Computational Biology and Chemistry 89 (2020) 107379

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/cbac

Research Article

DeepAdd: Protein function prediction from k-mer embedding and additional features

Zhihua Du [a,*], Yufeng He [a], Jianqiang Li [a], Vladimir N. Uversky [b,c,d,*]

[a] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, Guangdong Province, PR China
[b] Department of Molecular Medicine, Morsani College of Medicine, University of South Florida, 12901 Bruce B. Downs Blvd. MDC07, Tampa, FL, USA
[c] USF Health Byrd Alzheimer's Research Institute, Morsani College of Medicine, University of South Florida, 12901 Bruce B. Downs Blvd. MDC07, Tampa, FL, USA
[d] Laboratory of New Methods in Biology, Institute for Biological Instrumentation, Russian Academy of Sciences, Institutskaya Str., 7, Pushchino, Moscow Region, 142290, Russia

ARTICLE INFO

Keywords:
Protein function prediction
Convolution neural network
Natural language process
Protein-protein interaction network
Sequence similarity profile

ABSTRACT

With the application of new high throughput sequencing technology, a large number of protein sequences is becoming available. Determination of the functional characteristics of these proteins by experiments is an expensive endeavor that requires a lot of time. Furthermore, at the organismal level, such kind of experimental functional analyses can be conducted only for a very few selected model organisms. Computational function prediction methods can be used to fill this gap. The functions of proteins are classified by Gene Ontology (GO), which contains more than 40,000 classifications in three domains, Molecular Function (MF), Biological Process (BP), and Cellular Component (CC). Additionally, since proteins have many functions, function prediction represents a multi-label and multi-class problem. We developed a new method to predict protein function from sequence. To this end, natural language model was used to generate word embedding of sequence and learn features from it by deep learning, and additional features to locate every protein. Our method uses the dependencies between GO classes as background information to construct a deep learning model. We evaluate our method using the standards established by the Computational Assessment of Function Annotation (CAFA) and have noticeable improvement over several algorithms, such as FFPred, DeepGO, GoFDR and other methods compared on the CAFA3 datasets.

1. Introduction

The gap between the known sequences of proteins and their function become wider, because of the ever increasing universe of protein amino acid sequences (Radivojac et al., 2013). Many computational methods have been proposed to annotate functions for unknown proteins. In order to provide a standard description method, Gene Ontology (GO) was launched. Currently, Gene Ontology (GO) has over 40,000 biological conceptsgrouped into three domains: Molecular Function Ontology (MFO, or MF), Biological Process Ontology (BPO,or BP) and Cellular Component Ontology (CCO,or CC) (You et al., 2018a). On the other hand, the UniProt, which is the biggest protein sequence database, had an increase of the data from 553,232–558,590 for the manually annotated proteins, whereas the number of automatically annotated proteins increased from 7,187,988 in January of 2016 to 11,713,556 in October of 2018. In parallel, the proportion of protein with known functions in this database decreased from 7.70 % to 4.77 %. All these observations indicate that although the number of protein sequences is increasing fast, the understanding of their functions is falling far behind. To fill this deep gap, an imperative issue would be the development of efficient automated function prediction (AFP) tools (Jiang et al., 2016a).

To advance the performance of AFP, the Critical Assessment of Functional Annotation challenges (CAFA) have been held four times (Anon., 2013; Jiang et al., 2016b; Zhou et al., 2019). CAFA1 to CAFA3 were held in 2010–2011, 2013–2014, 2016–2017 respectively and most recent CAFA pi was held in 2017−2018. CAFA utilizes a time-delayed evaluation procedure to assess the accuracy of protein function prediction submitted by participants. To this end, CAFA uses proteins that have no publicly available experimental annotations for each GO domain (MFO, BPO or CCO), which is called no-knowledge protein (Jiang et al., 2016a). Importantly, at the start of the assessment, these proteins have only sequences, whereas their functional annotations will be released

* Corresponding author.
E-mail addresses: duzh@szu.edu.cn (Z. Du), vuversky@health.usf.edu (V.N. Uversky).

https://doi.org/10.1016/j.compbiolchem.2020.107379
Received 13 October 2019; Received in revised form 15 September 2020; Accepted 17 September 2020
Available online 23 September 2020

only after the deadline. Since in practice, over 95 % of proteins have only sequence information and no functional annotations, developing the efficient AFP for such no-knowledge proteins constitutes a very important task. These would be also the reasons why a well-performing AFP method is important. Since a protein is expected to have multiple functions, AFP method requires an eigenvalue that clearly represents each protein. To achieve this goal, researchers extracted many pieces of biological information, which can be useful in functional predictions, such as protein sequences, protein domains, protein structures, protein interactions, text information, informational spectrum, sequence similarity (Pérez et al., 2004; Raychaudhuri et al., 2002; Shatkay et al., 2007; Wong and Shatkay, 2013; Shatkay et al., 2015; Van et al., 2014; Deng and Huang, 2014; Huang and Hong-Jie, 2013; Kent, 2002), and various combinations of these features (Sokolov and Ben-Hur, 2010).

Below we provide a short description of several previously elaborated methods as shown in Table 1, which may fall into two classes.

The first and the most widely used class is information-based methods, which use the information of sequence alignment, amino acid content, sequence properties etc for protein function prediction. For example, COGIC (Cozzetto et al., 2013), GoFDR (Gong et al., 2016), and SMISS (Cao and Cheng, 2016) use combinations of different resources for protein function prediction

The second class is machine learning-basedmethods (Huang and Du, 2008; Shen et al., 2018) Most machine learning methods use features generated from the protein sequences for model training, and use that model to predict protein function. For example, PhosPred-RF (Wei et al., 2017a), CPPred-RF (Wei et al., 2017b), PANNZER (Koskinen et al., 2015), FEATURE (Halperin et al., 2008), NetGo (You et al., 2018b), FFPred3 (Cozzetto et al., 2016) use SVM, random forest, naïve Bayes, learn to rank to predict protein function. There are two crucial steps among these methods, building a meaningful feature set and choosing an appropriate algorithm. However, more errors could be involved for an inapposite feature set. The latest machine learning methods—deep learning which uses multiple layers representation and abstraction of data have proven their outstanding performance in image recognition and speech recognition It would be interesting to apply these latest machine learning methods for the protein function prediction problem, such as DeepGO (Kulmanov et al., 2017) and DeepText2Go (You and Zhu, 2017).

