Deep Learning the Protein Function in Protein Interaction Networks

Kire Trivodaliev[1(✉)], Martin Josifoski[2], and Slobodan Kalajdziski[1]

[1] Ss. Cyril and Methodius University, Skopje, Macedonia
{kire.trivodaliev, slobodan.kalajdziski}@finki.ukim.mk
[2] Ecole Polytechnique Fédérale de Lausanne EPFL, Lausanne, Switzerland
martin.josifoski@epfl.ch

Abstract. One of the essential challenges in proteomics is the computational function prediction. In Protein Interaction Networks (PINs) this problem is one of proper labeling of corresponding nodes. In this paper a novel three-step approach for supervised protein function learning in PINs is proposed. The first step derives continuous vector representation for the PIN nodes using semi-supervised learning. The vectors are constructed so that they maximize the likelihood of preservation of the graph topology locally and globally. The next step is to binarize the PIN graph nodes (proteins) i.e. for each protein function derived from Gene Ontology (GO) determine the positive and negative set of nodes. The challenge of determining the negative node sets is solved by random walking the GO acyclic graph weighted by a semantic similarity metric. A simple deep learning six-layer model is built for the protein function learning as the final step. Experiments are performed using a highly reliable human protein interaction network. Results indicate that the proposed approach can be very successful in determining protein function since the Area Under the Curve values are high (>0.79) even though the experimental setup is very simple, and its performance is comparable with state-of-the-art competing methods.

Keywords: Protein interaction network · Deep learning
Protein function prediction

1 Introduction

Proteins are the building blocks of life. Protein functions are at the core of understanding and solving crucial questions and problems in life sciences like disease mechanisms and proper drug development, design of new biochemicals, elucidation of unknown life phenomena, etc. With the upsurge in high-throughput technologies big data is taking center stage in life sciences, ranging from sequences to complex proteomic data, such as gene expression data sets and protein interaction networks (PINs). One of the main challenges is bridging the incompleteness of the data, especially in terms of building an effective and precise system for analyzing such data and uncovering their intrinsic functional meaning [5].

Protein-protein interaction (PPI) data are fundamental to biological processes [24] and in terms of single-source computational protein function prediction this data is the

S. Kalajdziski and N. Ackovska (Eds.): ICT 2018, CCIS 940, pp. 185–197, 2018.
https://doi.org/10.1007/978-3-030-00825-3_16

best choice. PPI data has the nature and organization of networks, with proteins and their interactions considered nodes and edges in the network. These networks are referred to as Protein Interaction Networks (PINs) and protein functions associated to a protein can be modeled as labels of the corresponding node. Taking this definition, the problem of computational function prediction of a protein is translated to a problem of proper labeling of its corresponding node in its PIN graph representation. The semantics of protein functions is usually defined using notational schemes organized as ontology, the most comprehensive one being the Gene Ontology (GO) [4]. GO defines three semantic contexts, stored as separate subontologies within the GO: Biological Process (BP), Molecular Function (MF), and Cellular Component (CC). Each subontology consists of a set of terms (GO terms), connected in a directed acyclic graph. GO is the most applied functional annotation scheme across a wide variety of biological data [14] and as such is the scheme used in this research.

Existing computational function prediction methods using PINs can be characterized based on what and how much information is used in the method: (1) neighborhood-based [10, 16], where the query protein "receives" its functions from the "dominant" terms in its immediate neighborhood, (2) global optimization-based [20, 25], where the neighboring information may be insufficient, so the functions of the query protein are inferred from the indirectly connected proteins, sometimes the entire network, (3) clustering-based [19, 24, 26, 32], where query protein's functions are chosen from "dominant" functions present in its determined network cluster, (4) association-based [11], similar to clustering approaches, but here functional modules are hypothesized from frequently occurring sets of interactions in PINs of protein complexes.

