METHODS
published: 22 September 2021
doi: 10.3389/fgene.2021.744334

Comparative Analysis of Unsupervised Protein Similarity Prediction Based on Graph Embedding

Yuanyuan Zhang[1,2]*, *Ziqi Wang*[1], *Shudong Wang*[2] *and Junliang Shang*[3]

[1] *School of Information and Control Engineering, Qingdao University of Technology, Qingdao, China,* [2] *College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China,* [3] *School of Information Science and Engineering, Qufu Normal University, Rizhao, China*

The study of protein–protein interaction and the determination of protein functions are important parts of proteomics. Computational methods are used to study the similarity between proteins based on Gene Ontology (GO) to explore their functions and possible interactions. GO is a series of standardized terms that describe gene products from molecular functions, biological processes, and cell components. Previous studies on assessing the similarity of GO terms were primarily based on Information Content (IC) between GO terms to measure the similarity of proteins. However, these methods tend to ignore the structural information between GO terms. Therefore, considering the structural information of GO terms, we systematically analyze the performance of the GO graph and GO Annotation (GOA) graph in calculating the similarity of proteins using different graph embedding methods. When applied to the actual Human and Yeast datasets, the feature vectors of GO terms and proteins are learned based on different graph embedding methods. To measure the similarity of the proteins annotated by different GO numbers, we used Dynamic Time Warping (DTW) and cosine to calculate protein similarity in GO graph and GOA graph, respectively. Link prediction experiments were then performed to evaluate the reliability of protein similarity networks constructed by different methods. It is shown that graph embedding methods have obvious advantages over the traditional IC-based methods. We found that random walk graph embedding methods, in particular, showed excellent performance in calculating the similarity of proteins. By comparing link prediction experiment results from GO(DTW) and GOA(cosine) methods, it is shown that GO(DTW) features provide highly effective information for analyzing the similarity among proteins.

Keywords: protein similarity, graph embedding, gene ontology, link prediction, DTW algorithm

OPEN ACCESS

Edited by:
Jianing Xi,
Northwestern Polytechnical University,
China

Reviewed by:
Shouheng Tuo,
Xi'an University of Posts
and Telecommunications, China
Cheng Liang,
Shandong Normal University, China
Yajun Liu,
Xi'an University of Technology, China

****Correspondence:***
Yuanyuan Zhang
yyzhang1217@163.com

Specialty section:
This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: *20 July 2021*
Accepted: *25 August 2021*
Published: *22 September 2021*

Citation:
Zhang Y, Wang Z, Wang S and
Shang J (2021) Comparative Analysis
of Unsupervised Protein Similarity
Prediction Based on Graph
Embedding.
Front. Genet. 12:744334.
doi: 10.3389/fgene.2021.744334

INTRODUCTION

Proteomics essentially refers to the study of the characteristics of proteins on a large scale, including the expression level of proteins, the functions of proteins, protein–protein interactions, and so forth. The study of proteome not only provides the material basis for the law of life activities but can also provide the theoretical basis and solutions for elucidating and solving the mechanism of many diseases (Xi et al., 2020a). However, at present, research on the function of proteins is lacking. The functions of proteins encoded by most of the newly discovered genes by genome sequencing are unknown. For those whose functions are known, their functions have mostly been inferred by methods such as homologous gene function analogy. Therefore, using computational methods to explore the similarity between proteins can effectively improve the efficiency of proteomic studies.

Gene Ontology (GO) (Harris, 2004) describes the function of genes It is a standardized description of the characteristics of genes and gene products, enabling bioinformatics researchers to uniformly summarize, process, interpret, and share the data of genes and gene products. It provides the representation of biological knowledge through structured and controlled terms. GO includes three kinds of ontologies: Biological Processes (BPs), Cell Components (CCs), and Molecular Functions (MFs). The words in the three kinds of ontologies are related to each other and form a Directed Acyclic Graph (DAG), wherein a node denotes a GO term, while an edge denotes a kind of relationship between two GO terms. Therefore, it is of great significance to study the similarity of proteins based on the graph characteristics of GO to explore the function of proteins.

GO has been widely studied in the field of biology (Xi et al., 2020b). GO terms have been used to annotate many biomedical databases [e.g., UniProt database (UniProt Consortium, 2015) and SwissProt database (Amos and Brigitte, 1999)]. The characteristics and structure of GO have made GO terms the basis of functional comparison between gene products (Pesaranghader et al., 2014). GO annotation defines the semantic similarity of genes (proteins) and provides a basis for measuring the functional similarity of proteins. The more information two GO terms share, the more similar they are, and the more the similarity between the proteins annotated by the two GO terms (Hu et al., 2021). In earlier studies, many researchers analyzed protein–protein interaction (PPI) based on GO (Sevilla et al., 2005). Studies on computing protein similarity using GO mainly focus on the IC of GO terms, which is widely used to identify relations between proteins. The uniqueness of GO terms is often evaluated by taking the average of the IC of two terms. The IC of a term depends on the annotating corpus (Sevilla et al., 2005). Three IC-based methods—Resnik's (Resnik, 1999), Rel's (Paul and Meeta, 2008), and Jiang and Conrath's (Jiang and Conrath, 1997)—have been introduced from natural language taxonomies by Lord et al. (2003) to compare genes (proteins). Although the abovementioned methods are used to calculate semantic similarity between two GO terms to achieve good results, they only consider the amount of information of common nodes. They do not consider the information differences between the nodes themselves and ignore the structural information of the terms. The result of term comparison is a rough estimate. For example, in Resnik's method, if the ancestors of two terms are the same, then the similarity of two terms in any layer is not different and cannot be compared. Obviously, this is unreasonable.

This study merged the three categories of ontologies and GO annotations into a large graph called the GO Annotation (GOA) graph. We used three categories of ontologies transformed into a GO graph. Effective graph analysis on GOA and GO graphs can improve our understanding of the structure and node information of GO and proteins. Using the GOA information of the proteins, the similarity among proteins can be calculated, and the relationship between proteins can be predicted. In recent years, graph learning-based analytical methods have made remarkable progress in bioinformatics and other fields (Xi et al., 2021). At present, graph learning-based analytical methods focuses on dynamic graphs. Methods such as SDNE (Wang et al., 2016), DeepWalk (Perozzi et al., 2014), LINE (Tang et al., 2015), Node2vec (Grover and Leskovec, 2016), and SINE (Wang et al., 2020) have been widely used for unsupervised feature learning in the field of data mining and natural language processing. The edge prediction task is applied to the PPI prediction to find new protein interaction relationships. They also provide a basis for calculating protein similarity based on GO, such as GO2vec (Zhong et al., 2019), which used the Node2vec algorithm to compute the functional similarity between proteins.