The DeepGO method combines two forms of representation learning based on multiple layers of neural networks to learn features that can be used in prediction of protein functions (Kulmanov et al., 2017). Here, one method learns features from protein sequence, whereas another learns representations of proteins based on their location within the protein–protein interaction networks for multiple species from the STRING database (Damian et al., 2015). Adding this PPI network-derived information to the sequence-based information increased the predictive performance of DeepGO (Kulmanov et al., 2017). Similarly, a NetGO approach (You et al., 2018b) combines various sequence information and massive network information of all species (>2000) in STRING (Damian et al., 2015). Furthermore, NetGO is able to use network information to annotate a protein by homology transfer, even if it is not covered in STRING (You et al., 2018b). It was shown that NetGO significantly outperformed GOLabeler (You et al., 2018a), DeepGO (Kulmanov et al., 2017), and several other methods for automatic functional annotation of proteins.

Furthermore, there are several methods, such as FFPred3 (Cozzetto et al., 2016), that show great performance based on the CAFA data (Anon., 2013; Jiang et al., 2016b; Zhou et al., 2019). These methods use protein sequences as their primary data and utilize homology-based transfer of information (GoFDR) (Gong et al., 2016) or scan the input sequences against a set of SVMs, each examining the relationship between protein function and biophysical attributes, such as secondary structure, transmembrane helices, intrinsically disordered regions, signal peptides and other motifs (FFPred3) (Cozzetto et al., 2016). Additionally, utilization of the Label-Space Dimensionality Reduction (LSDR) techniques based on the structure of the GO terms and on the semantic similarity of terms was shown to improve the CAFA performance of several function prediction algorithms (Makrodimitris et al., 2019).

Motivated by the success of the DeepGO method, we propose here a method called DeepAdd that can predict protein functions using a deep convolutional neural network (CNN) framework. To this end, we integrate a natural language method into the protein sequence representation. Instead of describing the protein sequence as a tri-gram embedding as DeepGO did, DeepAdd utilizes a Word2Vec method on defining the set of features to represent a protein. The vector representations of words learned by Word2Vec models has shown to carry semantic meanings and are useful in various natural language processing (NLP) tasks (Mikolov et al., 2013; Goldberg and Levy, 2014; Bengio et al., 2003; Asgari and Mofrad, 2015). Although DeepGo added network information to sequence-based information, network information has limitations for modeling protein function. For example, if a protein is annotated as unknown, then the corresponding features of PPI network will be set to zero, and as a result, the performance of DeepGo will be affected. To address this issue, DeepAdd incorporates sequence

Table 1
Comparison of several AFP methods.

Method	Means of prediction	Level	Properties
COGIC (Cozzetto et al., 2013)	Statistical scoring methods	Sequence similarity	Integrate multiple sources of biological information, need multiple methods
GoFDR (Gong et al., 2016)	Statistical scoring methods	Sequence similarity	Need PSI-BLAST; Slow in predicting
Smiss (Cao and Cheng, 2016)	Complex Statistical scoring methods	Sequence similarity,go term, PPIs, spatial gene–gene interaction networks	Combines information from different sources and calculates three different probability scores
PhosPred-RF (Wei et al., 2017a)	Random forest-based predictor	Phosphorylation and non- phosphorylation sites	Usage of random forest,simple sequence features
CPPred-RF (Wei et al., 2017b)	Random forest-based predictor	Sequence similarity	Multiple sequence-based feature
PANNZER (Koskinen et al., 2015)	Weighted k-nearest neighbor method	Sequence similarity, PPIs, and gene expressions	High annotation accuracy, not suitable classifier
FEATURE (Halperin et al., 2008)	Naïve Bayes	Structure information	Need known 3D structure
NetGo (You et al., 2018b)	Learn to rank	Sequence information and massive network information	Based on various sequence information and massive network information of all species (>2000)
FFPred3 (Cozzetto et al., 2016)	SVM	Sequence information, structure information	Utilize SVM to find biophysical attributes
DeepGO (Kulmanov et al., 2017)	Multiple layers of neural networks	Protein sequence, PPIs	Usage of deep learning, simple sequence coding
DeepText2Go (You and Zhu, 2017)	Combines neural network and basic method	Sequence information, sequence similarity	Need PSI-BLAST; Slow in predicting

similarity profile (SSP) to learn features that exploits functional relationships across all levels of similarity, and not only at high similarity levels (Makrodimitris et al., 2019). For each target protein, if the features of PPI network are empty, then we infer SSP features for that protein as added features instead of PPI network. For example, since a target protein PEFAH_ECOLI is an unknown protein, it does not have PPI features, then the SSP will be used as supplementary features instead of empty PPI network features.

2. Materials and methods

The main workflow of the DeepAdd algorithm is illustrated in Fig. 1. DeepAdd uses three steps to predict functions of protein. In testing, given the sequence of a query protein, we use the Word2Vec method to represent the protein sequence. After that, DeepAdd consists of two CNN models with multiple convolution blocks that map the presented protein sequence to two-feature vectors representation. One feature representation is for the sequence similarity profile by SSP model. The other feature representation is the PPI network by PPI model. Finally, DeepAdd uses a hierarchical classification method to classify all candidate GO terms of each query protein. All proteins in the training data set and their candidate GO terms are used for training the two CNN models. In this way, DeepAdd allows integration of the sequence similarity knowledge and protein- protein interaction information of query proteins.

