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

OXFORD

Systems Biology

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

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

[1]College of Computer Science and Electronic Engineering, Hunan University, Changsha 410000, China
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore
[3]Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), 138632, Singapore
[4]Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, 639798, Singapore
[5]School of Information Systems, Singapore Management University, 178902, Singapore

*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 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 designed 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 pre-training model to integrate different data sources for link prediction in biomedical networks. First, we design expressive deep learning methods (e.g., CNN and GCN) to learn features for individual nodes from sequence and structure data. Second, we further propose a GCN-based encoder to effectively refine the features of nodes by modelling the dependencies among nodes in the network data. Third, the model is pre-trained based on graph reconstruction 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 that the features generated by our pre-training model can help to improve the performance and reduce the training time for existing GNN models. In addition, fine-tuning the pre-trained model to a specific task can also achieve the performance comparable to the state-of-the-art methods.

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 microbe-drug association prediction (Long *et al.*, 2020b). We can classify these computational methods into three main categories, i.e., diffusion-based methods, matrix factorization methods and graph neural network (GNN) methods.

In particular, diffusion-based methods leverage random walks to full exploit the topological structure information of nodes in the network to infer potential links. For example, Chen *et al.* (2017) developed a KATZ-based computational method to predict potential microbe-disease associations by calculating walking number of node pairs in bipartite network. Following that, Luo and Long (2020) constructed a heterogeneous network, and further proposed a random walk-based model named NTSHMDA to predict microbe-disease associations, which uses network topological similarity to influence the walking preference of the walker. Chen *et al.* (2018) developed a bipartite network projection-based method of BNPMDA to infer latent microRNA-disease associations, which takes into account the bias preference degree of a node for different neighbors. In addition, Zong *et al.* (2017) proposed a similarity-based method to predict drug-target associations, which utilizes DeepWalk algorithm (Perozzi *et al.*, 2014) to calculate similarities between drugs and targets based on the topological information in a heterogeneous network.

Matrix factorization has shown promising performance in exploring intrinsic structure of various data (Zhang *et al.*, 2020) 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. Xiao *et al.* (2018) released a graph regularized non-negative matrix factorization model for miRNA-disease association prediction. More recently, 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.

Graph neural networks (GNNs, e.g., GCN and GAT) 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 named GCNMDA to predict microbe-drug associations by using GCN to aggregate representations of neighbors. After that, Long *et al.* (2020a) proposed another GAT-based model of EGATMDA for microbe-drug association prediction by leveraging GAT to capture hierarchical structure information. Finally, Cai *et al.* (2020) developed a dual-dropout GCN-based framework for synthetic lethality prediction.

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.* (2020b,a). However, such manual feature extraction requires domain-specific knowledge.

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 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 the node features generated by our pre-training model are effective and can help to improve the performance and reduce the training time for existing GNN models. In addition, fine-tuning the pre-trained model to a specific task can also achieve performance comparable to the state-of-the-art methods.

Overall, our main contributions are summarized as follows:

- A generic pre-training graph neural network framework 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., PPI network and 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 pre-training model is highly effective for downstream tasks.

2 Related work

In this section, we first present graph neural networks, including graph convolutional network (GCN) and graph attention network (GAT). Then, we 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, graph convolutional networks (GCNs), proposed by Kipf and Welling (2016), aim to learn node representations by aggregating the features of neighbours. Due to its great performance, GCN has attracted increasing attentions 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. Graph attention network (GAT) (Veličković *et al.*, 2018) is an extension of GCN, which focuses on more important neighbors by assigning greater weight values to them. Such operation enables the model to learn more informative representations. Graph attention network has been successfully applied for social influence analysis (Qiu *et al.*, 2018), recommender system (Wu *et al.*, 2019) 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.