To explore the performance of graph embedding methods in measuring protein similarity based on GO and GOA, we used four typical graph embedding methods to learn the features of GO terms and proteins. These methods can be divided into two categories. The first category is the random walk method, such as the DeepWalk and Node2Vec methods. The DeepWalk method uses the truncated random walk strategy to obtain the sequence of nodes and point embedding obtained from learning with Word2Vec (Goldberg and Levy, 2014). Node2Vec uses biased random walk to generate a node sequence by balancing the Breadth First Search (BFS) and Depth First Search (DFS) of the graph. The second category is based on deep learning, such as SDNE and LINE methods. SDNE uses an auto-encoder to optimize the first-order and second-order similarity simultaneously, while LINE optimizes the orders of similarity separately. As a result, their learned node embedding can retain the local and global graph structure and is robust to sparse networks. We introduce the overall flowchart of this paper in **Figure 1**, which is divided into two parts. Firstly, in Part A, the features of GO terms are learned based on the GO graph using graph embedding methods. The similarity of proteins is then calculated based on the features of their annotated GO terms by Dynamic Time Warping (DTW) distance (Lou et al., 2016). Secondly, in Part B, the features of proteins are learned based on the GOA graph directly. Then, the cosine similarity of the corresponding features is calculated to measure the similarity of protein. Finally, a link prediction (Li et al., 2018) experiment is performed in the screened-out protein similarity networks, using the area under the curve (AUC) (Lobo, 2010) and area under the precision-recall curve (AUCPR) (Yu and Park, 2014) to evaluate the reliability of the protein network constructed by learned vectors.

MATERIALS AND METHODS

Data Source and Preprocessing

We downloaded GO data in Open Biomedical Ontologies (OBO) format from the GO Consortium Website [1]. The GO protein

[1] http://geneontology.org/page/download-ontology

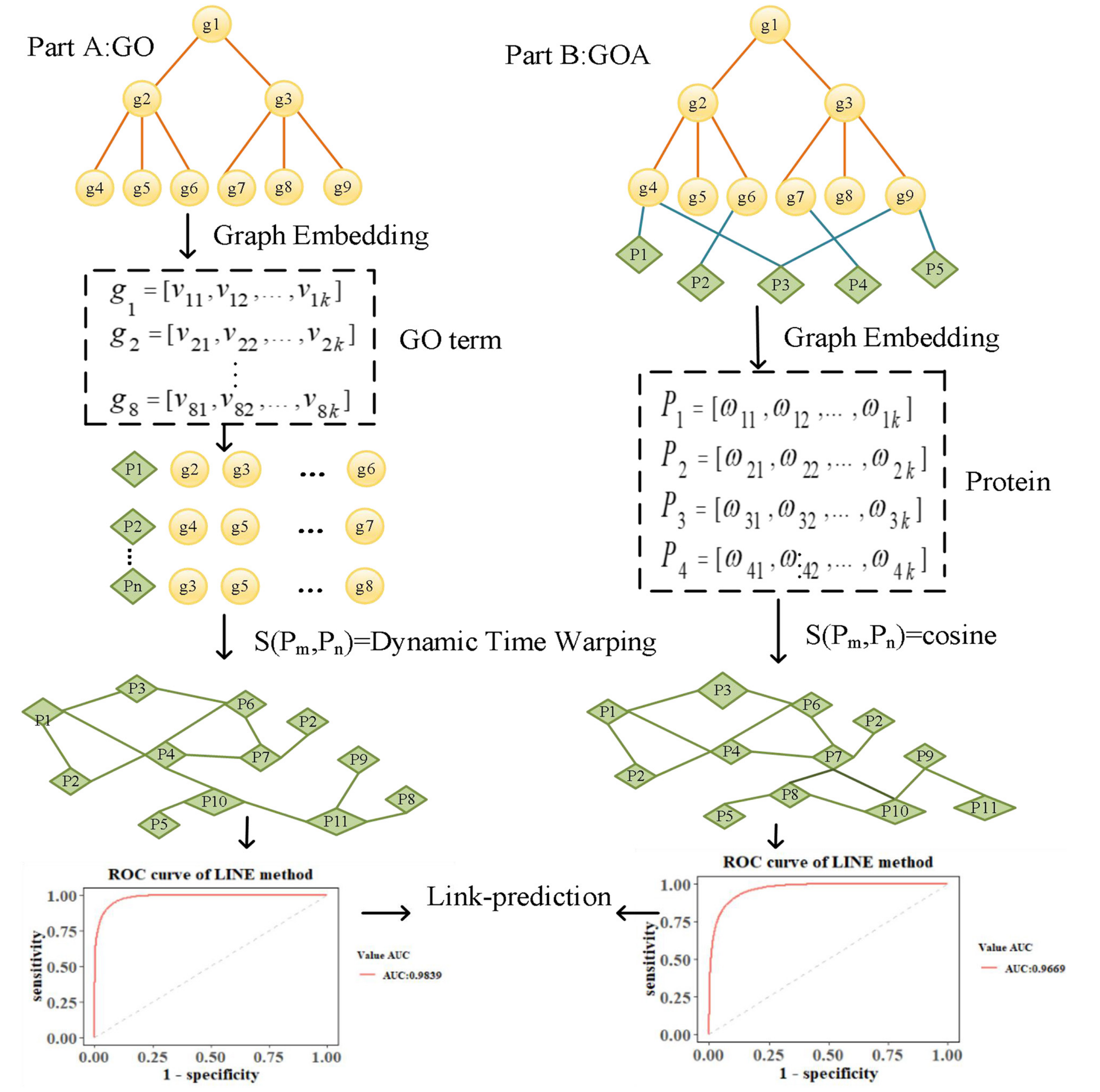

Part A:GO
g1
g2
g3
g4
g5
g6
g7
g8
g9
Graph Embedding
$g_1=[v_{11},v_{12},\dots,v_{1k}]$
$g_2=[v_{21},v_{22},\dots,v_{2k}]$
$g_8=[v_{81},v_{82},\dots,v_{8k}]$
GO term
P1
P2
Pn
S(P_m,P_n)=Dynamic Time Warping
Part B:GOA
P1
P2
P3
P4
P5
Graph Embedding
$P_1=[\omega_{11},\omega_{12},\dots,\omega_{1k}]$
$P_2=[\omega_{21},\omega_{22},\dots,\omega_{2k}]$
$P_3=[\omega_{31},\omega_{32},\dots,\omega_{3k}]$
$P_4=[\omega_{41},\omega_{42},\dots,\omega_{4k}]$
Protein
S(P_m,P_n)=cosine
P6
P7
P8
P9
P10
P11
ROC curve of LINE method
sensitivity
1 - specificity
0.00
0.25
0.50
0.75
1.00
Value AUC
AUC:0.9839
Link-prediction
ROC curve of LINE method
Value AUC
AUC:0.9669

FIGURE 1 | Framework for analyzing protein similarity.

annotations were obtained from the UniProt GOA website [2]. The Yeast dataset contained 2,887 proteins, and the Human dataset contained 9,677 proteins. The GO data were then preprocessed based on the following processes. First, since several GO terms annotate a protein, term–term relations of GO terms and term–protein annotations between GO terms and proteins were combined into a GOA graph. Second, the GO terms were then transformed into an undirected, unweighted GO graph, regardless of the type and direction of the relationship. We summarize the numbers of GO terms and edges in **Table 1**.

