DeepFunc: A deep learning framework for accurate prediction of protein functions from protein sequences and interactions

Fuhao Zhang[1], Hong Song[1], Min Zeng[1], Yaohang Li[1, 2], Lukasz Kurgan[3] and Min Li[1*]

[1]School of Computer Science and Engineering, Central South University, Changsha, 410083, P.R. China

[2]Department of Computer Science, Old Dominion University, Norfolk, USA.

[3]Department of Computer Science, Virginia Commonwealth University, USA.

Correspondence: Dr. Min Li, School of Computer Science and Engineering, Central South University, Changsha, 410083, China

E-mail: limin@mail.csu.edu

Abbreviations: **PPI**, Protein-Protein Interaction; **CAFA,** Critical Assessment of protein Function Annotation; **MF**, molecular function; **F_{max}**, max F-measure; **AvgPr**, average precision; **AvgRc**, average recall; **MCC**, Mathews Correlation Coefficient; **AUC**, Area Under Curve

Keywords: protein functions / deep learning / functional linkages / protein-protein interactions / protein sequences / protein domains.

Total number of words: about 5500 words

Abstract

Annotation of protein functions plays an important role in understanding life at the molecular level. High throughput sequencing produces massive numbers of raw proteins sequences and only about 1% of them have been manually annotated with functions. Experimental annotations of functions are expensive, time-consuming and do not keep up with the rapid growth of the sequence numbers. This motivates development of computational approaches that predict protein functions. We propose a novel deep learning framework, DeepFunc, which accurately predicts protein functions from protein sequence- and

Received: JANUARY 12, 2019; Revised: MARCH 18, 2019; Accepted: MARCH 29, 2019

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1002/pmic.201900019.

network-derived information. More precisely, DeepFunc uses a long and sparse binary vector to encode information concerning domains, families and motifs collected from the InterPro tool that are associated with the input protein sequence. We process this vector with two neural layers to obtain a low-dimensional and dense vector which is combined with topological information extracted from publically available protein-protein interactions (PPIs) and functional linkages. The combined information is processed by a deep neural network that predicts protein functions. We empirically and comparatively test DeepFunc on a benchmark testing dataset and the CAFA3 dataset. The experimental results demonstrate that DeepFunc outperforms current methods on the testing dataset and that it secures the highest F_{max} = 0.54 and AUC = 0.94 on the CAFA3 dataset.

Statement of significance of the study

Function annotation of proteins is crucial in molecular biology. However, existing computational methods usually focus on using one type of protein data (either protein sequences or PPI network) to predict protein functions, which may cause the loss of certain protein features. In this study, a powerful deep learning framework (DeepFunc) for predicting protein functions is proposed. By using the Deepwalk algorithm, InterProscan tool and deep learning architecture, DeepFunc extracts high-quality features of protein sequences and PPI networks. DeepFunc combines these features to predict protein functions and achieves better performance than BLAST and DeepGO.

1 Introduction

Proteins perform many cellular functions and play indispensable role in a large variety of biological processes [1]. Protein data is being produced at a fast and even increasing pace by high-throughput sequencing techniques but their functional understanding is lagging [2] [3]. Only about 1% of proteins has been probed experimentally and was manually annotated in the UniProt database [4]. Protein functions can be elucidated via in-vitro and in-vivo experiments [5]. However, these experimental methods are expensive, time-consuming, and do not scale with the growth of the number of protein data. This motivates the need to develop runtime-efficient and accurate computational methods that predict protein functions directly from protein data.

Many computational methods have been proposed to predict protein functions. Generally, researchers develop a pipeline to predict functions of proteins by using protein sequences according to the following steps: select useful features to encode input proteins, construct training and testing datasets, select an appropriate algorithm and evaluate the performance. One of the most popular computational methods is BLAST that uses functions of similar sequences to functionally annotate the input sequence. However, this approach has two limitations: 1) similar and functionally annotated proteins cannot be found for many input sequences; and 2) some proteins may have similar functions while having low sequence similarity. Thus, the results obtained by these homology-based approaches are not always accurate [6]. One way to overcome the challenge is to extract useful information from conserved subregions or residues in the input protein chain. For example, Das and his collaborators proposed a domain-based method to predict protein functions [7]. Wang and his collaborators proposed a motif-based protein function classifier [8]. Moreover, some methods predict protein functions utilizing residue-level information [9]. This information may include secondary structures extracted from input protein sequences [10], or secondary structure, disordered regions, signal peptides and motifs like in the case of the FFPred3 method [11]. Finally, several approaches rely on the PPI-derived information to accurately predict protein functions [12, 13] [14] [15] [16] [17] . The crucial idea behinds these methods is that proteins which share similar topological features in the PPI networks may share similar functions [18]. Moreover, some protein function predictors utilize other types of data, such as genetic interactions [5], genomic context [19], protein structure [20] [21] [22] [23], and gene expression [24] [25]. We focus on two classes of current predictors: sequence-based methods that cover the use of domains, motifs and residue-level information [26] [27], and PPI-based methods that rely on information extracted from these networks [17] [28] [29]. These two classes of methods utilize somehow complementary information. While topological information will be used to characterize protein functions based on protein-protein interactions, sequence-based methods could be effective in identifying proteins that incorporate signal peptides or transmembrane proteins [30], which are not necessarily easy to predict using PPIs.