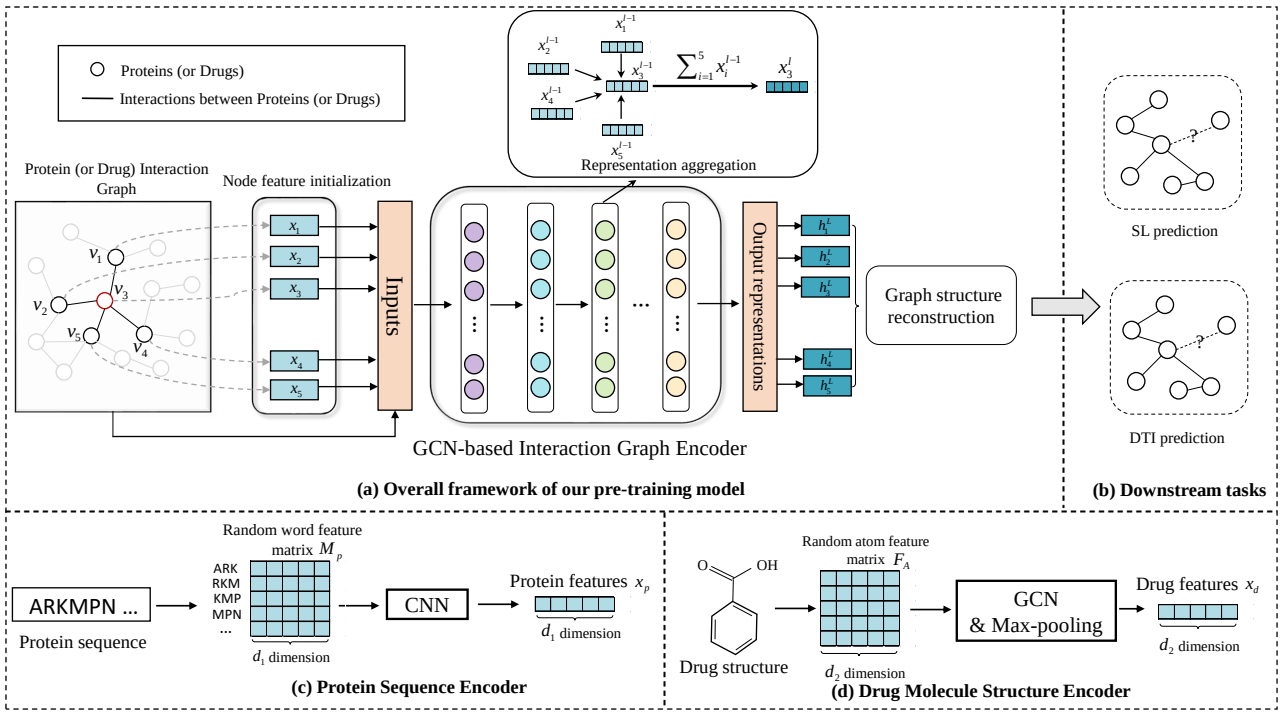


Fig. 1. The overall architecture of GCATSL for gene representation learning and SL prediction. (a) Overall framework of our pre-training model. (b) Downstream tasks. (c) Protein sequence encoder to learn initial features for proteins. (d) Drug molecule structure encoder to learn initial features for drugs.

More recently, pre-training has achieved significant success in multiple domains, such as natural language processing (Devlin *et al.*, 2018; Chung *et al.*, 2020), computer vision (Qi *et al.*, 2020; Li *et al.*, 2020) and link prediction (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. EPIVAN pre-trains models on genome sequences to learn DNA vectors, which are then used to encode enhancers and promoters. 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. While much effort has been made to use pre-training to solve different biological issues, few pre-training methods have been proposed 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 protein-protein interaction (PPI) data and the drug-drug interaction (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

For proteins, we firstly leverage the PPI data to build the PPI graph $G_{\text{PPI}} = \{V_{\text{P}}, E_{\text{PPI}}\}$, where V_{P} denotes the set of nodes (i.e., proteins) and E_{PPI} denotes the set of edges describing the interaction relationships between proteins. Moreover, we also calculate a semantic similarity matrix for proteins based on their GO terms. To extract more important association pairs, we apply the random walk with restart (RWR) algorithm on this similarity matrix and construct the protein GO similarity graph $G_{\text{GO}} = \{V_{\text{P}}, E_{\text{GO}}\}$, by selecting the top-$N$ neighbors as interaction pairs for a given protein. In addition, the protein sequence data is treated as the protein attribute information, and a protein sequence encoder, as shown in Figure 1(c), is developed to exploit this attribute information for pre-training the protein representations.