Method

Based on different graph embedding methods, the feature of GO terms and proteins was learned into vector representations by fusing GO and GOA graph topologies, respectively. Thus, we could capture the global information based on the graph embedding method, and its learned vectors could calculate

[2] http://www.ebi.ac.uk/GOA

TABLE 1 | Characteristics of GO graphs.

Gene ontology	Term	Edges
BP*	30,705	71,530
CC**	4,380	7,523
MF***	12,127	13,658

**Biological Processes, **Cell Components, and ***Molecular Functions.*

the similarity between proteins by the DTW distance and cosine similarity.

Introduction of Different Graph Embedding Methods

In this paper, we used the methods of graph embedding based on random walk and deep learning to learn the features of GO terms and proteins through fusing the topology of GO and GOA graphs, respectively. Random walk-based methods include DeepWalk (Perozzi et al., 2014) and Node2vec (Grover and Leskovec, 2016). The DeepWalk method is divided into two parts: random walk to obtain node sequences and to generate node embedding. Random walk is used to obtain the local information of the node in the graph, and the embedding reflects the local structure of the node in the graph. The path length is controlled by setting the parameter walk-length ($_L$). The more neighborhood nodes (higher-order neighborhood nodes) two nodes have, the more similar they are. **Figure 2A** illustrates the DeepWalk algorithm flow. Node2vec method sets two hyper-parameters p and q to control the random walk and adopts a flexible biased random walk procedure that smoothly combines BFS and DFS to generate node sequences. **Figure 2B** illustrates the Node2vec algorithm flow. Nodes c_i are generated based on the following distribution:

$$P(c_i = \mathrm{x}|\mathrm{c}_{\mathrm{i}-1} = t) = \begin{cases} \frac{\pi_{tx}}{Z} & \left(if\ (t, x) \in E\right) \\ 0 & \left(otherwise\right) \end{cases} \tag{1}$$

where π_{tx} is the transition probability between nodes t and x, and Z is the normalization constant. According to the node context information, node sequences are generated by setting the sizes of the hyper-parameters p and q to control the random walk strategy. The Skip-gram model is used to obtain the vector representation of the nodes. The random walk graph embedding of nodes reflects the local and global topology information of nodes in the graph.

The second kind of embedding method is SDNE, which proposed a new semi-supervised learning model. Combining the advantages of first-order and second-order estimation, SDNE can capture the global and local structural properties of the graph. The unsupervised part uses a deep auto-encoder to learn the second-order similarity, and the supervised part uses a Laplace feature map to capture the first-order similarity. **Figure 2C** illustrates the SDNE algorithm flow. By inputting the node embedding S_i in the model, where S_i is compressed by the auto-encoder, the feature is then reconstructed. Finally, its loss function is defined as follows:

$$O_2 = \Sigma \left|\left|S'_i - S_i\right|\right|_2^2 \tag{2}$$

LINE is another method based on deep learning, which optimizes the first-order and second-order similarities (**Figure 2D**). The first-order similarity is used to describe the local similarity between pairs of nodes in the graph. The second-order similarity is described as two nodes in the graph not having directly connected edges, but there are common neighbor nodes, which indicate that the two nodes are similar.

Introduction to IC-Based Method

In this paper, we chose two typical IC-based methods to measure the semantic similarity of GO terms, based on Jiang and Conrath (1997) and Rel (Paul and Meeta, 2008). The IC of a term is inversely proportional to the frequency of the term being used to annotate genes in a given corpus, such as the UniProt database. The IC of a GO term g is defined by the negative log-likelihood and is given by

$$IC(g) = -\log p(g) \tag{3}$$

$$p(g) = \frac{freq(g)}{N} \tag{4}$$

where $p(g)$ is the frequency of term g and its offspring in a specific GO annotated corpus. N represents the total number of annotated proteins in the corpus. If there are 50 annotated proteins in a corpus and 10 of them are annotated by term g, the annotation frequency of term g is $p(g) = 0.2$.

Jiang and Conrath and Rel's methods rely on comparing the attributes of terms in GO. Jiang and Conrath's method considered the fact that the semantic similarity between two terms is closely related to the nearest common ancestor corresponding to the two terms. The semantic similarity between two terms is estimated by calculating the amount of IC in the nearest common ancestor. Jiang and Conrath's and Rel's similarities are expressed as follows:

$$sim_{J\&C}(g_1, g_2) = 2^{*}IC(g_c) - IC(g_1) - IC(g_2) \tag{5}$$

$$sim_{Rel}(g_1, g_2) = \frac{2^{*}IC(g_c)}{IC(g_1) + IC(g_2)} + (1 - p(g_c)) \tag{6}$$

where g_c is the most informative common ancestor of g_1 and g_2 in the ontology. Given two proteins P_m and P_n annotated with GO terms $G_m = \left\{g_1, \cdots, g_i\right\}$ and $G_n = \left\{g_1^{'}, \cdots, g_j^{'}\right\}$, we used the Best Match Average (BMA) method to compute the similarity between two sets of GO terms, which can be expressed as follows:

$$\begin{aligned} BMA(P_m, P_n) = \frac{1}{2}(\frac{1}{n}\sum_{g_m \in G_m} \max_{g'_n \in G_n} sim(g_m, g'_n) \\ + \frac{1}{m}\sum_{g'_n \in G_n} \max_{g_m \in G_m} sim(g_m, g'_n)) \end{aligned} \tag{7}$$

where $sim(g_m, g'_n)$ is the similarity between term g_m and term g'_n, which could have been calculated using IC-based similarity methods.

