Graph embedding on biomedical networks: methods, applications and evaluations

Xiang Yue1,*, Zhen Wang1, Jingong Huang2, Srinivasan Parthasarathy1, Soheil Moosavinasab3, Yungui Huang3, Simon M. Lin3, Wen Zhang4, Ping Zhang1,5 and Huan Sun1,*

1Department of Computer Science and Engineering, 2Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA, 3Research Information Solutions and Innovation, The Research Institute at Nationwide Children’s Hospital, Columbus, OH, USA, 4College of Informatics, Huazhong Agricultural University, Wuhan, Hubei, China and 5Department of Biomedical Informatics, The Ohio State University, Columbus, OH, USA

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on April 9, 2019; revised on August 25, 2019; editorial decision on September 14, 2019; accepted on September 26, 2019

Abstract

Motivation: Graph embedding learning that aims to automatically learn low-dimensional node representations, has drawn increasing attention in recent years. To date, most recent graph embedding methods are evaluated on social and information networks and are not comprehensively studied on biomedical networks under systematic experiments and analyses. On the other hand, for a variety of biomedical network analysis tasks, traditional techniques such as matrix factorization (which can be seen as a type of graph embedding methods) have shown promising results, and hence there is a need to systematically evaluate the more recent graph embedding methods (e.g. random walk-based and neural network-based) in terms of their usability and potential to further the state-of-the-art.

Results: We select 11 representative graph embedding methods and conduct a systematic comparison on 3 important biomedical link prediction tasks: drug-disease association (DDA) prediction, drug–drug interaction (DDI) prediction, protein–protein interaction (PPI) prediction; and 2 node classification tasks: medical term semantic type classification, protein function prediction. Our experimental results demonstrate that the recent graph embedding methods achieve promising results and deserve more attention in the future biomedical graph analysis. Compared with three state-of-the-art methods for DDAs, DDIs and protein function predictions, the recent graph embedding methods achieve competitive performance without using any biological features and the learned embeddings can be treated as complementary representations for the biological features. By summarizing the experimental results, we provide general guidelines for properly selecting graph embedding methods and setting their hyper-parameters for different biomedical tasks.

Availability and implementation: As part of our contributions in the paper, we develop an easy-to-use Python package with detailed instructions, BioNEV, available at: https://github.com/xiangyue9607/BioNEV, including all source code and datasets, to facilitate studying various graph embedding methods on biomedical tasks.

Contact: yue.149@osu.edu or sun.397@osu.edu

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Graphs (a.k.a. networks) have been widely used to represent biomedical entities (as nodes) and their relations (as edges). Analyzing biomedical graphs can greatly benefit various important biomedical tasks, such as predicting potential drug indications (a.k.a. drug repositioning) based on disease association (DDA) graphs (Gottlieb et al., 2011), detecting long non-coding RNA (lncRNA) functions based on lncRNA–protein interaction networks (Zhang et al., 2018d), and assisting clinical decision making via disease-symptom graphs (Rotmensch et al., 2017).

In order to analyze the graph data, a surge of graph embedding (a.k.a. network embedding or graph representation learning) methods (Grover and Leskovec, 2016; Perozzi et al., 2014; Ribeiro et al., 2017; Tang et al., 2015) have been proposed, where their goal is to automatically learn a low-dimensional feature representation for
Fig. 1. Pipeline for applying graph embedding methods to biomedical tasks. Low-dimensional node representations are first learned from biomedical networks by graph embedding methods and then used as features to build specific classifiers for different tasks. For (a) matrix factorization-based methods, they use a data matrix (e.g., adjacency matrix) as the input to learn embeddings through matrix factorization. For (b) random walk-based methods, they first generate sequences of nodes through random walks and then feed the sequences into the word2vec model (Mikolov et al., 2013) to learn node representations. For (c) neural network-based methods, their architectures and inputs vary from different models (see Section 2 for details).

The results indicate that the graph embedding methods can achieve very competitive or better performance while being general (i.e. applied on different graphs and tasks). The learned embedding can also be treated as a complementary representation for the biological features. By summarizing the experimental results, we provide insightful observations as well as suggestions for selecting proper graph embedding methods and setting their hyper-parameters for biomedical prediction tasks. For instance, for MF-based methods, modeling higher-order proximity (Cowen et al., 2017) [e.g. HOPE (Ou et al., 2016) and GraRep (Cao et al., 2015)] is more useful in biomedical link prediction tasks compared with node classification tasks. For random walk-based methods, DeepWalk (Perozzi et al., 2014) and node2vec (Grover and Leskovec, 2016) perform better in node classification tasks while struc2vec achieves better results in biomedical link prediction tasks. We also discuss the connections between embedding methods and the recent network propagation and diffusion methods in biomedical graph analysis (Cowen et al., 2017). Additionally, we illustrate a few new trends and directions (e.g. transfer learning in biomedical graph embedding) to encourage future work.