For drugs, we utilize DDI data from the Drugbank database to construct the DDI graph $G_{\text{D}} = \{V_{\text{D}}, E_{\text{DDI}}\}$, where V_{D} denotes the set of nodes (i.e, drugs) and E_{DDI} denotes the set of edges describing the interaction relationships between drugs. For each drug, its drug molecule structure is considered as the drug attribute information, and a drug molecule structure encoder, as shown in Figure 1(d), is developed to exploit this drug attribute information for pre-training the drug representations.

3.2 Node Feature Initialization

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

3.2.1 Protein Sequence Encoder

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 $\mathbf{F}_{\mathrm{W}}$ are trainable parameters, which are randomly initialized and can be updated in the training phase for more accurately capturing the intrinsic features of sequences.

As shown in Figure 1(c), we can convert a protein sequence s_p into a feature matrix $\mathbf{M}_p$, where each row denotes a d_1-dimension word feature vector. To learn protein features from the sequence data, we design a two-layer convolutional neural network, including a 1D convolutional layer and a max-pooling layer. The input of the convolutional neural network is the word feature matrix $\mathbf{M}_p$. The convolutional layer is designed to learn local features, and the max-pooling layer is used to reduce dimension. 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). Moreover, 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. Subsequently, by applying this sequence encoder on all protein seqeuence data, we can obtain a feature matrix $\mathbf{X}_{\mathrm{P}} \in \mathbb{R}^{N_P \times d_1}$ for all proteins, where N_P denotes the number of proteins.

3.2.2 Drug Molecule Structure Encoder

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

In this work, we implement a graph convolutional network on the molecule graph G_{d} to learn 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_{d} to aggregate the features of multi-hop neighbors. More specifically, the k-th GCN layer can be formulated as follows,

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

where $\widetilde{\mathbf{A}}_{\mathrm{d}} = \mathbf{D}_{\mathrm{d}}{}^{-\frac{1}{2}}\mathbf{A}_{\mathrm{d}}\mathbf{D}_{\mathrm{d}}{}^{-\frac{1}{2}}$ is a normalized adjacency matrix. $\mathbf{D}_{\mathrm{d}}$ is a diagonal matrix with the diagonal element being $\mathbf{D}_{\mathrm{d}}(i,i) = \sum_{j=1}^{N_a}\mathbf{A}_{\mathrm{d}}(i,j)$. $\mathbf{W}_1^{(k-1)}$ 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. $\mathbf{R}_{\mathrm{d}}{}^{(k)}$ denotes the feature matrix of atoms at the k-th layer. Note that $\mathbf{R}_{\mathrm{d}}{}^{(0)}$ is the original feature matrix $\mathbf{F}_{\mathrm{A}}$ of atoms. After K GCN layers, we can obtain the atom representations $\mathbf{R}_{\mathrm{d}}{}^{(K)}$.

To learn the drug feature, we further implement a max-pooling layer on $\mathbf{R}_{\mathrm{d}}{}^{(K)}$ 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 o 1. By applying the drug structure encoder on the molecule 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, where N_D represents the number of 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 molecule 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 in Eq. (5). Such operation enables the proposed model to flexibly learn the properties of molecule 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 used 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. The only difference is the input interaction graph and the initial node features. In the following sections, we only describe the operations for learning protein representations with input graph G_{PPI} and initial node features $\mathbf{X}_{\mathrm{P}}$.

Let us denote the adjacency matrix of the PPI graph $G_{\mathrm{PPI}} = \{V_{\mathrm{P}}, E_{\mathrm{PPI}}\}$ by $\mathbf{A}_{\mathrm{PPI}} \in \mathbb{R}^{N_P \times N_P}$. For a node v_i in G_{PPI}, 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,

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

where $\mathbf{h}_j^{(\ell-1)}$ denotes the feature representations of the node v_j at the $(\ell-1)$-th layer, and $\mathcal{N}_i$ denotes the first-hop neighbors of v_i in the graph. Note that $\mathcal{N}_i$ also includes v_i in this work. $\mathrm{AGGREGATE}(\cdot)$ 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,

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

where $\widetilde{\mathbf{A}}_{\mathrm{PPI}}$ is the normalized diagonal adjacency matrix with self-connection, $\mathbf{H}_{\mathrm{PPI}}{}^{(\ell-1)}$ denotes the outputs of the model at the $(\ell-1)$-th layer. Note that $\mathbf{H}_{\mathrm{PPI}}{}^{(0)}$ 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 $\mathbf{H}_{\mathrm{PPI}} \in \mathbb{R}^{N_P \times d_3}$, where d_3 denotes the dimension of the protein representation features.