Computational function prediction has improved significantly in recent years [14], however there is lack of research taking into account the sparsity of GO annotations [27]. The problem arises due to GO providing mainly "positive" associations between a protein and a GO term (the protein has the function defined with the GO term). The specification of "negative" terms is very rare and in the context of computational function prediction the lack of a positive association can not be treated as a negative association. The binary classification problem requires for explicit positive and negative samples to learn the desired discriminative model. Recent approaches define negative examples directly or by using some heuristics. The direct approaches make assumptions based on the lack of annotations (associations) for a given term and a given protein, either by taking as negative proteins that lack a query annotation [8] or proteins that lack the annotation of sibling terms of a query term [3, 18]. The authors of [31] propose a parametrization Bayesian priors method that selects negative examples based on an approximation of the empirical conditional probability that a term will be annotated to a protein given that the protein is already annotated with another term. Two additional negative examples selection algorithms, selection of negatives through observed bias (SNOB) and negative examples from topic likelihood (NETL) are given in [30]. NegGOA [6] takes advantage of a hierarchical semantic similarity between GO terms and performs downward random walks with restart on the hierarchy and on the empirical conditional probability that two terms co-annotated to a protein, to determine the negative examples.

Recently, a lot of research has been focused on the problem of producing network embeddings. The DeepWalk [21] method uses short random walk sequences in the context of sentences from a natural language and using the Skip-Gram word representation model [17] learns a vector representation of the graph nodes. The LINE [23] method first learns node representations produced as concatenations of first- and second-order proximities. In a similar fashion, GraRep [2] uses various loss functions to capture k-order proximities and combines the learned representations. The TADW method [29] builds on the proof that DeepWalk is equivalent to matrix factorization by incorporating node level rich text information in the network representation learning. To capture the non-linear network structure, [28] proposes a deep learning model with non-linear functions and produces results that preserve first- and second-order proximities. The node2vec [7] method uses a biased random walk procedure with a flexible notion of a nodes neighborhood, which efficiently explores diverse neighborhoods. Node2vec treats the results of these walks as natural language sentences (same as DeepWalk) and using these "sentences" produces vectors that maximize the likelihood of preservation of the graph topology and semantics locally and globally.

In this paper a novel approach DeePin using deep learning is proposed for the computational protein function prediction. The problem is solved as term-centric i.e. the result of the prediction gives answer if a protein is annotated with a given term or not. In order to build the appropriate deep learning models, the PIN is first transformed into a continuous vector space (an instance per node) and for each term-model a set of positive and negative examples is chosen. The aim is to show that although very simple this approach can produce comparable results with other leading approaches.

The rest of the paper is organized as follows. Section 2 presents the steps in acquiring and building the data for the research and the technical details on the methods used. In the third section a detailed description of the performed experiments and the corresponding results is given. Discussion for the results and possible improvements of the method is also provided. Finally, the paper is concluded in the fourth section.

2 Materials and Methods

In this paper the graph representation of the PIN is used to derive a continuous vector representation. The approach of [7] is adopted and it employs a biased random walk procedure with a flexible notion of a nodes neighborhood, which efficiently explores diverse neighborhoods. The problem is formulated as a maximum likelihood optimization problem i.e. one that maximizes the log-probability of observing a neighborhood node for a target node, conditioned on its vector representation. The advantage of this approach over other algorithms arises in its scalability, as well as flexibility to easily custom-fit the representation for detecting node dependencies based on communities they belong to, structural equivalences based on the nodes role in the network, or a mixture of both. The produced vectors maximize the likelihood of preservation of the graph topology locally and globally. The next step is to binarize the PIN graph nodes (proteins) i.e. for each protein function derived from Gene Ontology (GO) determine the positive and negative set of nodes. Since positive set is predefined the only challenge is determining the negative node sets. Based on a random walk on the GO

acyclic graph weighted using Lin similarity metric computed from the available annotations, probabilities of not observing a label annotated to a protein will be associated on every pair of protein function and protein [6]. Using a threshold, the negative set of proteins for a given protein function is determined and combined with the positive set are used in the process of learning the protein model for the function. A simple deep learning six-layer model is built for the protein function learning. The following subsections present the steps in acquiring and building the data for the research. Additionally, technical details on the methods used are also provided.