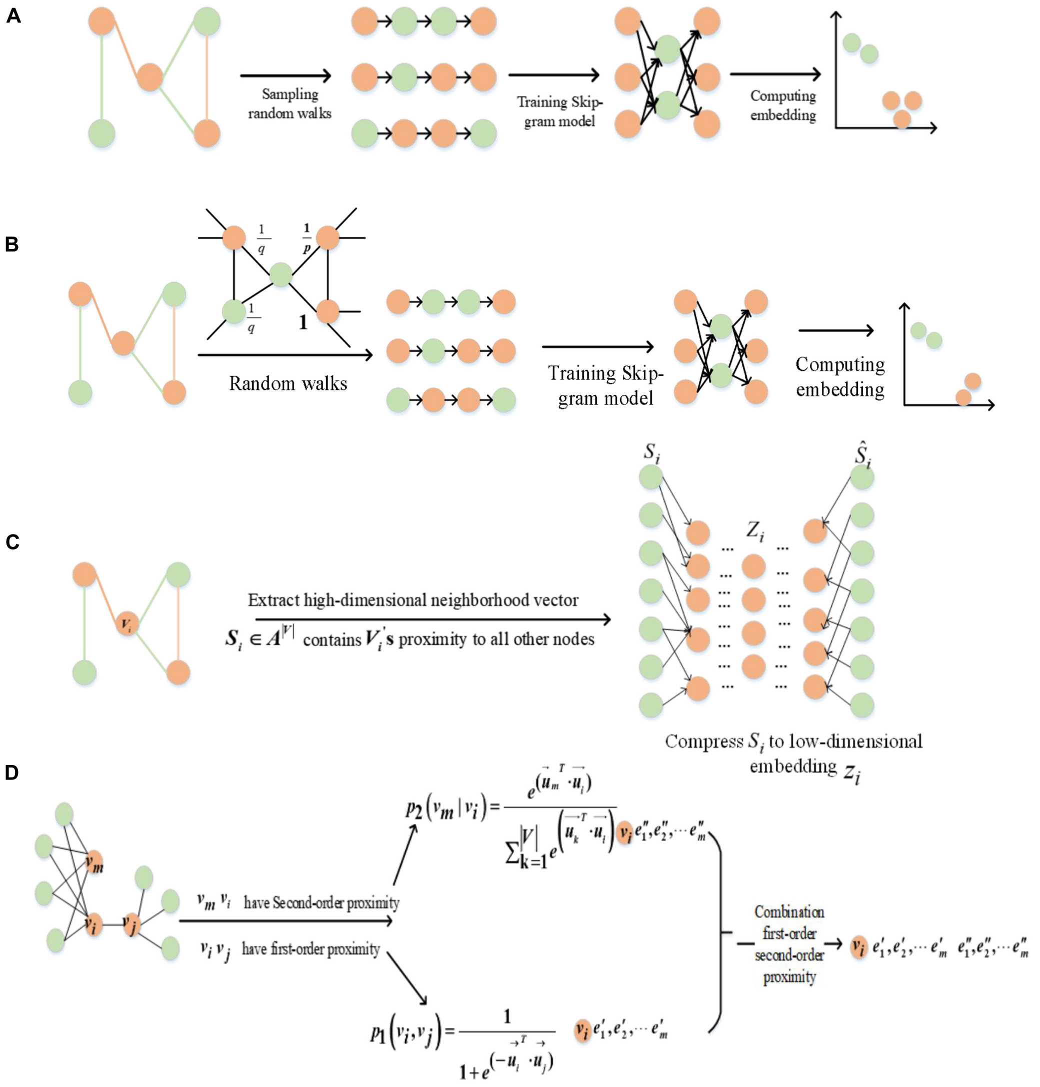

A
Sampling random walks
Training Skip-gram model
Computing embedding
B
1/q
1/p
1/q
1
Random walks
Training Skip-gram model
Computing embedding
C
S_i
$\hat{S}_i$
Z_i
v_i
Extract high-dimensional neighborhood vector
$S_i \in A^{|V|}$ contains V_i's proximity to all other nodes
Compress S_i to low-dimensional embedding z_i
D
v_m
v_i
v_j
v_m v_i have Second-order proximity
v_i v_j have first-order proximity
$p_2(v_m|v_i)=\frac{e^{(\vec{u}_m^T \cdot \vec{u}_i)}}{\sum_{k=1}^{|V|} e^{(\vec{u}_k^T \cdot \vec{u}_i)}}$
v_i $e''_1, e''_2, \cdots e''_m$
$p_1(v_i,v_j)=\frac{1}{1+e^{(-\vec{u}_i^T \cdot \vec{u}_j)}}$
v_i $e'_1, e'_2, \cdots e'_m$
Combination first-order second-order proximity
v_i $e'_1, e'_2, \cdots e'_m$ $e''_1, e''_2, \cdots e''_m$

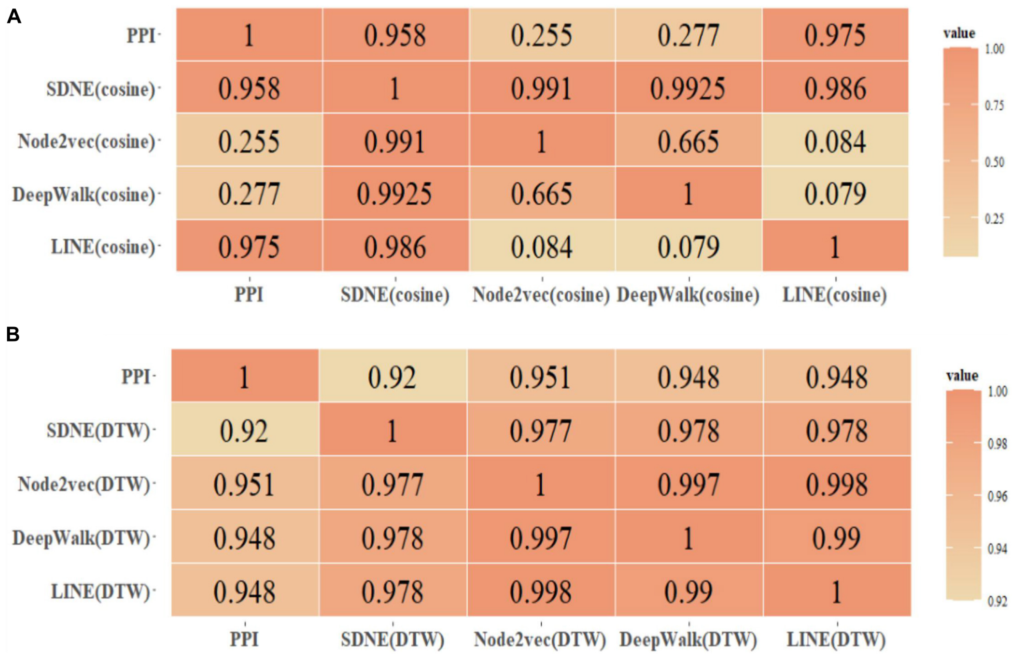
FIGURE 2 | Framework for graph embedding method. **(A)** DeepWalk, **(B)** Node2vec, **(C)** SDNE, and **(D)** LINE.

Protein Similarity Calculation

Each node in the GO graph is represented as a low-dimensional feature vector by considering the topology feature using a graph embedding method. Usually, a protein is annotated by several GO terms. For example, the protein "P03882" is annotated by the GO terms "GO:0004519," "GO:0005739," "GO:0006314," and "GO:0006397." Since a set of GO terms can be represented by its corresponding set of vectors, the similarity between proteins can be calculated based on the similarity of the two sets of GO vectors. Therefore, for any GO term g_i, we use SDNE (Wang et al., 2016), DeepWalk (Perozzi et al., 2014), LINE (Tang et al., 2015), and Node2vec (Grover and Leskovec, 2016) graph embedding methods to learn the low-dimensional feature vector v_i.

