Bioinformatics, 36(2), 2020, 422–429
doi: 10.1093/bioinformatics/btz595
Advance Access Publication Date: 27 July 2019
Original Paper

Sequence analysis

DeepGOPlus: improved protein function prediction from sequence

Maxat Kulmanov and Robert Hoehndorf *

Computational Bioscience Research Center, Computer, Electrical and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on April 25, 2019; revised on July 1, 2019; editorial decision on July 20, 2019; accepted on July 24, 2019

Abstract

Motivation: Protein function prediction is one of the major tasks of bioinformatics that can help in wide range of biological problems such as understanding disease mechanisms or finding drug targets. Many methods are available for predicting protein functions from sequence based features, protein–protein interaction networks, protein structure or literature. However, other than sequence, most of the features are difficult to obtain or not available for many proteins thereby limiting their scope. Furthermore, the performance of sequence-based function prediction methods is often lower than methods that incorporate multiple features and predicting protein functions may require a lot of time.

Results: We developed a novel method for predicting protein functions from sequence alone which combines deep convolutional neural network (CNN) model with sequence similarity based predictions. Our CNN model scans the sequence for motifs which are predictive for protein functions and combines this with functions of similar proteins (if available). We evaluate the performance of DeepGOPlus using the CAFA3 evaluation measures and achieve an F_{max} of 0.390, 0.557 and 0.614 for BPO, MFO and CCO evaluations, respectively. These results would have made DeepGOPlus one of the three best predictors in CCO and the second best performing method in the BPO and MFO evaluations. We also compare DeepGOPlus with state-of-the-art methods such as DeepText2GO and GOLabeler on another dataset. DeepGOPlus can annotate around 40 protein sequences per second on common hardware, thereby making fast and accurate function predictions available for a wide range of proteins.

Availability and implementation: `http://deepgoplus.bio2vec.net/`.

Contact: robert.hoehndorf@kaust.edu.sa

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Prediction of protein functions is a major task in bioinformatics that is important in understanding the role of proteins in disease pathobiology, the functions of metagenomes, or finding drug targets. A wide range of methods have been developed for predicting protein functions computationally (Fa *et al.*, 2018; Jiang *et al.*, 2016; Kahanda and Ben-Hur, 2017; Kulmanov *et al.*, 2018; Radivojac *et al.*, 2013; You *et al.*, 2018a,b; Zhou *et al.*, 2019). Protein functions can be predicted from protein sequences (Fa *et al.*, 2018; Jiang *et al.*, 2016; Kulmanov *et al.*, 2018; Radivojac *et al.*, 2013; You *et al.*, 2018a, b; Zhou *et al.*, 2019), protein–protein interactions (PPI) (Kulmanov *et al.*, 2018), protein structures (Yang *et al.*, 2015), biomedical literature and other features (Kahanda and Ben-Hur, 2017; You *et al.*, 2018a). Sequence-based methods employ sequence similarity, search for sequence domains, or multi-sequence alignments to infer functions. As proteins rarely function on their own, protein–protein interactions can be a good predictor for complex biological processes to which proteins contribute. Although it is

experimentally challenging to identify protein structures, they are crucial in understanding what proteins are capable of doing. Literature may contribute to function predicting because it may contain explicit descriptions of protein functions or describe properties of proteins that are predictive of protein functions indirectly. Overall, many of these features are available only for a small number of proteins, while a protein's amino acid sequence can be identified for most proteins. Therefore, methods that accurately predict protein functions from sequence alone may be the most general and applicable to proteins that have not been extensively studied.

Proteins with similar sequence tend to have similar functions (Radivojac *et al.*, 2013). Therefore, a basic way of predicting functions for new sequences is to find the most similar sequences with known functional annotations and transfer their annotations. Another approach is to search for specific sequence motifs which are associated with some function; for example, InterProScan (Mitchell *et al.*, 2014) is a tool which can help to find protein domains and families. The domains and families can be used to infer protein functions.

Recent developments in deep feature learning methods brought many methods which can learn protein sequence features. In 2017, we developed DeepGO (Kulmanov *et al.*, 2018) as one of first deep learning models which can predict protein functions using the protein amino acid sequence and interaction networks. Since 2017, many successor methods became available that achieve better predictive performance (You *et al.*, 2018a, b).

DeepGO suffers from several limitations. First, it can only predict functions for proteins with a sequence length less than 1002 and which do not contain 'ambiguous' amino acids such as unions or unknowns. While around 90% of protein sequences in UniProt satisfy these criteria, it also means that DeepGO could not predict functions for about 10% of proteins. Second, due to computational limitations, DeepGO can only predict around 2000 functions out of more than 45 000 which are currently in the Gene Ontology (GO) (Ashburner *et al.*, 2000). Third, DeepGO uses interaction network features which are not available for all proteins. Specifically, for novel or uncharacterized proteins, only the sequence may be known and not any additional information such as the protein's interactions or mentions in literature. Finally, DeepGO was trained and evaluated on randomly drawn training, validation and testing sets. However, such models may overfit to particular features in the training data and may not yield adequate results in real prediction scenarios. Consequently, challenges such as the Critical Assessment of Function Annotation (CAFA) (Jiang *et al.*, 2016; Radivojac *et al.*, 2013; Zhou *et al.*, 2019) use a time-based evaluation where training and predictions are fixed and evaluated after some time has elapsed on predictions that became available in that time. DeepGO did not achieve the same performance in the CAFA3 (Zhou *et al.*, 2019) challenge as it had in our own experiments.