To summarize, our contributions are 3-fold:

- We provide an overview of different types of graph embedding methods, and discuss how they can be used in three important biomedical link prediction tasks: DDA, DDIs and PPIs prediction; and two node classification tasks, protein function prediction and medical term semantic type classification.
- We compile 7 benchmark datasets for all the above prediction tasks, and use them to systematically evaluate 11 representative graph embedding methods selected from different categories (i.e. 5 MF-based, 3 random walk-based, 3 neural network-based). We discuss our observations from extensive experiments and provide some insights and guidelines for how to choose embedding methods (including their hyper-parameter settings).
- We develop an easy-to-use Python package with detailed instructions, BioNEV (Biomedical Network Embedding Evaluation), available at: https://github.com/xiangyue9607/BioNEV, including all source code and datasets, to facilitate studying various graph embedding methods on biomedical tasks.

2 Overview of graph embedding methods

In this section, we provide a brief overview of different graph embedding methods that are categorized into three groups: MF-based, random walk-based and neural network-based (Fig. 1 provides a high-level illustration).
2.1 MF-based methods
MF has been widely adopted for data analyses. Essentially, it aims to factorize a data matrix into lower dimensional matrices and still keep the manifold structure and topological properties hidden in the original data matrix. Pioneer work in this category dates back to the early 2000s, such as Isomap (Tenenbaum et al., 2000), Locally Linear Embedding (Roweis and Saul, 2000) and LEs (Belkin and Niyogi, 2003). Traditional MF has many variants, such as singular value decomposition (SVD) and graph factorization (GF) (Ahmed et al., 2013). And they often focus on factorizing the first-order data matrix (e.g. adjacency matrix).

More recently, researchers focus on designing various high-order data proximity matrices to preserve the graph structure and propose various MF-based graph embedding learning methods. For example, GraRep (Cao et al., 2015) considers the high-order proximity of the network and designs k-step transition probability matrices for factorization. HOPE (Ou et al., 2016) also considers the high-order proximity. But different from GraRep, it adopts some well-known network similarity measures such as Katz Index and Common Neighbors to preserve network structures.

2.2 Random walk-based methods
Inspired by the word2vec (Mikolov et al., 2013) model, a popular word embedding technique from Natural Language Processing (NLP), which tries to learn word representations from sentences, random walk-based methods are developed to learn node representations by generating ‘node sequences’ through random walks in graphs. Specifically, given a graph and a starting node, random walk-based methods first select one of the node’s neighbors randomly and then move to this neighbor. This procedure is repeated to obtain node sequences. Then the word2vec model is adopted to learn embeddings based on the generated sequences of nodes. In this way, structural and topological information can be preserved into latent features.

One of the initial works in this category is DeepWalk ( Perozi et al., 2014 ), which performs truncated random walks on a graph. Compared with DeepWalk, node2vec ( Grover and Leskovec, 2016 ) adopts a flexible biased random walk procedure that smoothly combines breadth-first sampling and depth-first sampling to generate node sequences. Furthermore, struc2vec ( Ribeiro et al., 2017 ) is proposed for better modeling the structural identity (e.g. nodes in the network may perform similar functions). Particularly, struc2vec first constructs a multi-layer weighted graph that encodes the structural similarity between nodes where each layer k is defined by using the k-hop neighborhoods of the nodes. DeepWalk is then performed on the multilayer graph to learn node representations in which nodes with high structural similarity are close to each other in the embedding space.

2.3 Neural network-based methods
Recent years have witnessed the success of neural network models in many fields. Various neural networks also have been introduced into graph embedding areas, such as multilayer perceptron (MLP) ( Tang et al., 2015 ), autoencoder ( Cao et al., 2016; Kipf and Welling, 2016; Wang et al., 2016 ), generative adversarial network (GAN) ( Wang et al., 2018 ) and graph convolutional network (GCN) ( Kipf and Welling, 2016, 2017 ). Different methods adopt different neural architectures and use different kinds of graph information as input. For example, LINE ( Tang et al., 2015 ) directly models node embedding vectors by approximating the first-order proximity and second-order proximity of nodes, which can be seen as a single-layer MLP model. DNGR ( Cao et al., 2016 ) applies the stacked denoising autoencoders on the positive pointwise mutual information (PPMI) matrix to learn deep low-dimensional node embeddings. SDNE ( Wang et al., 2016 ) adopts a deep autoencoder to preserve the second-order proximity by reconstructing the neighborhood structure of each node; meanwhile, it also incorporates LEs proximity measure into the learning framework to exploit the first-order proximity. GAE ( Kipf and Welling, 2016 ) utilizes a GCN encoder and an inner product decoder to learn node embeddings.

GraphGAN ( Wang et al., 2018 ) adopts GANs to model the connectivity of nodes. The GAN framework includes a generator and a discriminator where the generator approximates the true connectivity distribution over all other nodes and generates fake samples, while the discriminator model detects whether the sampled nodes are from ground truth or generated by the generator.

3 Applications of graph embedding on biomedical networks
In this section, we select 11 representative graph embedding methods (5 MF-based, 3 random walk-based, 3 neural network-based), and review how they are used on 3 popular biomedical link prediction applications: DDA prediction, DDI prediction, PPI prediction; and 2 biomedical node classification applications: protein function prediction and medical term semantic type classification.