We let $G_m = \{g_1, g_2, \cdots, g_m\}$ and $G_n = \{g'_1, g'_2, \cdots, g'_n\}$ denote the sets of GO terms that annotated proteins P_m and P_n; thus, $V_m = \{v_1, v_2, \cdots, v_m\}$ and $V_n = \{v'_1, v'_2, \cdots v'_n\}$ denote the sets of vectors that correspond to $G_m = \{g_1, g_2, \cdots, g_m\}$ and $G_n = \{g'_1, g'_2, \ldots g'_n\}$, respectively. In this paper, we use the idea of DTW to calculate the similarity between two sets of vectors, which is denoted as DTW distance. The smaller the value, the more similar the two proteins. The GO embedding of the two proteins' annotations is concatenated as V_m and V_n, and the lengths are m and n, respectively ($m \neq n$). For constructing the matrix $D_{m \times n}$, the element $D(v_m, v'_n)$ represents the distance between points v_m and v'_n and can be expressed as follows:

$$D(v_m, v'_n) = \min \begin{cases} D(v_{m-1}, v'_n) = Dist(v_{m-1}, v'_n) + d(v_m, v'_n) \\ D(v_m, v'_{n-1}) = Dist(v_m, v'_{n-1}) + d(v_m, v'_n) \\ D(v_{m-1}, v'_{n-1}) = Dist(v_{m-1}, v'_{n-1}) + 2d(v_m, v'_n) \end{cases} \tag{8}$$



A

	PPI	SDNE(cosine)	Node2vec(cosine)	DeepWalk(cosine)	LINE(cosine)
PPI	1	0.958	0.255	0.277	0.975
SDNE(cosine)	0.958	1	0.991	0.9925	0.986
Node2vec(cosine)	0.255	0.991	1	0.665	0.084
DeepWalk(cosine)	0.277	0.9925	0.665	1	0.079
LINE(cosine)	0.975	0.986	0.084	0.079	1

B

	PPI	SDNE(DTW)	Node2vec(DTW)	DeepWalk(DTW)	LINE(DTW)
PPI	1	0.92	0.951	0.948	0.948
SDNE(DTW)	0.92	1	0.977	0.978	0.978
Node2vec(DTW)	0.951	0.977	1	0.997	0.998
DeepWalk(DTW)	0.948	0.978	0.997	1	0.99
LINE(DTW)	0.948	0.978	0.998	0.99	1

FIGURE 3 | Human protein similarity network ($\tau > 0.4$) and PPI coincidence degree. **(A)** Cosine, **(B)** DTW.

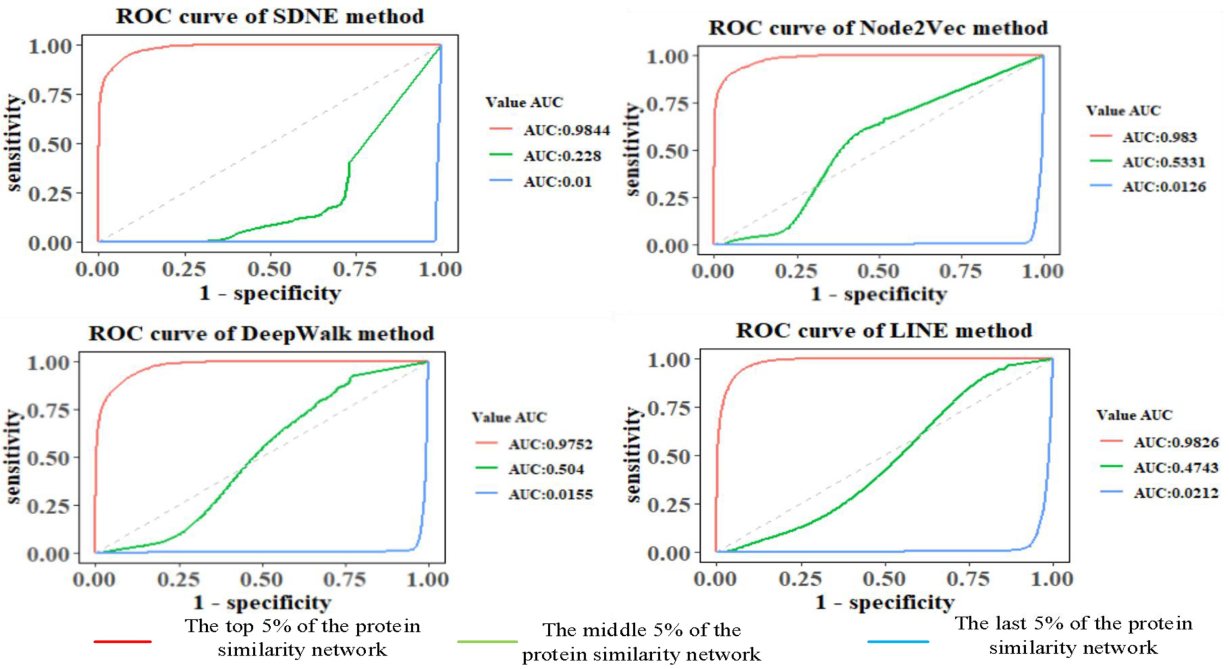


FIGURE 4 | Comparison of prediction results of Human protein similarity networks.

We used the DTW distance method to find a path W through several lattice points in the matrix. The shortest path is the distance between the set of vectors $V_m = \{v_1, v_2, \ldots v_m\}$ and $V_n = \left\{v_1', v_2', \ldots v_n'\right\}$. We then calculated the distance used to measure the similarity between the two proteins. The process for calculating the DTW distance is presented in **Supplementary Figure 1**.

For any protein P_i, the low-dimensional feature ω_i is directly learned from the GOA graph, which contains the information of term–term and term–protein relations. We use the cosine distance of the proteins' vector ω to measure the similarity of the proteins. Cosine distance can be expressed as follows:

$$D(P_m, P_n) = \text{cosine}(\omega_m, \omega_n) = \frac{\omega_m \cdot \omega_n}{||\ \omega_m||\ ||\omega_n||} \tag{9}$$

Link Prediction and Evaluation Metrics

When it is difficult to use a unified standard to measure the advantages and disadvantages of a network model, link prediction can be used as a unified comparison method for the similarity nodes in the network. It provides a standard to measure the reliability of the structure of the network. In the comprehensive evaluation, we use two commonly used evaluation indicators, AUC (Lobo, 2010) and AUCPR (Yu and Park, 2014), widely used in dichotomy. Therefore, to evaluate the available networks constructed based on different graph embedding methods in the GO graph and GOA graph, we perform link prediction experiments on the protein similarity network and evaluate the accuracy of the prediction results. For any undirected network $G(V, E)$, we let E be the complete set of $C^2_{|V|}$ node pairs. We first remove 20% of the existing edges E_r in the network. The remaining 80% of the edges E_s are then divided into E_p and E_t, where $E_\mathrm{s} = E_\mathrm{p} \cup E_t$, $E_P \cap E_\mathrm{t}$=∅, and $E = E_r \cup E_s$. Given a link prediction method, each pair of unconnected node pairs v_x and v_y is given a link probability of two nodes. Sorting all the node pairs according to the score value in descending order, we have the top node pair with the highest link probability. The calculation process of the AUC value is presented in **Supplementary Figure 2**. The value of AUCPR is affected by the precision and recall value. For a link prediction experiment, accuracy is defined as the proportion of accurate prediction among the top L prediction edges. If m prediction edges exist, sort the link probability score value in descending order. If m of the top L edges are in the E_t, the precision is defined as follows:

$$Precision = \frac{m}{L} \tag{10}$$

The number of existing edges in the network $M = E - E_r$, where m is the number of edges predicted by the prediction algorithm. The recall index is defined as follows:

$$\text{Recall } = \frac{m}{M} \tag{11}$$

The similarity between nodes is an essential precondition for link prediction, and the more similar the two nodes are, the more likely that a link exists between them. The similarity of network-based structural information definition is called structural similarity. Link prediction accuracy based on structure similarity depends on whether the structure similarity can grasp target structure characteristics. In the link prediction task, there are many methods to calculate the structural similarity between nodes, such as the following:

Common neighbors index

Common Neighbors (CN) (Li et al., 2018) similarity can be called structural equivalence, that is, if two nodes have multiple common neighbors, they are similar. In the link prediction experiment, CN index basic assumption is that if two unconnected nodes have more common neighbors, they are more likely to be connected. For nodes v_x and v_y in the protein similarity network, their neighbors are defined as $\Gamma\ (x)$ and $\Gamma\ (y)$, and the similarity of the two nodes is defined as the number of their CN. The index of CN is defined as follows:

$$S_{xy} = \left|\Gamma(x) \cap \Gamma(y)\right| = (A^2) \tag{12}$$

where S represents the similarity matrix and A represents the adjacency matrix of the graph. CN index is based on local information similarity index.