Here, we extend and improve DeepGO overcoming its main limitations related to sequence length, missing features and number of predicted classes. We increased the model's input length to 2000 amino acids and now cover more than 99% of sequences in UniProt. Furthermore, our new model's architecture allows us to split longer sequences and scan smaller chunks to predict functions. We also remove features derived from interaction networks because only a small number of proteins have such network information. Instead, we combine our neural network predictions with methods based on sequence similarity to capture orthology and, indirectly, some interaction information. Through this step we also overcome the limitation in the number of classes to predict and we can, in theory, predict any GO class that has ever been used in an experimental annotation. To avoid overfitting of our model, we substantially decreased our model's capacity by replacing the amino acid trigram embedding layer with a one-hot encoding and removing our hierarchical classification layer.

In our evaluation we exactly reproduce the CAFA3 evaluation by training our model using only data provided by CAFA3 as training data and evaluating on the CAFA3 testing data. Using the publicly available CAFA Assessment Tool, DeepGOPlus achieves an F_{max} of 0.390, 0.557 and 0.614 for BPO, MFO and CCO evaluations, respectively. These results would have made DeepGOPlus the one of three best predictors in CCO and the second best performing method in the BPO and MFO evaluations.

We also compare DeepGOPlus with our baseline methods including DeepGO and two of the best-performing protein function prediction methods, GOLabeler (You *et al.*, 2018b) and DeepText2GO (You *et al.*, 2018a), on another dataset. GOLabeler mainly uses sequence-based features, DeepGO uses interaction network features, and DeepText2GO uses features extracted from literature in addition to sequence-based ones. In terms of F_{max} measure, we outperform all methods in predicting biological processes and cellular components. Notably, our model significantly improves predictions of biological process annotations with an F_{max} of 0.474.

To provide an insight into what kind of features our model uses to predict functions, we analyze the convolutional filters of our model to understand what type of feature they recognize. We found that sequence regions that activate our filters are very similar to seed sequences of protein families and domains in the Pfam database (El-Gebali *et al.*, 2019). We were able to associate protein sequences in our test set with almost half of their InterPro (Finn *et al.*, 2017) annotations by using sequence regions which activate our convolutional filters.

By using a single model with few parameters, we also significantly improved the runtime of the model. In average, DeepGOPlus can annotate 40 proteins per second on ordinary hardware. Overall, with these improvements, our model can now rapidly perform function prediction for any protein with available sequence. Our online predictor is available at `http://deepgoplus.bio2vec.net` and DeepGOPlus.

2 Materials and methods

2.1 Datasets and gene ontology

We use two datasets to evaluate our approach. Firstly, we downloaded CAFA3 challenge training sequences and experimental annotations published on September, 2016 and test benchmark published on November 15, 2017 which was used to evaluate protein function prediction methods submitted to the challenge. According to CAFA3, the annotations with evidence codes: EXP, IDA, IPI, IMP, IGI, IEP, TAS, or IC are considered to be experimental. The training set includes all proteins with experimental annotations known before September, 2016 and the test benchmark contains no-knowledge proteins which gained experimental annotation between September, 2016 and November 2017. Similar time based splits were used in all previous CAFA challenges.

We propagate annotations using the hierarchical structure of the Gene Ontology (GO) (Ashburner *et al.*, 2000). We use the version of GO released on June 1, 2016. The version has 10 693 molecular function (MFO), classes, 29 264 biological process (BPO) classes and 4034 cellular component (CCO) classes. This version is also used to evaluate CAFA3 predictions. While propagating

Table 1. The number of protein sequences with experimental annotations in CAFA3 and 2016 datasets grouped by sub-ontologies

Dataset	Statistic	MFO	BPO	CCO	All
CAFA3	Training size	36 110	53 500	50 596	66 841
CAFA3	Testing size	1137	2392	1265	3328
CAFA3	Number of classes	677	3992	551	5220
2016	Training size	34 488	51 716	49 346	65 028
2016	Testing size	679	1434	1148	1788
2016	Number of classes	652	3904	545	5101

annotations, we consider all types of relations between classes. For instance, if a protein P is annotated with a class C which has a part-of relation to a class D, then we annotate P with the class D. This procedure is repeated until no further annotation can be propagated. After this step, we count the number of annotated proteins for each GO class and select all classes with 50 or more annotations for our prediction model. The statistics with the number of classes in Table 1 represent how many classes we can predict using our deep neural network model.

