Predicting Missing and Spurious Protein-Protein Interactions Using Graph Embeddings on GO Annotation Graph

Xiaoshi Zhong and Jagath C. Rajapakse
School of Computer Science and Engineering
Nanyang Technological University, Singapore
{xszhong, asjagath}@ntu.edu.sg

***Abstract*—Protein-protein interaction (PPI) prediction is a key step towards many bioinformatics applications including prediction of protein functions and drug-disease interactions. However, previous research on PPI prediction rarely considered missing and spurious interactions in PPI networks. To address these two issues, we define two corresponding tasks, namely missing PPI prediction and spurious PPI prediction, and propose a novel method that employs graph embeddings to learn vector representations from constructed Gene Ontology (GO) annotation graphs. Our method leverages the information from both term-term relations among GO terms and term-protein annotations between GO terms and proteins, and preserves properties of both local and global structural information of the GO annotation graph. We compare our method with methods based on information content and on word embeddings, using three PPI datasets from STRING database. Experimental results demonstrate that our method is more effective than those compared methods.**

***Index Terms*—GO annotation graph, graph embeddings, missing protein interactions, spurious protein interactions, protein-protein interactions**

I. Introduction

Protein-protein interactions (PPI) play an important role in understanding functional properties and biomarker potentials of proteins. Predicting interactions between proteins becomes a crucial step in many bioinformatics applications such as identifying drug-target interactions [1], [2], construction of PPI networks (PPIN) [3]–[5], and detection of functional modules [6], [7]. The task aiming at predicting interactions between proteins is often termed as PPI prediction [8], [9].

PPI prediction is well investigated problem in bioinformatics: for example, Struct2Net was used to integrate the structural information for PPI prediction [10], [11], PSOPIA leveraged on sequence information for PPI prediction [12], and several other research [9], [13]–[18]. However, these methods implicitly assume that known interactions between proteins are perfect and focus mainly on predicting existing PPI. However, existing PPIN are incomplete and contain missing and spurious PPI, and few existing PPI prediction methods consider missing and spurious (i.e., erroneous) interactions.

To address these two issues, we defined two specific tasks of PPI prediction: (i) missing PPI prediction and (ii) spurious PPI prediction. For the missing PPI prediction, we treat a real PPI dataset as the ground-truth PPI dataset, remove PPIs randomly, and attempt to predict them as missing PPI. The goal of missing PPI prediction is to see whether we could correctly predict the missing PPI. For the spurious PPI prediction, we add some PPIs to the ground-truth PPI dataset, treat them as spurious PPIs, and try to predict them. The goal of spurious PPI prediction is to see the extent of correctly predicting the spurious PPIs. See Section II-A for details about the definitions of these two tasks.

Among existing models that are developed for the PPI prediction, the majority leverages on the information from the structure of Gene Ontology (GO) that provides a set of structured and controlled vocabularies (or terms) describing gene products and molecular properties [19]. Proteins are generally annotated by a set of GO terms [20], [21]. For example, the protein 'P06182' is annotated by GO terms: 'GO:0004408', 'GO:0005743', 'GO:0005758', 'GO:0018063', and 'GO:0046872'. Based GO term-protein annotations, many research have employed information content (IC) of GO terms [22]–[25] to compute similarity between two proteins in order to predict PPI. These methods have succeeded in the development of protein-related tasks, including PPI prediction [26]–[41].

Recently, several researchers have employed word embeddings (e.g., word2vec [42] and GloVe [43]), which have been developed in the area of natural language processing, to learn vector representations of GO terms and proteins and then used learned vectors for the PPI prediction [44]–[46]. These methods mainly use the word2vec model [42] to learn vectors for each word from the corpus derived from descriptive axioms of GO terms and proteins; the descriptive axiom of a GO term is its textual description, for example, the descriptive axiom of the GO term "GO:0036388" is "pre-replicative complex assembly." Then, the learned word vectors are combined into vectors of GO terms and proteins, according to the words in the descriptive axioms of GO terms and proteins. Finally, the vectors of proteins are used to predict the protein interactions.

In this paper, we proposed a novel method extending GO2Vec [47], which employs graph embeddings to transform GO annotation (GOA) graphs into their vector representations in order to predict missing and spurious PPI. Specifically,

This research is supported by the Tier-2 grant MOE2016-T2-1-029 from the Ministry of Education, Singapore.

our method firstly combines term-term relations between GO terms and term-protein annotations between GO terms and proteins and constructs an undirected and unweighted graph; this constructed graph is called a GOA graph. Thereafter, node2vec model [48], one of graph embedding models, is applied on the GOA graphs to transform the nodes (including GO terms and proteins) into their vector representations. Finally, learned vectors of GO terms and proteins with the cosine distance and the modified Hausdorff distance [49] measures are used to predict the missing and spurious PPI.

Our method can capture the structural information connecting the nodes in the entire GOA graph. On the one hand, when compared with the structure-based IC methods that mainly consider the nearest common ancestors of two nodes, graph embeddings take into account the information from every path between two nodes. Graph embeddings therefore can fully portray the relationship of two nodes in the entire graph. On the other hand, when compared with the corpus-based methods, including the traditional IC based methods and word embedding based methods, graph embeddings can employ the expert knowledge (e.g., term-term relations and term-protein annotations) stored in the graphical structure. In our experiments, we used the node2vec model [48] as the representative of graph embedding techniques. The node2vec model adopts a strategy of random walk over an undirected graph to sample neighborhood nodes for a given node and preserves both neighborhood properties and structural features.