Jaccard index

Based on the common neighbors and considering the influence of the node degree at both ends, the Jaccard (JC) similarity index (Ran et al., 2015) is proposed. JC not only considers the number of two nodes' common neighbors but also considers the number of all their neighbors. JC is defined as follows:

$$S_{xy} = \frac{\left|\Gamma(x) \cap \Gamma(y)\right|}{\left|\Gamma(x) \cup \Gamma(y)\right|} = \frac{(A^2)_{xy}}{\left|\left|\Gamma(x) \cap \Gamma(y)\right|\right|} \tag{13}$$

Resource allocation index

Resource Allocation (RA) (Dianati et al., 2005) index considers the attribute information of the common neighbors of two nodes. In the link prediction process, the common neighbor nodes with higher degrees play a lesser role than those with lower degrees, and the weight of the common neighbor nodes decreases in the form of $^1/_k$. An example is presented in **Supplementary Figure 3**. RA index (Dianati et al., 2005) is defined as follows:

$$S_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{K_\mathrm{z}} \tag{14}$$

where K_z is the degree of the common neighbors of nodes v_x and v_y. The calculation process of the RA similarity index is shown in **Supplementary Figure 3**. Assuming that each node's resources are distributed equally to its neighbors, the RA index calculates a node's received resources, which is the similarity between nodes v_x and v_y .

RESULTS

Comparison of Protein Similarity and the Actual PPI Network Coincidence Degree

We downloaded the human yeast protein interaction network from the String database. We then mapped the proteins to the UniProt database, filtered out those proteins that could not be found in the UniProt database, and removed duplicate edges. After filtering, the Yeast dataset consisted of 2,877 proteins with 228,468 interactions, and the Human dataset consisted of 6,882 proteins with 892,054 interactions. Finally, to verify the validity of our calculated protein similarity network, we compared protein similarity and the actual PPI network coincidence degree.

This paper only shows the Human dataset experiment results in **Figure 3**, and the Yeast dataset results are shown in **Supplementary Figures 4**, **5**.

We selected the protein similarity networks ($\tau > 0.4$) and compared them with the PPI dataset downloaded from the

TABLE 2 | AUCPR value of protein similarity prediction in the Human dataset.

Method	The top 5% of the network	The middle 5% of the network	The last 5% of the network
SDNE	0.9105	0.0076	0.0052
Node2vec	**0.9115**	**0.0143**	**0.0055**
DeepWalk	0.8220	0.0127	0.0052
LINE	0.7117	0.0097	0.0052

Bold means the best result in the comparative experiment.

String database to analyze the coincidence degree of the Human and Yeast protein networks. Furthermore, we compared the edge coincidence of the protein similarity network based on different graph embedding methods (as shown in **Figure 3**). The calculation was based on $\frac{E_a \cap E_b}{E_a} (E_a > E_b)$.

By comparing the GO(DTW) and GOA(cosine) methods, it can be seen that the Node2vec graph embedding method performed best in the GO graph. SDNE and LINE methods performed better in the GOA graph, and there was little difference between them in the GOA graph and GO graph. However, Node2vec and DeepWalk performed better in the GO graph. In general, the performance of protein similarity calculation based on different graph embedding methods in the GO graph was better than in the GOA graph. As shown, using graph embedding methods can be effective in calculating protein similarity in GO and GOA graphs. We also proved that using the DTW method to calculate different dimensional protein vector similarities is feasible.

Comparison of Link Prediction Results Based on Different Graph Embedding Methods in GO Graph

The features of GO terms are learned from the GO graph based on different graph embedding methods, and the similarity among proteins is calculated. By selecting the top 5%, middle 5%, and the last 5% of the protein similarity network data, the link prediction is computed for the filtered protein similarity network, and the AUC and AUCPR values are calculated (as shown in **Figure 4** and **Table 2**). This paper only shows the Human dataset experiment result, and the Yeast dataset result is shown in **Supplementary Figure 6** and **Supplementary Table 1**.

We can see that as the similarity of network nodes decreases, the value of AUC decreases. In the top 5% of the protein similarity network, the proteins are more similar, but for AUCPR values, we can see that the performance of the Node2vec method is the best in all the top, middle, and the last 5% of the protein similarity networks. The Node2vec method introduces BFS and DFS into the generation process of the random walk sequence by introducing two parameters p and q. BFS focuses on the adjacent nodes and characterizes a relatively local graph representation; that is, the BFS can explore the local structural properties of the graph, while the DFS can explore the global similarity in context. We found that the AUC value of protein similarity calculated by the graph embedding method decreased gradually with the decrease in the value of the screening protein similarity. Furthermore, it is shown that the edge connection of the protein similarity network calculated by the graph embedding method is reliable.

We also found that the Node2vec graph embedding method performed well in calculating the Yeast protein similarity network (as shown in **Supplementary Figure 6** and **Supplementary Table 1**). Therefore, the GO term vectors fused the local and global information of nodes in the GO graph and contain more information, so the GO(DTW) method performs better in computing protein similarity.

Comparison of Link Prediction Results Based on Different Graph Embedding Methods in the GOA Graph

To reflect the influence of the structure information of the GO annotation on proteins, the features of proteins are learned from the GOA graph based on different graph embedding methods, and the similarity among proteins is calculated (as shown in **Figure 5** and **Table 3**). This paper only shows the Human dataset experiment result, and the Yeast dataset result is presented in **Supplementary Figure 7** and **Supplementary Table 2**.

We screened the top, middle, and last 5% of the protein similar networks and performed the link prediction experiments to observe the values of AUC and AUCPR under different methods. The AUC and AUCPR values decreased gradually with the decrease in the percentage selected. Therefore, it can be seen that the performance of the Node2vec method in the GOA(cosine) method is also better than other graph embedding methods. For the Yeast protein similarity network, we also performed the same experiment and obtained the same experimental conclusions as described above. We found that SDNE graph embedding methods also showed excellent performance in the Yeast dataset (as shown in **Supplementary Table 2**). This is because the SDNE method also defines first-order and second-order similarities. Therefore, calculating the protein similarity network based on these vectors achieved excellent results in the prediction task.

