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

Systems Biology

Pre-training Graph Neural Networks for Link Prediction in Biomedical Networks

Yahui Long [1], Min Wu [2], Yong Liu [3], Yuan Fang [4], Chee Keong Kwoh [5], Jinmiao Chen [1,], Jiawei Luo [6,*] and Xiaoli Li [2,*]

[1] Singapore Immunology Network (SIgN), Agency for Science, Technology and Research [2] Institute for Infocomm Research, Agency for Science, Technology and Research [3] Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, [4] School of Information Systems, Singapore Management University, 178902, [5] School of Computer Science and Engineering, Nanyang Technological University, [6] College of Computer Science and Electronic Engineering, Hunan University,

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Graphs or networks are widely utilized to model the interactions between different entities (e.g., proteins, drugs, etc) for biomedical applications. Predicting potential interactions/links in biomedical networks is important for understanding the pathological mechanisms of various complex human diseases, as well as screening compound targets for drug discovery. Graph neural networks (GNNs) have been utilized for link prediction in various biomedical networks, which rely on the node features extracted from different data sources, e.g., sequence, structure and network data. However, it is challenging to effectively integrate these data sources and automatically extract features for different link prediction tasks.
Results: In this paper, we propose a novel **P**re-**T**raining **G**raph **N**eural **N**etworks based framework named PT-GNN to integrate different data sources for link prediction in biomedical networks. First, we design expressive deep learning methods (e.g., convolutional neural network (CNN) and graph convolutional network (GCN)) to learn features for individual nodes from sequence and structure data. Second, we further propose a GCN-based encoder to effectively refine the node features by modelling the dependencies among nodes in the network. Third, the node features are pre-trained based on graph reconstruction tasks. The pre-trained features can be used for model initialization in downstream tasks. Extensive experiments have been conducted on two critical link prediction tasks, i.e., synthetic lethality (SL) prediction and drug-target interaction (DTI) prediction. Experimental results demonstrate PT-GNN outperforms the state-of-the-art methods for SL prediction and DTI prediction. In addition, the pre-trained features benefit improving the performance and reduce the training time of existing models.
Availability: Python codes and dataset are available at: https://github.com/longyahui/PT-GNN
Contact: luojiawei@hnu.edu.cn and xlli@i2r.a-star.edu.sg

1 Introduction

Advances in biomedical research boost the enormous accumulation of biological relational data (Su *et al.*, 2020). Graphs (or networks) have been extensively utilized to represent the relations (i.e., links or edges) between biomedical entities (i.e., nodes) (Yue *et al.*, 2020). The analysis of biomedical networks can provide great insights into the prevention, diagnosis, and treatment of various human complex diseases, as well as the screening of targeted compounds for drug discovery.

Identifying the potential relations/links between biomedical entities based on traditional wet-lab experiments often suffers from high cost and risk. In contrast, *in-silico methods* of predicting potential links in a biomedical network can be a rapid and cost-effective way to guide the experimental methods. Recently, biomedical network analysis has attracted much attention and a large number of computational methods have been developed to address various important link prediction tasks, such as drug-target interaction (**DTI**) prediction (Liu *et al.*, 2016), synthetic lethality (**SL**) prediction (Cai *et al.*, 2020) and lncRNA/miRNA/circRNA-disease association prediction (Chen *et al.*, 2017a, 2019; Wang *et al.*, 2021). We can classify these computational methods into three main categories, i.e., diffusion-based methods, matrix factorization (or completion) methods and graph neural network (GNN) methods.

In particular, diffusion-based methods leverage random walks (Codling *et al.*, 2008) to fully exploit the topological structure information of biomedical network to infer potential links. For example, Chen *et al.* (2017b) developed a Katz-measure-based computational method to predict potential microbe-disease associations by calculating the number of walks between nodes and walk lengths in bipartite network. Chen *et al.* (2018a) developed a bipartite network projection-based method named BNPMDA to infer latent microRNA-disease associations by fully considering the bias preference degree of a node for different neighbors. Zong *et al.* (2017) proposed a similarity-based method to predict drug-target associations. In addition, Luo and Long (2020) constructed a heterogeneous network, and further proposed a random walk-based model to predict microbe-disease associations, which uses network topological similarity to influence the walking preference of the walker.

Matrix factorization has shown promising performance in exploring intrinsic structure of various data and achieved success in various link prediction tasks, such as DTI prediction and SL prediction. The main idea behind matrix factorization is to learn node representations by exploring the latent patterns of interactive node pairs. For example, Zheng *et al.* (2013) developed a collaborative matrix factorization method to predict drug-target interactions. Liu *et al.* (2016) proposed a novel neighborhood regularized logistic matrix factorization method for drug-target prediction. Following that, Liu *et al.* (2019) further extended logistic matrix factorization to predict synthetic lethality interactions. Chen *et al.* (2018b) proposed an inductive matrix completion method for miRNA-disease association prediction. Zhang *et al.* (2020) developed a regularized generalized matrix factorization model called GRGMF for link prediction in various biomedical bipartite networks, e.g., DTI prediction and miRNA-disease association prediction. More recently, a logistic matrix factorization-based method was proposed to predict metabolite-disease interactions (Ma and Ma, 2021).

Graph neural networks, such as graph convolutional networks (GCNs) (Kipf and Welling, 2017) and graph attention networks (GATs) (Veličković *et al.*, 2018), have recently shown powerful capability in modeling graph-structured data. The main purpose of GNN-based methods is to learn node representations for downstream tasks, which preserve structural information of nodes. For example, Long *et al.* (2020b) proposed a novel GCN-based method named GCNMDA to predict microbe-drug associations by using a GCN to aggregate representations of neighbors. After that, Long *et al.* (2020a) proposed another GAT-based model for microbe-drug association prediction by leveraging GAT to capture hierarchical structure information. Liu *et al.* (2021a) proposed to use GCN to encode PPI network and protein-phenotype bipartite network to refine features for protein-phenotype association prediction. Nguyen *et al.* (2021) presented a novel framework for drug-target binding affinity prediction by using GNNs to encode drug structures. Fu *et al.* (2021) proposed a novel multi-view GCN model for three link prediction tasks.

In addition to the network data, other biological data sources (e.g., protein sequence data, drug structure data, gene ontology annotations, etc) are also valuable for link prediction tasks involving proteins or drugs. However, network-based methods mentioned above have different issues to integrate other data sources for link prediction. First, diffusion-based methods are usually not able to integrate the data sources other than network data. Second, matrix factorization methods need to first calculate the similarity matrices based on features manually extracted from other data sources (Zheng *et al.*, 2013; Liu *et al.*, 2016), and then define regularization terms based on the similarity matrices to improve the performance for link prediction. Third, GNN methods can take the node features, which are manually extracted from other data sources, as inputs for link prediction (Long *et al.*, 2020a,b). However, such manual feature extraction requires domain-specific knowledge. In addition, most of methods above are designed for single link prediction task, which limits their applications. And many of these methods calculate similarity matrix as input features depending on known associations and thus have to recalculate features when new nodes (i.e., proteins or drugs) are introduced.