To evaluate the quality of our proposed methods in addressing the issues of missing and spurious PPIs, we conducted experiments on three kinds of PPI datasets (i.e., HUMAN, MOUSE, and YEAST) from the STRING database [50], using three categories of GO, i.e., Biological Process (BP), Cellular Component (CC), and Molecular Function (MF), with the GO annotations collected from the UniProt database [51]. We compared our methods with the representative information content-based methods including Resnik [24], Lin [23], Jang&Conrath [22], simGIC [25], and simUI [52], and a recent corpus-based vector representation method Onto2Vec [44]. Experimental results demonstrate the effectiveness of our methods over these compared methods in both missing and spurious PPI predictions. This demonstrates the usefulness of combining term-term relations between GO terms and term-protein annotations between GO terms and proteins, and of employing techniques of graph embeddings on GOA.

To summarise, this paper makes the following contributions.

- We address the issues of missing and spurious PPI in PPIN, which have been neglected in earlier research.
- We apply graph embeddings on a constructed GOA graph to learn vector representations for predicting missing and spurious PPI.
- We conduct experiments on three PPI datasets for missing and spurious PPI predictions, and demonstrate the effectiveness of using graph embeddings on GOA graph over representative IC-based methods and corpus-based word vector method.

II. Methods

This section details our method for the missing and spurious PPI predictions. We firstly define the two tasks, then provide an overview of the framework of our proposed method, and finally illustrate technical details of its components.

A. Task Definitions

In this paper, we consider two kinds of PPI prediction tasks, namely missing PPI prediction and spurious PPI prediction. Figure 1 illustrates the constructions of missing PPIs and spurious PPIs. Graph (a) is given by a real-world PPI dataset and is treated as the ground-truth PPI graph. Graph (b) is derived from Graph (a) by removing some PPIs and these removed PPIs are treated as missing PPIs. Graph (c) is also derived from Graph (a), but instead of removing PPIs, some PPIs are added to Graph (a) and these added PPIs are treated as spurious PPIs.

1) Missing PPI Prediction: Given a ground-truth PPI graph with some PPI removed (e.g., Graph (b)), the goal of missing PPI prediction is to predict whether these removed PPIs are missing PPI.

2) Spurious PPI Prediction: Given a ground-truth PPI graph with some PPIs added (e.g., Graph (c)), the goal of spurious PPI prediction is to predict whether these added PPIs are spurious PPI.

Figure 2 shows the framework of our method for missing and spurious PPI predictions, which mainly consists of three components: (1) GOA graph construction, (2) transformation of GOA graph to vector representations, and (3) prediction of missing and spurious PPI.

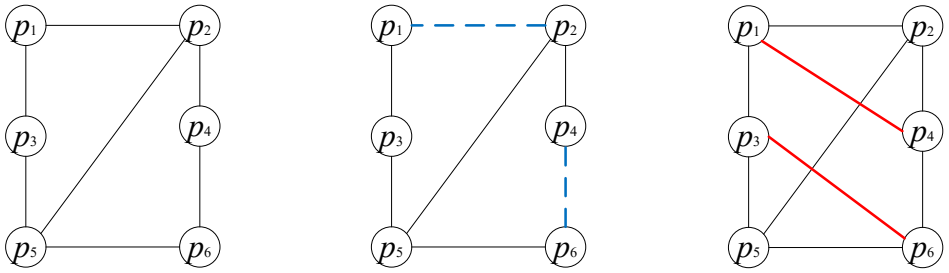
B. GOA Graph Construction

A GOA graph (or GO annotation graph) is an undirected and unweighted (or binary) graph, constructed from the GO and GOA. Specifically, we combine term-term relations between GO terms and term-protein annotations between GO terms and proteins together to form an undirected and unweighted graph where the nodes include both the GO terms and proteins, and the edges include both term-term relations and term-protein annotations.

Although GO is a directed acyclic graph (DAG) and transforming directed edges to undirected edges might result in a loss of some information, we found that graph embeddings working on undirected graphs achieves better performance than on directed graphs (see Section III-C). That is probably because the node2vec model we used adopts a strategy of random walks to sample neighborhood nodes, and such strategy works better on undirected graphs than on directed graphs. Therefore, in this paper, we constructed the GOA graph as an undirected graph by simply setting a directed edge as an undirected edge.

C. GOA Graph to Vector Representations

There are several graph embedding models that can be used to transform a graph to the vector representations, such as DeepWalk [53], LINE [54], and node2vec [48]. In our



(a) Ground-truth (b) Removed interactions as missing PPIs (c) Added interactions as spurious PPIs

Fig. 1: Illustration of missing and spurious PPI constructions. Graph (a) is the ground-truth PPI graph derived from a real-world PPI dataset where nodes are the proteins and edges represent PPI. Graph (b) is a derived PPI graph, with two PPIs (indicated by the blue dashed edges) removed from Graph (a), and is used for missing PPI prediction where the blue dashed edges are missing PPIs. Graph (c) is a derived PPI graph, with two PPIs (indicated by the red bold edges) added to Graph (a) and is used for spurious PPI prediction where the red bold edges are spurious PPIs.