Secondly, to compare with other methods for function prediction such as DeepText2GO (You *et al.*, 2018a) and GoLabeler (You *et al.*, 2018b) we downloaded SwissProt reviewed proteins published on January, 2016 and October, 2016. We use all experimental annotations before January 2016 as a training set and experimental annotations collected between January and October 2016 as testing set. We filter the testing set with 23 target species which are in CAFA3 evaluation set. Table 1 summarizes both datasets.

2.2 Baseline comparison methods

2.2.1 Naive approach

It is possible to get comparable prediction results just by assigning the same GO classes to all proteins based on annotation frequencies. This happens due to the hierarchical structure of GO which, after the propagation process, results in many annotations at high-level classes. In CAFA, this approach is called 'naive' approach and is used as one of the baseline methods to compare function predictions. Here, each query protein p is annotated with the GO classes with a prediction scores computed as:

$$S(p,f) = \frac{N_f}{N_{total}} \tag{1}$$

where f is a GO class, N_f is a number of training proteins annotated by GO class f and N_{total} is a total number of training proteins.

2.2.2 DiamondBLAST

Another baseline method is based on sequence similarity score obtained by BLAST (Altschul *et al.*, 1997). The idea is to find similar sequences from the training set and transfer an annotation from the most similar. We use the normalized bitscore as prediction score for a query sequence q:

$$S(q,f) = \frac{\max_{s\in E} bitscore(q,s) * I(f \in T_s)}{\max_{s\in E} bitscore(q,s)} \tag{2}$$

where E is a set of similar sequences filtered by e-value of 0.001, T_s is a set of true annotations of a protein with sequence s and I is an identity function which returns 1 if the condition is true and 0 otherwise.

2.2.3 DiamondScore

The DiamondScore is very similar to the DiamondBLAST approach. The only difference is that we normalize the sum of the bitscores of similar sequences. We compute prediction scores using the formula:

$$S(q,f) = \frac{\sum_{s\in E} bitscore(q,s) * I(f \in T_s)}{\sum_{s\in E} bitscore(q,s)} \tag{3}$$

2.2.4 DeepGO

DeepGO (Kulmanov *et al.*, 2018) was developed by us previously and it is one of the first methods which learns sequence features with a deep learning model and combines it with PPI network features to predict protein functions. It also uses a hierarchical classifier to output predictions consistent with structure of GO. Here we trained three separate models for three parts of GO mainly because of the computational costs involved in training larger models. We use our previously reported optimal parameters and set of functions to train new models with our current datasets. With DeepGO, we trained and predicted 932 BPO, 589 MFO and 436 CCO classes.

2.2.5 GOLabeler and DeepText2GO

Currently the best performing methods for function prediction task are GOLabeler (You *et al.*, 2018b) and DeepText2GO (You *et al.*, 2018a), both developed by the same group. GOLabeler achieved some of the best results in the preliminary evaluation for all three subontologies of GO in the CAFA3 challenge. It is an ensemble method which combines several approaches and predicts functions mainly from sequence features. DeepText2GO improves the results achieved by GOLabeler by extending their ensemble with models that predict functions from literature.

Our second dataset is specifically designed to compare our results with these two methods. Since we use same training and testing data, we directly compare our results with the results reported in their papers.

2.3 Model training and tuning

We use Tensorflow (Abadi *et al.*, 2016) to build and train our neural network model. Our model was trained on Nvidia Titan X and P6000 GPUs with 12–24 Gb of RAM.

Our neural network model has many hyperparameters such as convolutional filter lengths, number of convolutional filters, depth of fully connected layers, loss functions, activation functions, optimizers and learning rate. In addition, we use weighted sum model to combine sequence similarity method score with neural network model score which has α parameter to be tuned. In general, all parameters were tuned depending on their performance on a validation set which is a randomly split 10% of our training set. Since the parameter search space is quite large we evaluated several loss functions, activation functions, optimizers and learning rate on a simple model and selected binary cross-entropy loss and Adam (Kingma and Ba, 2014) with learning rate of 0.0003. We selected ReLU (Nair and Hinton, 2010) activations for intermediate layers and used Sigmoid function for our final classification layer. Then, we ran an extensive search for the other parameters. Our model uses multiple 1D convolutional layers with different filter lengths where the smallest filter starts from length 8 and the following filter is increased by 8 units. The tested settings were {{8, 16, 24, 32}, {8, 16, 24,. . ., 64}, {8, 16, 24,. . ., 128}, {8, 16, 24,. . ., 256}, {8, 16, 24,. . ., 512}} where each layer's number of filters were selected from 32, 64, 128, 256, 512. The depth of fully connected layers were selected from {1, 2, 3}. We tested all combinations of these parameters (in total 75) and

the best performing parameters were convolutional layers with filter lengths {8, 16, 24,..., 128} with 512 filters each and 1 fully connected layer. This setting generated 8192 (16×512) convolutional filter outputs which were used as a sequence features. The α parameter values which give best performance on a validation set are 0.55, 0.59 and 0.46 for MFO, BPO and CCO evaluations respectively.