3.1 Link prediction
Discovering new interactions (links) is one of the most important tasks in the biomedical area. A considerable amount of efforts has been devoted to developing computational methods to predict potential interactions in various biomedical networks, such as the DDA network (Liang et al., 2017), DDI network (Zhang et al., 2018b) and PPI network (Wang et al., 2014). Developing such computational methods can help generate hypotheses of potential associations or interactions in biological networks.

The link prediction task can be formulated as: given a set of biological entities and their known interactions, we aim to predict other potential interactions between entities (Liu and Zhou, 2011). Traditional methods in the biomedical field put much effort on feature engineering to develop biological features [e.g. chemical substructures (Liang et al., 2017), gene ontology (Gottlieb et al., 2011)] or graph properties [e.g. topological similarities (Hamilton et al., 2017)]. After that, supervised learning methods [e.g. support vector machine (SVM), Random Forest (Hamilton et al., 2017)] or semi-supervised graph inference model [e.g. label propagation (Cowan et al., 2017)] are utilized to predict potential interactions. The assumption behind these methods is that entities sharing similar biological features or graph features could have similar connections.

However, deploying methods based on biological features typically faces two problems: (i) biological features may not always be available and can be hard and costly to obtain. One popular approach to solve this problem is to remove those biological entities without features via pre-processing, which usually results in small-scale pruned datasets and thus is not pragmatic and useful in the real setting. (ii) Biological features, as well as hand-crafted graph features (e.g. node degrees), may not be precise enough to represent or characterize biomedical entities, and may fail to help build a robust and accurate model for many applications (Hamilton et al., 2017).

Graph embedding methods that seek to learn node representations automatically promise to solve the two problems mentioned above. Embedding ideas have also been employed in some recently proposed computational methods in the biomedical field. For example, MF-based techniques (Dai et al., 2015; Yang et al., 2014; Zhang et al., 2018c) are used for predictions of DDAs. Essentially, a DDA matrix is factorized to learn low-dimensional representations for drugs and diseases in the latent space. During factorization, regularization terms or constraints can be added to further improve the quality of latent representations. For predictions of DDIs, Zhang et al. (2018b) propose manifold regularized MF in which Laplacian regularization is incorporated to learn a better drug representation. Besides, graph neural network is introduced for DDIs prediction (Ma et al., 2018; Zitnik et al., 2018) and the intuitions are similar to the GAE (Kipf and Welling, 2016). PPIs are commonly predicted using Laplacian and SVD techniques (You et al., 2017; Zhu et al., 2013). More recently, Wang et al. (2017) propose an autoencoder-based model to learn embeddings of proteins, which has a similar design to SDNE (Wang et al., 2016).
In addition to the link prediction task, node classification which aims to predict the class of unlabeled nodes given a partially labeled graph, is also one of the most important applications in graph analyses. Here, we mainly focus on two node classification applications: protein function prediction and medical term semantic type classification.

Protein function prediction. The large-scale experimental functional annotation of proteins is often expensive (Gligorijević et al., 2018; Kulmanov et al., 2018), hence graph-based computational methods which widely incorporate the idea of graph embedding, have been proposed in recent years. For example, Lim et al. (2018) propose a regularized Laplacian kernel-based method to learn low-dimensional embeddings of proteins. Cho et al. (2016) develop Mashup, which first performs random walks with restart (RWR) on PPI networks and then learns embeddings for each protein via a low rank matrix approximation method (can be optimized by SVD). The feature vectors are then fed into classifiers to derive functional insights about genes or proteins. Kulmanov et al. (2018) propose DeepGO which learns joint representations of proteins based on protein sequences as well as PPI network via convolutional neural nets and a graph embedding method (Alshahrami et al., 2017) similar to DeepWalk (Perozzi et al., 2014). In node2vec, Grover and Leskovec (2016) test the effectiveness of the proposed embedding method on a PPI network. Furthermore, Zitnik and Leskovec (2017) develop OhmNet, which optimizes hierarchical dependency objectives based on node2vec to learn feature representations in multilayer tissue networks for function prediction. Gligorijević et al. (2018) develop deepNF, which learns embeddings of proteins via a deep autoencoder [similar to SDNE (Wang et al., 2016)].

Medical term semantic type classification. In the past few years, the increase of clinical texts have been encouraging data-driven models for improving the patient personal care and help clinical decision (Mullenbach et al., 2018). However, due to the privacy and security concerns, the access to raw clinical texts is often limited (Beam et al., 2018; Finlayson et al., 2014; Ta et al., 2018). To facilitate research on clinical texts, a popular substitute strategy for releasing raw clinical texts is to extract medical terms and their aggregated co-occurrence counts from the clinical texts (Finlayson et al., 2014; Ta et al., 2018). However, such released privacy-aware datasets only contain medical terms (words or phrases) extracted from clinical texts and do not reveal the semantic information (e.g. semantic types or categories). By referring to some medical knowledge bases, e.g. unified medical language system (UMLS) (Bodenreider, 2004), we can obtain semantic types (labels) medical terms. But due to mismatch and incomplete knowledge in UMLS, the semantic types of some medical terms remain unknown. Hence, we formulate a less-investigated but meaningful node classification task (Fig. 2): given a medical term co-occurrence graph where terms and co-occurrence statistics have been extracted from clinical texts, classify the semantic types of medical terms. In this work, we assume the clinical texts have been converted into a medical term–term co-occurrence graph as in Finlayson et al. (2014), where each node is an extracted medical term and each edge is the co-occurrence count of two terms in a context window. We apply graph embedding methods to the co-occurrence graph to learn representations of medical terms. Afterward, a multi-label classifier can be trained based on the learned embeddings to classify the semantic types of medical terms.

