.

40. Öztürk, H., E. Ozkirimli and A. Özgür, "WideDTA: Prediction of Drug-Target Binding Affinity", *arXiv preprint arXiv:1902.04166*, 2019.

41. Nguyen, T., H. Le and S. Venkatesh, "GraphDTA: Prediction of Drug-Target Binding Affinity Using Graph Convolutional Networks", *Biorxiv*, p. 684662, 2019.

42. Lee, I., J. Keum and H. Nam, "DeepConv-DTI: Prediction of Drug-Target Interactions via Deep Learning with Convolution on Protein Sequences", *PLoS Computational Biology*, Vol. 15, No. 6, p. E1007129, 2019.

43. Tsubaki, M., K. Tomii and J. Sese, "Compound-Protein Interaction Prediction with End-to-End Learning of Neural Networks for Graphs and Sequences", *Bioinformatics*, Vol. 35, No. 2, pp. 309–318, 2019.

44. Jiang, M., Z. Li, S. Zhang, S. Wang, X. Wang, Q. Yuan and Z. Wei, "Drug-Target Affinity Prediction Using Graph Neural Network and Contact Maps", *RSC Advances*, Vol. 10, No. 35, pp. 20701–20712, 2020.

45. Karimi, M., D. Wu, Z. Wang and Y. Shen, "Explainable Deep Relational Networks for Predicting Compound-Protein Affinities and Contacts", *Journal of Chemical Information and Modeling*, Vol. 61, No. 1, pp. 46–66, 2020.

46. Chen, L., X. Tan, D. Wang, F. Zhong, X. Liu, T. Yang, X. Luo, K. Chen, H. Jiang and M. Zheng, "TransformerCPI: Improving Compound-Protein Interaction Prediction by Sequence-Based Deep Learning with Self-Attention Mechanism and Label Reversal Experiments", *Bioinformatics*, Vol. 36, No. 16, pp. 4406–4414, 2020.

47. Agyemang, B., W.-P. Wu, M. Y. Kpiebaareh, Z. Lei, E. Nanor and L. Chen, "Multi-View Self-Attention for Interpretable Drug-Target Interaction Prediction", *Journal of Biomedical Informatics*, Vol. 110, p. 103547, 2020.

48. Yang, Z., W. Zhong, L. Zhao and C. Y.-C. Chen, "ML-DTI: Mutual Learning

Mechanism for Interpretable Drug-Target Interaction Prediction", *The Journal of Physical Chemistry Letters*, Vol. 12, No. 17, pp. 4247–4261, 2021.

49. Natarajan, N. and I. S. Dhillon, "Inductive Matrix Completion for Predicting Gene-Disease Associations", *Bioinformatics*, Vol. 30, No. 12, pp. I60–I68, 2014.

50. Zhao, T., Y. Hu, L. R. Valsdottir, T. Zang and J. Peng, "Identifying Drug-Target Interactions Based on Graph Convolutional Network and Deep Neural Network", *Briefings in Bioinformatics*, Vol. 22, No. 2, pp. 2141–2150, 2021.

51. Peng, J., Y. Wang, J. Guan, J. Li, R. Han, J. Hao, Z. Wei and X. Shang, "An End-to-End Heterogeneous Graph Representation Learning-Based Framework for Drug-Target Interaction Prediction", *Briefings in Bioinformatics*, 2021.

52. Thafar, M. A., R. S. Olayan, H. Ashoor, S. Albaradei, V. B. Bajic, X. Gao, T. Gojobori and M. Essack, "DTIGEMS+: Drug-Target Interaction Prediction Using Graph Embedding, Graph Mining, and Similarity-Based Techniques", *Journal of Cheminformatics*, Vol. 12, No. 1, pp. 1–17, 2020.

53. Grover, A. and J. Leskovec, "node2vec: Scalable Feature Learning for Networks", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016.

54. Thafar, M. A., R. S. Olayan, S. Albaradei, V. B. Bajic, T. Gojobori, M. Essack and X. Gao, "DTI2Vec: Drug-Target Interaction Prediction Using Network Embedding and Ensemble Learning", *Journal of Cheminformatics*, Vol. 13, No. 1, pp. 1–18, 2021.