2.1. Datasets

We trained DeepAdd on two datasets, a CAFA3 dataset and a SwissProt dataset (Boutet et al., 2016). As a sequence could be assigned more than one function, DeepAdd had to solve a multi-label classification task on both datasets.

In our experiments, we use Gene Ontology (Ashburner et al., 2000) to annotate functions of protein, which was downloaded on 23 April 2018 from the following link http://www.geneontology.org/page/download-go-annotations in the OBO format. By following the CAFA settings, we kept experimental annotations as our training and test data with following codes: 'EXP', 'IDA', 'IPI', 'IMP', 'IGI', 'IEP', 'TAS' and 'IC'.

As for the SwissProt dataset (Boutet et al., 2016), we downloaded reviewed and manually annotated proteins on 24 April 2018 from https://www.uniprot.org/downloads#uniprotkblink, to have a set which includes 558,590 proteins. Also, we test our method on CAFA3 dataset, which includes 130,787 protein sequences, downloaded from https://www.biofunctionprediction.org/cafa/. We filtered the protein sequences by length to uniform the input length of two deep learning models, and the length of protein sequence was no more than 1000 residues.

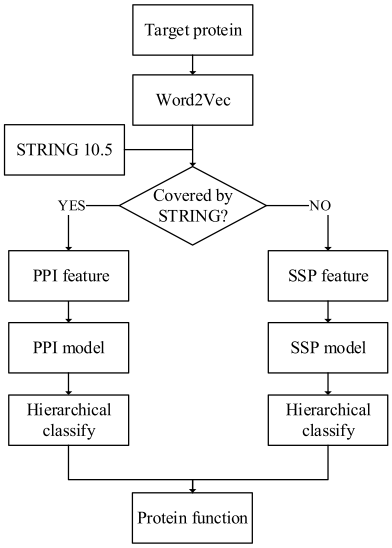

Fig. 1. Framework of the DeepAdd algorithm.

2.2. Utilization of Word2Vec

Algorithm 1

Input: *seq*: sequence of protein

Output: *emb*: word embedding of sequence

load the dictionary $D_i\psi$ and k-residues;
initial $emb\psi = 0$, $vec\psi = 0$ and $n\psi = 0$;
len ← length of *seq*
for $(word\psi / \psi each\psi k\text{-}residues)$ **do**
  if $word\psi \in \leftarrow D_i\psi$ **then**
    $vec\psi = D_i(word)$;
    $emb\psi / \psi concatenate \leftarrow emb\ ,\ vec \leftarrow \psi$
  else
    // Supplementary training for small corpora
    $D_{i \rightarrow 1}\psi = D_i\psi \cup \leftarrow word$;
    $Redo\psi$ ***for***;
  end if
end for

Traditional methods simply calculate the vector of k-mer frequencies without utilizing the co-occurrence relationship of k-mers. The k-mer feature is an order-less document, used in natural language processing and information retrieval. With the co-occurrence, we can get a word embedding with global statistical information, which may help us to construct better feature representations. Therefore, we use a Word2Vec method in DeepAdd. The superiority of Word2Vec over other method for learning word embedding lies in thesimplicity of its operation and capability to generate stable results.

Word2Vec include 2 models, continuous bag-of-words (CBOW) model and Skip-Gram model (Mikolov et al., 2013; Goldberg and Levy, 2014). These two models have opposite data flows, when one training the corpus. We use CBOW model in this experiment. The raw protein sequences are used as the input to the Word2Vec. Each protein is has a characteristic sequence composed of 20 amino acids. We are using the fixed word length of k for the sequence utilizing a sliding window (Fig. 2). Then these k-residues are trained to a dictionary with 20^k elements. We divide each sequence by the same window and annotate every k-residue according to the dictionary (see Algorithm 1). The

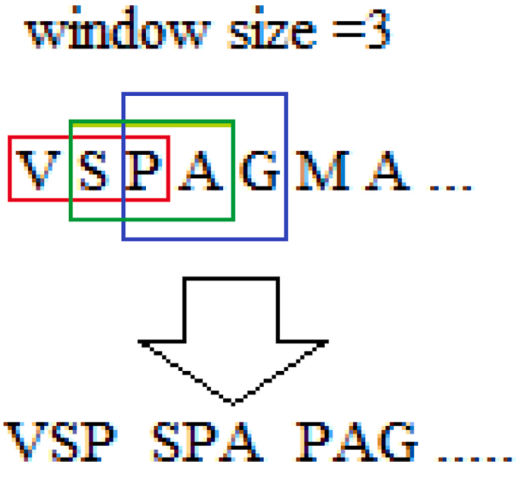

Fig. 2. Principles of using neural networks for predicting molecular traits from protein sequence.

corresponding outputs were used as the inputs for the two CNN models.

2.3. Additional features

In order to furtherly help the neural network to classify proteins and describe the relationship between proteins in training set and test set, we introduce two additional features: PPI network features and sequence similarity profile features.

2.3.1. Protein-protein interaction (PPI) network features

PPI network is composed of individual proteins engaged in their interactions with specific partners. PPI networks are crucial forvarious life processes such as biological signal transmission, gene expression regulation, energy and material metabolism and cell cycle regulation. We use it to describe the relationship between all proteins we use.

Since our experiment was based on SwissProt protein identifiers, we mapped the graph of proteins identifiers using the identifier mapping from Search Tool for the Retrieval of Interacting Genes (STRING) database (Damian et al., 2015).