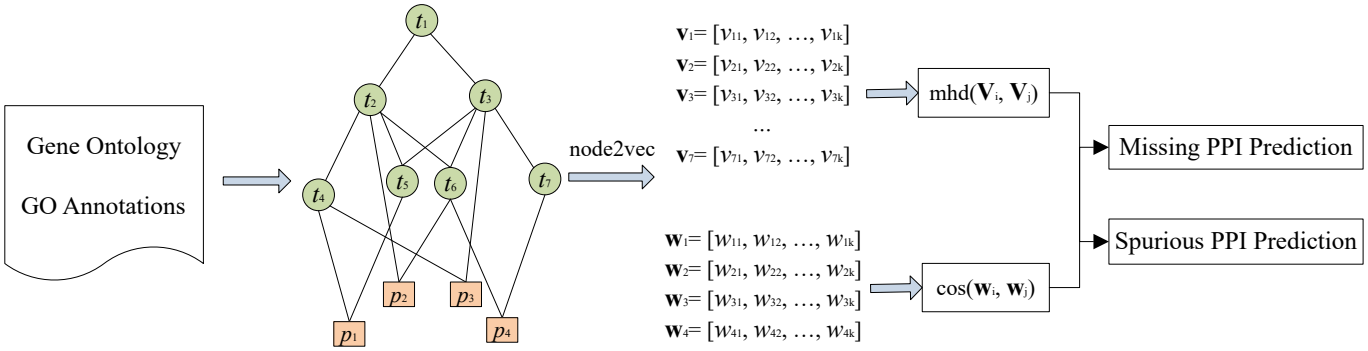


Fig. 2: Steps involved in our method of PPI prediction. Firstly, GO and GOA are combined together to construct a GOA graph, which is an undirected and unweighted graph. The node2vec model is then applied on the GOA graph to transform the nodes to their vector representations. Finally, the learned vectors are used for the tasks of missing and spurious PPI predictions. t_i denotes a GO term and $\mathbf{v}_i = (v_{ij})$ denotes its k-dimensional vector, p_m denotes a protein and $\mathbf{w}_m = (w_{mn})$ denotes its k-dimensional vector representing the protein. $\mathbf{V}_m$ denote a set of vectors of GO terms that annotate the protein.

experiments, we found that the node2vec model works better in our datasets than other models and therefore node2vec was used to convert GOA graph into vector representations. To make our paper self-contained, in what follows, we briefly introduce the node2vec model.

The node2vec Model: Let (T, E) represent a graph where T denotes the set of nodes and $E \subseteq (T \times T)$ denotes the set of edges. The goal is to learn a mapping function $f : T \rightarrow \mathbb{R}^k$ that transforms the nodes to vector representations in the space $\mathbb{R}^k$, where the parameter k specifies the dimensions of the vector representations. f can be represented by a matrix of parameters with the size $|T| \times k$. For each node $t \in T$, $N(t) \subset T$ denotes the set of neighbourhood nodes of node t, generated through a sampling strategy.

The node2vec model aims to optimize Equation (1), which maximizes the log-probability of observing a network neighborhood $N(t)$ for a node t conditioned on its vector representation, given by f.

$$\max_f \sum_{t \in T} \log P(N(t)|f(t)) \tag{1}$$

To make the optimization problem resolvable, the node2vec model makes two assumptions: conditional independence and symmetry in feature space.

Conditional independence: given the vector representation of the source node t, the likelihood of observing a neighborhood node t' is independent of observing any other neighborhood node. This is expressed by Equation (2).

$$P(N(t)|f(t)) = \prod_{t' \in N(t)} P(t'|f(t)) \tag{2}$$

Symmetry in feature space: the source node t and neighborhood node t' have a symmetric effect on each other in the feature space. This assumption is expressed by Equation (3).

$$P(t'|f(t)) = \frac{\exp(f(t') \cdot f(t))}{\sum_{t'' \in T} \exp(f(t'') \cdot f(t))} \tag{3}$$

With the above two assumptions, Equation (1) is simplified

to Equation (4):

$$\max_f \sum_{t\in T}\left(\sum_{t'\in N(t)} f(t')\cdot f(t) - \sum_{t''\in T}\exp(f(t'')\cdot f(t))\right) \tag{4}$$

Therefore, given a source node t, the node2vec model simulates a random walk of fixed length l. Let c_i denote the i-th node in the walk, starting with $c_0 = t$. Node c_i is generated by the following distribution:

$$P(c_i = x|c_{i-1} = t) = \begin{cases} \frac{\pi_{tx}}{Z} & \text{if}(t,x)\in E \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where π_{tx} is the transition probability between nodes t and x, and Z is the normalizing constant.

D. Missing and Spurious PPI Predictions

After applying the node2vec model on the GOA graph for transformation, we get the vector representations for the GO terms and proteins. Specifically, each of GO terms and proteins is represented by a k-dimensional vector. There are two ways to use these learned vectors for missing and spurious PPI predictions. One way is directly through the learned vectors of proteins, and the other way is through the learned vectors of GO terms.

1) From Learned Vectors of Proteins: Let $\mathbf{w}_m$ and $\mathbf{w}_n$ denote the learned vectors of protein p_m and p_n. The similarity between two proteins is defined by the cosine distance $cos(\mathbf{w}_m, \mathbf{w}_n)$ of their corresponding vectors $\mathbf{w}_m$ and $\mathbf{w}_n$, as shown by Equation (6).

$$sim(p_m, p_n) = cos(\mathbf{w}_m, \mathbf{w}_n) = \frac{\mathbf{w}_m \cdot \mathbf{w}_n}{||\mathbf{w}_m||||\mathbf{w}_n||} \tag{6}$$

2) From Learned Vectors of GO Terms: Since a protein is annotated by several GO terms under each category of GO graphs, we can view protein p as a set of GO terms that annotate p. Let T_m denote the set of GO terms that annotate protein p_m, and T_n denote the set of GO terms that annotate protein p_n. To compute the functional similarity between proteins p_m and p_n, we need only to compute the semantic similarity of their sets of GO terms (i.e., T_m and T_n). Since a set of GO terms can be represented by its corresponding set of vectors, semantic similarity of two proteins can be computed by the distance of the two sets of vectors. Let $\mathbf{V}_m$ denote the set of vectors that correspond to T_m, and $\mathbf{V}_n$ correspond to T_n. Then, the similarity between two proteins is given by the semantic similarity between two sets of vectors, that is, the distance between the corresponding sets of vectors:

$$sim(p_m, p_n) = sim(T_m, T_n) = dist(\mathbf{V}_m, \mathbf{V}_n) \tag{7}$$

There are several measures that can be used to compute the semantic similarity between two sets of vectors [35], [55]. In our experiments, we find that the modified Hausdorff distance [49] achieves much better performance than the linear combination of vectors. Therefore, in this paper we adopt the modified Hausdorff distance to compute the distance of two sets of vectors for the functional similarity between two proteins.