55. Biggs, N., N. L. Biggs and B. Norman, *Algebraic Graph Theory*, 67, Cambridge University Press, 1993.

56. Hamilton, W. L., "Graph Representation Learning", *Synthesis Lectures on Artifical Intelligence and Machine Learning*, Vol. 14, No. 3, pp. 1–159, 2020.

57. Gardiner, A., "Homogeneoys Graphs", *Journal of Combinatorial Theory, Series B*, Vol. 20, No. 1, pp. 94–102, 1976.

58. Gardiner, A., "Homogeneity Conditions in Graphs", *Journal of Combinatorial Theory, Series B*, Vol. 24, No. 3, pp. 301–310, 1978.

59. Ihlefeldt, F. S., F. B. Pettersen, A. Von Bonin, M. Zawadzka and C. H. Görbitz, "The Polymorphs of l-Phenylalanine", *Angewandte Chemie International Edition*, Vol. 53, No. 49, pp. 13600–13604, 2014.

60. Sun, Y. and J. Han, "Mining Heterogeneous Information Networks: A Structural Analysis Approach", *ACM SIGKDD Explorations Newsletter*, Vol. 14, No. 2, pp. 20–28, 2013.

61. Ruiz-Sánchez, J. G., M. Pazos Guerra, D. Meneses and I. Runkle, "Primary Hyperaldosteronism: When to Suspect It and How to Confirm Its Diagnosis", *Endocrines*, Vol. 3, No. 1, pp. 29–42, 2022.

62. Mason, O. and M. Verwoerd, "Graph Theory and Networks in Biology", *Iet Systems Biology*, Vol. 1, No. 2, pp. 89–119, 2007.

63. Pavlopoulos, G. A., M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider and P. G. Bagos, "Using Graph Theory to Analyze Biological Networks", *Biodata Mining*, Vol. 4, No. 1, pp. 1–27, 2011.

64. Ricard, J., "Biological Networks", *New Comprehensive Biochemistry*, Vol. 40, pp. 57–81, 2006.

65. Daminelli, S., V. J. Haupt, M. Reimann and M. Schroeder, "Drug Repositioning through Incomplete Bi-Cliques in an Integrated Drug-Target-Disease Network", *Integrative Biology*, Vol. 4, No. 7, pp. 778–788, 2012.

66. Davis, A. P., C. G. Murphy, C. A. Saraceni-Richards, M. C. Rosenstein, T. C.

Wiegers and C. J. Mattingly, "Comparative Toxicogenomics Database: A Knowledgebase and Discovery Tool for Chemical-Gene-Disease Networks", *Nucleic Acids Research*, Vol. 37, No. Suppl_1, pp. D786–D792, 2009.

67. Bronstein, M. M., J. Bruna, Y. Lecun, A. Szlam and P. Vandergheynst, "Geometric Deep Learning: Going Beyond Euclidean Data", *IEEE Signal Processing Magazine*, Vol. 34, No. 4, pp. 18–42, 2017.

68. Asif, N. A., Y. Sarker, R. K. Chakrabortty, M. J. Ryan, M. H. Ahamed, D. K. Saha, F. R. Badal, S. K. Das, M. F. Ali, S. I. Moyeen, M. R. Islam and Z. Tasneem, "Graph Neural Network: A Comprehensive Review on Non-Euclidean Space", *IEEE Access*, 2021.

69. Wu, Z., S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks", *arXiv preprint arXiv:1901.00596*, 2019.

70. Dong, Y., N. V. Chawla and A. Swami, "metapath2vec: Scalable Representation Learning for Heterogeneous Networks", *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 135–144, 2017.

71. Pal, S., Y. Dong, B. Thapa, N. V. Chawla, A. Swami and R. Ramanathan, "Deep Learning for Network Analysis: Problems, Approaches and Challenges", *Milcom 2016-2016 IEEE Military Communications Conference*, pp. 588–593, 2016.

72. Mikolov, T., K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space", *arXiv preprint arXiv:1301.3781*, 2013.

73. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality", *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

74. Perozzi, B., R. Al-Rfou and S. Skiena, "Deepwalk: Online Learning of Social Rep-

resentations", *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.