3.3 Summary
In order to show the current research status of evaluated graph embedding methods on the above biomedical applications, we summarize 11 graph embedding techniques by 3 categories and the existing works which have applied these techniques on certain tasks in Table 1. As can be seen, existing methods for the five representative biomedical applications primarily adopt the traditional techniques, e.g. LEs, MF. On the other hand, more recent advanced graph embedding methods have been demonstrated to outperform traditional techniques in social/information networks (Cao et al., 2015; Grover and Leskovec, 2016; Tang et al., 2015), but their performance in biomedical networks is not unknown. In addition, the comparison between these general graph embedding methods and state-of-the-arts in the individual prediction tasks should be explored to encourage future research. Hence, we conduct comprehensive experiments to evaluate those 11 graph embedding methods selected from 3 different categories on 5 representative biomedical tasks and compare them against the state-of-the-arts in each biomedical prediction task.

We follow the pipeline (shown in Fig. 1) of the widely adopted link prediction and node classification methods in general domains (Grover and Leskovec, 2016; Tang et al., 2015): graph embeddings are first learned and then used as feature inputs to build a binary classifier or multi-label classifier (e.g. Logistic Regression, SVM, MLP) to predict the unobserved links or the node labels.

4 Experiments
In this section, we introduce the details of seven compiled datasets, including two DDA graphs, a DDI graph, a PPI graph for link prediction and a medical term–term co-occurrence graph as well as two PPI graphs for node classification. Then, we conduct comprehensive comparisons of 11 selected graph embedding methods on these compiled datasets.

4.1 Datasets
We use the following datasets for Link Prediction:

1. DDA graph. We extract chemical-disease associations from the Comparative Toxicogenomics Database (CTD) (Davis et al., 2018). CTD offers two kinds of associations: curated (verified) and inferred. Since our task is to infer potential chemical-disease associations, we only use curated ones as our golden instances. Finally, we obtain 92 813 edges between 12 765 nodes (9580 chemicals and 3185 diseases) in this graph (named as ‘CTD DDA’).

   Also, we construct another DDA network from National Drug File Reference Terminology (NDF-RT) in UMLS (Bodenreider, 2004). NDF-RT is produced by the US Department of Veterans Affairs, and models drug characteristics including ingredients, physiologic effect and related diseases. We extract drug-disease treatment associations using the may treat and may be treated by relationships in NDF-RT. This graph (named ‘NDFRT DDA’) contains 13 545 nodes (12 337 drugs and 1208 diseases) and 56 515 edges.

2. DDI graph. We collect verified DDIs from DrugBank (Wishart et al., 2018 ), a comprehensive and freely accessible online database that contains detailed information about drugs and drug
targets. We obtain 242 027 DDIs between 2191 drugs and refer to this dataset as ‘DrugBank DDI’.  

3. PPI graph. We extract *Homo sapiens* PPIs from STRING database (Szklarczyk et al., 2015). Each PPI is associated with a confidence score that indicates its possibility to be a true positive interaction. To reduce noise, we only collect PPI whose confidence score is larger than 0.7 according to the guidelines of STRING database. Finally, we obtain 359 776 interactions among 15 131 proteins and name this dataset as ‘STRING PPI’. We use the following datasets for Node Classification:  

1. Medical term–term co-occurrence graph. We adopt a publicly available set of medical terms with their co-occurrence statistics which are extracted by Finlayson et al. (2014) from 20 million clinical notes collected from Stanford Hospitals and Clinics (Lowe et al., 2009) since 1995. Medical terms are extracted from raw clinical notes using an existing phrase mining tool (LePendu et al., 2012) by matching with 22 clinically relevant ontologies such as SNOMED-CT and MedDRA. Co-occurrence frequencies between two terms are counted based on how many times they co-occur in the same temporal frame; see Finlayson et al., 2014 for more details. We select perBin 1-day dataset since it contains more medical terms compared with other bins. To filter very common medical terms (e.g. ‘medical history’, ‘medication dose’) that may influence the quality of embeddings, we convert the co-occurrence counts to the PPMI value (Levy and Goldberg, 2014) and remove the edges whose PPMI value is <2. We also adopt a subsampling (Mikolov et al., 2013) strategy to further filter common terms and construct a medical term–term co-occurrence graph that contains 48 651 medical terms and 1 659 249 edges. We keep the medical terms that can be mapped to the UMLS Concept Unique Identifiers (CUI) and collect their corresponding semantic types (e.g. clinical drug, disease or syndrome) from UMLS. We select 31 different semantic types, with each having more than 20 samples. Finally, we obtain 25 120 nodes with label information. This dataset is called ‘Clin Term COOC’.  