2.1 Protein Interaction Network Data

The Protein Interaction Network (PIN) is constructed on data from the HIPPIE (v2.0) database [22], which is a highly reliable human PPI dataset, built from multiple experimental datasets, that integrates the amount and quality of evidence for a given interaction in a normalized scoring scheme. Data is first preprocessed to remove self-interactions, zero-confidence interactions and duplicate interactions. Duplicates are removed so that only the highest confidence score interaction remains. The preprocessing results in an undirected weighted graph, having proteins as nodes, their interactions as edges, and the confidence scores of interactions as weights associated to corresponding edges. The largest connected component of this graph is the final PIN graph used in the research and is consisted of 16,769 proteins and 277,055 protein-protein interactions.

To model the problem of function prediction as a label prediction problem the PIN needs to be enriched i.e. describe every protein with all its known functional annotations. To that aim a Gene Ontology Annotation (GOA) file from the European Bioinformatics Institute is used (archived date: May 3, 2013). The GOA file provides GO annotations which associate gene products with GO terms. This file is processed so that terms labeled as 'obsolete' and annotations with evidence code 'IEA' (Inferred from Electronic Annotation) are excluded. Additionally, duplicate annotations are also excluded. The remaining annotations are associated with their corresponding nodes in the final PIN. From the initial GOA file consisting of 369,199 annotations, in the final labeled PIN there are a total of 126,367 annotations.

2.2 Vector Representation of the PIN

To apply deep learning of the protein function one first needs to construct highly informative, discriminating and mutually independent feature representations of the PIN's nodes/edges. The node2vec algorithm [7] provides a semi-supervised method for learning continuous vector representation for nodes in a network, that map every node to a d-dimensional feature space in a process that maximizes the likelihood of preserving the network neighborhood of nodes. Formally, for a network with a graph representation $G = (V, E)$ let $f : V \rightarrow R^d$ be the mapping to the feature space. For every protein node $u \in V$, a neighborhood of node u is defined with $N(u) \subset V$. Now the problem is formulated as a maximum likelihood optimization problem with the following objective function:

$$\max \sum_{u \in V} \log \Pr(N(u)|f(u)) \tag{1}$$

Equation 1 maximizes the log-probability of observing a neighborhood node for a node u, conditioned on its vector representation, given by f. This procedure with certain parameter settings becomes equivalent to DeepWalk [21], and by transitivity incorporates the key features of other previously proposed methods for network embeddings.

Let $G = (V, E)$ be the graph representation for the PIN. The process of learning the representations starts by generating r random walks from every protein node u as a source, with fixed length l. Let c_i be the i-th protein in the walk and $c_0 = u$. The generation of a sequence of proteins is done using the distribution

$$P(c_i = x|c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & (u, v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The random walks are biased to provide a representation that captures the right mix of equivalences from the graph using the parameters p and q. Suppose we are at a protein v, and we have traversed there through edge (u, v) from protein u, the next protein t in the walk is decided using the transition probability π_{vt} (Fig. 1). Let w_{vt} be the weight and π_{vt} the transition probability on the edge (v, t) directed from v, then the transition probability is set to $\pi_{vt} = \alpha_{pq}(u, t) * w_{vt}$, such that

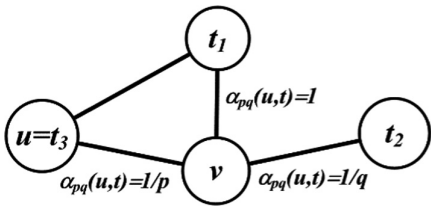

Fig. 1. Random walk transition probability illustration

$$\alpha_{pq}(u, t) = \begin{cases} 1/p & if \quad d_{ut} = 0 \\ 1 & if \quad d_{ut} = 1 \\ 1/q & if \quad d_{ut} = 2 \end{cases} \tag{3}$$

Using p, q we can control how far a random walk can get from a source node, thus direct (bias) the walk and redefine the context of a node's neighborhood. Higher values of p(>(max(1, q)) reduce the probability of revisiting a node, and therefore forces the walk in an "exploratory mode", conversely, low values of p(<(min(1, q)) result in going back to already visited nodes, hence constraining the walk close to the source node. On the other hand, having a higher value for $q(q > 1)$ makes the walk focused on nodes closer to the source and results in sample proteins within a small locality, while having

lower values for $q(q < 1)$ tend to explore interactions that lead to more distant protein nodes from the source protein.