75. Sun, Y., B. Norick, J. Han, X. Yan, P. S. Yu and X. Yu, "Pathselclus: Integrating Meta-Path Selection with User-Guided Object Clustering in Heterogeneous Information Networks", *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 7, No. 3, pp. 1–23, 2013.

76. Bengio, Y., A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp. 1798–1828, 2013.

77. Sun, Y. and J. Han, "Mining Heterogeneous Information Networks: Principles and Methodologies", *Synthesis Lectures on Data Mining and Knowledge Discovery*, Vol. 3, No. 2, pp. 1–159, 2012.

78. Lee, S., C. Park and H. Yu, "Bhin2vec: Balancing the Type of Relation in Heterogeneous Information Network", *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 619–628, 2019.

79. Wang, X., Y. Zhang and C. Shi, "Hyperbolic Heterogeneous Information Network Embedding", *Proceedings of the AAAI Conference on Artificial Intelligence*, 01, pp. 5337–5344, 2019.

80. Fu, T.-Y., W.-C. Lee and Z. Lei, "Hin2vec: Explore Meta-Paths in Heterogeneous Information Networks for Representation Learning", *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1797–1806, 2017.

81. Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

82. Liu, Z., Y. Lin and M. Sun, "Representation Learning and NLP", *Representation Learning for Natural Language Processing*, pp. 1–11, Springer, 2020.

83. Bengio, Y., R. Ducharme and P. Vincent, "A Neural Probabilistic Language Model", *Advances in Neural Information Processing Systems*, pp. 932–938, 2001.

84. Bojanowski, P., E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information", *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

85. Pennington, J., R. Socher and C. D. Manning, "Glove: Global Vectors for Word Representation", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

86. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding", *arXiv preprint arXiv:1810.04805*, 2018.

87. Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep Contextualized Word Representations", *arXiv preprint arXiv:1802.05365*, 2018.

88. "Protbert", `Https://Huggingface.Co/Rostlab/Prot_Bert`, accessed: 2022-16-01.

89. Elnaggar, A., M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik and B. Rost, "ProtTrans: Towards Cracking the Language of Life's Code through Self-Supervised Deep Learning and High Performance Computing", *Biorxiv*, 2020.

90. Chithrananda, S., G. Grand and B. Ramsundar, "Chemberta: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction", *arXiv preprint arXiv:2010.09885*, 2020.

91. Sennrich, R., B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units", *arXiv preprint arXiv:1508.07909*, 2015.

92. Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So and J. Kang, "BioBERT: A Pre-trained Biomedical Language Representation Model for Biomedical Text Mining", *Bioinformatics*, Vol. 36, No. 4, pp. 1234–1240, 2020.

93. Gönen, M. and G. Heller, "Concordance Probability and Discriminatory Power in Proportional Hazards Regression", *Biometrika*, Vol. 92, No. 4, pp. 965–970, 2005.

94. Pahikkala, T., 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.

95. Gilson, M. K., T. Liu, M. Baitaluk, G. Nicola, L. Hwang and J. Chong, "BindingDB in 2015: A Public Database for Medicinal Chemistry, Computational Chemistry and Systems Pharmacology", *Nucleic Acids Research*, Vol. 44, No. D1, pp. D1045–D1053, 2016.

96. He, T., 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.

97. Davies, M., M. Nowotka, G. Papadatos, N. Dedman, A. Gaulton, F. Atkinson, L. Bellis and J. P. Overington, "ChEMBL Web Services: Streamlining Access to Drug Discovery Data and Utilities", *Nucleic Acids Research*, Vol. 43, No. W1, pp. W612–W620, 2015.

98. Gaulton, A., A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, M. Davies, N. Dedman, A. Karlsson, M. P. Magariños, J. P. Overington, G. Papadatos, I. Smit and A. R. Leach, "The ChEMBL Database in 2017", *Nucleic Acids Research*, Vol. 45, No. D1, pp. D945–D954, 2017.

99. Davis, A. P., C. J. Grondin, R. J. Johnson, D. Sciaky, J. Wiegers, T. C. Wiegers and C. J. Mattingly, "Comparative Toxicogenomics Database (CTD): Update 2021", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D1138–D1143, 2021.