Comparison of Link Prediction Results of Protein Similarity Calculated by IC-Based Method and Based on Graph Embedding Methods

We studied the application of different graph embedding methods to calculate protein similarity in GO and GOA graphs. We screened the top 5% of the protein similarity networks for link prediction analysis (as shown in **Table 4**). Furthermore, we performed an experiment that calculated the density of the protein similarity network based on graph embedding and IC-based methods (as shown in **Table 5**). This paper only presents the Human dataset experiment results, and the Yeast dataset result is presented in **Supplementary Tables 3**, **4**.

The link prediction results from these methods are compared as follows. From **Table 4**, it can be seen that the similarity calculation of proteins based on different graph embedding methods is superior to that of the IC-based methods. We also performed the above experiment for Yeast datasets, and the same conclusion was obtained (as shown in **Supplementary Table 3**). It can be seen that the SDNE and Node2vec graph embedding methods show good performance in the GO graph. Analyzing the

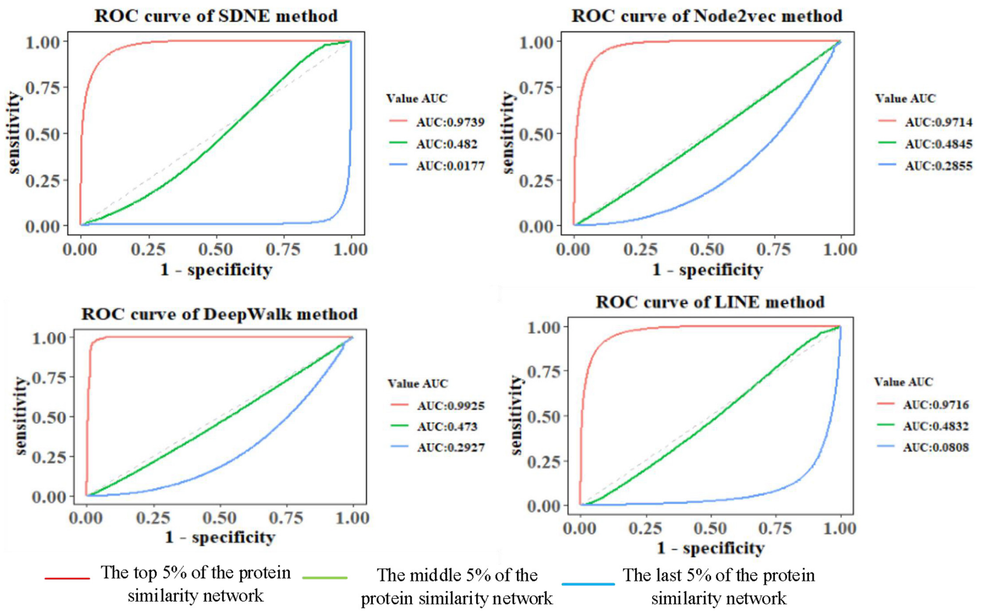


FIGURE 5 | Comparison of prediction results of Human protein similarity networks.

density of the top 5% of the human protein similarity networks, it can be seen that the density of the protein similarity network calculated by the graph embedding method is higher than that calculated by IC-based methods. Therefore, it is shown that the protein similarity network calculated by the IC-based method is sparse, and the similarity of proteins is not as high as that calculated by the graph embedding method. Thus, in the IC-based method, the AUCPR value obtained in link prediction is lower. We also verified this conclusion on the Yeast dataset (as shown in **Supplementary Table 4**).

Based on different graph embedding methods, the features of the GO terms were learned into the vector representations through fusing the topology of the GO graph. Thus, we could capture the global information based on the graph embedding method, and its learned vectors could calculate the similarity between proteins by the DTW distance similarity. As can be seen from the results of the link prediction, the GO(DTW) method performed better than GOA(cosine), and most of the protein similarity networks calculated by the GO(DTW) method are denser than those calculated by the GOA(cosine) method.

TABLE 3 | AUCPR value of Human protein similarity prediction.

Method	The top 5% of the network	The middle 5% of the network	The last 5% of the network
SDNE	0.6578	0.0100	0.0052
Node2vec	**0.8758**	**0.0105**	**0.0069**
DeepWalk	0.8719	0.0094	0.0053
LINE	0.8189	0.0095	0.0053

Bold means the best result in the comparative experiment.

TABLE 4 | AUCPR and AUC values of Human protein similarity prediction (the top 5% of the similarity network).

Method	AUC	AUCPR
SDNE (cosine/DTW)	0.9699/**0.9739**	0.9015/**0.9105**
Node2vec (cosine/DTW)	0.9714/**0.983**	0.8758/**0.9115**
DeepWalk (cosine/DTW)	**0.9925**/0.9752	**0.8719**/0.8220
LINE (cosine/DTW)	**0.9839**/0.9716	**0.8189**/0.7117
Rel.	0.9067	0.1519
Jiang and Conrath	0.8409	0.0669

Bold means the best result in the comparative experiment.

TABLE 5 | Comparison of Human protein similarity network density between different methods.

Method	Nodes	Edges	Density
SDNE (cosine/DTW)	4,797/2,024	1,183,801/713,961	0.1/**0.3**
Node2vec (cosine/DTW)	6,882/2,807	2,841,303/1,183,762	0.12/**0.3**
DeepWalk (cosine/DTW)	6,882/3,079	1,183,876/1,183,707	0.05/**0.2**
LINE (cosine/DTW)	5,586/1,660	1,183,815/206,650	0.07/**0.15**
Rel	5,902	870,987	0.05
Jiang and Conrath	5,883	870,986	0.05

Bold means the best result in the comparative experiment.

TABLE 6 | Prediction results under different similarity indexes (the top 5% of the Human protein similarity network).

Similarity index	CN	JC	RA
SDNE (cosine/DTW)	0.9694/0.981	0.9739/0.9843	**0.9818**/**0.9886**
Node2vec (cosine/DTW)	0.9598/0.9809	0.9714/0.9843	**0.9856**/**0.9886**
DeepWalk (cosine/DTW)	0.9772/0.981	0.9856/0.9842	**0.9885**/**0.9884**
LINE (cosine/DTW)	0.9703/0.9716	0.9716/0.9825	**0.9874**/**0.9853**

Bold means the best result in the comparative experiment.

Similarity Indexes' Results

We performed three different link prediction similarity index experiments on the top 5% of the protein similarity network and found that based on different similarity indexes, the difference in the AUC value is small, which indicates that the calculated protein similarity network structure has improved (as shown in **Table 6**). This paper only presents the Human dataset experiment result, and the Yeast dataset result is presented in **Supplementary Table 5**.

Among the three different similarity evaluation indexes, we found that the AUC value of the RA similarity index based on link prediction is slightly higher than the other two similarity indexes. Furthermore, the results showed that the top 5% of the protein similarity network had higher AUC values in different similarity indexes of link prediction, indicating that the graph embedding method effectively calculated protein similarity. We obtained the same conclusion in the experiment with the Yeast dataset (as shown in **Supplementary Table 5**).