The whole representation learning process can be summarized in three phases: preprocessing to compute transition probabilities, random walk simulations and optimization of the vector representations using stochastic gradient descent.

2.3 Negative Examples Selection

In this research the NegGOA [6] approach is adopted, and negative examples are selected based on the available protein annotations and the term hierarchy defined in GO. Initially, the pairwise semantic similarity of all terms is calculated using Lin's approach:

$$sim_H(t,s) = \frac{2 \times IC(LCA(t,s))}{IC(t) + IC(s)} \tag{4}$$

having $LCA(t, s)$ denote the lowest common ancestor of the two terms, while $IC(\cdot)$ is the information content defined with:

$$\mathrm{IC(t)} = \left(1 - \frac{log_2(|desc(t)|)}{log_2(|T|)}\right) \tag{5}$$

$|T|$ denotes the number of terms, and $desc(t)$ is the complete set of descendants of t.

In time, more annotations are added to proteins. New annotations often correspond to descendants of existing annotations. This observation is modeled using random walks with restarts on the GO hierarchy, having existing annotations as source nodes. The transition matrix $W'_H \in R^{|T|\times|T|}$ is modeled using the semantic similarities between term pairs in the following way:

$$W'_H(t,s) = sim_H(t,s) \times G(t,s) \tag{6}$$

where $G(t, s) = 1$ if s is child of t, and 0 otherwise. The transition matrix is further normalized:

$$W_H(t,s) = W'_H(t,s) / \sum_{v \in T} W'_H(t,v) \tag{7}$$

Using this transition matrix, the probability to reach a term v starting from a term t is defined as the random walk with restart probability in the 4^{th} iteration and is denoted with $R_H(t,v) = W^4_H(t,v)$.

To utilize the existing annotation set an empirical conditional probability is defined for terms t and s:

$$p(s|t) = \frac{|A_t \cap A_s|}{|A_t|} \tag{8}$$

where A_x is the set of proteins annotated with term x. The lower the value of the conditional probability $p(s|t)$, the higher the probability for s to be chosen as a negative example. Once again, a random walk with restart needs to be employed and the transition matrix to be defined

$$W_C(t,s) = p(s|t) / \sum_{v \in T} p(v|t) \tag{9}$$

having, $W_C^0(t,t) = 1$ and $W_C^0(t,s) = 0 (t \neq s)$.

As in the previous case the 4th iteration of the random walk with restart probability is the probability to reach term v starting from term t, i.e. $R_C(t,v) = W_C^4(t,v)$.

The probabilities derived from the GO hierarchy and the existing annotation in the dataset are used to calculate the following two metrics:

$$L_H(i,v) = 1 - \frac{1}{|T_i|} \sum_{t \in T_i} R_H(t,v) \tag{10}$$

$$L_C(i,v) = 1 - \frac{1}{|T_i|} \sum_{t \in T_i} R_C(t,v) \tag{11}$$

where T_i is the set of existing annotations of the i-th protein, including terms added with transitive closure. $L_H(i,v)$ is the predicted likelihood of term $v \notin T_i$ as a negative example of the i-th protein from $R_H(t,v)$. $L_C(i,v)$ is the predicted likelihood of negative example from $R_C(t,v)$.

The two metrics are combined in one:

$$L(i,v) = \beta L_H(i,v) + (1-\beta) L_C(i,v) \tag{12}$$

where $\beta \in [0,1]$ is a scalar used to control the influence of $L_H(i,v)$ and $L_C(i,v)$.

Finally, the i-th protein receives as negative the annotations that correspond the largest values of $L(i,\cdot) \in R^{|T|}$.

2.4 Deep Learning the Protein Function

The problem of computational protein function prediction i.e. learning the correct labels for the PIN graph nodes is modeled as a binary classification problem i.e. whether a specific functional term should be associated with a protein node or not. In this research a feed-forward deep neural network is used. A simple deep learning six layer model is built for the protein function learning: the first layer consisting of 128 fully connected ReLU (Rectified Linear Units) neurons, the second layer consisting of 128 fully connected sigmoid neurons, the third is a dropout layer (with a 0.25 rate) followed by a layer of 64 fully connected sigmoid neurons, the fifth is once again a dropout layer (0.25 rate) and the final layer is a single fully connected sigmoid neuron. The dropout layers are used since the number of instances used in the learning process is modest in terms of deep learning and overfitting needs to be avoided.