This article explores the use of deep learning to efficiently process and combine the sequence-based and PPI-based approaches. While deep learning was shown to improve predictive performance in several related prediction problems [31] [32] [33] [34] [35] [36] [37], it was used only once in the context of combining these two types of information to predict proteins functions in the DeepGO model [28]. We design and comparatively test a novel

deep learning model called DeepFunc. Our sequence-based approach relies on generation of a high-dimensional vector of information (35 thousand dimensions) that describes domains, families and motifs which are extracted by InterPro [38]. These data must be reduced before they can be combined with a relatively low-dimensional data extracted from the PPI network. We combine functional linkages from EggNOG [39] and interactions from STRING [40] to construct the PPI network. We use the Deepwalk algorithm [41] to extract a comprehensive collection of topological features that describe the underlying PPI network. The innovative aspect of DeepFunc is the use to the deep network for two distinct purposes: to convert the high-dimensional sequence-based approach into an information-rich, low-dimensional format, and to effectively combine these data with the topological information obtained from the PPI network. This design is arguably better that DeepGO where the deep network is used only to combine the sequence-derived and PPI-derived information, and where the sequence-derived data is low -dimensional and relies on simple counting of *k*-mers. Consequently, comparative empirical analysis on multiple benchmark datasets reveals that DeepFunc outperforms DeepGO, as well as a few other representative function predictors, such as FFPred3 and GOPDR. The results demonstrate that the improved predictive performance is directly attributed to the extraction of high-quality sequence-based and PPI-based features. Moreover, DeepFunc obtain comparable results when testing on particularly challenging low similarity proteins.

2 Materials and Methods

2.1 Datasets and assessment metrics

We use data introduced in the DeepGO article [28] that is available at https://github.com/bio-ontology-research-group/deepgo. This benchmark dataset contains 60,710 proteins annotated with functions based on experimental evidence codes (EXP, IDA, IPI, IMP, IGI, IEP, TAS, and IC) that were filtered to exclude long sequences and sequences that contain ambiguous amino acid codes (B, O, J, U, X, and Z). This dataset includes 31,530 proteins with annotated molecular functions (MFs) and focuses on the top 589 MF terms that are assigned to at least 50 proteins. The dataset is divided into a training dataset (80% of randomly selected proteins) and a testing dataset (the remaining 20% of proteins). The training dataset contains 25,224 protein sequences and the testing dataset contains 6,306 protein sequences. Only the training dataset is used to parametrize DeepFunc, while the testing dataset is used to evaluate the already parametrized model. We select a subset of 20% of the training proteins to create validation dataset that is

used to empirically select the best parameters for the models trained using the remaining 80% of the training dataset, i.e., parameters are optimized to maximize predictive performance on the validation dataset. Moreover, we provide an independent (blind) empirical assessment on the dataset from CAFA3 and download the dataset and results from selected other predictors from https://github.com/bio-ontology-research-group/deepgo.

We evaluate predictive performance with three commonly used measures that include, average precision (AvgPr), average recall (AvgRc), and F_{max} that are used in the CAFA challenge [42]. Moreover, we use two additional measures, AUC and Mathews Correlation Coefficient (MCC), which were utilized in recent related studies [28] [43] [44] [45] [46].

2.2 Architecture of DeepFunc

The architecture of the DeepFunc framework for the prediction of protein functions is shown in Figure 1. We process the InterPro outputs using two fully connected neural layers to extract small and dense vector of sequence-based features. Concurrently, we utilize the Deepwalk algorithm to capture topological features of the PPI network in the vicinity of the input protein sequences. The feature vectors produced from the PPI network and from the sequence are concatenated and fed into a fully connected deep network that predicts protein functions.

2.2.1 Extraction of the sequence-derived features

A variety of features computed from sequences-based features were used in the prediction of protein functions. They include sequence similarity [9] [47] , *k*-mer frequencies [48] and presence of certain sub-sequences [49]. Our approach is to encode the raw protein sequence by the vector of protein families, domains and motifs (sub-sequences) that are obtained by InterPro resource. InterPro releases 70.0 contains 35,020 entries and combines diverse information coming from 14 databases, such as CCD [50], Pfam [51], CATH-Gene3D [52], and SUPERFAMILY [53]. It provides InterProScan package (http://www.ebi.ac.uk/interpro/download.htm) that scans protein sequences and annotates information about the input sequences with 2,865 superfamilies, 21,695 families, 9,268 domains, 280 repeats and 912 sites. Then, we use InterPro to encode the families, domains and motifs information of input protein sequence into a 35,020-dimensional binary vector. In this vector, 1 means that this sequence was assigned to a given

