

# DeepGOA: Predicting Gene Ontology Annotations of Proteins via Graph Convolutional Network

Guangjie Zhou<sup>1</sup>, Jun Wang<sup>1</sup>, Xiangliang Zhang<sup>2</sup>, Guoxian Yu<sup>1,\*</sup>

<sup>1</sup>College of Computer and Information Sciences, Southwest University, China

<sup>2</sup>CEMSE, King Abdullah University of Science and Technology, Thuwal, SA.

Email: zgj524662639@email.swu.edu.cn, kingjun@swu.edu.cn, xiangliang.zhang@kaust.edu.sa, gxyu@swu.edu.cn

**Abstract**—Gene Ontology (GO) uses a series of standardized and controlled GO terms to describe the molecular functions, biological process roles and cellular locations of gene products (i.e., proteins and RNAs), it structurally organizes GO terms in a direct acyclic graph (DAG). GO has more than 40000 terms and each protein is only annotated with several or dozens of these terms. It is a difficult challenge to accurately annotate relevant GO terms to a protein from such a large number of candidate GO terms. Some deep learning models have been proposed to utilize the GO hierarchy for protein function prediction, but they inadequately utilize GO hierarchy. To use the knowledge encoded in the GO hierarchy, we propose a deep Graph Convolutional Network (GCN) based model (DeepGOA) to predict GO annotations of proteins. DeepGOA firstly utilizes GO annotations and hierarchy to measure the correlations between GO terms and to accordingly update the edge weights of the DAG, and then applies GCN on the updated DAG to learn the semantic representation and latent inter-relations of GO terms. At the same time, it uses Convolutional Neural Network (CNN) to learn the feature representation of amino acids sequences with respect to the semantic representations. After that, DeepGOA computes the dot product of two representations, which enables training the whole network end-to-end in a coherent fashion. Experiments on two model species (Human and Corn) show that DeepGOA outperforms the state-of-the-art deep learning based methods. The ablation study proves that GCN can employ the knowledge of GO and boost the performance.

**Index Terms**—Gene Ontology, GO terms, Protein function prediction, Graph Convolutional Network, Convolutional Neural Network

## I. INTRODUCTION

Protein is the main executor of life activities, such as the structures of biological tissues, the catalysis of biochemical reactions required for metabolism, the maintenance of cell environment, the recognition and transmission of biological signals [1], [2]. With the widespread use of high-throughput biotechnology, the collected proteomic data (i.e., amino acid sequences and interaction networks) have grown rapidly. Accurately annotating the functions of protein can greatly promote the research and development of biofuel, pathology analysis, drug development, and many others [1], [2]. However, due to the bias of biologists' research interests, only a very tiny part of newly obtained sequences have experimentally validated GO annotations [3]–[5]. The speed of newly collected sequences far exceeds the capability of

annotating protein functions by wet experiments. As such, automated function prediction (AFP) has become increasingly important [1], [6].

Gene Ontology (GO) is the major bioinformatics initiative to unify the representation of gene and gene product attributes across all species [7]. GO has been extensively used as a golden standard to annotate the function of gene products [8]. GO include plenty of GO terms and each GO term describes a distinct biological concept [9]. Till now, GO contains over 40000 terms, covering three different sub-ontologies, Molecular Function Ontology (MFO), Biological Process Ontology (BPO) and Cellular Component Ontology (CCO). GO structurally organizes GO terms in a direct acyclic graph (DAG). In the DAG, each node corresponds to a GO term and the edge describes the relationship between terms. Figure 1 gives an example of GO annotations of protein 'AOAVI2'. GO annotations follow a principle termed as True Path Rule [7], [10]: a child term is a further refinement of the function of its parental node. If a protein is annotated with a term, then the protein is also annotated with its ancestor (if any) terms. On the other hand, if a protein is not annotated with a GO term, the protein is not annotated with any of its descendant terms.