To address the above issues, we propose a generic pre-training model, as shown in Figure 1, to integrate different data sources for link prediction in biomedical networks. Our model consists of the following key components. First, we leverage biological data to construct interaction networks for nodes (e.g., proteins and drugs). We then implement expressive CNN or GNN methods to capture node features, e.g., from protein sequence data and drug structure data. Second, with the networks and node features as inputs, we further design a GCN-based interaction graph encoder to effectively preserve the dependencies between nodes to refine node features, which are transferable to different downstream tasks. Third, the model is pre-trained based on the graph reconstruction tasks. Extensive experiments were conducted on two link prediction tasks, i.e., SL prediction and DTI prediction. Experimental results demonstrate that our PT-GNN model outperforms state-of-the-art methods for SL prediction and DTI prediction. In addition, the node features pre-trained by our PT-GNN model are proved to be effective and can help to improve the performance and reduce the training time for existing models.

Overall, our main contributions are summarized as follows:

- A generic pre-training graph neural network framework called PT-GNN was proposed for link prediction in biomedical networks. *To the best of our knowledge, this is the first study in the area of pre-training graph neural network model for biomedical link prediction*.
- To enhance link prediction performance, we *fully leveraged rich biological data, including protein sequences, drug molecular structures and their networks (e.g., protein-protein interaction (PPI) network and drug-drug interaction (DDI) network), to learn their features in our pre-training model*. Moreover, the pre-trained features can provide the existing models in the downstream tasks with high-quality initialization to improve their performance.
- To validate the effectiveness of our model, we conducted extensive experiments on two critical link prediction tasks, i.e., SL prediction and DTI prediction. The results demonstrated that our proposed PT-GNN model outperformed state-of-the-art methods and the pre-trained features benefited existing downstream task models.

2 Related work

In this section, we first present some backgrounds about graph neural networks, and then introduce pre-training and its applications in biological domains.

2.1 Graph neural networks

Graph neural networks have shown powerful capability in modeling graph-structured data. In particular, the GCN, proposed by Kipf and Welling (2017), aims to learn node representations by aggregating the features of neighbours. Due to its great performance, GCN has attracted increasing attention and achieved remarkable success in various research domains, such as text classification (Yao *et al.*, 2019), recommender system (Liu *et al.*, 2020) and computer vision (Dhingra *et al.*, 2021). The GAT (Veličković *et al.*, 2018) is an extension of GCN, which focuses on more important neighbors by assigning greater weight values to them. Such

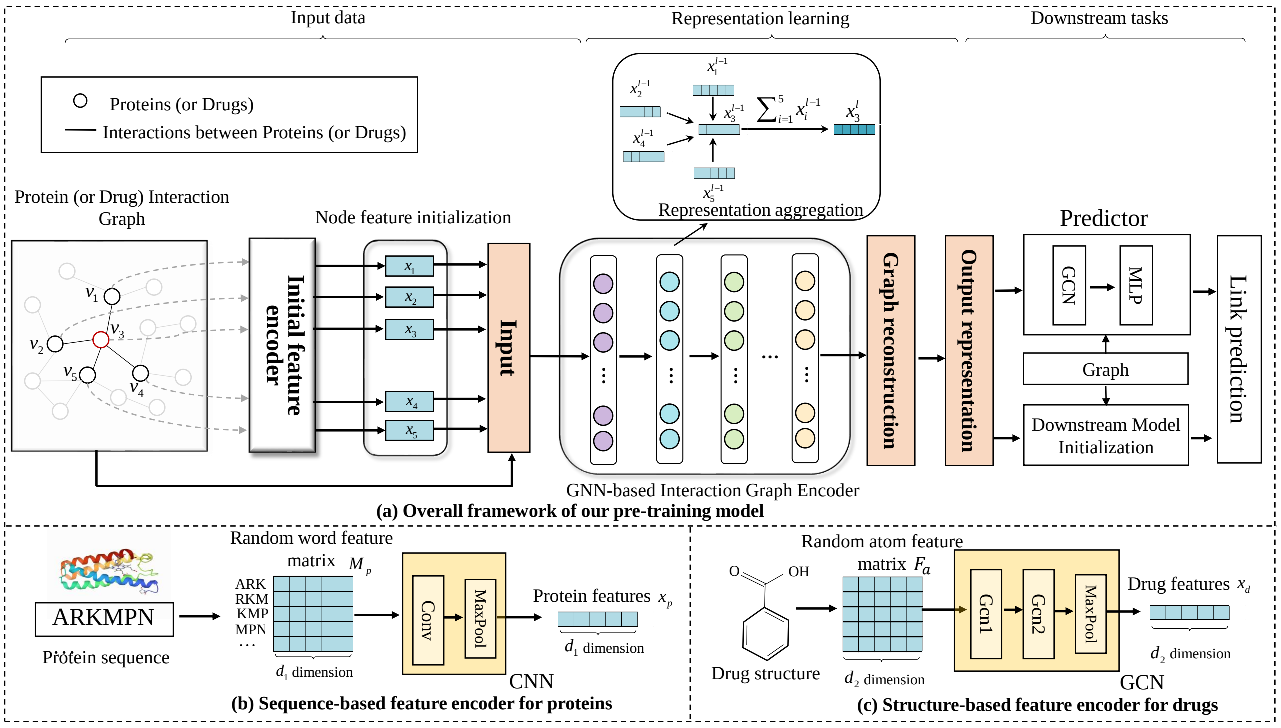


Fig. 1. The overall architecture of PT-GNN for link prediction in biomedical networks. (a) Overall framework of our PT-GNN model. It consists of three main steps. The first step is to encode node attribute information to initial features. The second step aims to learn node representations. The third step is to reconstruct interactive graph for link prediction. (b) Protein sequence encoder to learn initial features for proteins. (c) Drug molecule structure encoder to learn initial features for drugs.

operation enables the model to learn more informative representations. The GAT has attained wide applications on various tasks, such as computer vision systems (Mi and Chen, 2020), recommender systems (Liu *et al.*, 2021c) and bioinformatics (Long *et al.*, 2020a).

2.2 Pre-training

Pre-training is a type of transfer learning that aims to transform knowledge from a full domain to domain-specific tasks. Pre-training can provide a model with high-quality initialization and thus enhance its performance. In addition, it accelerates model convergence during training.

More recently, pre-training has achieved significant success in multiple domains, such as natural language processing (Devlin *et al.*, 2019; Qian *et al.*, 2021), computer vision (Li *et al.*, 2020) and bioinformatics (Hu *et al.*, 2020b; Lu *et al.*, 2021). Meanwhile, several pre-training models have been proposed to address biological tasks. For example, Navarin *et al.* (2018) developed a task-independent pre-training method that combines GNN with graph kernels to predict chemical compounds carcinogenicity. Hong *et al.* (2020) proposed a pre-training model named EPIVAN for enhancer-promoter interaction prediction. Hu *et al.* (2020a) developed a novel pre-training graph neural network model for protein function prediction, which pre-trains GNNs in a self-supervised way to learn protein features for protein classification tasks. Strodthoff *et al.* (2020) released a universal deep sequence model, which pre-trains the model on unlabeled protein sequences and fine-tunes it on protein classification tasks. Zhu *et al.* (2021) presented a graph neural networks based framework for drug response prediction, which uses pre-training strategy to learn drug features from large-scale molecular datasets. In this paper, we focus on pretraining graph neural networks for link prediction in biomedical networks.