2. PPI graphs with functional annotations. We also compile two PPI graphs with functional annotations from previous studies. One is from node2vec (Grover and Leskovec, 2016), which contains 3890 proteins, 76 584 interactions and 50 different function annotations (labels). This dataset is named as ‘node2vec PPI’. The other one is from Mashup (Cho et al., 2016), which is designed for integrating different information from multiple networks. The Mashup dataset contains six individual PPI networks (e.g. experimental, coexpression). Given that our selected graph embedding methods can only work on a single network, we select the experimental PPI network (there are six individual PPI networks in Mashup dataset, we select the experimental PPI network since Mashup achieves the best performance on it under single-network circumstance) to learn embeddings. The experimental PPI network contains 300 181 interactions among 16 143 proteins. Same to Mashup, we use the 3 grouped distinct levels of functional categories of varying specificity, each containing 28 100 and 262 different annotations, respectively. We only adopt the first level (28 labels) for the main comparison experiment for simplicity. Other label information is used in comparing the recent embedding methods with Mashup in Section 4.4. This dataset is called ‘Mashup PPI’.

The details of all datasets are summarized in Table 2.

### Table 1. A summary of 11 representative graph embedding methods and existing work (if any) using them for a certain task

<table>
<thead>
<tr>
<th>Method category</th>
<th>Method name</th>
<th>Link prediction tasks</th>
<th>Drug–drug interaction prediction</th>
<th>Drug–disease association prediction</th>
<th>Protein–protein interaction prediction</th>
<th>Node classification tasks</th>
<th>Medical term type classification</th>
<th>Protein function classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Matrix factorization-based</td>
<td>Laplacian (Belkin and Niyogi, 2003)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GF</td>
<td>(Ahmed et al., 2013)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recently Proposed</td>
<td>HOPE (Ou et al., 2016)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GraRep</td>
<td>(Cao et al., 2015)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DeepWalk</td>
<td>(Perozzi et al., 2014)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>node2vec</td>
<td>(Grover and Leskovec, 2016)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>struc2vec</td>
<td>(Ribeiro et al., 2017)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural network-based</td>
<td>LINE</td>
<td>(Tang et al., 2015)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SDNE</td>
<td>(Wang et al., 2016)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GAE</td>
<td>(Kipf and Welling, 2016)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: \(\times\) means that a method (row) has not been applied for a task (column).

We use OpenNE (https://github.com/thunlp/OpenNE), an open-source Python package for network embedding, to learn node embeddings for LEs (Belkin and Niyogi, 2003), HOPE (Ou et al., 2016), GF (Ahmed et al., 2013), DeepWalk (Perozzi et al., 2014), LINE (Tang et al., 2015) and SDNE (Wang et al., 2016). We run SVD using Numpy (http://www.numpy.org/) and obtain struc2vec (https://github.com/leoribeiro/struc2vec) (Ribeiro et al., 2017) and GAE (https://github.com/tkipf/gae) (Kipf and Welling, 2016)
embeddings using the source code provided by their authors. More implementation details can be found in Supplementary Material.

We conduct the link prediction task on the 4 compiled biomedical datasets: CTD DDA, NDFRT DDA, DrugBank DDI and STRING PPI. Table 3 shows the overall performance of different embedding methods on the four datasets.

Generally, compared with traditional techniques (e.g. LRs, SVD, GF), the recently proposed embedding methods have largely improved the link prediction performance. For example, LINE achieves 3–23% improvement in terms of AUC value on the four datasets compared with LRs. Struc2vec obtains 3–15% increment in the accuracy on the four datasets, respectively, when compared with GF. These results demonstrate that the recently proposed graph embedding methods are more effective and could be used on various biological link prediction tasks to improve the prediction performance.

4.3 Link prediction results

We conduct the link prediction task on the 4 compiled biomedical networks: CTD DDA, NDFRT DDA, DrugBank DDI and STRING PPI. Table 3 shows the overall performance of different embedding methods on the four datasets.

Generally, compared with traditional techniques (e.g. LRs, SVD, GF), the recently proposed embedding methods have largely improved the link prediction performance. For example, LINE achieves 3–23% improvement in terms of AUC value on the four datasets compared with LRs. Struc2vec obtains 3–15% increment in the accuracy on the four datasets, respectively, when compared with GF. These results demonstrate that the recently proposed graph embedding methods are more effective and could be used on various biological link prediction tasks to improve the prediction performance.