A protein usually participates in different life processes, executes multiple biological functions, and is annotated with multiple GO terms at the same time. Therefore, protein function prediction can be regarded as a multi-label learning problem [11]–[13]. However, due to a large amount of un-validated GO annotation of proteins, existing multi-label learning based function prediction methods face the issue of insufficient annotations and massive candidate GO terms. Furthermore, deep terms in the GO DAG describe more refined biological functions, and the shallow terms describe the broad functions, but the missing GO annotations of proteins usually correspond to deep terms, which makes accurately predicting the GO annotations of proteins more difficult than traditional multi-label learning. Some efforts have been made toward using the knowledge of GO. To name a few, Valentini [10] used the GO hierarchy to adjust the predictions made by binary classifier for each GO term. Pandey *et al.* [14] firstly defined a taxonomic similarity to hierarchically measure the correlations between GO terms, and then improved the prediction of deep

GO terms via the correlation. Yu *et al.* [15] applied downward random walks (dRW) on the GO hierarchy and took the terms already annotated to the protein as the initial walkers to predict new GO annotation of this protein, and to identify the negative GO annotations of this proteins [16]. Yu *et al.* [17] introduced a hybrid graph, composed of two types of nodes (proteins and GO terms), to encode interactions between proteins, GO hierarchy and available annotations of proteins, and then proposed a bi-random walk algorithm on the hybrid graph to predict GO annotations of proteins. Recently, Zhao *et al.* [18], [19] tried to compress massive GO terms by hashing and then achieved protein function prediction in the compressed hashing space and shows promising results.

All the above methods can be regarded as shallow solutions, and they are difficult to mine the deep (non-linear) relationship between proteins and GO terms. In recent years, deep learning has significantly sparked the development of image recognition and speech recognition [20]. When applying the deep learning model to protein function prediction, the challenge is its large and complex output space. Wehrmann *et al.* [21] proposed the Hierarchical Multi-Label Classification with Chained Neural Networks (HMC-CNN) model by establishing a series of fully connected neural networks for the GO terms of different levels in the GO hierarchy. They took each fully connected neural network as a classifier to predict a certain number of GO terms, respectively. Due to the imbalance between GO terms on the same level, which will impact the prediction accuracy, Zilke *et al.* [22] grouped GO terms based on the level of GO and the number of annotations, then established a series of fully connected neural network for each group. These two deep learning based approaches separate GO terms, they can not well respect the connection between GO terms that are not in the same group. Kulmanov *et al.* [23] first used convolution kernels to encode amino acids and incorporated the GO structure into the output layer. For each GO term, they generated one fully connected layer with a sigmoid activation function, which predicts whether the protein should be annotated with this GO term and all non-leaf nodes in the graph using a maximum merge layers that outputs the maximum value of the classification results for all child nodes and the internal nodes. Kulmanov *et al.* [24] further removed the maximum merge layers in the previous work to obtain better prediction accuracy. These aforementioned deep models optimistically assume that their models are suitable for multiple GO terms. But in fact, they do not well utilize the hierarchical relationship between GO terms and still suffer from the gap between amino acids and GO annotations, which is often similarly termed as the semantic gap in image classification [25].

In this paper, we present a novel method called DeepGOA to learn the knowledge of Gene Ontology and to reduce the semantic gaps between amino acids and Gene Ontology and the annotations. DeepGOA separately uses the Convolutional Neural Network (CNN) to extract the feature vectors of amino acids, and the Graph Convolution Network (GCN) [26] to learn the semantic representation of GO terms by referring to GO hierarchy and known annotations related with these GO terms.

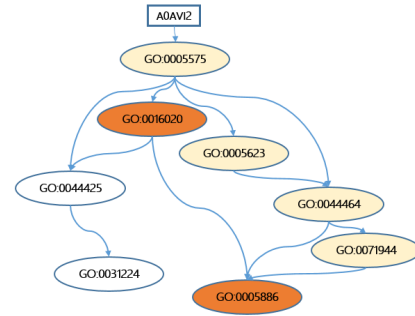


Fig. 1. An example of hierarchical GO annotations of proteins. ‘A0AVI2’ is a Human protein, it is annotated with ‘GO:0005886’ and ‘GO:0016020’. According to the True Path Rule, the protein ‘A0AVI2’ is also annotated with their ancestor terms (‘GO:0071944’, ‘GO:0044464’, ‘GO:0005623’ and ‘GO:0005575’).

After that, DeepGOA utilizes a multi-layer neural network to learn the mapping from sequence features to the semantic space of GO terms by the dot product, and reversely guide the mapping by the known GO annotations of proteins. We conduct experiments on two model species (Human and Corn) and observe that DeepGOA outperforms existing state-of-the-art methods [23], [24], [27]. In addition, DeepGOA retains more GO structure information.

This paper is organized as follows. In Section II we introduce the DeepGOA learning framework for predicting protein function. Section III presents and discusses the results of DeepGOA and comparing methods. Section IV gives the conclusion and future work.