100. Wishart, D. S., C. Knox, a. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang and J. Woolsey, "Drugbank: A Comprehensive Resource for in Silico Drug Discovery and Exploration", *Nucleic Acids Research*, Vol. 34, No. Suppl_1, pp. D668–D672, 2006.

101. Wishart, D. S., C. Knox, a. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam and M. Hassanali, "DrugBank: A Knowledgebase for Drugs, Drug Actions and Drug Targets", *Nucleic Acids Research*, Vol. 36, No. Suppl_1, pp. D901–D906, 2008.

102. Wishart, D. S., Y. D. Feunang, a. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, N. Assempour, I. Iynkkaran, Y. Liu, A. Maciejewski, N. Gale, A. Wilson, L. Chin, R. Cummings, D. Le, A. Pon, C. Knox and M. Wilson, "Drugbank 5.0: A Major Update to the Drugbank Database for 2018", *Nucleic Acids Research*, Vol. 46, No. D1, pp. D1074–D1082, 2018.

103. Kim, S., J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang and E. E. Bolton, "Pubchem in 2021: New Data Content and Improved Web Interfaces", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D1388–D1395, 2021.

104. Kuhn, M., M. Campillos, I. Letunic, L. J. Jensen and P. Bork, "A Side Effect Resource to Capture Phenotypic Effects of Drugs", *Molecular Systems Biology*, Vol. 6, No. 1, p. 343, 2010.

105. Kuhn, M., I. Letunic, L. J. Jensen and P. Bork, "The Sider Database of Drugs and Side Effects", *Nucleic Acids Research*, Vol. 44, No. D1, pp. D1075–D1079,

2016.

106. Szklarczyk, D., A. L. Gable, K. C. Nastou, D. Lyon, R. Kirsch, S. Pyysalo, N. T. Doncheva, M. Legeay, T. Fang, P. Bork, L. J. Jensen and C. von Mering, "The String Database in 2021: Customizable Protein-Protein Networks, and Functional Characterization of User-Uploaded Gene/Measurement Sets", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D605–D612, 2021.

107. Özçelik, R., A. Bağ, B. Atıl, A. Özgür and E. Özkırımlı, "DebiasedDTA: Model Debiasing to Boost Drug-Target Affinity Prediction", *arXiv preprint arXiv:2107.05556*, 2021.

108. Fey, M. and J. E. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric", *arXiv preprint arXiv:1903.02428*, 2019.

109. Carneiro, T., R. V. M. Da NÓBrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque and P. P. Reboucas Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications", *IEEE Access*, Vol. 6, pp. 61677–61685, 2018.

110. Nair, V. and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines", *International Conference on Machine Learning*, 2010.

111. Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization", *arXiv preprint arXiv:1412.6980*, 2014.

112. Flam-Shepherd, D., K. Zhu and A. Aspuru-Guzik, "Keeping It Simple: Language Models Can Learn Complex Molecular Distributions", *arXiv preprint arXiv:2112.03041*, 2021.

113. Choromanski, K., V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell and A. Weller, "Rethinking Attention with Performers", *arXiv preprint*

arXiv:2009.14794, 2020.

APPENDIX A: HOMOGENEOUS GRAPH RESULTS FOR LIGANDS

Table A.1. CI and R^2 scores of DDI models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.888 (0.009)	0.781 (0.028)
Model (1)		**0.896 (0.009)**	0.777 (0.026)
Model (2)		0.890 (0.014)	**0.782 (0.017)**
DeepDTA	Cold Ligand	**0.687 (0.096)**	**0.039 (0.243)**
Model (1)		0.664 (0.057)	-0.053 (0.210)
Model (2)		0.640 (0.066)	-0.125 (0.180)
DeepDTA	Cold Protein	0.759 (0.006)	0.315 (0.049)
Model (1)		**0.779 (0.030)**	**0.375 (0.104)**
Model (2)		0.768 (0.009)	0.327 (0.072)
DeepDTA	Cold	**0.554 (0.047)**	**-0.154 (0.164)**
Model (1)		0.554 (0.044)	-0.287 (0.184)
Model (2)		0.495 (0.037)	-0.496 (0.211)

Table A.2. MSE and RMSE scores of DDI models on test sets of BDB.