superfamily/family/domain or has a given repeat or site, otherwise, we assign the value of 0. This sparse and high-dimensional vector (all but a handful of the thousands of values equal 0) is not suitable as an input for the deep network. Thus, we use two fully connected neural layers to convert this vector into a substantially shorter and dense feature vector that can be effectively used to predict protein functions. The 35,020-dimensional binary input vector is processed by two fully connected layers of 1024 neurons that output 512-dimensional vector. The two layers are defined using the ReLU activation functions:

$$a^l = \sigma(wa^{l-1} + b) \quad (1)$$

where a^l is the output of given fully connected layer, a^{l-1} is the corresponding input, σ is a nonlinear activation function, *w* is a weight matrix, and b is the bias term. The values of *w* and *b* are optimized using the training dataset and the back-propagation algorithm.

2.2.2 Extraction of the PPI network-derived features

We construct the PPI network with the help of the STRING and EggNOG resources. We download STRING data on June 7, 2018 from https://string-db.orgm and the EggNOG data on June 10, 2018 from http://eggnogdb.embl.de/#/app/downloads. The network is based on the functionally annotated proteins collected from SwissProt on June 7, 2018. We map the SwissProt identifiers into the STRING records and we select a subset of these proteins that have interaction confidence score of at least 300. It is worthy noting that there are some proteins without interaction information in the STRING database. Thus, we use functional linkages from EggNOG database as missing interaction information to construct the PPI network. Specifically, if two nodes both have interaction information and functional linkage, we use the interaction information as the edge between them. In terms of two nodes without interaction information, we regard functional linkage as the interaction information. By using this strategy, we improve coverage of PPI interactions by adding functional linkages from the EggNOG database. The resulting network contains 354,687 nodes and 54,552,077 edges.

Networks have been widely used to model the structure of various biological systems and play an important role in biological prediction problems [54] [55] [56]. Thus, we aim to extract PPI network characteristics that are useful for the prediction of protein functions. Deep learning techniques were recently used to analyse

several network-based datasets [57] [58] [59]. A few new representations of such datasets have been proposed by drawing from the natural language processing area, such as node2vec [60], LINE [61] and Deepwalk techniques [41]. We apply the Deepwalk method motivated by a couple of recent studies that used similar random walk approaches to capture topological features of PPI networks [13, 62]. Deepwalk uses each vertex (protein) as the starting point to traverse nearby vertices by using a random walk algorithm. It applies the Skip-Gram model [63] to characterize the surrounding vertices for each given central vertex by maximizing the co-occurrence likelihood between the central vertex and its neighbours. This model generates a dense, low-dimensional vector for each vertex in the PPI network that represents topological features of the underlying PPI network. In order to cover all neighbors of a central vertex as many as possible, we use a sampling method. The Formula is as follows:

$$\left(1-\frac{1}{p}\right)^{k} \le \alpha \qquad (2)$$

p is the ratio of edges to vertices. After *k* iterations starting from a central vertex to perform random walks, the probability that one neighbor of the central vertex is not picked at least once is small than α. In this study, our PPI network has 354,687 vertices, and 54,552,077 edges. We set α=0.1, and the approximate value of walk number is 300. Using the training dataset, we iteratively compute the Deepwalk model that is parametrized with the walk-length = 20, window-size = 10 and the output vector size = 256. During training and testing process, we assign a zero vector to those proteins without topological features of PPI network.

2.2.3 Design of the deep neural network

We implement the deep neural network with PyTorch [64], a popular deep learning framework that was developed by Facebook. We optimize topology of the network (including the two neural layers used to produce the sequence-derived features) to maximize predictive performance on the validation dataset. The 512-dimensional vector of sequence-derived information is concatenated with the 256-dimensional vector of the PPI-derived information and the resulting 768 inputs are fed into the first fully connected hidden layer with 1024 nodes that use the ReLU activation function. The second hidden layer is also fully connected and includes 1024 neurons that utilize sigmoid function to map the outputs to the range that can be interpreted as

propensity for protein functions. We use the Adam optimizer with the batch size = 128 and the initial learning rate = 0.002 to train the deep network.

3 Results

3.1 Comparison on the testing dataset

We comparatively assess DeepFunc on the testing dataset against BLAST and the most related other method, DeepGO, which similarly relies on deep learning. While both DeepFunc and DeepGO provide numeric propensity scores (likelihood that a given protein has a given function), BLAST's predictions that are based on the function annotations of the most similar protein from the training dataset are binary (a given protein either has or has not a given function). The latter means that we cannot quantify MCC and AUC value for BLAST's predictions. Table 1 shows that the predictive performance of DeepFunc is consistently better (over all five quality measures) than the predictive performance of the other two predictors. Specifically, DeepFunc secures F_{max} = 0.56, AvgPr = 0.67, and AvgRc = 0.48, which are better by (0.56 – 0.37)/0.37 = 51.3%, 81.0% and 26.3% than the BLAST's predictions, respectively. Similarly, the relative improvements over DeepGO equal 19.1%, 15.5%, 20.0%, respectively. Moreover, the DeepFunc's MCC and AUC values are also substantially higher than the corresponding DeepGO's values (0.52 vs. 0.44 and 0.94 vs. 0.93).