3 Results and Discussions

In the experiments performed for the proposed approach two different vector representations are used for the PIN graph. The first representation is derived by using p = 1, q = 1 in the vector representation method, and the resulting representations will be close for nodes (proteins) in the same network community. The second representation corresponds to p = 1, q = 2, and the resulting representations are close for nodes that share similar structural roles in the graph.

The next step in creating proper experimental setup is the creation of corresponding datasets for each deep learning model that needs to be built. These datasets need to contain a sufficient number of positive examples as well as a corresponding number of negative examples. To that aim the 20 most frequent GO terms for each ontology (MF, BP, CC) present in the final GOA file are chosen as the initial target terms for the deep learning models. The initial target terms are further filtered based on their specificity since the aim is to be able to predict the most specific terms possible. A specificity metric is introduced:

$$C_t = \frac{|F_t|}{|A_t|} \tag{13}$$

where $|A_t|$ is the number of nodes annotated with term t in the final GOA file, while $|F_t|$ is the number of nodes annotated with term t when the annotation sets for each node are expanded using transitive closure in GO (if a node is annotated with term t then it is also annotated with every ancestral term of t). The initial 60 target terms are filtered so that each term that has a specificity metric $C_t > 1.3$ will be discarded. Using this filtering the final target term set is composed of 22 terms.

Positive examples are chosen based on the annotations in the final GOA file. Using the negative example selection method, the examples with the highest likelihood are chosen as negative by defining a lower bound on the likelihood so that the number of negative examples that survive the cutoff is comparable to number of positive examples. This procedure of creating the complete dataset of positive and negative examples is done for each target term separately and independently.

The neural network models are implemented using the deep learning library Keras with Tensorflow as a backend. The binary cross entropy is used as a loss function in the training process, and parameters are optimized using the Adaptive gradient descend (Adagrad) algorithm.

Binary classification is performed using 10-fold cross-validation and the following are of interest: True Positives (TP) – when the functional term is predicted for a protein and is part of the annotation set of the protein, True Negatives (TN) – when the functional term is not predicted for a protein and is not part of the annotation set of the protein, False Positives (FP) – when the functional term is predicted for a protein, but it is not part of the annotation set of the protein, False Negatives (FN) – when the functional term is not predicted for a protein, but it is part of the annotation set of the protein. Using these, the following classification quality measures can be defined:

$$Sensitivity\ (True\ Positive\ Rate) = \frac{TP}{TP + FN} \tag{14}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \tag{15}$$