Model	**Test Set**	**MSE**	**RMSE**
DeepDTA	Warm	0.288 (0.021)	0.536 (0.012)
Model (1)		0.295 (0.036)	0.542 (0.033)
Model (2)		**0.287 (0.014)**	**0.535 (0.014)**
DeepDTA	Cold Ligand	**1.448 (0.939)**	**1.152 (0.348)**
Model (1)		1.472 (0.650)	1.187 (0.249)
Model (2)		1.548 (0.602)	1.225 (0.220)
DeepDTA	Cold Protein	1.085 (0.146)	1.040 (0.146)
Model (1)		**0.992 (0.207)**	**0.991 (0.101)**
Model (2)		1.064 (0.154)	1.029 (0.077)
DeepDTA	Cold	**2.007 (1.223)**	**1.356 (0.410)**
Model (1)		2.013 (0.767)	1.395 (0.260)
Model (2)		2.348 (0.978)	1.504 (0.294)

Table A.3. CI and R^2 scores of DDS models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.888 (0.009)	0.781 (0.028)
Model (3)		**0.893 (0.005)**	0.787 (0.020)
Model (4)		0.890 (0.006)	**0.789 (0.008)**
DeepDTA	Cold Ligand	**0.687 (0.096)**	**0.039 (0.243)**
Model (3)		0.651 (0.085)	-0.139 (0.132)
Model (4)		0.666 (0.064)	-0.102 (0.330)
DeepDTA	Cold Protein	0.759 (0.006)	0.315 (0.049)
Model (3)		**0.775 (0.018)**	**0.340 (0.082)**
Model (4)		0.774 (0.015)	0.327 (0.075)
DeepDTA	Cold	**0.554 (0.047)**	**-0.154 (0.164)**
Model (3)		0.519 (0.036)	-0.390 (0.235)
Model (4)		0.536 (0.068)	-0.274 (0.269)

Table A.4. MSE and RMSE scores of DDS models on test sets of BDB.

Model	**Test Set**	**MSE**	**RMSE**
DeepDTA	Warm	0.288 (0.021)	0.536 (0.012)
Model (3)		0.280 (0.017)	0.529 (0.017)
Model (4)		**0.278 (0.011)**	**0.527 (0.010)**
DeepDTA	Cold Ligand	**1.448 (0.939)**	**1.152 (0.348)**
Model (3)		1.603 (0.645)	1.242 (0.246)
Model (4)		1.502 (0.615)	1.202 (0.239)
DeepDTA	Cold Protein	1.085 (0.146)	1.040 (0.146)
Model (3)		**1.045 (0.174)**	**1.019 (0.086)**
Model (4)		1.065 (0.162)	1.029 (0.079)
DeepDTA	Cold	**2.007 (1.223)**	**1.356 (0.410)**
Model (3)		2.278 (1.069)	1.467 (0.356)
Model (4)		2.021 (0.950)	1.388 (0.308)

APPENDIX B: HOMOGENEOUS GRAPH RESULTS FOR PROTEINS

Table B.1. CI and R^2 scores of PPI models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	**0.888 (0.009)**	**0.781 (0.028)**
Model (5)		0.888 (0.009)	0.773 (0.025)
Model (6)		0.888 (0.009)	0.777 (0.012)
DeepDTA	Cold Ligand	0.687 (0.096)	**0.039 (0.243)**
Model (5)		0.674 (0.095)	-0.031 (0.290)
Model (6)		**0.698 (0.083)**	-0.020 (0.347)
DeepDTA	Cold Protein	0.759 (0.006)	0.315 (0.049)
Model (5)		**0.765 (0.017)**	**0.323 (0.073)**
Model (6)		0.759 (0.019)	0.299 (0.084)
DeepDTA	Cold	0.554 (0.047)	**-0.154 (0.164)**
Model (5)		0.551 (0.049)	-0.269 (0.165)
Model (6)		**0.561 (0.031)**	-0.363 (0.279)

Table B.2. MSE and RMSE scores of PPI models on test sets of BDB.