DISCUSSION

Gene Ontology is one of the many biological ontology languages. Its emergence and development reduce the confusion of biological concepts and terms, provide a three-layer (BP, MF, and CC) structure of system definition, and describe the functions of proteins. Therefore, it is important to understand protein function based on GO terms to describe protein similarity.

In this paper, by fusing the GO terms' topology information, we learned the features of GO terms and proteins into vector representations in GO and GOA graph based on different graph embedding methods. Then, the similarity of proteins was calculated based on these vectors using DTW and cosine similarity. Finally, protein similarity networks were screened by selecting different percentages, and a link prediction experiment was used to evaluate the prediction accuracy of different networks. The experimental results indicate that the graph embedding method is better than the IC-based method in protein similarity calculation. Among the two graph embedding methods, the performance of the GO(DTW) method is better than that of the GOA(cosine) method. This is because the GO terms and proteins are treated equally in the GOA graph, and some information may be ignored when learning protein low-dimensional embedding. Therefore, the coincidence degree between the protein similarity network calculated by the GOA(cosine) method and the actual PPI data is not as high as that calculated by the GO(DTW) method. There are potential limitations to our method. First, we transformed directed graphs into undirected graphs, which might result in a loss of structural information. We also treated the GO terms and the proteins equally in the GOA graph, which may ignore some information. Therefore, in our future study, we plan to learn the protein representations in the graph by combining the information in the directed graph and by considering representation learning of heterogeneous graphs that contain GO terms and proteins.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

YZ conceived the idea and prepared the experimental data. ZW and YZ debugged the code, conducted the experiments, interpreted the results, and wrote and edited the manuscript. SW and JS advised the study and reviewed the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61902430, 61873281, and 61972226).

ACKNOWLEDGMENTS

We would like to thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.744334/full#supplementary-material

REFERENCES

Amos, B., and Brigitte, B. (1999). The SWISS-PROT protein sequence data bank. *Nucleic Acids Res.* 22, 49–54. doi: 10.1093/nar/22.17.3626

Dianati, M., Shen, X., and Naik, S. (2005). "A new fairness index for radio resource allocation in wireless networks," in *Proceedings of the Wireless Communications & Networking Conference* (New Orleans, LA: IEEE), 785–890.

Goldberg, Y., and Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *OALib J.* 14, 144–156. doi: 10.1017/S1351324916000334

Grover, A., and Leskovec, J. (2016). "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference*, (New York, NY: ACM), 855–864.

Harris, M. A. (2004). The gene ontology (GO) database and informatics resource. *Nucleic Acids Res.* 32, 258–261. doi: 10.1093/nar/gkh036

Hu, L., Wang, X., Huang, Y. A., Hu, P., and You, Z. H. (2021). A survey on computational models for predicting protein–protein interactions. *Bioinformatics.* 05, 77–85. doi: 10.1093/bib/bbab036

Jiang, J. J., and Conrath, D. W. (1997). "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proceedings of the 10th Research on Computational Linguistics International Conference*, Vol. 11, (Taipei: The

Association for Computational Linguistics and Chinese Language Processing (ACLCLP)), 115–123. doi: 10.1.1.269.3598

Li, S., Huang, J., Zhang, Z., Liu, J., Huang, T., and Chen, H. (2018). Similarity-based future common neighbors model for link prediction in complex networks. *Sci. Rep.* 19, 518–524. doi: 10.1038/s41598-018-35423-2

Lobo, J. M. (2010). AUC: a misleading measure of the performance of predictive distribution models. *Glob. Ecol.* 17, 145–151. doi: 10.1111/j.1466-8238.2007.00358

Lord, P. W., Stevens, R. D., Brass, A., and Goble, C. A. (2003). Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics.* 19, 1275–1283. doi: 10.1093/bioinformatics/btg153

Lou, Y., Ao, H., and Dong, Y. (2016). "Improvement of dynamic time warping (DTW) Algorithm," in *Proceedings of the 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, 14 (Guiyang: IEEE), 18–24. doi: 10.1109/DCABES.2015.103

Paul, P., and Meeta, M. (2008). Gene Ontology term overlap as a measure of gene functional similarity. *BMC Bioinformatics.* 9:327. doi: 10.1186/1471-2105-9-327

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). "DeepWalk: online learning of social representations," in *Proceedings of the 2014 ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, (New York, NY: ACM), 701–740.

Pesaranghader, A., Rezaei, A., and Davoodi, D. (2014). Gene functional similarity analysis by definition-based semantic similarity measurement of GO terms. *Lecture Notes Bioinformatics.* 12, 203–214. doi: 10.1007/978-3-319-06483-3_18

Ran, S., Ngan, K. N., and Li, S. (2015). "Jaccard index compensation for object segmentation evaluation," in *Proceedings of the 2014 IEEE International Conference on Image Processing*, (Paris: IEEE), 253–259.

Resnik, P. (1999). Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res.* 11, 95–130. doi: 10.1613/jair.514

Sevilla, J. L., Segura, V., Podhorski, A., Guruceaga, E., Mato, J., Martínez-Cruz, L. A., et al. (2005). Correlation between gene expression and GO semantic similarity. *IEEE/ACM Trans. Comput. Biol. Bioinformatics.* 24, 330–338. doi: 10.1109/TCBB.2005.50

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). "LINE: large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, (New York, NY: ACM), 1067–1077.

UniProt Consortium (2015). UniProt: a hub for protein information. *Nucleic Acids Res.* 32, 115–119. doi: 10.1093/nar/gkh131

Wang, D., Peng, C., and Zhu, W. (2016). "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD International Conference on Data Mining*, (New York, NY: ACM), 1225–1234.

Wang, Z., Zhang, Y., Wang, S., and Shang, J. (2020). SINE: second-order information network embedding. *IEEE Access* 1, 98–110. doi: 10.1109/ACCESS.2020.3007886

Xi, J., Li, A., and Wang, M. (2020a). HetRCNA: a novel method to identify recurrent copy number alternations from heterogeneous tumor samples based on matrix decomposition framework. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 17, 422–434. doi: 10.1109/TCBB.2018.2846599

Xi, J., Ye, L., and Huang, Q. (2021). "Tolerating data missing in breast cancer diagnosis from clinical ultrasound reports via knowledge graph inference," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*, (New York, NY: Association for Computing Machinery), 1–9. doi: 10.1145/3447548.3467106

Xi, J., Yuan, X., Wang, M., Li, A., and Huang, Q. (2020b). Inferring subgroup-specific driver genes from heterogeneous cancer samples via subspace learning with subgroup indication. *Bioinformatics* 36, 1855–1863.

Yu, W., and Park, T. (2014). AucPR: an AUC-based approach using penalized regression for disease prediction with high-dimensional omics data. *BMC Genomics.* 15:S1. doi: 10.1186/1471-2164-15-S10-S1

Zhong, X., Kaalia, R., and Rajapakse, J. C. (2019). GO2Vec: transforming GO terms and proteins to vector representations via graph embeddings. *BMC Genomics* 20:918. doi: 10.1186/s12864-019-6272-2

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Zhang, Wang, Wang and Shang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.