Furthermore, we have the following key observations and analyses:

- **For the MF-based methods**, since HOPE and GraRep are designed to capture the high-order proximity of graphs, they are usually more effective than traditional MF methods that only preserve the first-order of networks.
- **For the random walk-based methods**, generally, struc2vec performs better than DeepWalk and node2vec. This is because compared with DeepWalk and node2vec, struc2vec constructs a hierarchy weighted graph to measure the structural identity. Such hierarchy structure design incorporates both node degree distributions from the bottom as well as the entire network on the top, which can better capture the graph structure information and hence obtain better performance.
- **For the neural network-based methods**, LINE achieves competitive prediction performance consistently when compared with the best performing method on each dataset. It indicates that directly modeling edge information by a single-layer MLP is an effective way to learn node embeddings. SDNE and GAE also obtain satisfying prediction performance, which demonstrates that autoencoders and GCNs can also be useful for capturing graph structural information.

Comparison with state-of-the-art studies. To further demonstrate the effectiveness of graph embedding methods, we compare them with the state-of-the-art methods for two link prediction: DDA prediction and DDI prediction.

For the DDA prediction, we select LRSSL (Liang et al., 2017) as our baseline. LRSSL is a Laplacian regularized sparse subspace learning framework which aims to project different drug features into a common subspace. Three drug feature profiles (i.e. chemical substructure, target domain and target annotation) are used in the training process. To be fair, we adopt the code and dataset used in the LRSSL. To learn graph embeddings without modeling biological features, we run four representative graph embedding methods: GraRep, DeepWalk, LINE and struc2vec on LRSSL’s DDA graph. Following the same train/test split, training and evaluation process of link prediction in Section 4.2, we plot the ROC Curves to illustrate the performance of different methods. As seen in Figure 3a, graph embedding methods achieve competitive performance compared with LRSSL. Furthermore, we use the learned DeepWalk embedding vectors as the fourth feature for the LRSSL method and improve the LRSSL performance, which indicates that the learned node embedding can be used as a complementary representation for biological features.

For the DDI prediction, we compare the embedding methods with a recent method DeepDDI (Ryu et al., 2018). DeepDDI first adopts principal component analysis to reduce the dimension of the drug features (i.e. drug substructure) and then feeds these into a deep neural network (DNN) classifier. For a fair comparison with graph embedding methods and to reduce the bias caused by different classifiers, we compare these methods under four classifiers, Naive Bayes, Linear SVM, Logistic Regression and eight-layer DNN (the same as the original paper). More implement details can be found in Supplementary Material. As seen in Figure 3b, graph embedding methods outperform the drug features-based model or obtain very competitive performance under each classifier, which demonstrates the power of graph embedding methods.

4.4 Node classification results

Table 4 shows the performance of different embedding methods on medical term semantic type classification and protein function prediction. We make the following key observations:

- **For the MF-based methods**, it is a little surprising that the traditional method SVD achieves better performance, even surpassing HOPE and GraRep. This may indicate that directly modeling...
As can be seen, in each task, general graph embedding methods achieve competitive performance against them.

- For the random walk-based methods, node2vec performs better since it aims to capture different functions of nodes (i.e. homophily and structural equivalence) via a more flexible biased random walk. struc2vec performs not good as DeepWalk and node2vec as it mainly focuses on modeling the structural identity of nodes; however, a clear structural role may not exist in these biomedical graphs and struc2vec is not suitable on such graphs.

- For the neural network-based methods, LINE achieves better performance than SDNE, which demonstrates that directly modeling edge information is an effective way to learn the embedding for the node classification task. And GAE also achieve promising performance, which demonstrates the power of the graph neural networks.

Fig. 3. (a) Comparison with the state-of-the-arts for drug-disease association prediction (LRSSL) (Liang et al., 2017); (b) drug–drug interaction prediction (DeepDDI) (Ryu et al., 2018) and (c) gene (protein) function prediction (Mashup) (Cho et al., 2016). Same as Mashup, we evaluate their performance on three-level human Biological Process (BP) gene annotations (each containing GO terms with 101–300, 31–100 and 11–30 genes, respectively). As can be seen, in each task, general graph embedding methods achieve competitive performance against them.