3 Methods

This work focuses on pre-training the protein and drug representations that can fully exploit the protein and drug attribute information, as well as the PPI data and the DDI data, to benefit downstream tasks such as SL prediction and DTI prediction. Figure 1(a) shows the framework of the proposed pre-training model, which contains three main components: 1) node feature initialization, 2) GCN-based interaction graph encoder, and 3) interaction graph reconstruction. Before we detail each component in this section, we provide a preliminary background.

3.1 Preliminaries

We construct two graphs for proteins, namely the PPI graph $\mathcal{G}_\text{p}$ and the GO (Gene Ontology) graph $\mathcal{G}_\text{g}$. Firstly, we leverage PPI data to build a PPI graph $\mathcal{G}_\text{p} = \{\mathcal{V}_\text{p}, \mathcal{E}_\text{p}\}$, where $\mathcal{V}_\text{p} \in \mathbb{R}^{N_p}$ denotes the set of N_p protein nodes and $\mathcal{E}_\text{p}$ denotes the set of edges describing the interactions between proteins. Meanwhile, following Peng *et al.* (2016), we calculate semantic similarity matrix for proteins based on their GO terms. To extract more important interactive pairs, we implement random walk with restart (RWR) algorithm on the similarity matrix and then construct a protein GO similarity graph $\mathcal{G}_\text{g} = \{\mathcal{V}_\text{g}, \mathcal{E}_\text{g}\}$, by selecting the top-$t$ neighbors as interaction pairs for a given protein. More details for GO graph construction can be found in the supplementary file. Here we use protein sequence data as protein attribute information. Finally, we develop a convolutional neural network based encoder to learn initial protein features from the sequence data, as is shown in Figure 1(b).

We utilize DDI data to construct a DDI graph $\mathcal{G}_\text{d} = \{\mathcal{V}_\text{d}, \mathcal{E}_\text{d}\}$ for drugs, where $\mathcal{V}_\text{d} \in \mathbb{R}^{N_d}$ denotes the set of N_d drug nodes and $\mathcal{E}_\text{d}$ denotes the set of edges describing the interactions between drugs. For each drug, we consider its molecule structure as attribute information. Subsequently, we develop a graph convolutional network based encoder to learn drug initial features by fully considering DDI graph data and drug molecule structure information, as is shown in Figure 1(c).

3.2 Node Feature Initialization

In this section, we present the details of the protein sequence encoder and drug structure encoder, which are used to extract the initial features for proteins and drugs, respectively.

3.2.1 Protein Sequence Encoder

Protein sequences contain rich knowledge, which has been demonstrated by previous methods (Lee *et al.*, 2019; Nguyen *et al.*, 2021). Convolutional neural networks are able to learn high-order protein features from their sequences for various applications, such as DNA-protein binding prediction (Zeng *et al.*, 2016) and drug-target interaction prediction (Nguyen *et al.*, 2021). Following Nguyen *et al.* (2021), we adopt CNN to encode protein features from sequence data. In particular, for a protein sequence s_p, we firstly split it into a set of overlapping n-gram amino acid segments with r as the size of sliding window. For example, the sequence 'ARKMPN' can be split into 'ARK', 'RKM' ,'KMP', and 'MPN', when n and r are set to 3 and 1 respectively. Assume that a n-gram amino acid segment is considered as a word, and each word is represented by a d_1-dimension feature vector (empirically, we set d_1 as 100). The feature vectors of all words is denoted by $\mathbf{F}_{\mathrm{w}} \in \mathbb{R}^{N_w \times d_1}$, where N_w denotes the number of all possible words (i.e., corpus) in the dataset, and each row of $\mathbf{F}_{\mathrm{w}}$ is the feature vector of a possible word. Note that instead of invariant features, the word feature matrix $\mathbf{F}_{\mathrm{w}}$ is set as a trainable parameter matrix, which is randomly initialized and can be updated in the pre-training phase for more accurately capturing the intrinsic features of sequences.

After transforming protein sequences into words, we further design a CNN-based encoder to learn protein initial features. Specifically, as shown in Figure 1(b), we first convert a protein sequence s_p into a feature matrix $\mathbf{M}_p$, where each row denotes a d_1-dimension word feature vector. Then the feature matrix is fed into a two-layer convolutional neural network, including a 1D convolutional layer and a max-pooling layer. The convolutional layer is designed to learn local features, and the max-pooling layer aims to reduce dimension. Note that the average length of sequences of all proteins used in our experiments are 558. As the convolutional layer requires the same length of inputs, we set the maximal length of sequence to 800. The sequences with length less than 800 are padded with null label (i.e., Z). In the encoder, we use 16 filters with a kernel size of 10 in the convolutional layer. This indicates that the model will learn 16 different features for each sequence.

Following that, with the outputs of the convolutional layer as inputs, we further perform a max-pooling layer to reduce the feature dimension. Here both the pooling window size and stride are set to 60. As the outputs of the max-pooling layer, we can derive a feature vector preserving sequence semantic information for the input protein. Subsequently, we can obtain a feature matrix $\mathbf{X}_{\mathrm{p}} \in \mathbb{R}^{N_p \times d_1}$ for all proteins by applying this sequence encoder on all protein sequence data.

3.2.2 Drug Molecular Structure Encoder

The molecular structures are important components to achieve chemical functions of drugs. Essentially, the molecular structure of a drug d can be described by a graph $G_{\mathrm{a}} = (V_{\mathrm{a}}, E_{\mathrm{a}})$, where $V_{\mathrm{a}} \in \mathbb{R}^{N_a}$ represents the set of N_a nodes (i.e., atoms) and E_{a} represents the set of edges (i.e., bonds). The adjacency matrix of this graph is denoted by $\mathbf{A}_{\mathrm{a}} \in \mathbb{R}^{N_a \times N_a}$. Moreover, we denote the feature matrix of all atoms by $\mathbf{F}_{\mathrm{a}} \in \mathbb{R}^{N_a \times d_2}$, where each row of $\mathbf{F}_{\mathrm{a}}$ denotes the feature vector of an atom and d_2 represents the atom feature dimension.