TABLE I: Statistics of the three categories of gene ontologies. '#GO Terms' denotes the number of GO terms while '#Edges' denotes the number of edges.

Ontology	#GO Terms	#Edges
BP	30,705	71,530
CC	4,380	7,523
MF	12,127	13,658

Given two points in a vector space (e.g., the Euclidean space), $dist$ measures the distance of the two vectors in the space. The smaller the $dist$ score is, the closer the two vectors are. Since GO terms are transformed into vectors, the $dist(\mathbf{v}_i, \mathbf{v}_j)$ score can be used to estimate the spatial relation of two GO terms t_i and t_j. $dist(\mathbf{v}_i, \mathbf{v}_j)$ is defined by the opposite of the distance function: the larger the $dist(\mathbf{v}_i, \mathbf{v}_j)$ is, the closer the terms t_i and t_j are. Therefore, we get a variant of the modified Hausdorff distance [49] for computing the functional similarity between proteins p_m and p_n from two sets of vector representations of GO terms. Specifically, the modified Hausdorff distance of two proteins is defined by

$$mhd(\mathbf{V}_m, \mathbf{V}_n) = \min\left\{\frac{1}{|\mathbf{V}_m|}\sum_{\mathbf{v}_m\in\mathbf{V}_m}\max_{\mathbf{v}_n\in\mathbf{V}_n} cos(\mathbf{v}_m, \mathbf{v}_n), \frac{1}{|\mathbf{V}_n|}\sum_{\mathbf{v}_n\in\mathbf{V}_n}\max_{\mathbf{v}_m\in\mathbf{V}_m} cos(\mathbf{v}_m, \mathbf{v}_n)\right\} \tag{8}$$

where $|\mathbf{V}_m|$ denotes the number of vectors in $\mathbf{V}_m$.

III. Experiments and Results

We conducted experiments on the two tasks of missing PPI prediction and spurious PPI prediction, which are defined in Section II-A and evaluated the performance in comparison with representative IC-based methods including Resnik [24], Lin [23], Jang&Conrath [22], simGIC [25], and simUI [52]), and recent corpus-based vector representation method Onto2Vec [44] on three kinds of PPI datasets (HUMAN, MOUSE, and YEAST) from the STRING database [50].

A. Experimental Setup

1) Datasets: In this paper, we use three types of datasets: GO, GOA, and PPIN.

GO: The Gene Ontology [19] includes three independent categories of ontologies: BP, CC, and MF. The BP ontology includes GO terms that describe a series of events in biological processes. The CC ontology includes GO terms that describe molecular events in the components of a cell. The MF ontology includes GO terms that describe the chemical reactions (e.g., catalytic activity and receptor binding). These GO terms have been used to annotate biomedical entities (e.g., genes and proteins) and interpret biomedical experiments (e.g., genetic interactions and biological pathways). Table I summarise the statistics of the three gene ontologies.

GOA: GO annotations are statements about the functions of particular genes or proteins, and capture how a gene or protein functions at the molecular level, and what biological processes

TABLE II: Statistics on ground-truth PPI networks as well as the removed and added PPIs.

Dataset	#Proteins	#PPIs	#Remove PPIs	#Add PPIs
HUMAN	6,966	1,784,108	500,000	500,000
MOUSE	16,105	7,515,864	500,000	500,000
YEAST	2,851	456,936	100,000	100,000

it is associated with. Generally, a protein is annotated by several GO terms. For example, the protein 'P06182' is annotated by the GO terms 'GO:0004408', 'GO:0005743', 'GO:0005758', 'GO:0018063', 'GO:0046872'. We mapped the proteins to the UniProt[1] database [51] to obtain the GO annotations.

PPIN: From the STRING database [50], we downloaded three kinds of PPI datasets (v11.0 version): HUMAN (Homo sapiens), MOUSE (Mus musculus), and YEAST (Saccharomyces cerevisiae). The HUMAN dataset contains 9,677 proteins and 11,759,455 interactions, the MOUSE dataset contains 20,269 proteins and 8,780,518 interactions, and the YEAST dataset contains 3,287 proteins and 1,845,966 interactions. We mapped the proteins to the UniProt database and filter out those proteins that could not be found in the UniProt database; we also discarded those interactions involving the filtered proteins. After filtering, the HUMAN dataset remains 6,966 proteins and 1,784,108 interactions, the MOUSE dataset remains 16,105 proteins and 7,515,864 interactions, and the YEAST dataset remains 2,851 proteins and 456,936 interactions. The remaining proteins and interactions in the three datasets were treated as their ground-truth PPI graphs.

We randomly sampled 500,000 HUMAN interactions, 500,000 MOUSE interactions, and 100,000 YEAST interactions from the ground-truth PPI graphs, and removed these sampled interactions from the ground-truth PPI graphs and treated them as missing PPIs. This kind of derived datasets is used for the missing PPI prediction.