To avoid overfitting we use an early stopping strategy depending on the validation loss. Our CNN layers do not use any activation function or dropout because we use MaxPooling layer with maximum pool size. This means that every filter will return only a single value. The aim is to force the CNN filters to learn set of similar patterns (motifs) and if the filter finds the pattern in the sequence it returns a high value which is pooled with the MaxPooling layer. We obtained our best model with only one fully connected layer after the MaxPooling layer. This makes our model relatively simple and less prone to overfitting.

2.4 DeepGOPlus versus DeepGO

There are three main differences of our model from the original DeepGO model (Kulmanov *et al.*, 2018). First, DeepGO uses a trigram embedding layer to represent the sequence. The embedding layer has vectors of size 128 for each trigram ($20 \times 20 \times 20$ in total). This layer adds 128×8000 parameters to the model. We replaced this representation with a parameter-free one-hot encoding which allowed us to significantly reduce the number of parameters of the new model. We noticed during our experiments that the models with embedding layer easily memorize the training data and overfit to it. Using one-hot encoding helped to avoid this problem. Second, DeepGO has one CNN layer with fixed a filter length which was extended to several CNN layers with different filter lengths. Finally, in DeepGOPlus, we use a flat classification layer instead of hierarchical classifier in DeepGO. The reason for this choice is that we built a single model for all three ontologies with more than 5000 classes and we were not able to build a hierarchical classifier due to memory limitations and time complexities.

2.5 Evaluation

To evaluate our predictions we use the CAFA (Radivojac *et al.*, 2013) evaluation metrics $F_{\max}$ and $S_{\min}$ (Radivojac and Clark, 2013). In addition, we report the area under the precision–recall curve (AUPR) which is a reasonable measure for evaluating predictions with high class imbalance (Davis and Goadrich, 2006).

$F_{\max}$ is a maximum protein-centric F-measure computed over all prediction thresholds. First, we compute average precision and recall using the following formulas:

$$pr_i(t) = \frac{\sum_f I(f \in P_i(t) \wedge f \in T_i)}{\sum_f I(f \in P_i(t))} \tag{4}$$

$$rc_i(t) = \frac{\sum_f I(f \in P_i(t) \wedge f \in T_i)}{\sum_f I(f \in T_i)} \tag{5}$$

$$AvgPr(t) = \frac{1}{m(t)} \cdot \sum_{i=1}^{m(t)} pr_i(t) \tag{6}$$

$$AvgRc(t) = \frac{1}{n} \cdot \sum_{i=1}^{n} rc_i(t) \tag{7}$$

where f is a GO class, T_i is a set of true annotations, $P_i(t)$ is a set of predicted annotations for a protein i and threshold t, $m(t)$ is a number of proteins for which we predict at least one class, n is a total number of proteins and I is an identity function which returns 1 if the condition is true and 0 otherwise. Then, we compute the $F_{\max}$ for prediction thresholds $t \in [0, 1]$ with a step size of 0.01. We count a class as a prediction if its prediction score is higher than t:

$$F_{\max} = \max_t \left\{ \frac{2 \cdot AvgPr(t) \cdot AvgRc(t)}{AvgPr(t) + AvgRc(t)} \right\} \tag{8}$$

$S_{\min}$ computes the semantic distance between real and predicted annotations based on information content of the classes. The information content $IC(c)$ is computed based on the annotation probability of the class c:

$$IC(c) = -\log(Pr(c|P(c)) \tag{9}$$

where $P(c)$ is a set of parent classes of the class c. The $S_{\min}$ is computed using the following formulas:

$$S_{\min} = \min_t \sqrt{ru(t)^2 + mi(t)^2} \tag{10}$$

where $ru(t)$ is the average remaining uncertainty and $mi(t)$ is average misinformation:

$$ru(t) = \frac{1}{n} \sum_{i=1}^{n} \sum_{c \in T_i - P_i(t)} IC(c) \tag{11}$$

$$mi(t) = \frac{1}{n} \sum_{i=1}^{n} \sum_{c \in P_i(t) - T_i} IC(c) \tag{12}$$

In our evaluation, we consider the complete GO ontology when computing parent and child classes, and then separate classes into their individual sub-ontologies (MFO, BPO, CCO) based on the `namespace` attribute associated with classes in GO. This method is also used by GOLabeler (You *et al.*, 2018b) and DeepText2GO (You *et al.*, 2018a). In the CAFA3 evaluation (Zhou *et al.*, 2019), individual classes are first separated by their sub-ontology and parent and child classes are then computed locally within the sub-ontology. GO has relations between classes in the three sub-ontologies, and many MFO classes stand in a 'part-of' relation to BPO classes or in 'occurs-in' relations with CCO classes. For example, the MFO class *acyl carrier activity* (`GO: 0000036`) stands in a 'part-of' relation to *fatty acid biosynthetic process* (`GO: 0006633`) in the BPO ontology, and while we take this class into account when computing our evaluation measures, the CAFA3 evaluation does not. To compare our results with the CAFA3 evaluation results (Zhou *et al.*, 2019), we perform the evaluation twice, using the complete ontology and using the separate evaluation as in CAFA3 (using the publicly available CAFA3 evaluation tool).