As mentioned above, graph convolutional networks, as a typical graph neural networks, have powerful capability in modeling graph-structured data and achieve wide applications in various fields, such as text classification (Yao *et al.*, 2019), recommender system (Liu *et al.*, 2020). Thus, here we implement a graph convolutional network on the molecular graph G_{a} to learn the initial feature for drug d. As a single-layer GCN can only capture limited features from one-hop (or immediate) neighbors, we design a multi-layer GCN on the molecule graph G_{a} to aggregate the features of multi-hop neighbors. More specifically, the k-th GCN layer can be formulated as follows,

$$\mathbf{R}_{\mathrm{a}}{}^{(k)} = \mathrm{ReLU}\Big(\widetilde{\mathbf{A}}_{\mathrm{a}}\mathbf{R}_{\mathrm{a}}{}^{(k-1)}\mathbf{W}_1^{(k-1)} + \mathbf{b}_1^{(k-1)}\Big), \tag{1}$$

where $\widetilde{\mathbf{A}}_{\mathrm{a}} = \mathbf{D}_{\mathrm{a}}{}^{-\frac{1}{2}}\mathbf{A}_{\mathrm{a}}\mathbf{D}_{\mathrm{a}}{}^{-\frac{1}{2}}$ is a normalized adjacency matrix. $\mathbf{D}_{\mathrm{a}}$ is a diagonal matrix with the diagonal element being $\mathbf{D}_{\mathrm{a}}(i,i) = \sum_{j=1}^{N_a}\mathbf{A}_{\mathrm{a}}(i,j)$. $\mathbf{W}_1^{(k-1)}$ and $\mathbf{b}_1^{(k-1)}$ are the trainable weight matrix and bias vector respectively. $\mathrm{ReLU}(\cdot)$ is the Rectified Linear Unit activation function. $\mathbf{R}_{\mathrm{a}}{}^{(k)}$ denotes the feature matrix of atoms at the k-th layer. Note that $\mathbf{R}_{\mathrm{a}}{}^{(0)}$ is the original feature matrix $\mathbf{F}_{\mathrm{a}}$ of atoms. After K GCN layers, we can obtain the atom representations $\mathbf{R}_{\mathrm{a}}{}^{(K)}$.

To learn the drug feature, we further implement a max-pooling layer on $\mathbf{R}_{\mathrm{a}}{}^{(K)}$ to form the initial feature vector $\mathbf{x}_d \in \mathbb{R}^{1 \times d_2}$ for the drug d. Here, we set the size of pooling window as the number of atoms N_a, and set the step size to 1. By applying the drug structure encoder on the molecular structures of all drugs, we can derive a feature matrix $\mathbf{X}_{\mathrm{d}} \in \mathbb{R}^{N_d \times d_2}$ for all drugs. In the experiments, we empirically set K to 2.

In the literature, there are several existing studies (Öztürk *et al.*, 2018; Lee *et al.*, 2019) that use drug molecular structure information to learn representations for drugs. However, most of them use fixed invariant values (e.g., one-hot encoding) to initialize atom features. Thus, they cannot adaptively learn the structure features of drugs. Instead of setting invariant values, we treat the atom feature matrix $\mathbf{F}_{\mathrm{a}}$ as trainable parameters, which are randomly initialized and would be learned through graph structure reconstruction. Such operation enables the proposed model to flexibly learn the properties of molecular structures.

3.3 GCN-based Interaction Graph Encoder

In Section 3.2, we make full advantage of the protein and drug attribute information to extract initial features $\mathbf{X}_{\mathrm{p}}$ and $\mathbf{X}_{\mathrm{d}}$ for proteins and drugs, respectively. As shown in Figure 1(a), a GNN-based interaction graph encoder is then designed to exploit the structures of the protein/drug interaction graph for learning the protein/drug representations. Note that this interaction graph encoder is a unified structure that can be used to learn both the protein and the drug representations, while it takes different input interaction graph and initial features for drugs and proteins. In the following sections, we only describe the operations for learning protein representations with input graph $\mathcal{G}_{\mathrm{p}}$ and initial node features $\mathbf{X}_{\mathrm{p}}$.

Let us denote the PPI graph as $\mathcal{G}_{\mathrm{p}} = \{\mathcal{V}_{\mathrm{p}}, \mathcal{E}_{\mathrm{p}}\}$. For a node υ_i in $\mathcal{G}_{\mathrm{p}}$, the main purpose of the graph encoder is to learn its representation by iteratively aggregating the representations of its neighbors. Formally, the ℓ-th layer of a GNN-based graph encoder is as follows,

$$\mathbf{h}_i^{(\ell)} = \mathrm{AGGREGATE}\Big(\Big\{\mathbf{h}_j^{(\ell-1)} : \upsilon_j \in \mathcal{N}_i\Big\}\Big), \tag{2}$$

where $\mathbf{h}_j^{(\ell-1)}$ denotes the feature representations of the node υ_j at the $(\ell-1)$-th layer, and $\mathcal{N}_i$ denotes the first-hop neighbors of υ_i in the graph. Note that $\mathcal{N}_i$ also includes υ_i in this work. AGGREGATE$(\cdot)$ denotes aggregator function, which can be defined by various different graph neural architectures, such as GCN and GAT.

In this work, we leverage GCN as the aggregator function to integrate the representations of nodes in the interaction graph. The ℓ-th layer of the graph convolutional network can be formulated as follows,

$$\mathbf{H}_{\mathrm{p}}{}^{(\ell)} = \mathrm{ReLU}\Big(\widetilde{\mathbf{A}}_{\mathrm{p}}\mathbf{H}_{\mathrm{p}}{}^{(\ell-1)}\mathbf{W}_2^{(\ell-1)} + \mathbf{b}_2^{(\ell-1)}\Big), \tag{3}$$

where $\widetilde{\mathbf{A}}_{\mathrm{p}}$ is the normalized diagonal adjacency matrix with self-connection, $\mathbf{H}_{\mathrm{p}}{}^{(\ell-1)}$ denotes the outputs of the model at the $(\ell-1)$-th layer. Note that $\mathbf{H}_{\mathrm{p}}{}^{(0)}$ is defined as the input feature matrix $\mathbf{X}_{\mathrm{p}}$. Moreover, $\mathbf{W}_2^{\ell-1}$ and $\mathbf{b}_2^{\ell-1}$ are trainable weight matrix and bias vector respectively. After L GCN layers, we adopt the output of the last layer as the final representations of proteins $\mathbf{H}_{\mathrm{p}} \in \mathbb{R}^{N_P \times d_3}$, where d_3 denotes the dimension of the protein representations.

Note that $\mathbf{H}_{\mathrm{p}}$ is the protein representations obtained from the PPI graph $\mathcal{G}_{\mathrm{p}}$ with node features $\mathbf{X}_{\mathrm{p}}$. Similarly, we can obtain the protein representations $\mathbf{H}_{\mathrm{g}} \in \mathbb{R}^{N_P \times d_3}$ from the protein GO graph $\mathcal{G}_{\mathrm{g}}$ with node features $\mathbf{X}_{\mathrm{p}}$, and the drug representations $\mathbf{H}_{\mathrm{d}} \in \mathbb{R}^{N_d \times d_3}$ from the DDI graph $\mathcal{G}_{\mathrm{d}}$ with node features $\mathbf{X}_{\mathrm{d}}$.