From the ground-truth PPI datasets, we randomly sampled the same number of pairs of proteins (i.e., 500,000 interactions for HUMAN proteins, 500,000 interactions for MOUSE proteins, and 100,000 interactions for YEAST proteins), between which there are no interactions, and added them to the ground-truth PPI datasets. These added interactions were treated as spurious PPIs, and this kind of derived datasets is used for the spurious PPI prediction.

Table II summarizes the statistics of the proteins and interactions of the ground-truth PPI graphs, as well as the number of the removed PPIs and the added PPIs.

2) Implementation Details: We implemented several versions of our method in both ways described in Section II-D1 and II-D2. The version that uses the learned vectors of proteins with cosine distance (see Equation (6)) is denoted by "cos." The version that uses the learned vectors of GO terms with modified Hausdorff distance (see Equation (8)) is denoted by "mhd."

[1]https://www.uniprot.org/

To investigate the effect of using undirected graphs, we also implemented two versions of GE4LP working on directed graphs. Their corresponding versions are denoted by "d_cos" and "d_mhd," where "d" indicates using **d**irected graphs. Except using directed graphs, d_cos is the same as cos and d_mhd is the same as mhd.

For the node2vec model, we applied its code in our experiments with trying different settings and reported the best performance. The setting that achieves the best results is as follows: 100 dimensions, 20 walks per node, 100-length per walk and 20 walks per node, unweighted and undirected edges.

3) Existing Methods: Our method was compared with existing methods including the representative information content-based methods, namely Resnik [24], Lin [23], Jang&Conrath [22], simGIC [25], and simUI [52], and the corpus-based vector representation method Onto2Vec [44].

Resnik's semantic similarity is based on the information content (IC) of a given term in an ontology. The IC of a term t is defined by the negative log-likelihood in Equation (9).

$$IC(t) = -\log p(t) \tag{9}$$

where $p(t)$ is the probability of encountering an instance of the term t. According to this information, Resnik similarity is defined as

$$sim_{Resnik}(t_1, t_2) = -\log p(t_m) \tag{10}$$

where t_m is the most informative common ancestor of t_1 and t_2 in the ontology.

Lin similarity [23] is defined as

$$sim_{Lin}(t_1, t_2) = \frac{2 * \log p(t_m)}{\log p(t_1) + \log p(t_2)} \tag{11}$$

Jang&Conrath similarity [22] is instead defined as

$$sim_{J\&C}(t_1, t_2) = 2 * \log p(t_m) - \log p(t_1) - \log p(t_2) \tag{12}$$

simGIC similarity [25] and simUI similarity [52] compute the functional similarity between proteins. Let T_1 and T_2 be the set of GO terms that annotate proteins p_1 and p_2, respectively. simGIC similarity is defined by the Jaccard index as Equation (13) while simUI is defined by the universal index as Equation (14).

$$fun_{GIC}(p_1, p_2) = \frac{\sum_{t \in T_1 \cap T_2} IC(t)}{\sum_{t \in T_1 \cup T_2} IC(t)} \tag{13}$$

$$fun_{UI}(p_1, p_2) = \frac{\sum_{t \in T_1 \cap T_2} IC(t)}{\max\{\sum_{t \in T_1} IC(t), \sum_{t \in T_2} IC(t)\}} \tag{14}$$

The three kinds of combinations for Resnik, Lin, and Jang&Conrath similarities include average (AVG), maximum (MAX), and best-match average (BMA), and they are defined by Equations (15), (16), and (17), respectively.

$$fun_{AVG}(p_1, p_2) = \frac{1}{|T_1||T_2|} \sum_{t_1 \in T_1, t_2 \in T_2} IC(\{t_1, t_2\}) \tag{15}$$

$$fun_{MAX}(p_1, p_2) = \max\{IC(\{t_1, t_2\}) | t_1 \in T_1, t_2 \in T_2\} \tag{16}$$

TABLE III: AUC of ROC curve for **missing** PPI prediction

Ontology	Model	HUMAN	MOUSE	YEAST
BP	Resnik	0.8257	0.8154	0.8224
	Lin	0.8065	0.7831	0.7752
	Jang&Conrath	0.7973	0.7694	0.7610
	simGIC	0.8147	0.7775	0.7914
	Onto2Vec	0.8458	0.8316	0.8416
	Our method (cos)	0.8513	0.8419	0.8674
	Our method (mhd)	**0.8676**	**0.8527**	**0.8718**
CC	Resnik	0.7776	0.7826	0.7916
	Lin	0.7165	0.7251	0.7435
	Jang&Conrath	0.7134	0.7295	0.7201
	simGIC	0.7658	0.7761	0.7715
	Onto2Vec	0.7984	0.8016	0.8068
	Our method (cos)	0.8027	0.8196	0.8035
	Our method (mhd)	**0.8237**	**0.8349**	**0.8146**
MF	Resnik	0.7934	0.7815	0.7916
	Lin	0.7335	0.7428	0.7432
	Jang&Conrath	0.7129	0.7349	0.7216
	simGIC	0.7618	0.7796	0.7794
	Onto2Vec	0.7953	0.7954	0.8059
	Our method (cos)	0.8115	0.8145	0.8243
	Our method (mhd)	**0.8223**	**0.8316**	**0.8253**