## II. MATERIALS AND METHOD

Effectively using GO hierarchy and known annotation is important for protein function prediction [8], [9], [18], [19]. Deep learning has been widely used in protein function prediction [21]–[23]. However, how to properly use the knowledge of GO in the deep model has not been well resolved. Most deep models simply try to learn the mapping of protein sequences to GO terms directly, without respecting to the GO hierarchy when optimizing the mapping. Different from these methods, DeepGOA firstly learns the semantic representation of Gene Ontology via GCN and simultaneously optimizes the representation of protein sequence through CNN. After that, DeepGOA computes the dot product of two sub-nets to learn the mapping from feature representation to semantic representation in an end-to-end style. At the same time, it uses the collected annotations of proteins and back propagation to refine the mapping coefficients, and to achieve coherent representations. Figure 2 illustrates the basic architecture of our model.

### A. Datasets

For experiments, we downloaded the latest Gene Ontology data (June 2019) from GO official site<sup>1</sup>. GO data includes three branches [7], in which there are 4169 terms in CCO;

<sup>1</sup><http://geneontology.org/page/download-ontology>

29,462 terms in BPO; 11155 terms in MFO. We collected the reviewed and manually annotated protein sequences with GO annotations of Human from SwissProt<sup>2</sup>. The human protein data contains 20,431 proteins. We collected 18,533 protein sequences and GO annotations of Corn from Phytozome<sup>3</sup>.

We trained a model for each subontology in GO. Particularly, we ranked GO terms by their number of annotations and selected terms with the minimum number of annotations 25, 150 and 25 for CCO, BPO and MFO, respectively. The adopted cutoff values are only half of those used by DeepGO [23], and thus our datasets include much more GO terms. After that, we applied the True Path Rule to expand the direct annotations. If a protein is annotated with a GO term, then the protein is also annotated with its all ancestor terms. We create a binary label vector for each amino acid sequence. If a sequence is annotated with a GO term from our list of selected terms, then we assign 1 to the term position in the binary vector and use it as a positive sample for this term. Otherwise, we assign 0 and use it as a negative sample. In our model training process, we exclude proteins not annotated by any of the selected GO terms.

### B. Extracting features from Amino Acids via CNN

Since the computer cannot directly process the amino acid sequences and different proteins have different number of amino acids, we need to numerically encode each protein sequence. Kulmanov *et al.* [24] confirmed that using one hot encoding in deep networks can achieve a good predictive effect. Therefore, the input of our model is the one-hot encoding of amino acids. Each amino acid can be represented via a one-hot encoding vector of length of 21, since there are twenty types of amino acids and some amino acid sequences have undetermined amino acids at certain positions, which we specifically use an additional one-hot bit to represent them. We treat each amino acid as a one-hot encoding and using a combination of one-hot vectors to represent the protein sequence. It is indicated that a protein of more than 2000 amino acids takes the first 2000 amino acids, and a protein of less than 2000 amino acids is zero-padded to obtain the final feature vector. We finally got the feature vector with size  $2000 \times 21$ . Each protein sequence can be presented by a matrix:

$$\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{i2000}] \quad (1)$$

where  $\mathbf{X}_i \in \mathbb{R}^{2000 \times 21}$  represents the  $i$ -th protein in the data set,  $x_{ij}$  is the one-hot encoding of the  $j$ -th amino acid of the  $i$ -th protein. In this paper,  $n$  represents the number of proteins in the training set,  $\mathcal{T}$  represents the set of studied GO terms,  $|\mathcal{T}|$  counts the number of selected GO terms.

For each protein sequence feature vector, we use CNN to learn its low-dimensional representation. CNN is a kind of feedforward neural network with convolutional computation and deep structure. It is a representative deep learning model and has very strong ability in extracting feature when dealing with

fixed-size data. The convolution portion takes  $\mathbf{X}$  as input and extracts protein sequence features using a series of differently sized 1D convolution kernels. The convolution kernel is  $\mathbf{w} \in \mathbb{R}^{21 \times h}$  and  $h$  is the sliding window length. The convolution operation is operated as follows:

$$\mathbf{c}_{im} = f(\mathbf{w} * \mathbf{x}_{im:m+h}), m \in [1, k-h] \quad (2)$$

where  $*$  is a convolution operation,  $f(\cdot)$  is a non-linear operation,  $\mathbf{w}$  is a convolution kernel,  $\mathbf{x}$  is our model input vector,  $k$  is the input feature vector length. The new feature vector of  $\mathbf{c}_i$  is defined as:

$$\mathbf{c}_i = [\mathbf{c}_{i1}, \mathbf{c}_{i2}, \dots, \mathbf{c}_{ip}] \quad (3)$$

where  $p = k - h + 1$ . To this end, we learn the feature representation of each protein.