<table>
<thead>
<tr>
<th>Method category</th>
<th>Method name</th>
<th>CTD DDA</th>
<th>DDA NDFRT DDA</th>
<th>DrugBank DDI</th>
<th>STRING PPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Matrix factorization-based</td>
<td>Laplacian (Belkin and Niyogi, 2003)</td>
<td>0.856 ± 0.004</td>
<td>0.930 ± 0.003</td>
<td>0.796 ± 0.002</td>
<td>0.639 ± 0.021</td>
</tr>
<tr>
<td></td>
<td>SVD</td>
<td>0.936 ± 0.002</td>
<td>0.779 ± 0.003</td>
<td>0.919 ± 0.001</td>
<td>0.867 ± 0.001</td>
</tr>
<tr>
<td>Recently proposed</td>
<td>GF (Ahmed et al., 2013)</td>
<td>0.884 ± 0.004</td>
<td>0.720 ± 0.006</td>
<td>0.882 ± 0.003</td>
<td>0.817 ± 0.005</td>
</tr>
<tr>
<td></td>
<td>HOPE (Ou et al., 2016)</td>
<td>0.951 ± 0.001</td>
<td>0.949 ± 0.001</td>
<td>0.923 ± 0.001</td>
<td>0.839 ± 0.001</td>
</tr>
<tr>
<td></td>
<td>GraRep (Cao et al., 2015)</td>
<td>0.960 ± 0.001</td>
<td>0.963 ± 0.001</td>
<td>0.925 ± 0.001</td>
<td>0.894 ± 0.001</td>
</tr>
<tr>
<td>Random walk-based</td>
<td>DeepWalk (Perozzi et al., 2014)</td>
<td>0.929 ± 0.002</td>
<td>0.783 ± 0.004</td>
<td>0.921 ± 0.001</td>
<td>0.884 ± 0.001</td>
</tr>
<tr>
<td></td>
<td>node2vec (Grover and Leskovec, 2016)</td>
<td>0.911 ± 0.002</td>
<td>0.819 ± 0.005</td>
<td>0.902 ± 0.001</td>
<td>0.828 ± 0.003</td>
</tr>
<tr>
<td></td>
<td>struc2vec (Ribeiro et al., 2017)</td>
<td>0.965 ± 0.001</td>
<td>0.958 ± 0.001</td>
<td>0.904 ± 0.001</td>
<td>0.909 ± 0.001</td>
</tr>
<tr>
<td>Neural network-based</td>
<td>LINE (Tang et al., 2015)</td>
<td>0.965 ± 0.001</td>
<td>0.962 ± 0.002</td>
<td>0.905 ± 0.002</td>
<td>0.859 ± 0.003</td>
</tr>
<tr>
<td></td>
<td>SDNE (Wang et al., 2016)</td>
<td>0.935 ± 0.010</td>
<td>0.944 ± 0.004</td>
<td>0.911 ± 0.006</td>
<td>0.884 ± 0.008</td>
</tr>
<tr>
<td></td>
<td>GAE (Kipf and Welling, 2016)</td>
<td>0.937 ± 0.001</td>
<td>0.813 ± 0.007</td>
<td>0.917 ± 0.001</td>
<td>0.900 ± 0.001</td>
</tr>
</tbody>
</table>

Note: Due to the limited space, we only show the AUC value. Other evaluation metrics can be found in Supplementary Material. The best performing method in each category is in bold.

4.5 Influence of hyper-parameters

Hyper-parameters play essential roles in machine learning models. However, selecting proper hyper-parameters is often time-consuming. We investigate the influence of some important hyper-parameters in various embedding methods. By running grid-search of these important hyper-parameters of each method, we expect to summarize some general guidelines for helping researchers better set the hyper-parameters, so as to save their time and efforts.

We first evaluate how different embedding dimensions can affect the prediction performance and time efficiency. Figure 4 shows the impact of embedding dimensionality on the prediction performance and time efficiency for ‘CTD DDA’ dataset. Generally, the prediction performance becomes better when the embedding dimensionality increases, which is intuitive since higher dimensionality can...

the first-order proximity would be good enough to classify the nodes.

- For the random walk-based methods, node2vec performs better since it aims to capture different functions of nodes (i.e. homophily and structural equivalence) via a more flexible biased random walk. struc2vec performs not good as DeepWalk and node2vec as it mainly focuses on modeling the structural identity of nodes; however, a clear structural role may not exist in these biomedical graphs and struc2vec is not suitable on such graphs.

- For the neural network-based methods, LINE achieves better performance than SDNE, which demonstrates that directly modeling edge information is an effective way to learn the embedding for the node classification task. And GAE also achieve promising performance, which demonstrates the power of the graph neural networks.
Random walk-based DeepWalk (Perozzi authors. Table 5 shows the selected hyper-parameters in different embedding methods, which have been pointed out to be important by their authors. In particular, for MF-based methods, we observe that modeling neighbors have similar functions and tend to interact with each other. By simply applying the recent graph embedding methods on biomedical graphs, we can achieve very competitive or better performance compared with state-of-the-arts. Future model design for biomedical prediction tasks but less satisfying performance on node classification. GAE performs well in relatively large-scale network but may not perform well on small-scale datasets. More details of the datasets, implementation, experiment results, and assumption are similar. They both assume that nodes with similar functions and tend to interact with each other. Hence, the performance tends to saturate when the dimension reaches to a threshold (e.g. 100). As for the time cost, it first increases gradually below 100 but tends to boost sharply (the y-axis is log-based) if the dimensionality continues to increase. So we would not suggest to set the dimensionality to be too large (e.g. around 100 is a good option) for the practitioners when considering both performance and time efficiency. The results of dimensionality’s influence on other datasets can be found in Supplementary Figures S1 and S2.

Furthermore, we choose sensitive hyper-parameters for 7 embedding methods, which have been pointed out to be important by their authors. Table 5 shows the selected hyper-parameters in different embedding methods as well as their meanings. We spend a lot of efforts on carefully tuning these hyper-parameters by grid search. The influence of the hyper-parameters on each embedding method is shown in Supplementary Figures S3–S9, respectively. By summarizing these results, we provide some high-level guidelines on setting hyper-parameters for practitioners in Table 5.