In order to represent nodes and the topological relationship between nodes, Walking-RDF (Alshahrani et al., 2017) method was used to extract the knowledge graph embedding with size of 256 for each protein. Walking-RDF is an improved DeepWalk method, which uses language modeling in social network (Perozzi et al., 2014). For proteins those missing in the graph, a vector of 256 zeros were set to indicate features missing. We mapped 5,570,349 proteins randomly in UniProt for our work and the knowledge graph embedding.

2.3.2. Sequence similarity profile (SSP) features

For unknown proteins or some proteins which are not covered in STRING and SwissProt, they may have no any PPI features. In this case, the performance of the PPI features based model will be poor for the PPI-related feature vectors are empty. In order to fix the problem, we included the sequence similarity profile (SSP) features as supplementary features. The SSP feature is a list of sequence similarity score s, computed with respect to all of the sequences in the training set. It represented each protein i in the dataset as a vector x_i, which jth element shows the sequence identity between ith and jth training protein. This means that each protein is represented by a SSP in the training set. The SSP feature offers two primary advantages. First, it is a helpful feature when the PPI features are empty. Second, the SSP feature is simpler because it does not need to be trained separately for each GO term.

We utilized the Needleman-Wunsch global sequence alignment algorithm (Heringa, 2004) to conduct alignment, which use BLOSUM62 as scoring matrix. Additionally, the SwissProt database may contain orthologous proteins, which are almost identical and might have similar functions. To ensure that our dataset does not contain sequences with high similarity levels, we use SSP to filter our dataset. The highest similarity in our dataset is 0.45. Summarizing since using zeros for proteins that are not in PPI graph has bad effect on evaluation parameters, we used SSP for these proteins to improve performance of our models.

2.4. Convolutional neural network (CNN) models

We built up the deep learning framework by Keras, which uses Tensorflow as backend. As we have two kinds of additional features, two CNN models were built.

For the SSP model, an embedding layer was used to facilitate data input and matrix shape transformation. After the embedding layer, three pairs of convolution layers with max pooling layers were designed to extract eigenvalues. Then, one full connection layer (dense layer) was used for further data extraction and a fixed length of output vector.

For the PPI model, an embedding layer was used for input and three 1D-convolution layers for eigenvalue extraction, followed by a max-pooling layer to simplify computing complexity of network. The detail parameters are shown in Fig. 3.

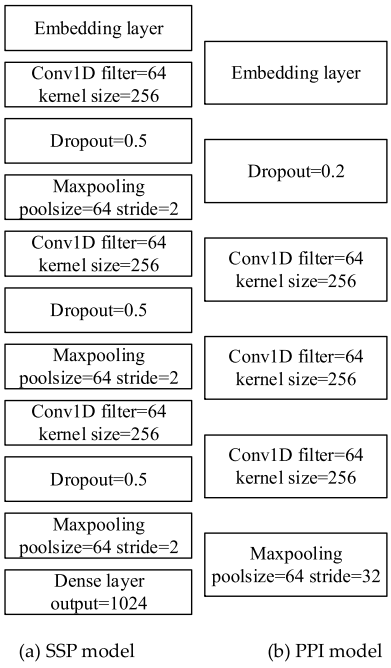

Fig. 3. Convolutional neural network structure.

2.5. Hierarchical classifier

In order to code the functional dependencies between classes in GO and optimize the classification accuracy on the hierarchy of GO at the same time, each GO class is encoded instead of optimizing a local model for each class. The intention is that this model can identify both explicit and implicit dependencies. We generate a series of full connection layers with a sigmoid activation function for each classes in GO, each of these layers has one connection to an output neuron. For each subclass, we have a connection from its ancestors to represent this relationship.

The concatenated sequence and additional features passed to a fully connected layer with 1024 nodes and the output is used as the input of the hierarchical classifier for further classification.

2.6. Evaluation

For comparison, we use two sets of parameters to evaluate the model. One set include AUC of ROC curve and Mathews Correlation Coefficient (MCC). The sensitivity and specificity for ROC curve is computed by Eqs. (1) and (2). For all threshold parameter t mentioned below, we have $t\in$ [0,1].

$$sen_f(t) = \frac{\sum_i |f \in P_i(t) \cap f \in T_i|}{\sum_i |f \in T_i|} \tag{1}$$

$$spe_f(t) = \frac{\sum_i |f \notin P_i(t) \cap f \notin T_i|}{\sum_i |f \notin \mathrm{T}|} \tag{2}$$

In these equations, f is the annotation for proteins from Gene Ontology, P_i (t) is a set of predicted classes for a given protein i with threshold t, T_i is a set of accurate annotation of a protein i.

The AUC is computed by Eq. (3). The MCC is computed by Eq. (4). Here, TP stands for true positives, FN for false negatives, FP for false positives and TN for true negatives.

$$\text{AUC} = \int_{-\infty}^{\infty} \frac{TP(t)}{TP(t)+FN(t)} \cdot \left(\frac{-FP(t)}{Fp(t)+TN(t)} \right) dt \tag{3}$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4}$$

The other set of parameters are average precision (*pr*) and recall (*re*) (Clark and Predrag, 2013), which are computed by Eqs. (5)–(8). Where precision is averaged over the proteins where we at least predict one term and $m(t)$ is the total number of such proteins and n is the number of protein in the test set.

$$p_k(t) = \frac{\sum_f |f \in P_k(t) \cap f \in T_k|}{\sum_{fun} |f \in P_k(t)|} \tag{5}$$

$$r_k(t) = \frac{\sum_f |f \in P_k(t) \cap f \in T_k|}{\sum_f |f \in A_k|} \tag{6}$$

$$pr(t) = \frac{1}{m(t)} \sum_{k=1}^{m(t)} p_k(t) \tag{7}$$

$$re(t) = \frac{1}{n} \sum_{k=1}^{n} r_k(t) \tag{8}$$

Additionally, we use F_{max}, which can compute by Eq. (9) to get a general watch on precision and recall for a threshold parameter $t \in [0,1]$ using the average precision and recall.

$$F_{max} = max\left\{ \frac{2 \cdot pr(t) \cdot re(t)}{re(t)+pr(t)} \right\} \tag{9}$$

3. Result

In order to verify the feasibility of DeepAdd, we run a series of comparative experiments, including comparison of this tool with the new and effective external methods and internal control comparison.