### C. Graph Convolutional Network

Incorporating the GO structure into the deep model is a very challenging problem, but can boost the performance of protein function prediction. Graph Convolutional Network (GCN) [26] can learn the node representation of a graph (or network) using the graph structure. The core idea of GCN is to learn the representation of GO terms by propagating information between GO terms using the connections between GO terms. Unlike standard convolution for fixed-size input operations, GCN takes the feature descriptions  $H^0 \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$  with one-hot coding and the corresponding correlation matrix  $A \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$  of GO terms as input, and updates the representation  $H^l \in \mathbb{R}^{|\mathcal{T}| \times d_l}$  of  $|\mathcal{T}|$  GO terms. The GCN layer operation is defined as follows:

$$H^{l+1} = f(\hat{A}H^lW^l) \quad (4)$$

where  $\hat{A} \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$  is the normalized version of the correlation matrix  $A$ , which will be given later.  $W^l \in \mathbb{R}^{d_l \times d_{l+1}}$  is a transformation matrix to be learned, and  $f(\cdot)$  is a non-linear operation. We can learn the information on the GO DAG by stacking the GCN layers.

One straightforward solution to estimate the correlation between GO terms is to count the frequency that two terms annotated to the same proteins, which has been widely adopted in multi-label learning based protein function prediction [11]–[13]. However, due to the available annotations of proteins are rather imbalance and incompletely, this simple estimation can not well reflect the underlying correlation between GO terms. Furthermore, the GO hierarchy between GO terms is overlooked in the estimation process, this hierarchy is independent from the studied species but carry important guidance for accurate protein function. Given that, we combine the GO hierarchy and collected annotations of proteins to estimate the correlations between the parental term  $t$  and its child term  $s$  as follows

$$A(t, s) = \frac{n_s}{n_t} + \frac{IC(s)}{\sum_{s' \in ch(t)} IC(s')} \quad (5)$$

where  $ch(t)$  is an aggregate of all direct child terms of  $t$ ,  $n_s$  and  $n_t$  represent the number of proteins annotated with term  $s$

<sup>2</sup><http://www.uniprot.org/uniprot/>

<sup>3</sup><https://phytozome.jgi.doe.gov/pz/portal.html>

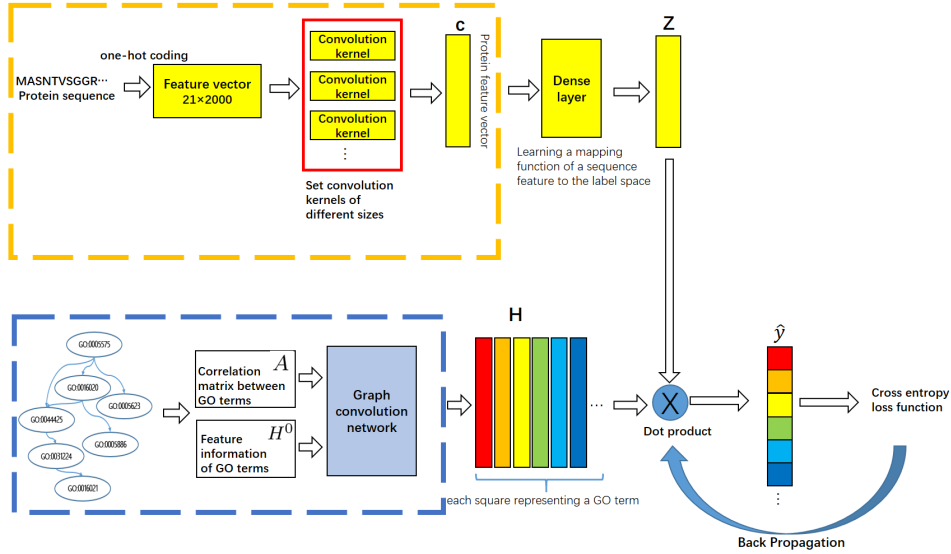


Fig. 2. The network architecture of DeepGOA. The upper yellow subnetwork is the convolutional network part. The amino acids are extracted by convolution kernels of different sizes, and the fully connected layer is used to learn the mapping from sequence features to semantic representations of GO terms. The lower blue subnetwork is the graph convolution part, it uses the GO hierarchy  $H^0 \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$  and empirical correlations between GO terms stored in  $A \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$  to learn the semantic representation of each GO term. The dot product is finally used to guide the mapping between proteins and GO terms and to reversely adjust the representations of proteins and GO terms. In this way, the associations between GO terms and proteins are also predicted.