3 Results

3.1 DeepGOPlus learning model

In DeepGOPlus, we combine sequence similarity and sequence motifs in a single predictive model. To learn sequence motifs that are predictive of protein functions, we use one-dimensional convolutional neural networks (CNNs) over protein amino acid sequence to learn sequence patterns or motifs. Figure 1 describes the architecture of our deep learning model. First, the input sequence is converted to a one-hot encoded representation of size 21×2000, where a one-hot vector of length 21 represents an amino acid (AA) and 2000 is the input length. Sequences with a length less than 2000 are padded with zeros and longer sequences are split into smaller chunks with less than 2000 AAs. This input is passed to a set of CNN layers with different filter sizes of 8, 16, ..., 128. Each of the CNN layers has 512 filters which learn specific sequence motifs of a particular size.

Each filter is scanning the sequence and their maximum score is pooled using a MaxPooling layer. In total, we generate a feature vector of size 8192 where each value represents a score that indicates the presence of a relevant sequence motif. This vector is passed to the fully connected classification layer which outputs the predictions. To select the best parameters and hyperparameters for our deep learning model, we extensively searched for optimal combinations of parameters such as filter sizes, number of filters and depth of dense layers based on a validation set loss. We report the list of parameters and validation losses in Supplementary Table S1.

DeepGOPlus combines the neural network model predictions with predictions based on sequence similarity. First, we find similar sequences from a training set using Diamond (Buchfink *et al.*, 2015) with an *e*-value of 0.001 and obtain a bitscore for every similar sequence. We transfer all annotations of similar sequences to a query sequence with prediction scores computed using the bitscores. For a set of similar sequences E of the query sequence q, we compute the prediction score for a GO class f as

$$S(q,f) = \frac{\sum_{s \in E} I(f \in T_s) * bitscore(q,s)}{\sum_{s \in E} bitscore(q,s)},$$

where T_s is a set of true annotations of the protein with sequence s. Then, to compute the final prediction scores of DeepGOPlus, we combine the two prediction scores using a weighted sum model (Fishburn, 1967):

$$S = \alpha * S_{DiamondScore} + (1 - \alpha) * S_{DeepGOCNN},$$

where $0 \leq \alpha \leq 1$ is a weight parameter which balances the relative importance of the two prediction methods.

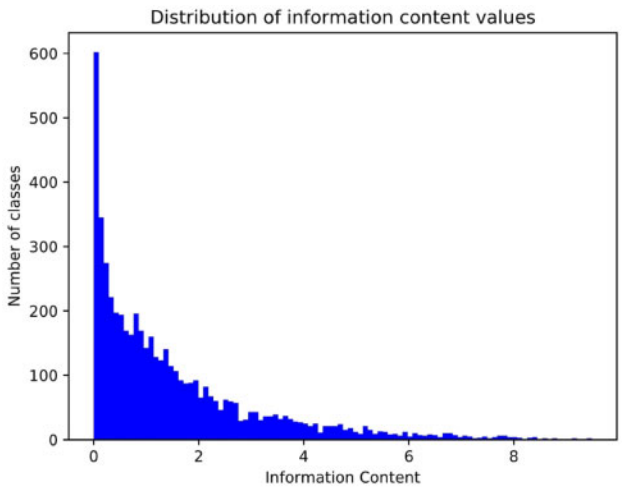

Fig. 1. Overview of the CNN in DeepGOPlus. The CNN uses multiple filters of variable size to detect the presence of sequence motifs in the input amino acid sequence

3.2 Evaluation and comparison

We evaluate DeepGOPlus using two datasets. First, we use the latest CAFA3 (Radivojac *et al.*, 2013) challenge dataset and compare our method with baseline methods such as Naive predictions, BLAST and our previous deep learning model DeepGO. We use two strategies for predicting functions based on sequence similarity computed with the Diamond tool (Buchfink *et al.*, 2015) (which is a faster implementation of the BLAST algorithm). We call them DiamondBLAST and DiamondScore. DiamondBLAST considers only the most similar sequence whereas DiamondScore predicts functions using all similar sequences returned by Diamond. We also report the performance of using only our neural network model (labeled as DeepGOCNN). We find that with the DiamondScore approach, we can outperform DeepGO predictions in MFO and achieve comparable results in BPO and CCO evaluations while DeepGOCNN gives better predictions in CCO. We achieve the best performance in all three subontologies with our DeepGOPlus model which combines the DiamondScore and DeepGOCNN. Table 2 summarizes the performance of the models.