3.4 Model Optimization

The proposed model is pre-trained with the graph structure reconstruction task. More specifically, for a given input interaction graph $\mathcal{G}$ with the adjacency matrix $\mathbf{A}$ and the output of the GCN-based graph encoder $\mathbf{H}$, we reconstruct the adjacency matrix in Eq. (4) and derive the reconstruction loss in Eq. (5),

$$\mathbf{P} = \mathrm{ReLU}\Big(\mathbf{H}\mathbf{H}^{\top}\Big), \tag{4}$$

$$\mathcal{L} = \sum_{(i,j)\in\Omega^{+}\cup\Omega^{-}} \Phi\Big(\mathbf{P}(i,j), \mathbf{A}(i,j)\Big) + \delta\big\|\mathbf{\Theta}\big\|_F^2, \tag{5}$$

where ReLU is activation function, and $\mathbf{P}$ is the reconstructed score matrix where each element describes the interaction score for a node pair (e.g., protein-protein pair). $\mathbf{\Theta}$ is the parameter matrix of the pre-training model. δ is weight factor that is used to control the influence of $\mathbf{\Theta}$ on our model. In addition, $\Phi(\cdot)$ is the MSE (i.e., mean square error) loss. We employ the Adam optimizer (Kingma and Ba, 2015) for the optimization. Note that when pre-training the protein and drug representations, the parameters of the protein sequence encoder and the drug molecule structure encoder are also updated simultaneously. In this work, for better training, we adopt a negative sampling strategy to train our model. Ω^{+} and Ω^{-} represent the sets of positive and negative samples for model training, respectively.

3.5 Link prediction

After training our model, we can derive the pre-trained node representations. Specifically, the pre-trained representations can be used as input features of link prediction models. In this paper, we feed the pre-trained representations into a two-layer predictor (as shown in Figure 1(a)), consisting of a GCN layer and a MLP (Multi-Layer Perception) layer, for link prediction. Also, we can use the pre-trained representations to initialize other models in downstream link prediction tasks. Note that the parameters in the protein sequence encoder and drug structure encoder are fixed when the pre-trained representations are used as the inputs of either the predictor or the model in downstream tasks.

4 Experimental Results

In this section, we first present the experimental settings, and then conduct extensive experiments to demonstrate the performance of our model for two downstream tasks, i.e., SL prediction and DTI prediction.

4.1 Experimental setups

4.1.1 Datasets

SL prediction. To pre-train protein features, we first downloaded the whole genome sequences of 20,375 human proteins from Uniprot (Consortium, 2019). Moreover, we constructed two gene-gene interaction graphs from PPI and Gene Ontology (GO) data, respectively. In particular, we collected 383,122 interactions associated with these 20,375 proteins from the latest version of BioGrid (Oughtred *et al.*, 2019), which was used to construct PPI graph. In addition, we first downloaded the ontology and annotation files from http://geneontology.org/. Then a semantic similarity matrix was calculated based on the sub-ontology "biological process (BP)". Given a node, we further prioritized all the neighbors according to their similarity scores and selected the top-t neighbors to construct the GO similarity graph. We empirically set t as 50. As a result, the GO similarity graph (or GO graph for short) contains 917,393 interactions between 20,375 proteins. For the first downstream task, i.e., SL prediction, we utilized SL pairs derived from SynLethDB (Guo *et al.*, 2016) to construct a SL graph, which includes 19,667 SL interactions between 6,375 genes. Note that during pre-training we use PPI and GO graphs as two different views to learn protein representations. Here we use factor λ to weight the influences of PPI and GO graphs on our model, respectively. More specifically, during pre-training, we assume that two graph-specific representations (i.e., $\mathbf{H}_{\mathrm{p}}$ and $\mathbf{H}_{\mathrm{g}}$) are derived for proteins when the PPI and GO graphs are fed into the model respectively. Therefore, the final protein representations are formulated as $\mathbf{H} = \lambda\mathbf{H}_{\mathrm{p}} + (1-\lambda)\mathbf{H}_{\mathrm{g}}$.

DTI prediction. We collected 1,113,252 drug-drug interactions (DDI) involving 3,543 drugs from Drugbank (Wishart *et al.*, 2018) to learn drug representations. Meanwhile, we downloaded the SMILES (Simplified Molecular Input Line Entry System) for these 3,543 drugs from Drugbank to construct their drug molecule graphs. We first extracted the initial features for drugs from their SMILES data via the molecular graph encoder, and then pre-trained drug features from DDI graph via the GCN-based interaction graph encoder. For the second downstream task, i.e., DTI prediction, we derived drug-target interaction data from Drugbank. In particular, we selected 9,679 drug-target interactions between 1,971 drugs and 1,899 targets on condition that drugs/targets have SMILES/sequences. Overall, the statistics of the datasets above are shown in Tables 1 and 2.

Table 1. The dataset statistics for synthetic lethality prediction.

	Graphs	# Proteins	# Interactions
Pre-train	PPI	20,375	383,122
	GO	20,375	917,393
Prediction	PPI	6,375	19,667

Table 2. The dataset statistics for drug-target prediction.

	Graphs	Drugs	# Targets	# Interactions
Pre-train	DDI	3,543	-	1,113,252
Prediction	DTI	1,971	1,899	9,679

4.1.2 Experimental settings

In this work, we conducted 5-fold cross validation (CV) to evaluate the performance of our model. Specifically, taking SL as example, we first randomly divide all known SL pairs into five groups. Then one group of SL pairs are in turn selected for model testing while the rest of SL pairs are used for model training. Following previous methods (Öztürk *et al.*, 2018; Long *et al.*, 2020b), we adopt a negative sampling strategy to better train the model. Negative SL pairs are randomly sampled from unknown SL pairs and the same numbers of negatives and positives are used for model training (including pre-training and downstream task) and testing. We adopt two well-known metrics for performance evaluation, i.e., area under ROC curve (AUC) and area under precision recall curve (AUPR). To offset the bias of random division, we repeat each experiment for 10 times and take their average as final AUC and AUPR values.

For the pre-training of both proteins and drugs, the training epoch is set to 200 and the learning rate is set to 0.005. To learn initial features for proteins, the length of amino acid segment n and sliding window size r are set as 3 and 1 respectively. Since there are totally 20 types of amino acids, the number of corpus N_w is 8001 (including one null label 'Z'). In the drug structure encoder, we set the number of GCN layers as 2. The dimensions of hidden layers are set to 256 and 128 (i.e., d_2) respectively. While the above parameters are empirically set, we also make parameter analysis for several other important parameters, including the dimension of representation d_3, the number of layers in the GCN-based graph encoder L and weight factor λ, in the following sections.

4.1.3 Baseline methods

In this work, we validate the performance of our model via two tasks, i.e., SL prediction and DTI prediction. We introduce eight state-of-the-art baseline methods for SL prediction as follows:

- SL^{2}MF (Liu *et al.*, 2019) proposes a logic matrix factorization-based method to identify SL pairs.
- GRSMF (Huang *et al.*, 2019) is a graph regularized self-representative matrix factorization algorithm for SL prediction.
- GCATSL (Long *et al.*, 2021) is a novel graph attention network-based model developed for SL prediction.
- SLMGAE (Hao *et al.*, 2021) is a multi-view graph auto-encoder based method to predict SL pairs.
- DDGCN (Cai *et al.*, 2020) presents a dual-dropout graph convolutional network model for SL prediction.

Meanwhile, we introduce five state-of-the-art deep learning methods for DTI prediction as follows:

- NeoDTI (Wan *et al.*, 2019) develops an end-to-end deep learning model to predict drug-target interactions by integrating heterogeneous biological data.
- DeepDTA (Öztürk *et al.*, 2018) is a deep learning model that uses drug structures and proteins sequences to predict drug-target binding affinity.
- MolTrans (Huang *et al.*, 2021) is a Transformer-based framework for DTI prediction.
- DeepConv-DTI (Lee *et al.*, 2019) uses convolutional neural network to learn representations to predict drug-target interactions.
- GraphDTA (Nguyen *et al.*, 2021) proposes a graph neural network based method for drug-target binding affinity prediction.