and  $t$ , respectively.  $IC(t)$  is the information content of  $t$  and it is measured as:

$$IC(t) = 1 - \frac{\log(1 + |desc(t)|)}{\log |\mathcal{T}|} \quad (6)$$

where  $desc(t)$  includes all the descendants of  $t$  and itself. This type of information content is widely adopted to measure the semantic similarity between GO terms [28], [29], [16]. Obviously, the bigger the  $desc(t)$  is, the smaller the information content  $t$  has, since  $t$  has a larger number of descendant terms, which convey more specific biological functions than  $t$ . This measurement is independent from known GO annotations of proteins. Therefore, it is less sensitive to the incomplete and sparse GO annotations of proteins. In this way, we can differentiate the edges between parental terms and child terms.

#### D. DeepGOA classifier learning

Till now, we can obtain the representation  $H \in \mathbb{R}^{|\mathcal{T}| \times d}$  for GO terms via the GCN, and the representation  $Z \in \mathbb{R}^{n \times d}$  of  $n$  protein sequences (after dense layer of  $C$  in Figure 2) in the  $d$ -dimensional semantic space encoded by  $H$ . Finally, we get the dot product of  $H$  and  $Z$  as the predicted association probabilities as follows:

$$\hat{Y} = HZ^T \quad (7)$$

Since it is a binary problem to predict the association between a GO term and a protein, and the semantic representation already encodes the latent relationships between GO terms, our multi-label loss function can be defined by cross-entropy as follows:

$$Loss = \sum_{s=1}^{|\mathcal{T}|} y_s \log(\sigma(\hat{y}_s)) + (1 - y_s) \log(1 - \sigma(\hat{y}_s)) \quad (8)$$

where  $\mathbf{y} \in \mathbb{R}^{|\mathcal{T}|}$  stores the truth annotations of a protein,  $y_s \in \{0, 1\}$  denotes whether GO term  $s$  annotated to the protein or not,  $\sigma(\cdot)$  is the sigmoid activation function.

By minimizing the above loss and back propagating the loss to the subnetwork of learning  $H$  and to the subnetwork of learning  $Z$ , we can achieve the optimization of  $H$  and  $Z$ , and protein function prediction in the semantic space in a coherent end-to-end fashion.

### III. RESULTS AND ANALYSIS

For experimental validation, we compare DeepGOA against Naive [1], [6], BLAST [27], DeepGO [23] and DeepGOPlus [24]. All the input parameters are kept the same as the author reported or optimized in the suggested range. For DeepGOPlus, we reduce the number of convolution kernels from 512 to 128, since DeepGOPlus has too many parameters to run in our experimental environment.

For a comprehensive evaluation, we use five widely-used evaluation metrics: AUC, AUPRC, PR50,  $F_{max}$  and  $S_{min}$  [30]. AUPRC (area under the precision-recall curve) and AUC (area under the receiver operator characteristics curve) are widely adopted for binary classification, here we compute the AUPRC and AUC for each term and then takes average AUPRC and AUC of all terms. AUPRC is more sensitive to class-imbalance than AUC. PR50 is the average precision of all GO terms when the recall rate equal to 50%.  $F_{max}$  is the overall maximum harmonic mean of precision and recall across all possible thresholds on the predicted protein-term association matrix  $\hat{Y}$ .  $S_{min}$  uses the information theoretic analogs of precision and recall based on the GO hierarchy to measure the minimum semantic distance between the predictions and ground-truths across all possible thresholds. The first three evaluation metrics

are term-centric ones and the last two are protein-centric ones. These metrics quantify the performance of protein function prediction from different perspectives, it is difficult for an approach consistently outperforming another one across all the metrics. We want to remark that unlike other evaluation metric, the smaller the value of  $S_{min}$ , the better the performance is.

#### A. Results of protein function prediction

To quantitatively study the performance of the proposed method and that of comparing algorithms, we conducted experiment on GO annotations and amino acids of Human and Corn. We firstly sorted GO terms in descending order based on the number of proteins annotated to the GO term. Then we selected the most frequent terms for our experiments. Particularly, we selected 1190, 661 and 540 GO terms in BPO, MFO and CCO for experiments on Human; and 117, 251 and 112 GO terms in BPO, MFO and CCO for experiments on Corn. In the following experiment, we randomly partition the proteins into a training set (80%) and a validation set (20%). Table I reveals the prediction results of DeepGOA and those of comparing methods in 10 rounds of independent partitions.