The testing dataset includes highly similar (nearly identical) proteins when compared to the proteins in the training dataset, which are rather trivial to predict given that they would share the same functions. In order to investigate whether DeepFunc can perform well on the low similar protein sequences, raw testing set removes all protein sequences that are highly similar to the sequences in the training dataset. Specifically, we create a low similarity subset from the raw testing dataset by using BLAST which calculates the pair-wise sequence identity of all proteins with experimental annotations. A sequence having less a certain sequence identity value is selected from the raw testing dataset and placed in the low similarity subset as the low similarity testing set. The selection of this certain value draws on prior studies that observe that functional similarity is characteristic for proteins that share > 50% similarity [65] [66] [67], and thus use of lower similarity sequences would rely on non-trivial relationships. The raw testing dataset contains 6,306 protein sequences. After pre-processing, the low similarity testing set contains 1,835 protein sequences. Table 2 summarizes results for these challenging testing proteins. The results show that the performance of evaluating new testing set is slightly lower than the performance of evaluating raw testing set. When evaluating new testing set, the

F_{max}, AvgPr, MCC and AUC drops from 0.56, 0.67, 0.52 and 0.94 to 0.55, 0.64, 0.51 and 0.93, respectively. In conclusion, DeepFunc obtains satisfactory results no matter when evaluating raw testing set or low similarity testing set.

3.2 Comparison on the CAFA3 dataset

We empirically compare DeepFunc on the CAFA3 dataset with DeepGO and two recently published and relatively highly-cited methods: FFPred3 (published in Aug 2016; 19 citations in Google Scholar as of Jan. 2019) [5] and GoFDR (published in Jan 2016; 18 citations in Google Scholar as of Jan. 2019) [9]. We use public source code of GoFDR to run its predictions. FFPred3's prediction was downloaded from http://bioinfadmin.cs.ucl.ac.uk/downloads/ffpred/cafa3/. None of the four methods (DeepFunc, DeepGO, FFPred3, and GoFDR) has used protein annotations from the CAFA3 dataset during their training.

Table 3 compares predictive performance on the CAFA3 dataset. DeepFunc secures the best values of F_{max}, AvgRc and AUC. DeepFunc obtains F_{max} = 0.54 and AUC = 0.94 outperforming FFPred3 (0.38 and 0.86, respectively), GoFDR (0.52 and 0.84, respectively) and DeepGO (0.47 and 0.90, respectively). The corresponding relative improvements in F_{max} range between (0.54 – 0.52) / 0.52 = 3.8% compared to GoFDR and 42.1% when compared to FFPred3, while the increases in AUC range between 4.4% when contrasted with DeepGO and 11.9% when compared to GoFDR. Moreover, we observe that DeepFunc's performance is better than the performance of FFPred3 and DeepGO in all assessment metrics. Comparison with GoFDR reveals a trade-off in the average precision (0.89 for GoFDR vs. 0.62 for DeepFunc) and the average recall (0.36 for GoFDR vs. 0.48 for DeepFunc). However, an arguably most informative AUC value, which is independent of the somehow arbitrary binarization threshold, reveals a large advantage for DeepFunc (0.94 vs. 0.84). Overall, we conclude that DeepFunc outperforms the other three recently published predictors.

3.3 Ablation study

We also dissect the DeepFunc model to investigate impact of the two types of its inputs: sequence-derived and PPI network-derived. We empirically compare predictions of the corresponding three version of DeepFunc: complete DeepFunc model, DeepFunc_Seq that applies only the sequence-derived inputs, and DeepFunc_PPI that uses only the PPI network-derived inputs. We contrast these predictions with the outputs produced by DeepGO and DeepGO_Seq that applies only the sequence-derived inputs (we were not able to implement the

DeepGO_PPI version). The results produced by these five models on the testing dataset are compared in Table 4. We observe that DeepFunc outperforms all other considered models on all five metrics.

The comparison of the three versions of DeepFunc reveals that inclusion of each of the two inputs improves the resulting predictions. The removal of the sequence-derived inputs results in in drop in AUC from 0.94 to 0.93 and from in F_{max} from 0.56 to 0.48. The exclusion of the PPI network-derived features also has a strong negative impact. It lowers AUC from 0.94 to 0.91 and F_{max} from 0.56 to 0.54. The fact that combining the two input types together improves predictive performance suggests that these two inputs are complementary.