3.1. Experiment setup

For our experiments, we trained three models for each sub-ontology on GO (BP, CC and MF). First, we use GO to annotate proteins in SwissProt database and randomly group the proteins. Then 80 % of them are used as a training set and 20 % as a test set. Furthermore, 20 % of the training set are extracted as validation set.

To reduce the amount of computation without affecting the training efficiency, we set the window size to 3 in the Word2Vec operation, then a dictionary of 8000 words is created. The dimension of word embedding was set to 4. Since the length of sequence is smaller than 1001, the input to embedding layer has no more than 4000 features for each protein.

During the training, we use Rmsprop optimizer to minimize the binary cross entropy loss, with the learning rate of 0.001. We monitor the loss value during training to ensure the best model is saved. To accelerate the training process, we use NVIDIA GeForce GTX 1080 Ti GPU that noticeably speeds up the training process.

3.2. Performance with additional SSP features

In the previous work (Kulmanov et al., 2017), DeepGO demonstrated that PPI features improved performance of function prediction. In order to verify the performance of SSP features, we generated a special test set that included all unknown proteins.

This means that all the PPI features for these proteins were zero vectors. Table 2 shows the overall performance of DeepAdd-SSP (that relies only on SSP features) in comparison with the DeepAdd-PPI (that relies only on PPI features).

We find that DeepAdd-SSP outperforms DeepAdd-PPI in the special test set (filtered dataset). This is the reason why we can improve the prediction performance of the method by combining th PPI and SSP features. This analysis also shows that SSP features are effective supplement features for unknown proteins.

3.3. Comparison with baseline models

To evaluate the performance of our method, we compared our method with DeepGo (Kulmanov et al., 2017). For DeepGo, we use the source code downloaded from https://github.com/bio-ontology-research-group/deepgo. To make the experiment closer to practical application, the test set for this experiment contained all kinds of PPI features.

From the data summarized in Table 3, we find that DeepAdd shows higher F_{max} and AUC values than DeepGO on all three sub-ontologies, BP, MF, and CC. Therefore, Word2Vec performs better than NPLM, when a protein sequence act as an input to a neural network. In addition, PPI features and SSP features together helped us to improve the DeepAdd performance for the protein function prediction.

3.4. Performance on CAFA3 dataset

As aforementioned, AFP for no-knowledge proteins from CAFA challenge [4,5,6] is important. We compare DeepAdd with two top-performing methods from the previous CAFA challenges, GoFDR (Zhou et al., 2019) and FFPred3 (Cozzetto et al., 2016), also with the baseline methods DeepGO (Kulmanov et al., 2017) and SSP-LSDR (Makrodimitris et al., 2019). The data used here are on the benchmark released on November 2017 for CAFA3 competition. During training, the protein has no annotations. Table 4 shows the performance results of DeepAdd in comparison to other four methods on this benchmark dataset. DeepAdd performed well and received the highest Fmax and AUC score AUC in all three GO sub-ontologies.

Also, we compared our method with other CAFA methods, and the results are shown as Fig. 5. The figure shows DeepAdd did not achieve the best performance in the CAFA3 challenge as compared with the top CAFA methods in Fmax value. GoLabeler(Zhu Lab) got the best overall performance because it integrated five different types of sequence-based information and an LTR regression model to predict protein function. All these 5 types of information are not easy to get. It need BLAST-kNN sequence alignment, three long features has 1170, 8000, 33,879 features and a frequency feature. The combination of these features makes GOlabeler get better performance.

Among deep learning based methods, DeepGOPlus got better Fmax value. Due to computational limitations, DeepAdd and DeepGo can only predict around 2000 functions out of more than 45 000 which are currently in the GO. While DeepGOPlus has predicted more than 5000

Table 2
Classification performance on filtered dataset.

	BP					MF					CC				
	F_{max}	AUC	MCC	*pr*	*re*	F_{max}	AUC	MCC	*pr*	*re*	F_{max}	AUC	MCC	*pr*	*re*
DeepAdd-ssp	0.358	0.846	0.346	0.368	0.348	0.552	0.910	0.565	0.656	0.476	0.539	0.924	0.546	0.531	0.547
DeepAdd-PPI	0.233	0.754	0.222	0.184	0.315	0.188	0.761	0.210	0.217	0.166	0.321	0.773	0.491	0.281	0.372

Table 3
Classification performance on Swiss-Prot's dataset.

	BP					MF					CC				
	Fmax	AUC	MCC	pr	re	Fmax	AUC	MCC	pr	re	Fmax	AUC	MCC	pr	re
DeepGo	0.385	0.893	0.384	0.415	0.359	0.546	0.928	0.570	0.673	0.459	0.633	0.967	0.592	0.643	0.624
DeepAdd	**0.393**	**0.907**	**0.395**	0.400	**0.386**	**0.580**	**0.947**	**0.606**	**0.684**	**0.504**	0.619	**0.968**	**0.592**	0.638	0.601

Table 4
Performance on CAFA3 dataset.