TABLE IV: AUC of ROC curve for **spurious** PPI prediction

Ontology	Model	HUMAN	MOUSE	YEAST
BP	Resnik	0.8243	0.7935	0.7917
	Lin	0.7758	0.7514	0.7572
	Jang&Conrath	0.7494	0.7427	0.7348
	simGIC	0.7965	0.7638	0.7823
	Onto2Vec	0.8426	0.8167	0.8051
	Our method (cos)	0.8613	0.8207	0.8324
	Our method (mhd)	**0.8725**	**0.8439**	**0.8467**
CC	Resnik	0.7827	0.7758	0.8016
	Lin	0.7334	0.7364	0.7452
	Jang&Conrath	0.7157	0.7296	0.7291
	simGIC	0.7608	0.7710	0.7776
	Onto2Vec	0.8016	0.7913	0.7935
	Our method (cos)	0.8191	0.8142	0.8117
	Our method (mhd)	**0.8360**	**0.8207**	**0.8254**
MF	Resnik	0.7903	0.7817	0.7834
	Lin	0.7317	0.7298	0.7265
	Jang&Conrath	0.7186	0.7215	0.7184
	simGIC	0.7636	0.7716	0.7716
	Onto2Vec	0.8137	0.7903	**0.8216**
	Our method (cos)	0.8116	0.8134	0.8177
	Our method (mhd)	**0.8209**	**0.8275**	0.8194

$$fun_{BMA}(p_1,p_2) = \frac{1}{2}(\frac{1}{|T_1|}\sum_{t_1\in T_1} IC(\{t_1,t_2\}) + \frac{1}{|T_2|}\sum_{t_2\in T_2} IC(\{t_1,t_2\})) \quad (17)$$

Onto2Vec [44] uses the word2vec model [42] with the skip-gram algorithm to learn from the descriptive axioms of GO terms and proteins. Given a sequence S of training words s_1, s_2, ..., s_K, the skip-gram model aims to maximize the average log-likelihood of Function (18),

$$Loss = \frac{1}{K}\sum_{k=1}^{K}\sum_{-|S|\le i\le |S|, i\neq 0} \log p(s_{t+i}|s_t) \quad (18)$$

where $|S|$ is the size of the training text and K is the size of the vocabulary. After getting the word vectors from the word2vec model, Onto2Vec linearly combines the word vectors for proteins according to the words appearing in the descriptive axioms of proteins

$$v(p) = \sum_{s_i\in S} v(s_i) \quad (19)$$

where $v(p)$ is the vector of protein p, $v(s_i)$ is the vector of word s_i, and S represents the set of words in the descriptive axiom of protein p.

4) Evaluation Metrics: The performances of missing and spurious PPI predictions were evaluated under the metric of area (AUC) under the ROC (Receiver Operating Characteristic) curve, which is widely used to evaluate the performance of classification and prediction tasks. ROC is defined by the relation between the true-positive rate (TPR) and the false-positive rate (FPR). TPR is defined as $TPR = \frac{TP}{TP+FN}$ and FPR is defined as $FPR = \frac{FP}{FP+TN}$, where TP denotes the number of true positives, FP denotes the number of false positives, TN denotes the number of true negatives, and FN denotes the number of false negatives.

B. Experimental Results

Table III reports overall performance of our proposed methods and existing methods for the task of missing PPI prediction. Table IV reports overall performance of our models and existing methods for the spurious PPI prediction. For each PPI dataset using each GO category, the best result is highlighted in boldface.

1) Missing PPI Prediction: From Table III we can see that 'cos' and 'mhd' achieve the best results on the missing PPI prediction, in comparison with the information content-based methods and the corpus-based vector representation method on all the three PPI datasets. This indicates that graph embeddings can capture structural information from GOA graphs that is useful for predicting the missing PPIs, and both the learned vectors of proteins and the ones of GO terms are effective for the missing PPI prediction.

Particularly, our proposed methods significantly outperform the traditional IC-based methods; the possible reason is that the IC-based methods consider only the information from the partial or local structure of a graph, while 'cos' and 'mhd' take into account the information from both the local and globe structure of the GOA graphs, which incorporates the knowledge of both term-term relations between GO terms and term-protein annotations between GO terms and proteins. 'cos' and 'mhd' also outperform the corpus-based vector representation method Onto2Vec. The possible reason is that our proposed methods leverage the domain knowledge stored in the structure of Gene Ontology and GO annotations.

Let us look at the comparison between the performance of 'cos' and the one of 'mhd'. 'mhd' achieves slightly better performance than 'cos' does. The possible reason is that 'mhd' levarages the fact that a protein is annotated by a set of GO terms and uses the modified Hausdorff distance [49] to estimate the distance of two sets of data points (i.e., two sets of vectors of GO terms) in the Euclidian space. Our experimental results also justify the usefulness of the functional annotation relationships between GO terms and proteins.

2) Spurious PPI Prediction: From Table IV we can see that, 'cos' and 'mhd' outperform both the IC-based methods

TABLE V: Comparison between our method between using undirected graphs and using directed graphs for **missing** PPI prediction. 'd' stands for directed graph, 'cos' stands for cosine similarity, and 'mhd' stands for modified Hausdorff distance. Evaluation metric: AUC of ROC curve.