For all the above methods, we adopt the default parameters from their original implementations. The introduction of original features for each method can be found in supplementary file. Note that GCN and GAT are used as baselines for both SL and DTI prediction tasks.

4.2 Performance evaluation

In this section, we evaluate the performance of our pre-training model on two downstream tasks, i.e., SL prediction and DTI prediction.

4.2.1 SL prediction

In this section, we evaluate our model on SL prediction task. Table 3 shows the comparison results of various methods on SL prediction task. We can observe PT-GNN outperforms baseline methods consistently. Particularly, PT-GNN achieves an average AUC of 0.9525±0.0022 and an average AUPR of 0.9551±0.0007, which are 1.6% and 0.72% higher than that of the second-best method GCATSL. The results in Table 3 indicates that our proposed PG-GNN model is effective in predicting potential SL pairs.

To demonstrate the effectiveness of the pre-trained features, we compare the performance of different methods on SL prediction task by using original features (i.e., features used in baseline methods) and pre-trained features as their inputs respectively. It can be found from Table 4 that the pre-trained features are able to improve baseline methods. For example, GCATSL with pre-trained representations obtains an average AUC of 0.9576 and average AUPR of 0.9620, which are 2.14% and 1.44% higher than its original model. These results demonstrate that the pre-trained features learned from sequence data, PPI and GO networks are effective and informative. In particular, DDGCN does not integrate any data sources other than the SL graph and thus the pre-trained features can significantly improve its performance. Meanwhile, other methods already exploit additional data sources, e.g., GCATSL and SLMGAE utilize PPI and GO graphs, and GCN and GAT utilize PPI network as original inputs. Therefore, their performances with original features are already very good. Nevertheless, the pre-trained features, which effectively integrate protein sequence data and PPI/GO network data, can still improve their performance.

Table 3. Performance comparison of PT-GNN with baseline methods on SL prediction in 5-fold CV. The best results are marked in bold and the second best is underlined.

Methods	AUC	AUPR
SL^{2}MF	0.8454±0.0109	0.8986±0.0059
GRSMF	0.8853±0.0021	0.9187±0.0006
GCATSL	0.9375±0.0024	0.9483±0.0018
SLMGAE	0.9140±0.0049	0.9405±0.0030
DDGCN	0.8796±0.0080	0.9161±0.0046
GCN	0.9083±0.0034	0.9203±0.0027
GAT	0.8964±0.0136	0.8981±0.0157
PT-GNN	**0.9525±0.0022**	**0.9551±0.0007**

Table 4. Performance comparison of baseline methods with different feature initialization on SL prediction in 5-fold CV.

Method	Feature	Epoch	AUC	AUPR
SL^{2}MF	Original	200	0.8454±0.0109	0.8986±0.0059
	Pre-trained	100	**0.8553±0.0065**	**0.9017±0.0048**
GRSMF	Original	200	0.8853±0.0021	0.9187±0.0006
	Pre-trained	100	**0.9252±0.0054**	**0.9417±0.0041**
GCATSL	Original	600	0.9375±0.0024	0.9483±0.0018
	Pre-trained	100	**0.9576±0.0016**	**0.9620±0.0018**
SLMGAE	Original	300	0.9140±0.0049	0.9405±0.0030
	Pre-trained	200	**0.9279±0.0040**	**0.9465±0.0032**
DDGCN	Original	2000	0.8796±0.0080	0.9161±0.0046
	Pre-trained	10	**0.9204±0.0103**	**0.9305±0.0075**
GCN	Original	200	0.9083±0.0034	0.9203±0.0027
	Pre-trained	100	**0.9286±0.0056**	**0.9345±0.0052**
GAT	Original	200	0.8964±0.0136	0.8981±0.0157
	Pre-trained	100	**0.9087±0.0091**	**0.9097±0.0130**

In addition, we analyse the influences of our pre-trained features on the training time of various baseline models. As shown in Table 4, all the methods with pre-trained features take less epochs than using original features. Therefore, we can conclude that our pre-training model is helpful to reduce the training time of various baseline models.

4.2.2 DTI prediction

For further validating the performance of PT-GNN, we compare our model with state-of-the-art methods on DTI prediction. In particular, we first pre-train our model on DDI graph to derive drug representations while on PPI & GO graphs to derive protein representations. Similar to SL prediction task, we treat the learned representations as inputs of the predictor mentioned above to predict DTI. Here both GCN and GAT use drug structure similarity and target sequence similarity as input features. Table 5 indicates that PT-GNN performs better than all baseline methods in terms of AUC and AUPR. Specifically, PT-GNN attains average AUC of 0.9131±0.0052 and average AUPR of 0.9245±0.0057. The results demonstrate our pre-training model has powerful ability in identifying novel DTI. In our pre-training model, we consider simultaneously the interaction graphs and attributes for proteins and drugs to learn their representations. In particular, we integrate multiple sources of biological data (e.g., protein sequences, PPI and GO annotations) to refine the features for proteins. Besides, we take into account full domain knowledge (e.g., all sequences of human proteins) to learn protein features. This is the main reason why our PT-GNN model outperforms baseline methods. Note that since GraphDTA uses different methods (e.g., GCN and GAT) to encode drug structures, Table 5 includes four variants of GraphDTA.

Table 5. Performance comparison of PT-GNN with baseline methods on DTI prediction in 5-fold CV. The best results are marked in bold and the second best is underlined.

Methods	AUC	AUPR
NeoDTI	0.8343±0.0103	0.8497±0.0091
DeepDTA	0.8466±0.0068	0.8368±0.0096
MolTrans	0.8791±0.0134	0.8618±0.0172
DeepConv-DTI	0.9070±0.0065	0.8982±0.0034
GraphDTA-GCN	0.8632±0.0108	0.8261±0.0094
GraphDTA-GIN	0.8815±0.0156	0.8652±0.0103
GraphDTA-GAT	0.8476±0.0171	0.8138±0.0087
GraphDTA-GAT_GCN	0.8746±0.0135	0.8293±0.0157
GCN	0.8908±0.0073	0.8961±0.0068
GAT	0.8564±0.0092	0.8720±0.0093
PT-GNN	**0.9131±0.0052**	**0.9245±0.0057**

Here the baseline methods including DeepDTA, MolTrans, DeepConv-DTI and GraphDTA, take the drug structures and target sequences as inputs to learn the features for drugs and targets respectively. Therefore, the pre-trained features cannot be used to initialize these baselines, and we are not able to run these baselines under different initialization settings (e.g., original features vs. pre-trained features as shown in Table 4). Nevertheless, Table 5 shows that the pre-trained features for drugs and proteins are informative and effective for DTI prediction.

4.3 Ablation study

Recall that we use two types of data sources (i.e., PPI and GO) to construct graphs for proteins to pre-train our model. Here we conduct ablation studies to measure their influences on our pre-trained model for SL prediction.