Note that $\mathbf{H}_{\mathrm{PPI}}$ is the protein representations obtained from the PPI graph G_{PPI} with node features $\mathbf{X}_{\mathrm{P}}$. Similarly, we can obtain the protein representations $\mathbf{H}_{\mathrm{GO}} \in \mathbb{R}^{N_P \times d_3}$ from the protein GO graph G_{GO} with node features $\mathbf{X}_{\mathrm{P}}$, and the drug representations $\mathbf{H}_{\mathrm{DDI}} \in \mathbb{R}^{N_D \times d_3}$ from the DDI graph G_{DDI} 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 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),

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

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

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. 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. In this work, for better training, we adopt negative sampling strategy to train our model. Ω^{+} and Ω^{-} represent the sets of positive and negative samples for model training, respectively.

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. For pre-training, we 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 task of 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. It should be noted that we use simultaneously PPI and GO graphs to pre-train our model to learn protein features. Here we use factors γ and $(1-\gamma)$ to weight the influences of PPI and GO graphs on our model, respectively.

DTI prediction. We collected 1,113,252 drug-drug interactions (DDI) involving 3,543 drugs from Drugbank (Wishart *et al.*, 2018) to pre-train the model to learn drug features. Meanwhile, we downloaded the SMILES (Simplified Molecular Input Line Entry System) for these 3,543 drugs from Drugbank to construct the drug molecule graphs. In addition, we derived drug-target interaction data from Drugbank for experimental validation in DTI prediction task. In particular, we selected 9,679 drug-target interactions between 1,971 drugs and 1,899 targets from Drugbank, where all the drugs and targets here have SMILES and sequences respectively.

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 Long *et al.* (2020b), we adopt 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 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 numbers of neurons for the first and second hidden layers are set to 256 and 128 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 of GCN-based encoder L and weight factor γ, in the following section.

4.1.3 Baseline methods

In this work, we validate the performance of our model via fine-tuning and two downstream tasks. We introduce seven state-of-the-art baseline methods for downstream tasks as follows:

- GCATSL (Long *et al.*, 2020b) is a novel graph attention network-based model developed for SL prediction.
- SLMGAE (Huang *et al.*, 2019) 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.
- NeoDTI (Wan *et al.*, 2019) develops a end-to-end deep learning model to predict drug-target interactions by integrating heterogeneous biological data.
- NRLMF (Liu *et al.*, 2016) utilizes neighborhood regularized logistic matrix factorization to learn node representations for drug-target interaction prediction.
- GCN (Kipf and Welling, 2016) is a benchmark graph convolutional network.
- GAT (Veličković *et al.*, 2018) is a benchmark graph attention network.

For all the above methods, we adopt the default parameters from their original implementations. GCN and GAT are used as baseline models for both SL and DTI prediction tasks. GCATSL, SLMGAE and DDGCN are applied for SL prediction task while NeoDTI and NRLMF are applied for drug-target prediction task.

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. We first pre-train our model and subsequently use the pre-trained representations to initialize features of both genes (or targets) and drugs in downstream tasks. Hence, each baseline method (i.e., using original features) has an additional variant (i.e., using our pre-trained features). We can thus validate the effectiveness of our pre-training model by comparing different baseline methods with their variants.

4.2.1 SL prediction

Table 1 shows the comparison results of various methods on SL prediction task. As mentioned above, each baseline method used two types of features as inputs, i.e., original features and our pre-trained features. We can observe that for all the methods, pre-trained features achieved better performance than the original features consistently. For example, GCATSL

with pre-trained representations obtained 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 can enhance prediction performance for SL prediction. 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 performances.