While Section 3.1 already compares DeepFunc and DeepGO, here we focus on the side-by-side comparison of their versions that utilize only the sequence-derived inputs. DeepGO primarily relies on a simple 3-mer-based representation of the input sequence while DeepFunc uses likely more informative features that encode information about domains, families and motifs associated with the input protein chain. The higher quality of the DeepFunc's sequence-derived inputs results in a substantially higher predictive performance. The improvements are present across all five metrics, with AUC = 0.91 for DeepFunc_Seq vs. 0.87 for DeepGO_Seq, F_{max} = 0.54 vs. 0.36 and AUC = 0.50 vs. 0.33. These results indicate that the domain/family/motif information is effective for the prediction of protein functions. We also compare DeepFunc_PPI with DeepGO. Table 4 shows that DeepFunc_PPI has a slight advantage although unlike DeepGO it does not use the sequence-derived input. The corresponding F_{max} and MCC for DeepFunc_PPI equals 0.48 and 0.46 vs. 0.47 and 0.44 for DeepGO, respectively. This suggests that the PPI network and its encoding utilized by DeepFunc are better than the network used in DeepGO. In summary, DeepFunc effectively combines protein sequences and PPI networks, and extract higher-quality features for protein function prediction, which is the main factor behind the improvements offered by DeepFunc over the other deep learning-based predictor, DeepGO.

4 Summary and conclusions

We design, test and comparatively assess a deep learning framework for protein function prediction, DeepFunc. Our method uses deep neural network to make accurate predictions from the protein sequence- and network derived information. The DeepFunc combines topological features of PPI network and subsequence-based features concerning motifs, domains and family assignments associated with the protein sequences. The topological features and protein sequence- used in DeepFunc have been previously used

individually or in combination with other features for the protein function prediction. They do not bias our model towards the GO terms any more than in the other published methods. The main contribution in our article is related to the use of the deep learning techniques to effectively represent the high-dimensional vector of the InterProScan-derived information and to combine the topological features extracted from the "enhanced" PPI network with this reduced InterProScan-derived information. These advances are responsible for the favourable predictive performance of DeepFunc.

Empirical tests show that DeepFunc secures comparable results on the two benchmark datasets: AUC = 0.94 and F_{max} = 0.56 on the testing dataset and AUC = 0.94 and F_{max} = 0.54 on the CAFA3 dataset. These tests demonstrate that DeepFunc outperforms the other deep learning-based solution, DeepGO, and predictions that rely on the sequence alignment with BLAST, including a challenging scenario where the test proteins share relatively low similarity to the training proteins. Comparison with three recently published methods (Deep GO, FFPred3 and GoFDR) on CAFA3 dataset reveals that DeepFunc obtains the highest values of F_{max} and AUC. The improvements over the second best result on this dataset are 0.94 vs. 0.90 in AUC and 0.54 vs. 0.52 in F_{max}. Overall, these empirical results suggest that DeepFunc provides the most accurate predictions.

The ablation study shows that extracting higher-quality features from protein sequences and PPI network that are utilized by DeepFunc contribute to its high predictive performance. Moreover, detailed comparison of the two deep learning-based tools suggests that the sequence-derived inputs and PPI network used by DeepFunc are superior to the same type of inputs used by the DeepGO predictor.

As part of future work, we will consider inclusion of additional sources of sequence-derived information to study whether this can lead to further improvements in the predictive performance. Example of potentially useful inputs that were successfully used by some of the older predictors include co-expression data [68], phylogenetic information [69], and quantitative biophysical properties [70].

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants (No. 61832019, No. 61622213 and No. 61728211), the 111 Project (No. B18059), and the Hunan Provincial Science and Technology Program (2018WK4001).

Competing interests

The authors declare that they have no competing interests.

5 Reference

[1] M. Li, W. Li, F. X. Wu, Y. Pan, J. Wang, Journal of theoretical biology 2018, 447, 65.

[2] B. Rekapalli, K. Wuichet, G. D. Peterson, I. B. Zhulin, BMC genomics 2012, 13, 634.

[3] M. Li, X. Meng, R. Zheng, F. X. Wu, Y. Li, Y. Pan, J. Wang, IEEE/ACM Trans Comput Biol Bioinform 2017.

[4] E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, P. Bansal, A. J. Bridge, S. Poux, L. Bougueleret, I. Xenarios, in *Plant Bioinformatics*, Springer, 2016, 23.

[5] M. Costanzo, B. VanderSluis, E. N. Koch, A. Baryshnikova, C. Pons, G. Tan, W. Wang, M. Usaj, J. Hanchard, S. D. Lee, Science 2016, 353, aaf1420.

[6] G. Pandey, V. Kumar, M. Steinbach, Twin Cities: Department of Computer Science and Engineering, University of Minnesota 2006.

[7] S. Das, D. Lee, I. Sillitoe, N. L. Dawson, J. G. Lees, C. A. Orengo, Bioinformatics 2015, 31, 3460.

[8] X. Wang, D. Schroeder, D. Dobbs, V. Honavar, Information Sciences 2003, 155, 1.

[9] Q. Gong, W. Ning, W. Tian, Methods 2016, 93, 3.

[10] R. D. King, A. Karwath, A. Clare, L. Dehaspe, International Journal of Genomics 2000, 1, 283.