Fig. 2 show that the methods, which use pre-trained features learned from PPI and GO, consistently outperform the original methods in terms of AUC and AUPR, indicating that both PPI and GO can contribute to enrich the protein features. Moreover, all the methods also achieve higher AUC and AUPR values when using the pre-trained features learned from either PPI or GO network than their original methods. Finally, we can conclude that both PPI and GO networks are important for pre-training protein features.

4.4 Parameter analysis

There are several important parameters that influence the performance of our model, such as the dimension of representation d_3 in the GCN-based interaction graph encoder, the number of GCN layers in the interaction graph encoder L and weight factor λ. Here, we fine-tune the pre-trained model with different parameter values to analyze their impacts for the task of SL prediction.

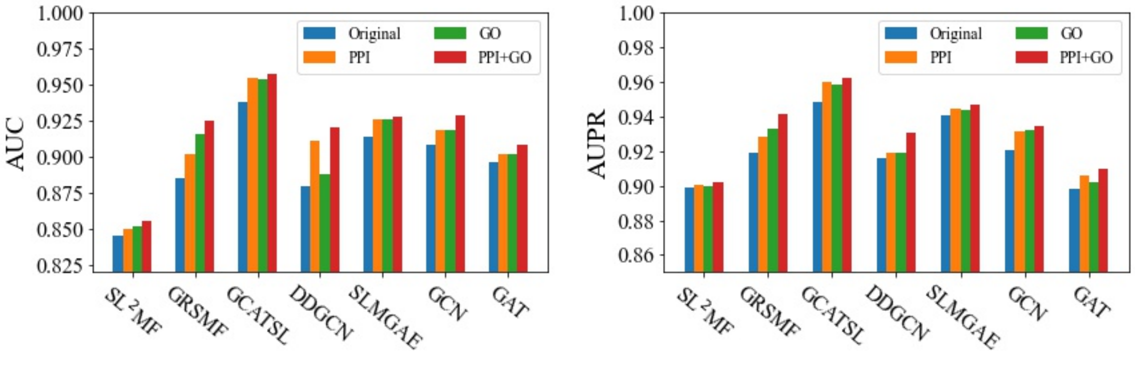
The representation dimension d_3 is important to our model. We select its values from $\{16, 32, 64, 128, 256, 512, 1024\}$. As shown in Figure 3 (a), a small or large value of representation dimension d_3 is not good for the model performance and the best performance is achieved when d_3 is set to 128. In the GCN-based interaction graph encoder, the number of layers L determines the aggregation of neighbors' features. To evaluate its influences on our pre-training model, we change its value from 1 to 5 with a step size of 1. It can be observed in Figure 3 (b) that as L increases, the performance first increases and then decreases. In particular, our model achieves the best performance when L is set as 2. We note that more layers do not help improve the performance. This is because too many layers can lead to the problem of “over-smoothing", which is faced by most of GNN models (Chen *et al.*, 2020).

In addition, weight factor λ controls the contributions of two different gene interaction graphs (i.e., PPI graph and GO graph). To determine its influences, we evaluate our model by ranging its value from 0 to 1 with a step value of 0.1. It should be noted that $\lambda = 0$ means only GO similarity data are used for pre-training and $\lambda = 1$ means only PPI data are used for pre-training. The results in Figure 3 indicate that our pre-training model is relatively robust against λ, and thus we set it as 0.5 in our experiments.

5 Conclusion and Future Work

In this work, we propose a novel universal pre-training framework based on graph neural networks for critical link prediction in biomedical networks - this is the first work in this area. Firstly, we leverage multiple sources of biological data to construct interaction graphs for nodes (i.e., proteins and drugs). In particular, we introduce CNN to capture latent features of protein sequences to generate initial features for proteins. Meanwhile, we adopt GCN to model drug molecular structures and learn initial drug features. Secondly, with the interaction graphs and initial features as inputs, we further design a GCN-based interaction graph encoder to aggregate the features of a node and its neighbors in the graph. Finally, our model is pre-trained on graph reconstruction tasks. We conducted extensive experiments on two important downstream tasks, i.e., SL prediction and DTI prediction, experimental results demonstrate our pre-trained model outperforms existing state-of-the-art techniques significantly for both tasks, in term of both accuracy and efficiency. Moreover, the node features pre-trained from our PT-GNN model are able to improve the performance and reduce training times for existing models in downstream link prediction tasks.

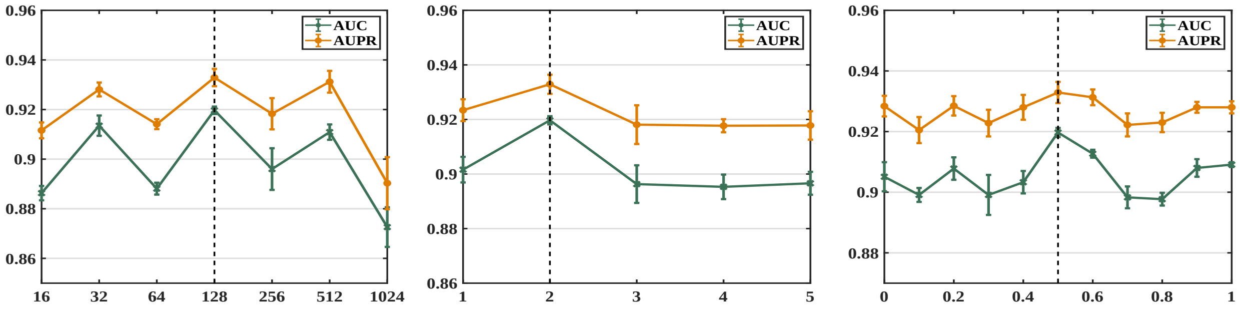
While our pre-trained model achieves great performance, there are still some limitations expected to be addressed. First, the node representations learned based on graph reconstruction task are sub-optimal, as the known ground truth data are sparse. In the future, we would combine with self-supervised learning and integrate more prior domain knowledge to learn more valuable and robust node representations. In particular, we can further enrich the representations of drugs and proteins by fully exploiting cell line data containing gene-drug association patterns (Chi *et al.*, 2021; Liu *et al.*, 2021b). Second, since our PT-GNN model has better performance on SL



(a) AUC

(b) AUPR

Fig. 2. Comparison between different methods with their variants on SL prediction in terms of AUC and AUPR. ‘Original’ refer to using default features. ‘PPI’, ‘GO’, and ‘PPI + GO’ refer to using the features pre-trained on PPI graph, GO graph and both graphs respectively.



(a) Dimension of representation d_3

(b) Number of layers of encoder L

(c) Weight factor λ

Fig. 3. Parameter sensitivity analysis for our pre-training model in terms of dimension of representation d_3, number of layers of encoder L and weight factor λ.

prediction and DTI prediction than state-of-the-art methods, it is desirable to predict novel SL pairs and DTI pairs with PT-GNN and then collaborate with biologists to validate them.

Acknowledgements

This work was supported by the National Natural Science Foundation of China [61873089] and the Key Program of National Natural Science Foundation of China [62032007].