	BP					MF					CC				
	F_{max}	AUC	MCC	*pr*	*re*	F_{max}	AUC	MCC	*pr*	*re*	F_{max}	AUC	MCC	*pr*	*re*
SSP-LSDR	0.298	0.761	0.317	**0.320**	0.276	0.291	0.767	0.457	0.335	0.255	0.303	0.783	0.417	0.345	0.267
FFPred3	0.288	0.841	0.232	0.311	0.267	0.376	0.861	0.287	0.349	0.407	0.446	0.891	0.393	0.462	0.431
DeepGo	0.343	0.884	0.324	0.313	0.379	0.475	0.906	0.574	0.614	0.387	0.522	0.953	0.504	**0.557**	0.492
GoFDR	0.193	0.621	0.024	0.283	0.146	0.513	0.847	**0.615**	**0.889**	0.361	0.413	0.734	0.320	0.400	0.426
DeepAdd	**0.345**	**0.896**	**0.335**	0.315	**0.381**	**0.516**	**0.912**	0.585	0.641	**0.432**	**0.547**	**0.958**	**0.511**	0.536	**0.558**

protein functions. DeepGOPlus will get more than two times of recall when it hits the 5000 functions. That the reason why DeepAdd and DeepGo can't outperform it. And we compared the number of network parameters in DeepAdd and DeepGOPlus, as shown in Fig. 6. It can be seen that the number of parameters in DeepGOPlus are several times of DeepAdd. In the future work, we need to improve our method computational ability in order to predict more protein functions.

3.5. Performance on proteins sets from different organisms

Fig. 4 shows the AUC scores of DeepAdd and DeepGo on protein sets from different organisms for three sub-ontologies. We find that performance of DeepAdd varies greatly among proteins from different organisms, in particular between eukaryotic and prokaryotic organisms.

We also find that DeepAdd shows competitive performance for many organisms in BP and MF sub-ontologies, probably due to the organism complexity, which can bring more features to their proteins. Yet this advantage is not obvious in CC sub-ontology. However, compared with DeepGO, we found that our method get an overall better performance.

3.6. Sensitivity analysis

To check the robustness of the newly developed tool, we carry out sensitivity analysis of the DeepAdd. In Word2Vec, word embedding varies with output dimensions and the length of a word that is, *k* of *k*-residues is variable too. We ran a series of experiments to find out if these parameters have any influence on the outputs of our experiment. Several corresponding results are shown in Table 5.

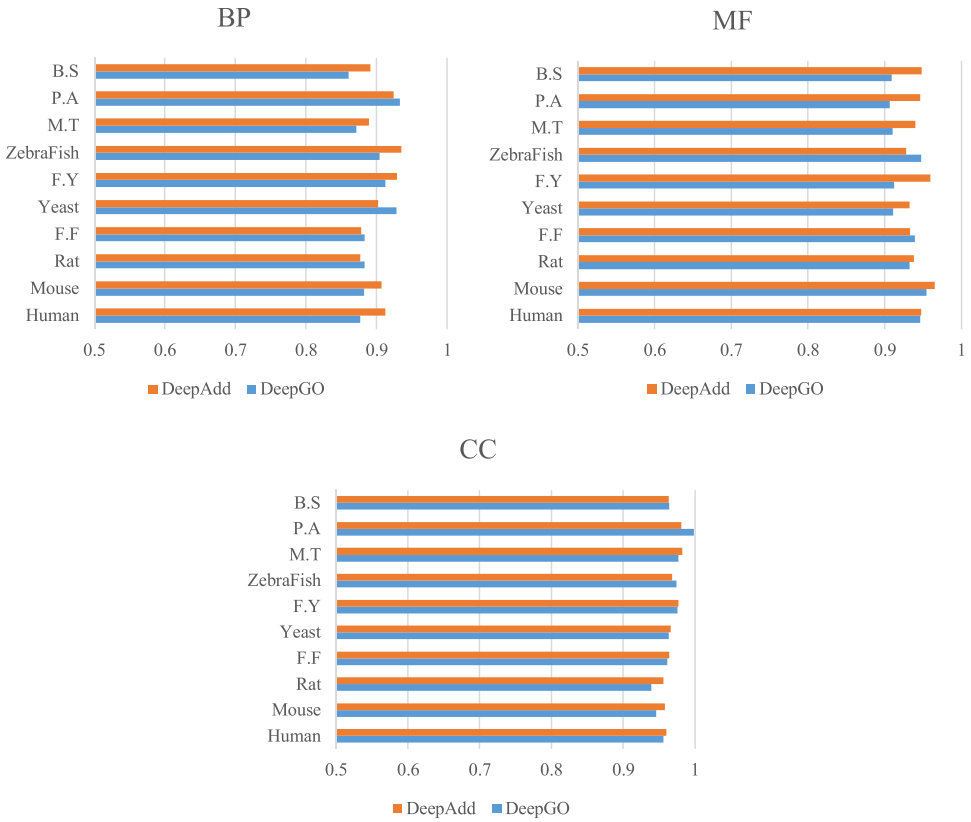

Fig. 4. AUC of different organisms. F.F: Fruit Fly, F.Y: Fission Yeast, M.T: Mycobacterium tuber-s, P.A: Pseudomonas aeruginosa, B.S: *Bacillus subtilis*.