Model	**Test Set**	**MSE**	**RMSE**
DeepDTA	Warm	**0.288 (0.021)**	**0.536 (0.012)**
Model (5)		0.299 (0.020)	0.546 (0.018)
Model (6)		0.295 (0.017)	0.543 (0.015)
DeepDTA	Cold Ligand	**1.448 (0.939)**	**1.152 (0.348)**
Model (5)		1.561 (1.059)	1.192 (0.373)
Model (6)		1.542 (1.146)	1.179 (0.390)
DeepDTA	Cold Protein	1.085 (0.146)	1.040 (0.146)
Model (5)		**1.069 (0.144)**	**1.031 (0.071)**
Model (6)		1.109 (0.176)	1.050 (0.083)
DeepDTA	Cold	**2.007 (1.223)**	**1.356 (0.410)**
Model (5)		2.186 (1.310)	1.419 (0.415)
Model (6)		2.288 (1.303)	1.457 (0.407)

Table B.3. CI and R^2 scores of PPS models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.888 (0.009)	0.781 (0.028)
Model (7)		**0.893 (0.006)**	0.775 (0.018)
Model (8)		0.892 (0.008)	**0.785 (0.019)**
DeepDTA	Cold Ligand	0.687 (0.096)	0.039 (0.243)
Model (7)		0.675 (0.083)	-0.067 (0.211)
Model (8)		**0.728 (0.046)**	**0.155 (0.162)**
DeepDTA	Cold Protein	0.759 (0.006)	0.315 (0.049)
Model (7)		**0.783 (0.010)**	**0.362 (0.047)**
Model (8)		0.773 (0.022)	0.339 (0.086)
DeepDTA	Cold	0.554 (0.047)	-0.154 (0.164)
Model (7)		0.557 (0.036)	-0.260 (0.090)
Model (8)		**0.598 (0.061)**	**-0.065 (0.297)**

Table B.4. MSE and RMSE scores of PPS models on test sets of BDB.

Model	**Test Set**	**MSE**	**RMSE**
DeepDTA	Warm	0.288 (0.021)	0.536 (0.012)
Model (7)		0.296 (0.018)	0.544 (0.016)
Model (8)		**0.283 (0.015)**	**0.532 (0.015)**
DeepDTA	Cold Ligand	1.448 (0.939)	1.152 (0.348)
Model (7)		1.538 (0.816)	1.204 (0.298)
Model (8)		**1.162 (0.451)**	**1.060 (0.193)**
DeepDTA	Cold Protein	1.085 (0.146)	1.040 (0.146)
Model (7)		**1.008 (0.115)**	**1.002 (0.056)**
Model (8)		1.046 (0.171)	1.019 (0.081)
DeepDTA	Cold	2.007 (1.223)	1.356 (0.410)
Model (7)		2.033 (0.908)	1.393 (0.305)
Model (8)		**1.590 (0.510)**	**1.246 (0.191)**

APPENDIX C: HETEROGENEOUS GRAPH RESULTS WITH DISEASES

Table C.1. CI and R^2 scores of DDiA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.888 (0.009)	0.781 (0.028)
Model (9)		0.895 (0.013)	**0.786 (0.023)**
Model (10)		**0.896 (0.006)**	0.784 (0.019)
DeepDTA	Cold Ligand	0.687 (0.096)	0.039 (0.243)
Model (9)		0.690 (0.050)	-0.025 (0.202)
Model (10)		**0.695 (0.056)**	**0.066 (0.122)**
DeepDTA	Cold Protein	0.759 (0.006)	0.315 (0.049)
Model (9)		**0.784 (0.010)**	**0.349 (0.072)**
Model (10)		0.779 (0.014)	0.333 (0.089)
DeepDTA	Cold	0.554 (0.047)	**-0.154 (0.164)**
Model (9)		**0.567 (0.047)**	-0.187 (0.177)
Model (10)		0.564 (0.031)	-0.201 (0.095)

Table C.2. MSE and RMSE scores of DDiA models on test sets of BDB.