To compare our approach with the state of the art methods GOLabeler (You *et al.*, 2018b) and DeepText2GO (You *et al.*, 2018a), we generate a second dataset which uses data obtained at the same dates as the other methods so that we can generate a time-based split of training and testing data. Both methods train on experimental function annotations that appeared before January 2016 and test on annotations which were asserted between January 2016 and October 2016. Furthermore, we use the same version of GO and follow the CAFA3 challenge procedures to process the data. As a result, we can directly compare our evaluation results with the other methods. In this evaluation, DeepGOPlus gives the best results for BPO and CCO in terms of F_{max} measure and ranks second in the MFO evaluation (after DeepText2GO). However, it is important to note that DeepText2GO uses features extracted from literature in addition to sequence based features while DeepGOPlus predictions are only based on protein sequence. Notably, our method significantly increased performance of predictions of BPO classes in both evaluation datasets (Table 3).

Due to large number of available sequences, analyzing sequences require both accurate and fast prediction methods. Specifically, function prediction is a crucial step in interpretation of newly sequenced genomes or meta-genomes. While we have compared DeepGOPlus in terms of prediction performance, we could not compare the running time of the models because the runtime of prediction models is rarely reported. With DeepGOPlus, 40 protein sequences can be annotated per second using a single Intel(R) Xeon(R) E5-2680 CPU and Nvidia P6000 GPU.

3.3 Comparison with CAFA3 methods

The CAFA3 challenge results (Zhou *et al.*, 2019) became available recently which allowed us to evaluate our method on the same

Table 2. The comparison of performance on the first CAFA3 challenge dataset

Method	F_{max}			S_{min}			AUPR		
	MFO	BPO	CCO	MFO	BPO	CCO	MFO	BPO	CCO
Naive	0.290	0.357	0.562	10.733	25.028	8.465	0.130	0.254	0.456
DiamondBLAST	0.431	0.399	0.506	10.233	25.320	8.800	0.178	0.116	0.142
DiamondScore	0.509	0.427	0.557	9.031	22.860	8.198	0.340	0.267	0.335
DeepGO	0.393	0.435	0.565	9.635	24.181	9.199	0.303	0.385	0.579
DeepGOCNN	0.420	0.378	0.607	9.711	24.234	8.153	0.355	0.323	0.616
DeepGOPlus	**0.544**	**0.469**	**0.623**	**8.724**	**22.573**	**7.823**	**0.487**	**0.404**	**0.627**

Best performance in bold. F_{max} and AUPR, highest; S_{min}, lowest.

Table 3. The comparison of performance on the second dataset generated by a time-based split

Method	F_{max}			S_{min}			AUPR		
	MFO	BPO	CCO	MFO	BPO	CCO	MFO	BPO	CCO
Naive	0.306	0.318	0.605	12.105	38.890	9.646	0.150	0.219	0.512
DiamondBLAST	0.525	0.436	0.591	9.291	39.544	8.721	0.101	0.070	0.089
DiamondScore	0.548	0.439	0.621	8.736	34.060	7.997	0.362	0.240	0.363
DeepGO	0.449	0.398	0.667	10.722	35.085	7.861	0.409	0.328	0.696
DeepGOCNN	0.409	0.383	0.663	11.296	36.451	8.642	0.350	0.316	0.688
DeepText2GO	**0.627**	0.441	0.694	5.240	17.713	**4.531**	**0.605**	0.336	**0.729**
GOLabeler	0.580	0.370	0.687	**5.077**	**15.177**	5.518	0.546	0.225	0.700
DeepGOPlus	0.585	**0.474**	**0.699**	8.824	33.576	7.693	0.536	**0.407**	0.726

Best performance in bold. F_{max} and AUPR, highest; S_{min}, lowest.

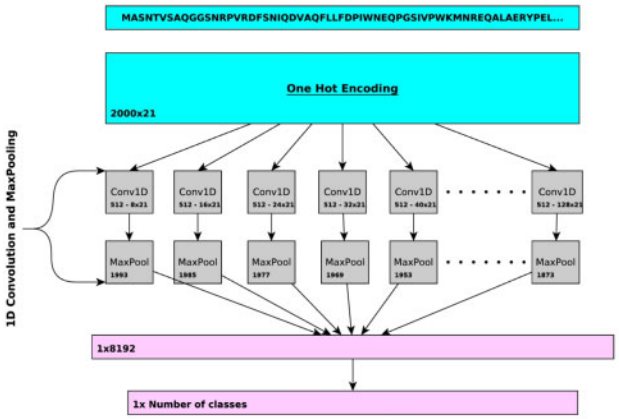

Fig. 2. Comparison of DeepGOPlus with CAFA3 top 10 methods

dataset used in CAFA3. We use the method used by CAFA3 to evaluate DeepGOPlus (see Methods) and compare against other methods that were evaluated in CAFA3. According to the CAFA3 evaluation measures, DeepGOPlus achieves an F_{max} of 0.390, 0.557 and 0.614 for BPO, MFO and CCO evaluations, respectively. These results would have made DeepGOPlus one of the three best predictors in CCO and the second best performing method in the BPO and MFO evaluations. Figure 2 shows the comparison of DeepGOPlus with all CAFA3 top performing methods.