### 4.6 Summary of experimental results

To better help the practitioners select proper embedding methods for their biomedical prediction task, we summarize the experimental results and discuss our observations:

- Generally, the recently proposed graph embedding methods achieve very promising results in various biomedical prediction tasks. They deserve more attention for future biomedical graph analysis.
- By simply applying the recent graph embedding methods on biomedical graphs and then feeding into a classifier, we can achieve very competitive or better performance compared with state-of-the-arts. Future model design for biomedical prediction tasks may begin at these embedding methods or integrate them as one module into the proposed method, which is expected to gain better results.
- In particular, for MF-based methods, we observe that modeling high-order proximity (e.g. HOPE, GraRep) is generally useful for link prediction tasks on medical graphs but may be less meaningful for the node classification tasks. For random walk-based methods, struc2vec is more suitable for link prediction tasks (when there is a lack of structural identity in graphs) while node2vec and DeepWalk are more suitable for node classification tasks. For neural network-based methods, LINE usually achieves competitive performance against the best performing method on each dataset. SDNE can achieve good performance on link prediction tasks but less satisfying performance on node classification.

### 5 Discussions and future directions

**Connections of network embedding and network propagation.** In the recent biomedical network analyses, a very popular paradigm is network propagation (Cowen et al., 2017), which amplifies a biological signal (e.g. label, association) based on the assumption that nodes with similar neighbors (e.g. genes underlying similar phenotypes) tend to interact with one another (Mench et al., 2015). Specifically, the information of one node is propagated through the edges to their neighbors in an iterative manner for a fixed number of steps or until convergence (Cowen et al., 2017). The core of these propagation methods is random walk, which is also adopted in many embedding methods (e.g. Deepwalk, node2vec and struc2vec). But different from network diffusion, which propagates the ‘signal’ in the network directly, the random walk-based embedding methods treat the ‘walk’ as a kind of node similarity or proximity characterizing method. They expect to preserve the network structural information as much as possible through a fixed number of random walks. These ‘walking histories’ (i.e. node sequences) are then fed into word2vec (Mikolov et al., 2013) to learn low-dimensional embeddings. Though the pipeline of the random walk-based embedding methods and network propagation methods is different, their idea and assumption are similar. They both assume that nodes with similar neighbors have similar functions and tend to interact with each other...
Transfer learning for graph embedding. Recent studies in Computer Vision and NLP show that transfer learning helps improve model performance on different tasks (Howard and Ruder, 2018; Shin et al., 2016). General patterns are captured during pre-trained processes and can be 'transferred' into new prediction tasks. There also exist some pre-trained embeddings of biomedical entities (Beam et al., 2018; Choi et al., 2016) which allow us to adopt similar ideas of ‘transfer learning’ to learn graph embeddings. We can initialize the embedding vector for each node on a graph with its pre-trained embedding (e.g. by looking for the corresponding entity in Choi et al., 2016; Beam et al., 2018) rather than by random initialization, and then continue training various graph embedding methods as before (which is often referred to as ‘fine-tuning’). The pre-trained embeddings can be seen as ‘coarse embeddings’ since they are usually pre-trained on a large general corpus and have not been optimized for downstream tasks yet. Nevertheless, they contain some additional semantic information that may not be able to be learned from a downstream task graph (e.g. due to its small scale). By fine-tuning, such additional semantic information can be ‘transferred’ into the finally learned embeddings. We conduct experiment with this transfer learning idea on the ‘CTD DDA’ graph. As seen in Supplementary Table S3, the link prediction performance has been improved using the pre-trained embeddings from Beam et al. (2018). Currently, the number of released biomedical entities with pre-trained embeddings is still limited and entities without pre-trained embeddings have to be initialized randomly. However, with the increasing volume of biomedical data, more and more entities can have pre-trained embeddings, and the idea of pre-training—then—fine-tuning can be more promising.

6 Conclusion

This paper provides an overview of various graph embedding techniques and evaluates their performance on two important biomedical tasks, link prediction and node classification. Specifically, we compile 7 datasets from public database or previous studies and use them to benchmark 11 representative graph embedding methods. Through extensive experiments, we find that generally the recent graph embedding methods can perform well in various biomedical prediction tasks and can also achieve very competitive or better performance compared with state-of-the-arts. Hence, these recent graph embedding methods can be considered as a starting point when designing advanced models for future biomedical prediction tasks. Additionally, we tune some important hyper-parameters of graph embedding methods and provide general guidelines for setting hyper-parameters for practitioners. We also discuss the connections between the recent network propagation (diffusion) methods and the graph embedding methods as well as potential directions (e.g. transfer learning for graph embedding) to inspire the future work.

Acknowledgements

We thank Dr Deborah A. Petrone, Kaushik Mani and anonymous reviewers for their helpful comments and suggestions on our work, and Ohio Supercomputer Center (OSC) (Ohio Supercomputer Center, 1987) for providing us computing resources.

Funding

This work has been supported by Patient-Centered Outcomes Research Institute (PCORI) under grant ME-2017C1-6413.

Conflict of Interest: none declared.