Ontology	Model	HUMAN	MOUSE	YEAST
BP	mhd	**0.8676**	**0.8527**	**0.8718**
	cos	0.8513	0.8419	0.8674
	d_mhd	0.8134	0.8038	0.8295
	d_cos	0.8027	0.7924	0.8246
CC	mhd	**0.8237**	**0.8349**	**0.8146**
	cos	0.8027	0.8196	0.8035
	d_mhd	0.7837	0.8001	0.7766
	d_cos	0.7712	0.7931	0.7613
MF	mhd	**0.8223**	**0.8316**	**0.8253**
	cos	0.8115	0.8145	0.8243
	d_mhd	0.7884	0.7765	0.7835
	d_cos	0.7716	0.7664	0.7769

TABLE VI: Comparison between different methods between using undirected graphs and using directed graphs for **spurious** PPI prediction

Ontology	Model	HUMAN	MOUSE	YEAST
BP	mhd	**0.8725**	**0.8439**	**0.8467**
	cos	0.8613	0.8207	0.8324
	d_mhd	0.8203	0.8034	0.8101
	d_cos	0.8167	0.7908	0.7964
CC	mhd	**0.8360**	**0.8207**	**0.8254**
	cos	0.8191	0.8142	0.8117
	d_mhd	0.8002	0.7834	0.7749
	d_cos	0.7763	0.7771	0.7746
MF	mhd	**0.8209**	**0.8275**	**0.8194**
	cos	0.8116	0.8134	0.8177
	d_mhd	0.7768	0.7824	0.7658
	d_cos	0.7834	0.7739	0.7549

and the corpus-based vector representation method on almost all the datasets except on the YEAST PPI dataset using the MF ontology. Similar to the performance on missing PPI prediction, this indicates again that graph embeddings can capture useful information from the structure of GOA graphs for the spurious PPI prediction, and that both the learned vectors of proteins and the ones of GO terms are effective for the spurious PPI prediction. In addition, 'mhd' performs slightly better than 'cos' on spurious PPI prediction, similar to their performance on missing PPI prediction. This justifies again the relationships between GO terms and proteins as term-protein annotations.

C. Undirected Graphs vs. Directed Graphs

Tables V and VI report comparisons between our proposed methods using undirected graphs and the ones using directed graphs for the missing and spurious PPI predictions. We can see that using undirected graphs achieves much better performance than using directed graphs does. The possible reason is that the node2vec model we used in this paper adopts a strategy of random walk over an undirected graph to sample neighborhood nodes for a given node and this strategy works better on undirected graphs than on directed graphs.

IV. Conclusion

In this paper, we defined two kinds of PPI prediction tasks, namely missing PPI prediction and spurious PPI prediction, to address the issues of missing and spurious interactions in PPIN. To investigate our defined tasks, we proposed a method that employs the technique of graph embeddings on the constructed GO annotation graph, which incorporates both the term-term relations between GO terms and term-protein annotations between GO terms and proteins. Our proposed method is able capture simultaneously the information from the local and globe structure of the constructed GOA graph. To evaluate the quality of our method, we conducted experiments on three kinds of PPI datasets, and compared our method with the representative information content-based methods and corpus-based word embeddings methods. Experimental results demonstrate the effectiveness of using graph embeddings to learn vector representations from GO annotation graph for missing and spurious PPI predictions.

References

[1] Wang, Y. & Zeng, J. Predicting drug-target interactions using restricted boltzmann machines. *Bioinformatics* **29**, i126–i134 (2013).

[2] Lu, Y., Guo, Y. & Korhonen, A. Link prediction in drug-target interactions network using similarity indices. *BMC bioinformatics* **18**, 39 (2017).

[3] Wang, J., Peng, X., Peng, W. & Wu, F.-X. Dynamic protein interaction network construction and applications. *Proteomics* **14**, 338–352 (2014).

[4] Wang, J., Peng, X., Li, M. & Pan, Y. Construction and application of dynamic protein interaction network based on time course gene expression data. *Proteomics* **13**, 301–312 (2013).

[5] De Las Rivas, J. & Fontanillo, C. Protein–protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS computational biology* **6**, e1000807 (2010).

[6] Pawson, T. Protein modules and signalling networks. *Nature* **373**, 573 (1995).

[7] Chen, J. & Yuan, B. Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics* **22**, 2283–2290 (2006).

[8] Marcotte, E. M. *et al.* Detecting protein function and protein-protein interactions from genome sequences. *Science* **285**, 751–753 (1999).

[9] Rao, V. S., Srinivas, K., Sujini, G. & Kumar, G. Protein-protein interaction detection: methods and analysis. *International journal of proteomics* **2014** (2014).

[10] Singh, R., Xu, J. & Berger, B. Struct2net: integrating structure into protein-protein interaction prediction. In *Biocomputing 2006*, 403–414 (World Scientific, 2006).

[11] Singh, R., Park, D., Xu, J., Hosur, R. & Berger, B. Struct2net: a web service to predict protein–protein interactions using a structure-based approach. *Nucleic acids research* **38**, W508–W515 (2010).

[12] Murakami, Y. & Mizuguchi, K. Psopia: Toward more reliable protein-protein interaction prediction from sequence information. In *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 255–261 (IEEE, 2017).

[13] Phizicky, E. M. & Fields, S. Protein-protein interactions: methods for detection and analysis. *Microbiol. Mol. Biol. Rev.* **59**, 94–123 (1995).

[14] Chen, X.-W. & Liu, M. Prediction of protein–protein interactions using random decision forest framework. *Bioinformatics* **21**, 4394–4400 (2005).

[15] Hosur, R., Xu, J., Bienkowska, J. & Berger, B. iwrap: an interface threading approach with application to prediction of cancer-related protein–protein interactions. *Journal of molecular biology* **405**, 1295–1310 (2011).

[16] Kotlyar, M. *et al.* In silico prediction of physical protein interactions and characterization of interactome orphans. *Nature methods* **12**, 79 (2015).

[17] Tastan, O., Qi, Y., Carbonell, J. G. & Klein-Seetharaman, J. Prediction of interactions between hiv-1 and human proteins by information integration. In *Biocomputing 2009*, 516–527 (World Scientific, 2009).

[18] Sun, T., Zhou, B., Lai, L. & Pei, J. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC bioinformatics* **18**, 277 (2017).