Using our own evaluation (based on inferring parent and child classes over the complete GO ontology instead of separately in the sub-ontologies), we obtain similar results for MFO and CCO as with the CAFA3 evaluation method while the difference in the BPO evaluation is quite large.

3.4 S_{min} performance analysis

Although our method performed among the top-ranking methods in several F_{max} evaluations, both GOLabeler and DeepText2GO perform significantly better when considering the evaluation based on the S_{min} measure. DeepGOPlus can potentially predict any GO class, including classes that are very specific. The S_{min} evaluation depend on the number of false negatives, false positives and the information content (IC) of GO classes. Figure 3 shows the distribution of IC values for the classes that are predicted by the DeepGOPlus model. The IC for general classes is close to zero and more specific classes have IC values of close to 10. Consequently, the methods which attempt to predict many specific classes will, in general, have a higher S_{min}. To test if this is true for our method, we evaluated the false positive predictions for the CAFA3 test set. In average, our method predicts 6.1 false positive classes per protein with a total IC of 8.3 for MFO, demonstrating that our false positive predictions are quite specific.

3.5 Convolutional filters

To understand what is being learned by the convolutional filters of the CNN model we performed an experiment where we analyze sequence regions which activate our filters. We selected the specific molecular function class 'enzyme activator activity' (`GO: 0008047`) and filtered out all proteins annotated to this class. In total, 623 proteins have been experimentally annotated to this class. The reason we selected this class is that this class is very specific and it is referenced by multiple InterPRO functional domains and families. Our hypothesis is that our CNN filters recognize sequence regions that are similar to sequence functional domains.

First, we extracted the scores of all 8192 filters and ordered them in descending order for all sequences. We found that filter number 8048 (0-based) gives the highest score for all sequences and many other top 10 filters are active in more than 600 sequences. This shows that the filters learned similar motifs that are related to our selected function. We then extracted sequence regions which give the highest score for top 10 CNN filters and compare them to Pfam (El-Gebali *et al.*, 2019) protein families database seeds. We use Diamond BLAST (Buchfink *et al.*, 2015) with an e-value of 0.001 and associate similar sequence regions to Pfam families. Furthermore, we map Pfam family IDs to InterPRO IDs. Through this process, we associated protein sequences to InterPRO domains using sequence regions recognized by our CNN filters. In total, 563 sequences have been associated to at least one InterPRO ID. Finally, we compare the InterPRO associations with InterPRO annotations in UniProt (The Uniprot Consortium, 2007) database using a protein-centric F measure and obtain an F-score of 0.62 with precision 0.9 and recall of 0.47. This experiment demonstrates that our CNN filters are learning meaningful sequence motifs and can, for enzyme activators, accurately recognize almost half of the currently known functional domain annotations.

3.6 Implementation and availability

DeepGOPlus is available as free software at `https://github.com/bio-ontology-research-group/deepgoplus`. We also publish training and testing data used to generate evaluation and results at `http://deepgoplus.bio2vec.net/data/`. Furthermore, DeepGOPlus is available through a web interface and REST API at `http://deepgoplus.bio2vec.net`.

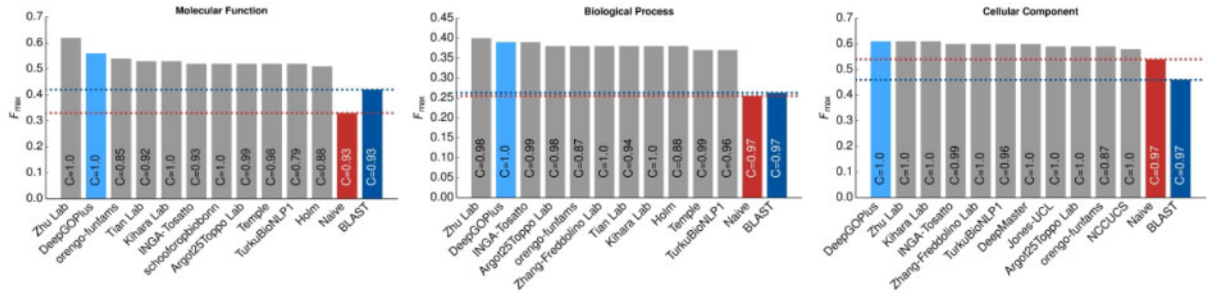

Fig. 3. Distribution of Information Content (IC) values for 5220 GO classes that are predicted for CAFA3 dataset