[11] D. Cozzetto, F. Minneci, H. Currant, D. T. Jones, Scientific reports 2016, 6, 31865.

[12] J. Q. Jiang, L. J. McQuay, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 2012, 9, 1059.

[13] W. Peng, M. Li, L. Chen, L. Wang, IEEE/ACM transactions on computational biology and bioinformatics 2017, 14, 360.

[14] M. Kirac, G. Ozsoyoglu, "Protein function prediction based on patterns in biological networks", presented at *Annual International Conference on Research in Computational Molecular Biology*, 2008.

[15] J. Hou, *New Approaches of Protein Function Prediction from Protein Interaction Networks*, Academic Press, 2017.

[16] C. D. Nguyen, K. J. Gardiner, K. J. Cios, Journal of biomedical informatics 2011, 44, 824.

[17] H. Rahmani, H. Blockeel, A. Bender, "Predicting the functions of proteins in protein-protein interaction networks from global information", presented at *JMLR: Workshop and Conference Proceedings*, 2010.

[18] V. Gligorijević, M. Barot, R. Bonneau, Bioinformatics 2018.

[19] J. Li, S. K. Halgamuge, C. I. Kells, S.-L. Tang, "Gene function prediction based on genomic context clustering and discriminative learning: an application to bacteriophages", presented at *BMC bioinformatics*, 2007.

[20] J. Konc, M. Hodošček, M. Ogrizek, J. T. Konc, D. Janežič, PLoS computational biology 2013, 9, e1003341.

[21] E. W. Stawiski, A. E. Baucom, S. C. Lohr, L. M. Gregoret, Proceedings of the National Academy of Sciences 2000, 97, 3954.

[22] C. Zhang, P. L. Freddolino, Y. Zhang, Nucleic acids research 2017, 45, W291.

[23] H. A. Maghawry, M. G. Mostafa, T. F. Gharib, Journal of Computational Biology 2014, 21, 936.

[24] X.-L. Li, Y.-C. Tan, S.-K. Ng, BMC bioinformatics 2006, 7, S23.

[25] L. Tran, arXiv preprint arXiv:1212.0388 2012.

[26] C. Cai, L. Han, Z. L. Ji, X. Chen, Y. Z. Chen, Nucleic acids research 2003, 31, 3692.

[27] W. Peng, J. Wang, J. Cai, L. Chen, M. Li, F.-X. Wu, BMC systems biology 2014, 8, 35.

[28] M. Kulmanov, M. A. Khan, R. Hoehndorf, Bioinformatics 2017, 34, 660.

[29] R. Sharan, I. Ulitsky, R. Shamir, Molecular systems biology 2007, 3, 88.

[30] Ö. S. Saraç, V. Atalay, R. Cetin-Atalay, PloS one 2010, 5, e12382.

[31] L. Wei, Y. Ding, R. Su, J. Tang, Q. Zou, Journal of Parallel and Distributed Computing 2018, 117, 212.

[32] Q. Zou, P. Xing, L. Wei, B. Liu, RNA 2019, 25, 205.

[33] L. Wei, R. Su, B. Wang, X. Li, Q. Zou, X. Gao, Neurocomputing 2019, 324, 3.

[34] M. Zeng, M. Li, Z. Fei, F.-X. Wu, Y. Li, Y. Pan, "A deep learning framework for identifying essential proteins based on protein-protein interaction network and gene expression data", presented at *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2018.

[35] M. Zeng, M. Li, Z. Fei, F. Wu, Y. Li, Y. Pan, J. Wang, IEEE/ACM Trans Comput Biol Bioinform 2019.

[36] S. Seo, M. Oh, Y. Park, S. Kim, Bioinformatics 2018, 34, i254.

[37] R. Vinayakumar, K. Soman, K. Naveenkumar, bioRxiv 2018, 414128.

[38] A. Mitchell, H.-Y. Chang, L. Daugherty, M. Fraser, S. Hunter, R. Lopez, C. McAnulla, C. McMenamin, G. Nuka, S. Pesseat, Nucleic acids research 2014, 43, D213.

[39] J. Huerta-Cepas, D. Szklarczyk, K. Forslund, H. Cook, D. Heller, M. C. Walter, T. Rattei, D. R. Mende, S. Sunagawa, M. Kuhn, Nucleic acids research 2015, 44, D286.

[40] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou, Nucleic acids research 2014, 43, D447.

[41] B. Perozzi, R. Al-Rfou, S. Skiena, "Deepwalk: Online learning of social representations", presented at *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[42] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur, Nature methods 2013, 10, 221.

[43] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, H. Tang, Frontiers in genetics 2018, 9.

[44] C. Wang, L. Kurgan, Brief Bioinform 2018.

[45] F. Meng, V. N. Uversky, L. Kurgan, Cell Mol Life Sci 2017, 74, 3069.

[46] J. Zhang, L. Kurgan, Brief Bioinform 2018, 19, 821.