DeepGOA almost always achieves a better performance than these methods across the five evaluation metrics. The improvement of DeepGOA to other comparing methods with respect to AUPRC and PR50 is more prominent, which suggests the effectiveness of DeepGOA in handling the imbalanced GO terms, which is contributed by the semantic representation of GO terms. DeepGO uses the structure between parent-child terms at the final output layer, but it still loses to DeepGOA, which suggests our choice of GCN for GO hierarchy representation learning is more effective. DeepGOPlus does not use any structural information and obtains a better performance than DeepGO, this fact suggests that the structural regularization in the final layer of DeepGO does not make a full use of the GO hierarchy. The performance margin between DeepGOA and DeepGOPlus again indicates the effectiveness of our coherent learning the semantic representation of GO terms and the feature representations of amino acids. Naive always has a value of AUC lower than 0.5, since it predicts the GO annotations of a protein based on the frequency of GO terms and prefers to assign the most frequent terms to a protein. BLAST often loses other comparing methods (except Naive). This fact proves the effectiveness of learning the representation of amino acids by CNN for protein function prediction.

#### B. Component analysis

In order to investigate which component of DeepGOA contribute to the improved performance of DeepGOA, we introduced three variants: DeepGOA-LABEL only uses the co-annotation patterns without GO hierarchy; DeepGOA-GO only uses the GO hierarchy; DeepGOA-CNN directly uses the representation of amino acids and dot product to make function prediction, without using the semantic representation of GO terms. Table II lists the results of DeepGOA and its three variants on Human genome with the experiments configured the same as in the previous subsection.

DeepGOA generally has a better performance than its three variants. Under the same experimental setting, DeepGOA-GO and DeepGOA-Label give better results than DeepGOA-CNN. This observation proves that it is important and beneficial to learn the semantic representation of GO terms and optimize the mapping of feature representation of amino acids to the semantic representation. DeepGOA-GO achieves better results than DeepGOA-Label with respect to  $S_{min}$ , since it utilizes the GO hierarchy while DeepGOA-Label mainly uses the co-annotation pattern of GO terms to the same proteins, and  $S_{min}$  is defined with respect to the GO hierarchy. On the other hand, DeepGOA-Label has a better AUPRC and AUC by modeling GO term co-annotation. DeepGOA leverages the GO hierarchy and GO terms' co-annotation pattern, and thus it obtains better results than DeepGOA-GO and DeepGOA-Label. This ablation study further confirms the necessity of incorporating GCN for exploring and exploiting the latent hierarchical relationship between GO terms, and thus to improve the prediction accuracy.

TABLE II  
PREDICTION RESULTS OF DEEPCGOA AND ITS VARIANTS.

		AUC	AUPRC	$S_{min} \downarrow$	$F_{max}$
BP	DeepGOA	69.79	<b>62.20</b>	<b>19.7772</b>	<b>38.52</b>
	DeepGOA-GO	69.72	60.69	20.1579	36.79
	DeepGOA-Label	<b>70.12</b>	61.72	20.2206	38.14
	DeepGOA-CNN	69.19	61.06	20.2332	36.12
CC	DeepGOA	75.69	49.97	<b>4.9029</b>	<b>62.92</b>
	DeepGOA-GO	75.94	48.64	4.9127	62.43
	DeepGOA-Label	<b>76.83</b>	<b>55.87</b>	4.9707	62.67
	DeepGOA-CNN	74.85	49.19	5.0134	61.43
MF	DeepGOA	<b>82.03</b>	<b>70.98</b>	<b>4.7571</b>	<b>47.71</b>
	DeepGOA-GO	81.75	70.28	4.8201	46.98
	DeepGOA-Label	81.46	70.81	4.9661	46.88
	DeepGOA-CNN	77.65	63.12	5.2867	41.54

## IV. CONCLUSIONS

Protein function prediction is one of the fundamental challenges in the post genomic era. Incorporating the knowledge of GO hierarchy into protein function can improve the prediction performance. For this purpose, we developed DeepGOA based on GCN and CNN. DeepGOA utilizes the GCN to learn the semantic representation of GO terms using GO hierarchy and annotations related to GO terms, and the CNN to learn the representation of amino acids. It then jointly seeks the mapping from the feature representation of amino acids to semantic representation of GO terms, and the accomplishment of protein function prediction in a coherent and end-to-end manner. Experimental results on archived GO annotations of Human and Corn show that DeepGOA performs better than other related deep learning based solutions. Our ablation study further confirms that it is beneficial to learn the semantic representations of GO terms for function prediction. We will extend our work to predict the functional roles of diverse noncoding RNAs.