4 Discussion

DeepGOPlus is a fast and accurate tool to predict protein functions from protein sequence alone. Our model overcomes several limitations of other methods and our own DeepGO model (Kulmanov *et al.*, 2018). In particular, DeepGOPlus has no limits on the length of the amino acid sequence and can therefore be used for the genome-scale annotation of protein functions, in particular in newly sequenced organisms. DeepGOPlus also makes no assumptions on the taxa or kingdom to which a protein belongs, therefore enabling, for example, function prediction for meta-genomics in which proteins from different kingdoms may be mixed. Furthermore, DeepGOPlus is fast and can annotate several thousand proteins in minutes even on single CPUs, further enabling its application in meta-genomics or for projects in which a very large number of proteins with unknown functions are identified. While we initially expected the absence of features derived from interaction networks to impact predictive performance, we found that we can achieve even higher prediction accuracy with our current model; additionally, our model is not limited by unbalanced or missing information about protein–protein interactions.

In DeepGOPlus, we combine similarity-based search to proteins with known functions and motif-based function prediction, and this combination gives us overall the best predictive performance. However, DeepGOPlus can also be applied using only sequence motifs; in particular when annotating novel proteins for which no similar proteins with known functions exist, our motif-based model would be most suitable.

In the future, we plan to incorporate additional features and test other types of deep neural network models. While related methods use features that can be derived only for known proteins, such as information obtained from literature or interaction networks, DeepGOPlus will rely primarily on features that can be derived from amino acid sequences to ensure that the model can be applied as widely as possible. Possible additional information that may improve DeepGOPlus in the future is information about protein structure, in particular as structure prediction methods are improving significantly (Wang *et al.*, 2017). We have already experimented with several types of neural networks such as recurrent neural networks, long-short term memory networks and autoencoders to learn seqeunce features. However, our attempts were unsuccessful and CNNs gave us the best results. Recently, attention networks have been successfully applied to protein sequences (Rives *et al.*, 2019) and we plan to test them on function prediction.

Acknowledgements

We acknowledge the use of computational resources from the KAUST Supercomputing Core Laboratory.

Funding

This work was supported by funding from King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. URF/1/3454-01-01, URF/1/3790-01-01, FCC/1/1976-08-01 and FCS/1/3657-02-01.

Conflict of Interest: none declared.

References

Abadi,M. *et al.* (2016) Tensorflow: a system for large-scale machine learning. In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pp. 265–283. USENIX Association, Berkeley, CA, USA.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Buchfink,B. *et al.* (2015) Fast and sensitive protein alignment using diamond. *Nat. Methods*, **12**, 59.

Davis,J. and Goadrich,M. (2006) The relationship between precision–recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 233–240. ACM, New York, NY, USA.

El-Gebali,S. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.

Fa,R. *et al.* (2018) Predicting human protein function with multi-task deep neural networks. *PLoS One*, **13**, e0198216–16.

Finn,R.D. *et al.* (2017) Interpro in 2017—beyond protein family and domain annotations. *Nucleic Acids Res.*, **45**, D190.

Fishburn,P.C. (1967) Additive utilities with incomplete product sets: application to priorities and assignments. *Operations Res.*, **15**, 537–542.

Jiang,Y. *et al.* (2016) An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.*, **17**, 184.

Kahanda,I. and Ben-Hur,A. (2017) Gostruct 2.0: Automated protein function prediction for annotated proteins. In: *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, ACM-BCB '17, pp. 60–66. ACM, New York, NY, USA.

Kingma,D.P. and Ba,J. (2014) Adam: A method for stochastic optimization. In: *Published as a Conference Paper at the 3rd International Conference for Learning Representations*, San Diego, 2015.

Kulmanov,M. *et al.* (2018) DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, **34**, 660–668.

Mitchell,A. *et al.* (2014) InterProScan 5: genome-scale protein function classification. *Bioinformatics*, **30**, 1236–1240.

Nair,V. and Hinton,G.E. (2010). Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 807–814. Omnipress, USA.

Radivojac,P. and Clark,W.T. (2013) Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics*, **29**, i53–i61.

Radivojac,P. *et al.* (2013) A large-scale evaluation of computational protein function prediction. *Nat. Methods*, **10**, 221–227.

Rives,A. *et al*. (2019) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Cold Spring Harbor Laboratory*, In press.

The Uniprot Consortium (2007) The universal protein resource (uniprot). *Nucleic Acids Res*., **35**,

Wang,S. *et al*. (2017) Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Comput. Biol*., **13**, e1005324–34.

Yang,J. *et al*. (2015) The I-TASSER Suite: protein structure and function prediction. *Nat. Methods*, **12**, 7.

You,R. *et al*. (2018a) DeepText2GO: improving large-scale protein function prediction with deep semantic text representation. *Methods*, **145**, 82–90.

You,R. *et al*. (2018b) GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, **34**, 2465–2473.

Zhou,N. *et al*. (2019) The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *bioRxiv*.