In addition, we analyse the influences of our pre-trained features on the training time of various baseline models. As shown in Table 1, 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.

Table 1. Performance comparison of baseline methods with different feature initialization on SL prediction in 5-fold CV.

Methods	Features	Epochs	AUC	AUPR
GCATSL	Pre-trained	100	**0.9576±0.0016**	**0.9620±0.0018**
	Original	600	0.9375±0.0024	0.9483±0.0018
SLMGAE	Pre-trained	200	**0.9279±0.0040**	**0.9465±0.0032**
	Original	300	0.9140±0.0049	0.9405±0.0030
DDGCN	Pre-trained	10	**0.9204±0.0103**	**0.9305±0.0075**
	Original	2000	0.8796±0.0080	0.9161±0.0046
GCN	Pre-trained	100	**0.9286±0.0056**	**0.9345±0.0052**
	Original	200	0.9083±0.0034	0.9203±0.0027
GAT	Pre-trained	100	**0.9087±0.0091**	**0.9097±0.0130**
	Original	200	0.8964±0.0136	0.8981±0.0157

4.2.2 DTI prediction

Similarly, we pre-train our model on DDI and PPI networks to derive drug and protein features respectively for the downstream task of DTI prediction. Here, we use the Gaussian kernel similarity (van Laarhoven *et al.*, 2011) as original features in various baseline models for DTI prediction. As shown in Table 2, pre-trained features outperform the original features (i.e., Gaussian kernel similarity) consistently, which demonstrates once again that our pre-trained features can help improve the performances of various methods for downstream tasks. Similarly, it could be found in Table 2 that the pre-trained features can help to reduce the training time of various baseline methods significantly.

Table 2. Performance comparison of baseline methods with different feature initialization on DTI prediction in 5-fold CV.

Methods	Features	Epochs	AUC	AUPR
NeoDTI	Pre-trained	100	**0.8386±0.0116**	**0.8626±0.0111**
	Gaussian	300	0.7903±0.0103	0.8281±0.0112
NRLMF	Pre-trained	50	**0.9223±0.0030**	**0.9388±0.0020**
	Gaussian	100	0.8962±0.0066	0.9240±0.0040
GCN	Pre-trained	100	**0.9052±0.0036**	**0.9097±0.0053**
	Gaussian	200	0.8885±0.0047	0.9036±0.0032
GAT	Pre-trained	100	**0.8716±0.0052**	**0.8954±0.0051**
	Gaussian	200	0.8657±0.0082	0.8827±0.0036

4.2.3 Impact of fine-tuning

In this section, we further show the performance of fine-tuning our pre-trained model in the task of SL prediction. In particular, we first pre-train our model by optimizing the loss function in Equation (5) to reconstruct both PPI and GO networks. Then, we fine-tune both the protein sequence encoder and GCN based graph encoder by reconstructing the SL graph instead. We also feed the pre-trained features to a GCN model for SL prediction. As shown in Table 3, we can observe that our fine-tuned model achieved an average AUC of 0.9197 and average AUPR of 0.9329, which are 1.26 % and 1.37 % better than that of GCN. Besides, our fine-tuned model takes less training time than GCN. The results demonstrate once again the effectiveness of our pre-training model.

In general, input features for the graph neural network models (e.g., GCN) need to be manually extracted, which require domain-specific knowledge and are time-consuming. Instead, our pre-training model is able to automatically learn features for nodes from a comprehensive knowledge domain, which can be used for different downstream tasks. Therefore, we believe that our model has a powerful capability in real-life applications.

Table 3. Performance comparison between fine-tuning and GCN in SL prediction task.

Methods	Epochs	AUC	AUPR
Fine-tuning	20	**0.9197±0.0015**	**0.9329±0.0035**
GCN	200	0.9083±0.0034	0.9203±0.0027

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.

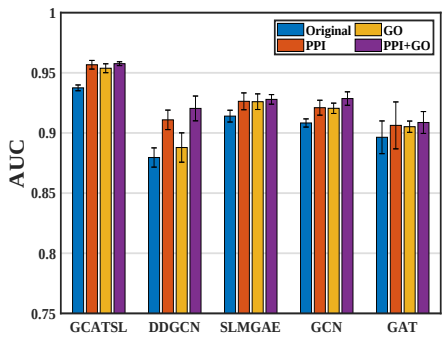


Fig. 2. Comparison between different methods and their variants on SL prediction in terms of AUC.