## V. ACKNOWLEDGEMENTS

This work is supported by Natural Science Foundation of China (61872300, 61873214 and 61871020), Funda-

TABLE I  
EXPERIMENTAL RESULTS OF PREDICTING GO ANNOTATIONS OF HUMAN AND CORN GENOME.

		Human					Corn				
		PR50	AUC	AUPRC	$S_{min} \downarrow$	$F_{max}$	PR50	AUC	AUPRC	$S_{min} \downarrow$	$F_{max}$
BPO	DeepGOA	<b>55.20</b>	<b>69.79</b>	<b>62.20</b>	<b>19.7772</b>	<b>38.52</b>	<b>82.44</b>	<b>89.56</b>	<b>70.18</b>	<b>0.9965</b>	<b>67.53</b>
	DeepGOPlus	53.61	68.74	60.75	20.0152	36.23	76.47	89.52	69.64	1.2193	59.61
	DeepGO	50.25	63.85	57.13	20.6061	32.71	64.41	85.39	62.91	1.2586	59.49
	BLAST	46.50	57.72	48.94	20.2695	33.92	32.61	71.77	28.96	1.1745	61.10
	Navie	51.94	49.98	56.61	20.4729	34.45	27.08	49.93	27.67	1.8957	29.32
CCO	DeepGOA	<b>50.88</b>	<b>75.69</b>	<b>49.97</b>	4.9029	<b>62.92</b>	<b>96.33</b>	<b>87.73</b>	75.78	<b>0.6603</b>	<b>75.74</b>
	DeepGOPlus	50.18	65.15	48.70	4.9488	62.75	91.21	82.51	<b>77.84</b>	0.8105	70.82
	DeepGO	43.60	69.51	44.81	5.1828	58.86	86.57	82.91	72.07	0.7759	71.08
	BLAST	21.25	56.27	26.91	5.0593	44.18	39.48	70.82	39.18	0.7904	62.02
	Navie	36.27	48.69	37.70	5.4474	55.15	48.14	49.98	43.74	1.2458	49.84
MFO	DeepGOA	<b>68.64</b>	<b>82.03</b>	<b>70.98</b>	<b>4.7571</b>	<b>47.71</b>	<b>83.63</b>	<b>92.51</b>	<b>68.63</b>	1.7024	<b>58.10</b>
	DeepGOPlus	67.84	81.86	69.38	4.8426	46.82	72.70	83.67	64.42	<b>1.6777</b>	51.25
	DeepGO	54.56	75.98	62.47	5.2581	40.43	68.78	88.22	59.91	1.8551	52.82
	BLAST	42.33	62.34	46.11	4.9195	41.07	27.40	67.76	32.92	1.8274	51.40
	Navie	46.86	49.87	52.77	5.7466	32.59	28.44	51.04	28.84	2.7430	26.13

mental Research Funds for the Central Universities (XDJK2019B024), Natural Science Foundation of CQ CSTC (c-stc2018jcyjAX0228).