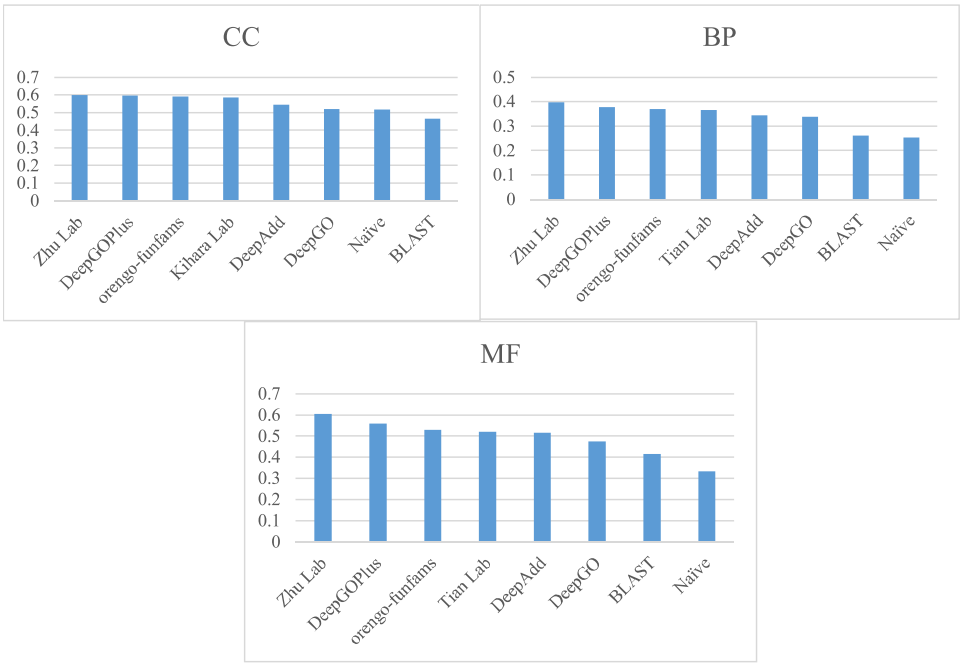

Fig. 5. Comparison results on CAFA3.

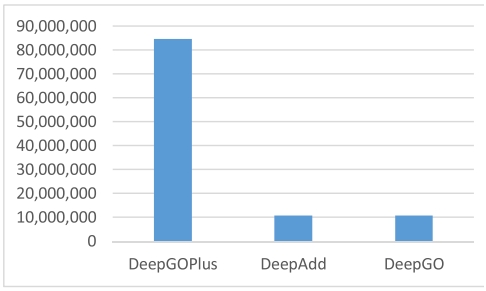

Fig. 6. the number of network parameters in DeepGOPlus, DeepAdd and DeepGO.

Table 5
Different dimension word embedding performance of BP.

Dimensions	4	64	4
k-residues	4	4	5
AUC	**0.910**	**0.911**	**0.908**
MCC	**0.392**	**0.394**	**0.389**
F_{max}	**0.394**	**0.387**	**0.395**
Pr	**0.401**	**0.397**	**0.402**
Re	**0.384**	**0.378**	**0.387**
Time per epoch(s)	**1867s**	**2219s**	**1947s**

NOTE: Dimensions show the output dimension of Word2Vec.

3.6.1. Word embedding dimension

From the Table 5 we can find that larger dimension of word embedding needs more time and memory to train the model. Although the performance of the model is partly improved, the increase in time is unacceptable. When the dimension of word embedding comes to 128 or higher, the training will cost more than 64GB of memory.

3.6.2. Length of the word

Due to the curse of dimensionality and the lack of a large enough corpus, we only use 5-residues to compare. The dimension of word embedding were set to 4 in this experiment. We found that the size of dictionary had little effect on the performance of our model.

According to the Table 5, our model is insensitive to the choice of D and k. However large *k* will bring an explosive growth of the dictionary, which will cause a big waste of calculation. To reduce the amount of computation complexity and get better results, we finally selected the 4 dimensional word embedding.

4. Discussion

DeepAdd extended the application of deep learning approaches for protein function prediction in three ways. First, we regard protein sequences as natural language and extract word embedding with Word2-Vec. Second, we apply feature learning using two CNN models, which include sequence similarity profile and PPI network features. Third, two CNN models data sources were used in a single model. The advantages of our method are its scalability, the potential for the end-to-end learning and the potential to predict class that can provide enough training data.

However, our model also has disadvantages. First, a large corpus is needed to generate a complete k-mer dictionary for Word2Vec. This means that we need a large enough protein database. Second, data we used to annotate the protein from GO have been accumulated artificially for many years, which limits applications in other areas, such as predicting phenotype annotations or effects of variants. Furthermore, we use deep learning to process data, which is extremely slow on CPU.

In the future, we intend to extend our method in several directions. First, we will introduce more influential, additional features, which may be helpful for prediction. Then we plan to explore more efficient algorithm to speed up the training while improving prediction results.

CRediT authorship contribution statement

Zhihua Du: Conceptualization, Methodology, Software. **Yufeng He:** Data curation, Writing - original draft. **Jianqiang Li:** Investigation. **Vladimir N. Uversky:** Supervision, Writing - review & editing.

Declaration of Competing Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or

kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, “DeepAdd：Protein function prediction from k-mer embedding and additional features”.

Acknowledgments

The authors gratefully acknowledge the contribution of the National Science Foundation of China [U1713212] [61572330] [61836005] [61702341], the Technology Planning Project of Shenzhen City [JCYJ20170302143118519] [GGFW2018021118145859] [JSGG201 80507182904693].

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.compbiolchem.2020.107379.