[19] Consortium, G. O. The gene ontology (go) database and informatics resource. *Nucleic Acids Research* **32**, D258–D261 (2004). URL http://dx.doi.org/10.1093/nar/gkh036.

[20] Hill, D. P., Smith, B., McAndrews-Hill, M. S. & Blake, J. A. Gene ontology annotations: what they mean and where they come from. In *BMC bioinformatics*, vol. 9, S2 (BioMed Central, 2008).

[21] Barrell, D. *et al.* The goa database in 2009—an integrated gene ontology annotation resource. *Nucleic acids research* **37**, D396–D403 (2008).

[22] Jiang, J. J. & Conrath, D. W. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Computational Linguistics*, 19–33 (1997).

[23] Lin, D. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, 296–304 (1998).

[24] Resnik, P. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 448–453 (1999).

[25] Pesquita, C., Faria, D., Bastos, H., Falcao, A. O. & Couto, F. M. Evaluating go-based semantic similarity measures. In *Proceedings of the 10th Annual Bio-Ontologies Meeting*, vol. 37, 38 (2007).

[26] Lord, P. W., Steven, R. D., Brass, A. & Goble, C. A. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics* **19**, 1275–1283 (2003).

[27] Couto, F. M., Silva, M. J. & Coutinho, P. Implementation of a functional semantic similarity measure between gene-products. Tech. Rep., University of Lisbon (2003).

[28] Lee, S. G., Hur, J. U. & Kim, Y. S. A graph-theoretic modeling on go space for biological interpretation of gene clusters. *Bioinformatics* **20**, 381–388 (2004).

[29] Sevilla, J. L. *et al.* Correlation between gene expression and go semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **2**, 330–338 (2005).

[30] Guo, X., Liu, R., Shriver, C. D., Hu, H. & Liebman, M. N. Assessing semantic similarity measures for the characterization of human regulatory pathways. *Bioinformatics* **22**, 967–973 (2006).

[31] Schlicker, A., Domingues, F. S., Rahnenfuhrer, J. & Lengauer, T. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics* **7** (2006).

[32] Couto, F. M., Silva, M. J. & Coutinho, P. M. Measuring semantic similarity between gene ontology terms. *Data & Knowledge Engineering* **61**, 137–152 (2007).

[33] Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S. & Chen, C.-F. A new method to measure the semantic similarity of go terms. *Bioinformatics* **23**, 1274–1281 (2007).

[34] Xu, T., Du, L. & Zhou, Y. Evaluation of go-based functional similarity measures using s.cerevisiae protein interaction and expression profile data. *BMC Bioinformatics* **9**, 1–10 (2008).

[35] Pesquita, C., Faria, D., Falcao, A. O., Lord, P. & Couto, F. M. Semantic similarity in biomedical ontologies. *PLoS Computational Biology* **5**, 1–12 (2009).

[36] Li, B., Wang, J. Z., Feltus, F. A., Zhou, J. & Luo, F. Effectively integrating information content and structural relationship to improve the go-based similarity measure between proteins. In *Proceedings of International Conference Bioinformatics and Computational Biology*, 166–172 (2010).

[37] Yang, H., Nepusz, T. & Paccanaro, A. Improving go semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. *Bioinformatics* **28**, 1383–1389 (2012).

[38] Li, M., Wu, X., Pan, Y. & Wang, J. hf-measure: A new measurement for evaluating clusters in protein-protein interaction networks. *Proteomics* **13**, 291–300 (2012).

[39] Teng, Z. *et al.* Measuring gene functional similarity based on group-wise comparison of go terms. *Bioinformatics* **29**, 1424–1432 (2013).

[40] Song, X., Li, L., Srimani, P. K., Yu, P. S. & Wang, J. Z. Measure the semantic similarity of go terms using aggregate information content. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **11**, 468–476 (2014).

[41] Zhong, X. & Cambria, E. Time expression recognition using a constituent-based tagging scheme. In *Proceedings of the 2018 World Wide Web Conference*, 983–992 (2018).

[42] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119 (2013).

[43] Pennington, J., Socher, R. & Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543 (2014).

[44] Smaili, F. Z., Gao, X. & Hoehndorf, R. Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations. *Bioinformatics* **34**, i52–i60 (2018).

[45] Smaili, F. Z., Gao, X. & Hoehndorf, R. Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinformatics* .

[46] Duong, D., Ahmad, W. U., Eskin, E., Chang, K.-W. & Li, J. J. Word and sentence embedding tools to measure semantic similarity of gene ontology terms by their definitions. *Journal of Computational Biology* **26**, 38–52 (2018).

[47] Zhong, X., Kaalia, R. & Rajapakse, J. C. Go2vec: Transforming go terms and proteins to vector representations via graph embeddings. *BMC Genomics* (2019).

[48] Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864 (2016).

[49] Dubuisson, M.-P. & Jain, A. K. A modified hausdorff distance for object matching. In *Proceedings of the 12th International Conference on Pattern Recognition*, 566–568 (1994).

[50] Mering, C. v. *et al.* String: a database of predicted functional associations between proteins. *Nucleic acids research* **31**, 258–261 (2003). URL https://doi.org/10.1093/nar/gkg034.

[51] Consortium, U. Uniprot: a hub for protein information. *Nucleic acids research* **43**, D204–D212 (2014). URL https://doi.org/10.1093/nar/gku989.

[52] Gentleman. Manual for r (2005).

[53] Perozzi, B., AL-Rfou, R. & Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710 (2014).

[54] Tang, J. *et al.* Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, 1067–1077 (2015).

[55] Mazandu, G. K. & Mulder, N. J. Information content-based gene ontology functional similarity measures: Which one to use for a given biological data type? *PLoS ONE* **9** (2014).