Model	**Test Set**	**MSE**	**RMSE**
DeepDTA	Warm	0.288 (0.021)	0.536 (0.012)
Model (9)		**0.282 (0.023)**	**0.530 (0.022)**
Model (10)		0.284 (0.015)	0.533 (0.014)
DeepDTA	Cold Ligand	1.448 (0.939)	1.152 (0.348)
Model (9)		1.381 (0.443)	1.162 (0.177)
Model (10)		**1.294 (0.476)**	**1.120 (0.202)**
DeepDTA	Cold Protein	1.085 (0.146)	1.040 (0.146)
Model (9)		**1.032 (0.161)**	**1.013 (0.080)**
Model (10)		1.057 (0.191)	1.024 (0.093)
DeepDTA	Cold	2.007 (1.223)	1.356 (0.410)
Model (9)		**1.873 (0.792)**	**1.341 (0.271)**
Model (10)		1.939 (0.854)	1.360 (0.298)

Table C.3. CI and R^2 scores of PDiA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.888 (0.009)	0.781 (0.028)
Model (11)		0.881 (0.007)	0.759 (0.028)
Model (12)		**0.895 (0.007)**	**0.794 (0.015)**
DeepDTA	Cold Ligand	0.687 (0.096)	**0.039 (0.243)**
Model (11)		0.695 (0.036)	-0.010 (0.201)
Model (12)		**0.702 (0.046)**	0.030 (0.169)
DeepDTA	Cold Protein	0.759 (0.006)	0.315 (0.049)
Model (11)		0.762 (0.017)	0.238 (0.233)
Model (12)		**0.785 (0.011)**	**0.377 (0.078)**
DeepDTA	Cold	0.554 (0.047)	-0.154 (0.164)
Model (11)		0.543 (0.068)	-0.475 (0.506)
Model (12)		**0.568 (0.027)**	**-0.060 (0.146)**

Table C.4. MSE and RMSE scores of PDiA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.288 (0.021)	0.536 (0.012)
Model (11)		0.317 (0.034)	0.563 (0.030)
Model (12)		**0.271 (0.014)**	**0.520 (0.014)**
DeepDTA	Cold Ligand	1.448 (0.939)	1.152 (0.348)
Model (11)		**1.340 (0.353)**	1.149 (0.143)
Model (12)		1.349 (0.546)	**1.140 (0.222)**
DeepDTA	Cold Protein	1.085 (0.146)	1.040 (0.146)
Model (11)		1.211 (0.411)	1.087 (0.172)
Model (12)		**0.987 (0.166)**	**0.990 (0.082)**
DeepDTA	Cold	2.007 (1.223)	1.356 (0.410)
Model (11)		2.140 (0.576)	1.449 (0.201)
Model (12)		**1.689 (0.680)**	**1.272 (0.265)**

Table C.5. CI and R^2 scores of DDiPA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	**0.888 (0.009)**	**0.781 (0.028)**
Model (13)		0.878 (0.009)	0.753 (0.020)
Model (14)		0.882 (0.008)	0.755 (0.022)
DeepDTA	Cold Ligand	0.687 (0.096)	**0.039 (0.243)**
Model (13)		0.664 (0.087)	-0.040 (0.097)
Model (14)		**0.703 (0.033)**	0.030 (0.082)
DeepDTA	Cold Protein	**0.759 (0.006)**	**0.315 (0.049)**
Model (13)		0.749 (0.018)	0.268 (0.061)
Model (14)		0.752 (0.013)	0.282 (0.073)
DeepDTA	Cold	0.554 (0.047)	**-0.154 (0.164)**
Model (13)		0.533 (0.063)	-0.300 (0.161)
Model (14)		**0.578 (0.050)**	-0.275 (0.123)

Table C.6. MSE and RMSE scores of DDiPA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	**0.288 (0.021)**	**0.536 (0.012)**
Model (13)		0.325 (0.025)	0.570 (0.022)
Model (14)		0.323 (0.029)	0.568 (0.026)
DeepDTA	Cold Ligand	1.448 (0.939)	1.152 (0.348)
Model (13)		1.479 (0.637)	1.190 (0.250)
Model (14)		**1.319 (0.385)**	**1.137 (0.161)**
DeepDTA	Cold Protein	**1.085 (0.146)**	**1.040 (0.146)**
Model (13)		1.156 (0.141)	1.073 (0.065)
Model (14)		1.138 (0.173)	1.064 (0.081)
DeepDTA	Cold	2.007 (1.223)	**1.356 (0.410)**
Model (13)		2.096 (0.890)	1.414 (0.310)
Model (14)		**2.005 (0.743)**	1.392 (0.260)