References

Alshahrani, M., Khan, M.A., Maddouri, O., Kinjo, A.R., Queralt-Rosinach, N., Hoehndorf, R., 2017. Neuro-symbolic representation learning on biological knowledge graphs. Bioinformatics 33.
Anon, 2013. A large-scale evaluation of computational protein function prediction. Nat. Methods 10 (3), 221–227.
Asgari, Ehsaneddin, Mofrad, M.R.K., 2015. Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS One 10.
Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., et al., 2000. Gene Ontology: tool for the unification of biology. Nat. Genet. 25, 25, 05/01/online.
Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155.
Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bansal, P., Bridge, A.J., et al., 2016. UniprotKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: how to use the entry view. Methods Mol. Biol. 1374, 23–54.
Cao, R., Cheng, J., 2016. Integrated protein function prediction by mining function associations, sequences, and protein–protein and gene–gene interaction networks. Methods 93, 84–91.
Clark, W.T., Predrag, R., 2013. Information-theoretic evaluation of predicted ontological annotations. Bioinformatics 29, i53–i61.
Cozzetto, D., Buchan, D.W., Bryson, K., Jones, D.T., 2013. Protein function prediction by massive integration of evolutionary; analyses and multiple data sources. BMC Bioinformatics 14, 1–11.
Cozzetto, D., Minneci, F., Currant, H., Jones, D.T., 2016. FFPred 3: feature-based function prediction for all Gene Ontology domains. Sci. Rep. 6, 31865.
Damian, S., Andrea, F., Stefan, W., Kristoffer, F., Davide, H., Jaime, H.C., et al., 2015. STRING v10: protein-protein interaction networks, integrated over the tree of life. Nucleic Acids Res. 43, D447.
Deng, Su-Ping, Huang, D.S., 2014. SFAPS: an R package for structure/function analysis of protein sequences based on informational spectrum method. Methods 69 (3), 207–212.
Goldberg, Y., Levy, O.J.C., 2014. word2vec Explained: Deriving Mikolov Et al.’S Negative-sampling Word-embedding Method. vol. abs/1402.3722.
Gong, Q., Ning, W., Tian, W., 2016. GoFDR: a sequence alignment based method for predicting protein functions. Methods 93, 3–14.
Halperin, I., Glazer, D.S., Wu, S., Altman, R.B., 2008. The FEATURE framework for protein function annotation: modeling new functions, improving performance, and extending to novel applications. BMC Genomics 9, S2.
Heringa, Jaap, 2004. Needleman-Wunsch Algorithm. Dictionary of Bioinformatics and Computational Biology. John Wiley & Sons, Inc.
Huang, D.S., Du, J.-X., 2008. A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. IEEE Trans. Neural Netw. 19 (12), 2099–2115.
Huang, D.S., Hong-Jie, Yu, 2013. Normalized feature vectors: a novel alignment-free sequence comparison method based on the numbers of adjacent amino acids. IEEEACM Trans. Comput. Biol. Bioinform. 10 (2), 457–467.
Jiang, Y., Oron, T.R., Clark, W.T., Bankapur, A.R., D’Andrea, D., Lepore, R., et al., 2016a. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biol. 17, 184.
Jiang, Yuxiang, et al., 2016b. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biol. 17 (1), 184.
Kent, W.J., 2002. BLAT - The BLAST-like alignment tool. Genome Res. 12 (4), 656–664.
Koskinen, P., Törönen, P., Nokso-Koivisto, J., Holm, L., 2015. PANNZER: high-throughput functional annotation of uncharacterized proteins in an error-prone environment. Bioinformatics 31, 1544–1552.
Kulmanov, M., Khan, M.A., Hoehndorf, R., 2017. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. Bioinformatics 34, 660–668.
Makrodimitris, S., Van Ham, R.C.H.J., Reinders, M.J.T., 2019. Improving protein function prediction using protein sequence and GO-term similarities. Bioinformatics.
Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space.
Pérez, A.J., Perez-Iratxeta, C., Thode, G., Andrade, M.A., Bork, P., 2004. Gene annotation from scientific literature using mappings between keyword systems. Bioinformatics 20, 2084–2091.
Perozzi, Bryan, Al-Rfou, R., Skiena, S., 2014. DeepWalk: Online Learning of Social Representations.
Radivojac, P., Clark, W.T., Oron, T.R., Schnoes, A.M., Wittkop, T., Sokolov, A., et al., 2013. A large-scale evaluation of computational protein function prediction. Nat. Methods 10, 221–227.
Raychaudhuri, S., Chang, J.T., Sutphin, P.D., Altman, R.B., 2002. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. Genome Res. 12 (January), 203–214.
Shatkay, H., Brady, H.S., Blum, T., Doennes, P., Kohlbacher, O., 2007. SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. Bioinformatics 23, 1410–1417.
Shatkay, H., Brady, S., Wong, A., 2015. Text as data: using text-based features for proteins representation and for computational prediction of their characteristics. Methods 74, 54–64, 2015/03/01/.
Shen, Zhen, Bao, Wen-Zheng, Huang, D.S., 2018. Recurrent neural network for predicting transcription factor binding sites. Sci. Rep. 8, 15270.
Sokolov, A., Ben-Hur, A., 2010. Hierarchical classification of gene ontology terms using the gostruct method. J. Bioinform. Comput. Biol. 08, 357–376.
Van, L.S., Hakala, K., Rönnqvist, S., Salakoski, T., Van, d.P.Y., Ginter, F., 2014. Exploring biomolecular literature with EVEX: connecting genes through events, homology, and indirect associations. Adv. Bioinformatics 2012, 582765.
Wei, L., Xing, P., Tang, J., Zou, Q., 2017a. PhosPred-RF: a novel sequence-based predictor for phosphorylation sites using sequential information only. IEEE Trans. Nanobioscience 16, 240–247.
Wei, L., Xing, P., Su, R., Shi, G., Ma, Z.S., Zou, Q., 2017b. CPPred-RF: a sequence-based predictor for identifying cell-penetrating peptides and their uptake efficiency. J. Proteome Res. 16, 2044–2053.
Wong, A., Shatkay, H., 2013. Protein function prediction using text-based features extracted from the; biomedical literature: the CAFA challenge. BMC Bioinformatics 14, S14.
You, R., Zhang, Z., Xiong, Y., Sun, F., Mamitsuka, H., Zhu, S., 2018a. GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. Bioinformatics 34.
You, R., Yao, S., Huang, X., Sun, F., Mamitsuka, H., Zhu, S., 2018b. NetGO: Improving Large-scale Protein Function Prediction With Massive Network Information, p. 439554.
You, Ronghui, Zhu, S., 2017. DeepText2Go: improving large-scale protein function prediction with deep semantic text representation. 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) IEEE.
Zhou, N., Jiang, Y., Bergquist, T.R., et al., 2019. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. Genome Biol. 20, 244.