[47] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman, Nucleic acids research 1997, 25, 3389.

[48] D. Cozzetto, D. W. Buchan, K. Bryson, D. T. Jones, "Protein function prediction by massive integration of evolutionary analyses and multiple data sources", presented at *BMC bioinformatics*, 2013.

[49] R. Cao, J. Cheng, Methods 2016, 93, 84.

[50] A. Marchler-Bauer, M. K. Derbyshire, N. R. Gonzales, S. Lu, F. Chitsaz, L. Y. Geer, R. C. Geer, J. He, M. Gwadz, D. I. Hurwitz, Nucleic acids research 2014, 43, D222.

[51] E. L. Sonnhammer, S. R. Eddy, R. Durbin, Proteins: Structure, Function, and Bioinformatics 1997, 28, 405.

[52] I. Sillitoe, T. E. Lewis, A. Cuff, S. Das, P. Ashford, N. L. Dawson, N. Furnham, R. A. Laskowski, D. Lee, J. G. Lees, Nucleic acids research 2014, 43, D376.

[53] D. A. de Lima Morais, H. Fang, O. J. Rackham, D. Wilson, R. Pethica, C. Chothia, J. Gough, Nucleic acids research 2010, 39, D427.

[54] M. Li, P. Ni, X. Chen, J. Wang, F. Wu, Y. Pan, IEEE/ACM transactions on computational biology and bioinformatics 2017, 1.

[55] G. Li, M. Li, J. Wang, Y. Li, Y. Pan, IEEE/ACM transactions on computational biology and bioinformatics 2018.

[56] M. Li, H. Gao, J. Wang, F. X. Wu, Brief Bioinform 2018.

[57] M. Li, Z. Fei, M. Zeng, F. Wu, Y. Li, Y. Pan, J. Wang, IEEE/ACM Transactions on Computational Biology and Bioinformatics 2018, 1.

[58] X. Qin, Y. Luo, N. Tang, G. Li, Big Data Mining and Analytics 2018, 1, 75.

[59] M. Zeng, M. Li, Z. Fei, Y. Yu, Y. Pan, J. Wang, Neurocomputing 2019, 324, 43.

[60] A. Grover, J. Leskovec, "node2vec: Scalable feature learning for networks", presented at *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

[61] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, "Line: Large-scale information network embedding", presented at *Proceedings of the 24th International Conference on World Wide Web*, 2015.

[62] M. Xie, T. Hwang, R. Kuang, Bioinformatics 2012, 1, 1.

[63] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, "Distributed representations of words and phrases and their compositionality", presented at *Advances in neural information processing systems*, 2013.

[64] A. Paszke, S. Gross, S. Chintala, G. Chanan, 2017.

[65] S. Addou, R. Rentzsch, D. Lee, C. A. Orengo, Journal of molecular biology 2009, 387, 416.

[66] M. J. Mizianty, X. Fan, J. Yan, E. Chalmers, C. Woloschuk, A. Joachimiak, L. Kurgan, Acta Crystallographica Section D 2014, 70, 2781.

[67] R. Rentzsch, C. A. Orengo, "Protein function prediction using domain families", presented at *BMC bioinformatics*, 2013.

[68] M. N. Wass, G. Barton, M. J. Sternberg, Nucleic acids research 2012, 40, W466.

[69] B. E. Engelhardt, M. I. Jordan, J. R. Srouji, S. E. Brenner, Genome research 2011.

[70] D. Ofer, M. Linial, Bioinformatics 2015, 31, 3429.

Table 1. The predictive performance of DeepFunc and other two methods on the testing dataset. MCC and AUC cannot be computed for the binary predictions generated with BLAST. Best results for each quality measure are highlighted in bold.

Method	F_{max}	AvgPr	AvgRc	MCC	AUC
BLAST	0.37	0.37	0.38	-	-
DeepGO	0.47	0.58	0.40	0.44	0.93
DeepFunc	**0.56**	**0.67**	**0.48**	**0.52**	**0.94**

Table 2. The predictive performance of DeepFunc on the testing dataset and low similarity testing dataset that only includes sequences that share low (<50%) similarity to the training dataset. Best results for each quality measure are highlighted in bold.

Dataset	F_{max}	AvgPr	AvgRc	MCC	AUC
Raw testing dataset	**0.56**	**0.67**	**0.48**	**0.52**	**0.94**
low similarity testing dataset	0.55	0.64	**0.48**	0.51	0.93

Table 3. The predictive performance of DeepFunc, DeepGO, FFPred3 and GoFDR on the CAFA3 dataset. Best results for each quality measure are highlighted in bold.