APPENDIX D: HETEROGENEOUS GRAPH RESULTS WITH SIDE EFFECTS

Table D.1. CI and R^2 scores of DSA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.888 (0.009)	0.781 (0.028)
Model (15)		**0.898 (0.009)**	**0.791 (0.016)**
Model (16)		0.889 (0.011)	0.776 (0.021)
DeepDTA	Cold Ligand	**0.687 (0.096)**	**0.039 (0.243)**
Model (15)		0.683 (0.040)	-0.074 (0.265)
Model (16)		0.664 (0.068)	-0.196 (0.196)
DeepDTA	Cold Protein	0.759 (0.006)	0.315 (0.049)
Model (15)		**0.774 (0.015)**	**0.358 (0.059)**
Model (16)		0.766 (0.022)	0.323 (0.097)
DeepDTA	Cold	0.554 (0.047)	**-0.154 (0.164)**
Model (15)		0.561 (0.045)	-0.278 (0.293)
Model (16)		**0.580 (0.052)**	-0.445 (0.284)

Table D.2. MSE and RMSE scores of DSA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.288 (0.021)	0.536 (0.012)
Model (15)		**0.275 (0.022)**	**0.524 (0.021)**
Model (16)		0.295 (0.021)	0.543 (0.019)
DeepDTA	Cold Ligand	1.448 (0.939)	**1.152 (0.348)**
Model (15)		**1.443 (0.461)**	1.186 (0.190)
Model (16)		1.621 (0.555)	1.258 (0.199)
DeepDTA	Cold Protein	1.085 (0.146)	1.040 (0.146)
Model (15)		**1.015 (0.131)**	**1.005 (0.064)**
Model (16)		1.076 (0.209)	1.032 (0.102)
DeepDTA	Cold	2.007 (1.223)	**1.356 (0.410)**
Model (15)		**1.943 (0.647)**	1.375 (0.228)
Model (16)		2.241 (0.905)	1.470 (0.281)

Table D.3. CI and R^2 scores of DDI-DSA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.888 (0.009)	0.781 (0.028)
Model (17)		0.892 (0.013)	0.781 (0.021)
Model (18)		**0.899 (0.008)**	**0.785 (0.017)**
DeepDTA	Cold Ligand	0.687 (0.096)	0.039 (0.243)
Model (17)		0.688 (0.080)	-0.034 (0.245)
Model (18)		**0.711 (0.057)**	**0.217 (0.153)**
DeepDTA	Cold Protein	0.759 (0.006)	0.315 (0.049)
Model (17)		0.769 (0.019)	0.352 (0.084)
Model (18)		**0.778 (0.014)**	**0.365 (0.069)**
DeepDTA	Cold	0.554 (0.047)	-0.154 (0.164)
Model (17)		0.569 (0.041)	-0.303 (0.244)
Model (18)		**0.607 (0.061)**	**-0.018 (0.108)**

Table D.4. MSE and RMSE scores of DDI-DSA models on test sets of BDB.

Model	**Test Set**	**CI**	**R^2**
DeepDTA	Warm	0.288 (0.021)	0.536 (0.012)
Model (17)		0.289 (0.037)	0.537 (0.034)
Model (18)		**0.285 (0.026)**	**0.533 (0.025)**
DeepDTA	Cold Ligand	1.448 (0.939)	1.152 (0.348)
Model (17)		1.435 (0.640)	1.171 (0.250)
Model (18)		**1.082 (0.397)**	**1.022 (0.194)**
DeepDTA	Cold Protein	1.085 (0.146)	1.040 (0.146)
Model (17)		1.026 (0.177)	1.009 (0.086)
Model (18)		**1.006 (0.162)**	**1.000 (0.081)**
DeepDTA	Cold	2.007 (1.223)	1.356 (0.410)
Model (17)		2.071 (0.927)	1.406 (0.306)
Model (18)		**1.608 (0.617)**	**1.245 (0.241)**