References

Cai, R. *et al.* (2020). Dual-dropout graph convolutional network for predicting synthetic lethality in human cancers. *Bioinformatics*, **36**(16), 4458–4465.

Chen, D. *et al.* (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3438–3445.

Chen, X. *et al.* (2017a). Long non-coding rnas and complex diseases: from experimental results to computational models. *Briefings in bioinformatics*, **18**(4), 558–576.

Chen, X. *et al.* (2017b). A novel approach based on katz measure to predict associations of human microbiota with non-infectious diseases. *Bioinformatics*, **33**(5), 733–739.

Chen, X. *et al.* (2018a). Bnpmda: bipartite network projection for mirna–disease association prediction. *Bioinformatics*, **34**(18), 3178–3186.

Chen, X. *et al.* (2018b). Predicting mirna–disease association based on inductive matrix completion. *Bioinformatics*, **34**(24), 4256–4265.

Chen, X. *et al.* (2019). Micrornas and complex diseases: from experimental results to computational models. *Briefings in bioinformatics*, **20**(2), 515–539.

Chi, C. *et al.* (2021). Bipartite graph-based approach for clustering of cell lines by gene expression-drug response associations. *Bioinformatics*.

Codling, E. A. *et al.* (2008). Random walk models in biology. *Journal of the Royal society interface*, **5**(25), 813–834.

Consortium, U. (2019). Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, **47**(D1), D506–D515.

Devlin, J. *et al.* (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Dhingra, N. *et al.* (2021). Border-seggcn: Improving semantic segmentation by refining the border outline using graph convolutional network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 865–875.

Fu, H. *et al.* (2021). Mvgcn: data integration through multi-view graph convolutional network for predicting links in biomedical bipartite networks. *Bioinformatics*.

Guo, J. *et al.* (2016). Synlethdb: synthetic lethality database toward discovery of selective and sensitive anticancer drug targets. *Nucleic acids research*, **44**(D1), D1011–D1017.

Hao, Z. *et al.* (2021). Prediction of synthetic lethal interactions in human cancers using multi-view graph auto-encoder. *IEEE Journal of Biomedical and Health Informatics*, **25**(10), 4041–4051.

Hong, Z. *et al.* (2020). Identifying enhancer–promoter interactions with neural network based on pre-trained dna vectors and attention mechanism. *Bioinformatics*, **36**(4), 1037–1043.

Hu, W. *et al.* (2020a). Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*.

Hu, Z. *et al.* (2020b). Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867.

Huang, J. *et al.* (2019). Predicting synthetic lethal interactions in human cancers using graph regularized self-representative matrix factorization. *BMC bioinformatics*, **20**(19), 1–8.

Huang, K. *et al.* (2021). Moltrans: Molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*, **37**(6), 830–836.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Lee, I. *et al.* (2019). Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS computational biology*, **15**(6), e1007129.

Li, X. *et al.* (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

Liu, L. *et al.* (2021a). Hpofiller: identifying missing protein–phenotype associations by graph convolutional network. *Bioinformatics*, page btab224.

Liu, X. *et al.* (2021b). Graphcdr: a graph neural network method with contrastive learning for cancer drug response prediction. *Briefings in Bioinformatics*.

Liu, Y. *et al.* (2016). Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. *PLoS computational biology*, **12**(2), e1004760.

Liu, Y. *et al.* (2019). Sl 2 mf: Predicting synthetic lethality in human cancers via logistic matrix factorization. *IEEE/ACM transactions on computational biology and bioinformatics*, **17**(3), 748–757.

Liu, Y. *et al.* (2021c). Contextualized graph attention network for recommendation with item knowledge graph. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, Z. *et al.* (2020). Basconv: Aggregating heterogeneous interactions for basket recommendation with graph convolutional neural network. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 64–72. SIAM.

Long, Y. *et al.* (2020a). Ensembling graph attention networks for human microbe–drug association prediction. *Bioinformatics*, **36**(Supplement_2), i779–i786.

Long, Y. *et al.* (2020b). Predicting human microbe-drug associations via graph convolutional network with conditional random field. *Bioinformatics*, **36**(19), 4918–4927.

Long, Y. *et al.* (2021). Graph contextualized attention network for predicting synthetic lethality in human cancers. *Bioinformatics*, **37**(16), 2432–2440.

Lu, Y., Jiang, X., Fang, Y., and Shi, C. (2021). Learning to pre-train graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4276–4284.

Luo, J. and Long, Y. (2020). Ntshmda: Prediction of human microbe-disease association based on random walk by integrating network topological similarity. *IEEE/ACM transactions on computational biology and bioinformatics*, **17**(4), 1341–1351.

Ma, Y. and Ma, Y. (2021). Hypergraph-based logistic matrix factorization for metabolite-disease interaction prediction. *Bioinformatics*.

Mi, L. and Chen, Z. (2020). Hierarchical graph attention network for visual relationship detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13886–13895.

Navarin, N. *et al.* (2018). Pre-training graph neural networks with kernels. *arXiv preprint arXiv:1811.06930*.

Nguyen, T. *et al.* (2021). Graphdta: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, **37**(8), 1140–1147.

Oughtred, R. *et al.* (2019). The biogrid interaction database: 2019 update. *Nucleic acids research*, **47**(D1), D529–D541.

Öztürk, H. *et al.* (2018). Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, **34**(17), i821–i829.

Peng, J. *et al.* (2016). Intego2: a web tool for measuring and visualizing gene semantic similarities using gene ontology. *BMC genomics*, **17**(5), 553–560.

Qian, Y. *et al.* (2021). Speech-language pre-training for end-to-end spoken language understanding. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7458–7462. IEEE.

Strodthoff, N. *et al.* (2020). Udsmprot: universal deep sequence models for protein classification. *Bioinformatics*, **36**(8), 2401–2409.

Su, C. *et al.* (2020). Network embedding in biomedical data science. *Briefings in bioinformatics*, **21**(1), 182–197.

Veličković, P. *et al.* (2018). Graph attention networks. In *International Conference on Learning Representations*.

Wan, F. *et al.* (2019). Neodti: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions. *Bioinformatics*, **35**(1), 104–111.

Wang, C.-C. *et al.* (2021). Circular rnas and complex diseases: from experimental results to computational models. *Briefings in bioinformatics*, **22**(6), bbab286.

Wishart, D. S. *et al.* (2018). Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, **46**(D1), D1074–D1082.

Yao, L. *et al.* (2019). Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Yue, X. *et al.* (2020). Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, **36**(4), 1241–1251.

Zeng, H. *et al.* (2016). Convolutional neural network architectures for predicting dna–protein binding. *Bioinformatics*, **32**(12), i121–i127.

Zhang, Z.-C. *et al.* (2020). A graph regularized generalized matrix factorization model for predicting links in biomedical bipartite networks. *Bioinformatics*, **36**(11), 3474–3481.

Zheng, X. *et al.* (2013). Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1025–1033.

Zhu, Y. *et al.* (2021). Tgsa: Protein-protein association-based twin graph neural networks for drug response prediction with similarity augmentation. *Bioinformatics*.

Zong, N. *et al.* (2017). Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations. *Bioinformatics*, **33**(15), 2337–2344.