Method	F_{max}	AvgPr	AvgRc	MCC	AUC
FFPred3	0.38	0.35	0.40	0.29	0.86
GoFDR	0.52	**0.89**	0.36	**0.60**	0.84
DeepGO	0.47	0.61	0.39	0.37	0.90
DeepFunc	**0.54**	0.62	**0.48**	0.44	**0.94**

Table 4. The predictive performance of DeepFunc and comparison to DeepGO_Seq (DeepGO that applies only the sequence-derived inputs), DeepGO, DeepFunc_Seq (DeepFunc that applies only the sequence-derived inputs), and DeepFunc_PPI (DeepFunc that applies only the PPI network-derived inputs) on the testing dataset. Best results for each quality measure are highlighted in bold.

Method	F_{max}	AvgPr	AvgRc	MCC	AUC
DeepGO_Seq	0.36	0.45	0.30	0.33	0.87
DeepGO	0.47	0.58	0.40	0.44	0.93
DeepFunc_Seq	0.54	**0.67**	0.46	0.50	0.91
DeepFunc_PPI	0.48	0.58	0.42	0.46	0.93
DeepFunc	**0.56**	**0.67**	**0.48**	**0.52**	**0.94**

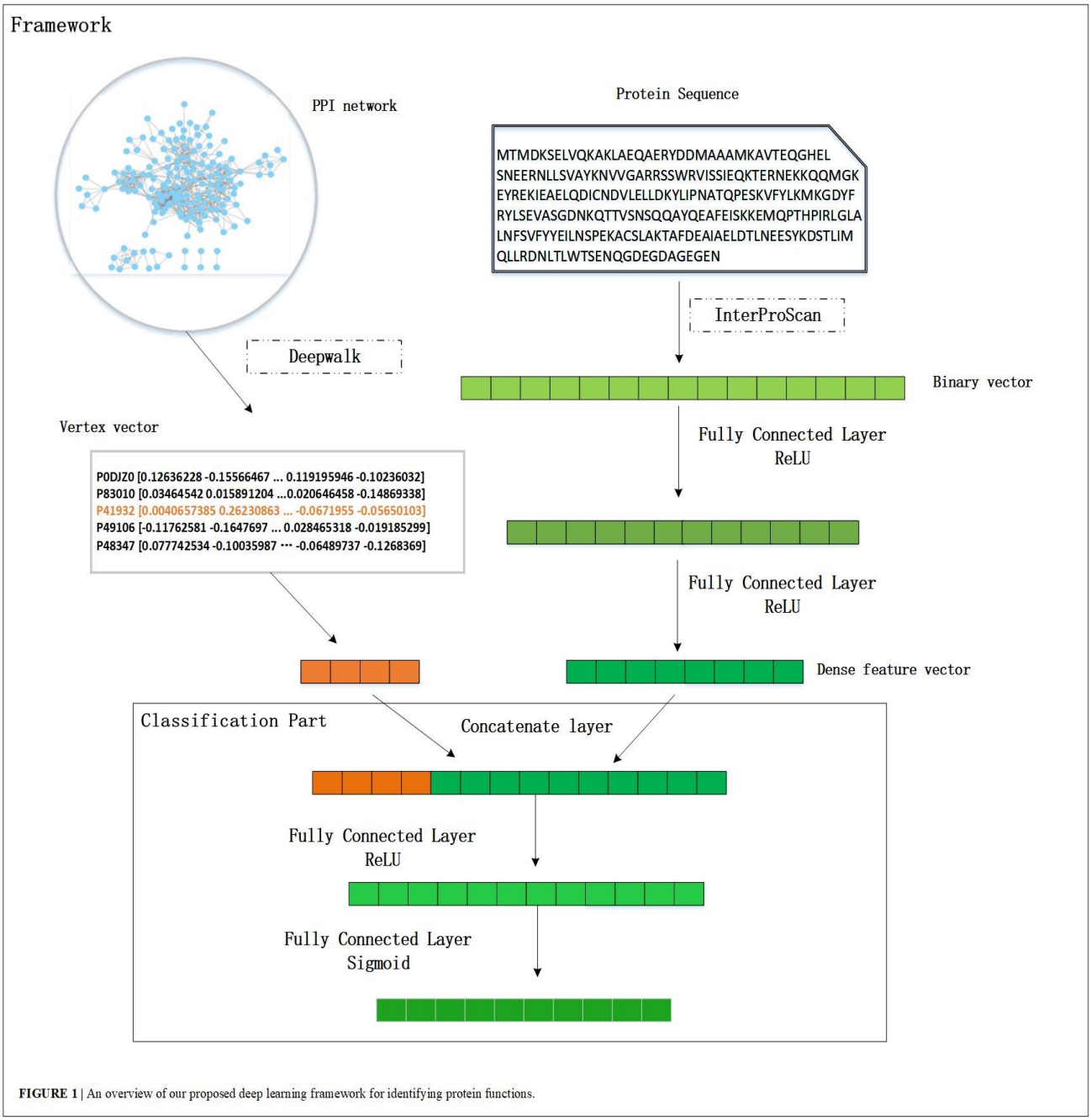

FIGURE 1 | An overview of our proposed deep learning framework for identifying protein functions.

Fig.1. An overview of our proposed deep learning framework for identifying protein functions.