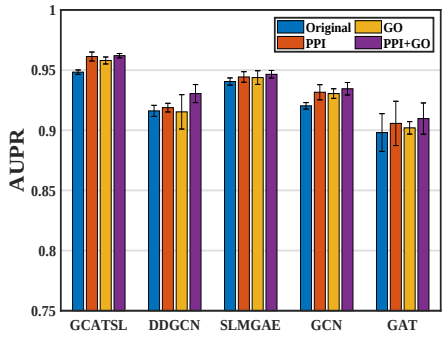
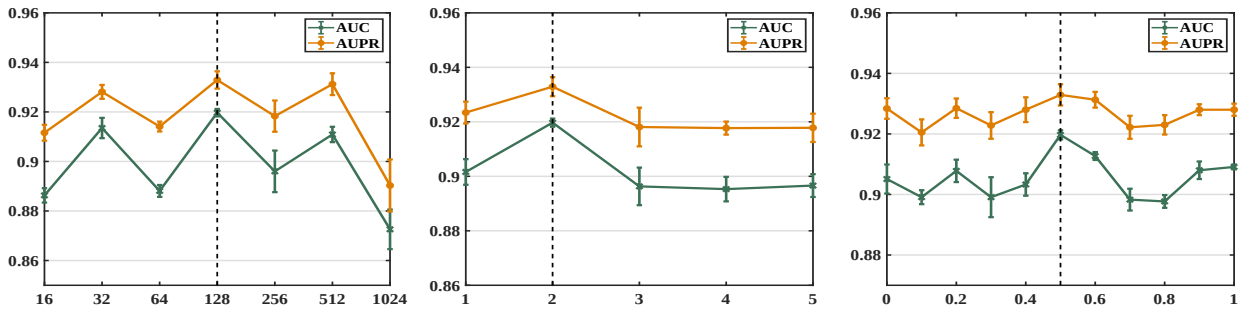


Fig. 3. Comparison between different methods and their variants on SL prediction in terms of AUPR.



(a) Dimension of representation d_3 (b) Number of layers of encoder L (c) Weight factor γ

Fig. 4. 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 γ.

Table 2 and Table 3 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 our pre-training.

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 γ. Here, we fine-tune the pre-trained model with different parameter values to analyze their impact 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 4 (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 4 (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 γ 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 $\gamma = 0$ means only GO similarity data are used for pre-training and $\gamma = 1$ means only PPI data are used for pre-training. The results in Figure 4 indicate that our pre-training model is relatively robust against γ, and thus we set it as 0.5 in our experiments.

5 Conclusion

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.

Acknowledgements

This work was supported by the National Natural Science Foundation of China [61873089], the Key Program of National Natural Science Foundation of China [62032007] and the Chinese Scholarship Council (CSC) [201906130027].

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.* (2017). 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.* (2018). Bnpmda: bipartite network projection for mirna–disease association prediction. *Bioinformatics*, **34**(18), 3178–3186.

Chung, Y.-A. *et al.* (2020). Semi-supervised speech-language joint pre-training for spoken language understanding. *arXiv preprint arXiv:2010.02295*.

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

Devlin, J. *et al.* (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

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.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks.

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, 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). Sl2mf: Predicting synthetic lethality in human cancers via logistic matrix factorization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **17**(3), 748–757.

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.

Lu, Y. *et al.* (2021). Learning to pre-train graph neural networks.

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.

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

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.

Perozzi, B. *et al.* (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Qi, D. *et al.* (2020). Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*.

Qiu, J. *et al.* (2018). Deepinf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2110–2119.

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.

van Laarhoven, T. *et al.* (2011). Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics*, **27**(21), 3036–3043.

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.

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.

Wu, Q. *et al.* (2019). Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*, pages 2091–2102.

Xiao, Q. *et al.* (2018). A graph regularized non-negative matrix factorization method for identifying microrna-disease associations. *Bioinformatics*, **34**(2), 239–248.

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.

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.

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