## REFERENCES

- [1] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schoes, T. Wittkop, A. Sokolov, K. Graim, K. Funk, K. Verspoor, and A. Ben-Hur, "A large-scale evaluation of computational protein function prediction," *Nature Methods*, vol. 10, no. 3, p. 221, 2013.
- [2] A. Shehu, D. Barabara, and K. Molloy, "A survey of computational methods for protein function prediction," in *Big Data Analytics in Genomics*. Springer, 2016, pp. 225–298.
- [3] A. M. Schoes, D. C. Ream, A. W. Thorman, P. C. Babbitt, and I. Friedberg, "Biases in the experimental annotations of protein function and their effect on our understanding of protein function space," *PLoS computational biology*, vol. 9, no. 5, p. e1003063, 2013.
- [4] P. C. Biol, "The gene ontology's reference genome project: a unified framework for functional annotation across species," *PLoS Computational Biology*, vol. 5, no. 7, p. e1000431, 2009.
- [5] P. D. Thomas, V. Wood, C. J. Mungall, S. E. Lewis, and J. A. Blake, "On the use of gene ontology annotations to assess functional similarity among orthologs and paralogs: A short report," *PLoS Computational Biology*, vol. 8, no. 2, p. e1002386, 2012.
- [6] Y. Jiang, T. R. Oron, W. T. Clark, A. R. Bankapur, D. DAndrea, R. Lepore, C. S. Funk, I. Kahanda, K. M. Verspoor, and A. Benhur, "An expanded evaluation of protein function prediction methods shows an improvement in accuracy," *Genome Biology*, vol. 17, no. 1, p. 184, 2016.
- [7] G. O. Consortium, "The gene ontology in 2010: extensions and refinements," *Nucleic acids research*, vol. 38, no. S1, pp. D331–D335, 2009.
- [8] R. P. Huntley, T. Sawford, M. J. Martin, and C. ODonovan, "Understanding how and why the gene ontology and its annotations evolve: the go within uniprot," *GigaScience*, vol. 3, no. 1, p. 4, 2014.
- [9] C. Dessimoz and N. Škunca, "The gene ontology handbook," 2017.
- [10] G. Valentini, "True path rule hierarchical ensembles for genome-wide gene function prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 3, pp. 832–847, 2011.
- [11] X.-F. Zhang and D.-Q. Dai, "A framework for incorporating functional interrelationships into protein function prediction algorithms," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 9, no. 3, pp. 740–753, 2012.
- [12] G. Yu, H. Rangwala, C. Domeniconi, G. Zhang, and Z. Yu, "Protein function prediction using multi-label ensemble classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 4, pp. 1045–1057, 2013.
- [13] —, "Protein function prediction with incomplete annotations," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 3, pp. 579–591, 2014.
- [14] G. Pandey, C. L. Myers, and V. Kumar, "Incorporating functional inter-relationships into protein function prediction algorithms," *BMC Bioinformatics*, vol. 10, no. 1, p. 142, 2009.
- [15] G. Yu, H. Zhu, and C. Domeniconi, "Predicting protein functions using incomplete hierarchical labels," *BMC Bioinformatics*, vol. 16, no. 1, p. 1, 2015.
- [16] G. Fu, J. Wang, B. Yang, and G. Yu, "Neggo: Negative go annotations selection using ontology structure," *Bioinformatics*, vol. 32, no. 19, pp. 2996–3004, 2016.
- [17] G. Yu, G. Fu, J. Wang, and Y. Zhao, "Newgo: Predicting new go annotations of proteins by bi-random walks on a hybrid graph," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 4, pp. 1390–1402, 2018.
- [18] Y. Zhao, G. Fu, J. Wang, M. Guo, and G. Yu, "Gene function prediction based on gene ontology hierarchy preserving hashing," *Genomics*, vol. 111, no. 3, pp. 334–342, 2019.
- [19] G. Yu, Y. Zhao, C. Lu, and J. Wang, "Hashgo: hashing gene ontology for protein function prediction," *Computational Biology and Chemistry*, vol. 71, p. 264, 2017.
- [20] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3, pp. 197–387, 2014.
- [21] J. Wehrmann, R. C. Barros, S. N. d. Dôres, and R. Cerri, "Hierarchical multi-label classification with chained neural networks," in *Proceedings of the ACM Symposium on Applied Computing*, 2017, pp. 790–795.
- [22] A. S. Rifaioğlu, T. Doğan, M. J. Martin, R. Cetin-Atalay, and M. V. Atalay, "Multi-task deep neural networks in automated protein function prediction," *arXiv preprint arXiv:1705.04802*, 2017.
- [23] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "Deepgo: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, 2017.
- [24] M. Kulmanov and R. Hoehndorf, "DeepgoPlus: Improved protein function prediction from sequence," *bioRxiv*, p. 615260, 2019.
- [25] C. Wang, L. Zhang, and H.-J. Zhang, "Learning to reduce the semantic gap in web image retrieval and annotation," in *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2008, pp. 355–362.
- [26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [27] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, pp. 3389–3402, 1997.
- [28] Y. Tao, L. Sam, J. Li, C. Friedman, and Y. A. Lussier, "Information theory applied to the sparse gene ontology annotation network to predict novel gene function," *Bioinformatics*, vol. 23, no. 13, pp. i529–i538, 2007.
- [29] Z. Teng, M. Guo, X. Liu, Q. Dai, C. Wang, and P. Xuan, "Measuring gene functional similarity based on group-wise comparison of go terms," *Bioinformatics*, vol. 29, no. 11, pp. 1424–1432, 2013.
- [30] W. T. Clark and P. Radivojac, "Information-theoretic evaluation of predicted ontological annotations," *Bioinformatics*, vol. 29, no. 13, pp. i53–i61, 2013.