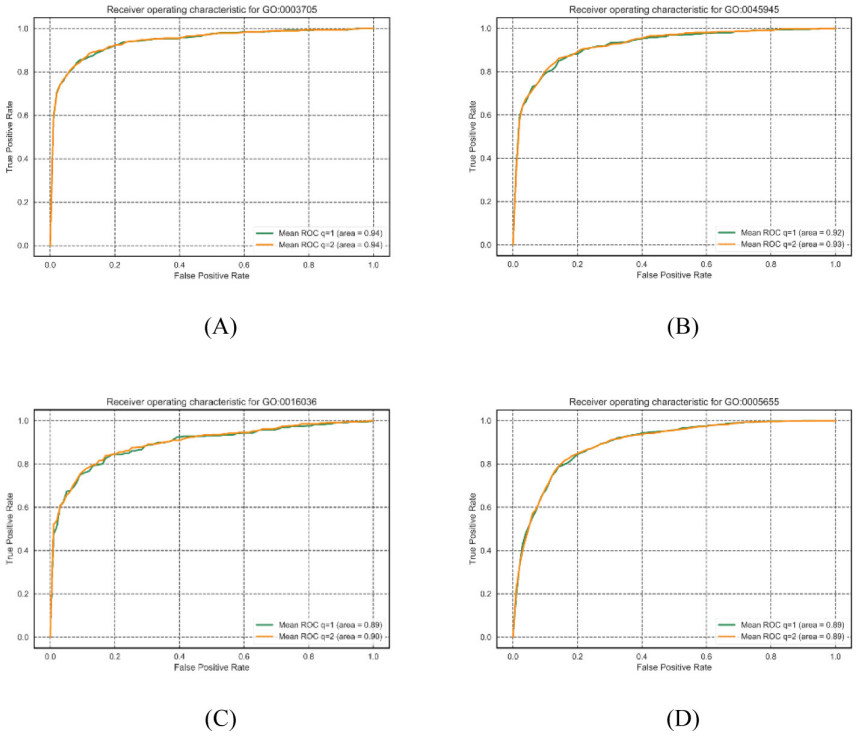

Fig. 2. Receiver operator characteristics for the four top performing deep learning models for (A) GO:0003705, (B) GO:0045945, (C) GO:0016036, (D) GO:0005655

Plotted as coordinate pairs Sensitivity and FalsePositiveRate define the Receiver Operating Curve (ROC). The ROC curve describes the model performance over the complete range of classification thresholds. The Area Under the Curve (AUC) for a classifier model is equivalent to the probability that the classifier will rank a random positive example higher than a random negative example. Figure 2 depicts the top four ROC curves for the deep learning models with best performances.

The proposed approach DeePin is compared with existing methods like topological-feature based prediction (TopFeat) [15], diffusion state distance (DSD) [1], scale-aware topological measures (STM) [13], PPI information (PPIi) [12]. TopFeat considers weighted PIN network topology features with local and global information to characterize proteins and identify protein function. DSD uses graph-diffusion based

metric to capture detailed distinctions in proximity and use them in the process of transferring functional annotations in a PPI network. STM uses scale-invariant description of the topology around or between proteins with a network smoothing operation and diffusion kernels. PPIi takes into account function information in the neighborhood of a query protein and the weights of interactions with neighbors and infers specific function for the query protein using a so-called "inclined potential". Comparison is performed based on the AUC values for predicting the target GO terms and results are given in Table 1.

Table 1. Comparison of AUC values for the proposed approach and existing methods

GO term	GO category	DeePin (q = 1)	DeePin (q = 2)	TopFeat	DSD	STM	PPIi
GO:0045945	BP	0.92	**0.93**	0.92	0.74	0.65	0.76
GO:0016036	BP	0.89	**0.9**	**0.9**	0.78	0.71	0.78
GO:0000128	BP	0.84	0.84	**0.85**	0.73	0.67	0.73
GO:0007269	BP	0.84	0.84	**0.86**	0.73	0.67	0.72
GO:0045088	BP	0.83	0.83	**0.84**	0.73	0.66	0.83
GO:0043067	BP	0.8	**0.81**	**0.81**	0.69	0.62	0.69
GO:0055086	BP	**0.82**	0.81	0.8	0.68	0.62	0.67
GO:0007597	BP	**0.79**	**0.79**	0.77	0.66	0.61	0.64
GO:0044282	BP	**0.78**	**0.78**	0.77	0.71	0.66	0.69
GO:0005655	CC	**0.89**	**0.89**	0.87	0.78	0.72	0.75
GO:0005889	CC	**0.86**	**0.86**	0.85	0.75	0.69	0.74
GO:0005635	CC	**0.85**	**0.85**	0.84	0.7	0.65	0.69
GO:0005616	CC	**0.85**	0.84	0.84	0.76	0.71	0.73
GO:0005790	CC	**0.84**	**0.84**	**0.84**	0.74	0.68	0.75
GO:0005831	CC	**0.82**	**0.82**	0.8	0.74	0.7	0.74
GO:0005887	CC	**0.82**	**0.82**	0.81	0.73	0.68	0.71
GO:0005814	CC	**0.79**	**0.79**	0.77	0.67	0.61	0.69
GO:0005731	CC	**0.76**	**0.76**	0.73	0.61	0.57	0.63
GO:0003705	MF	0.94	0.94	**0.99**	0.92	0.83	0.91
GO:0003714	MF	0.88	0.88	**0.95**	0.88	0.78	0.88
GO:0005525	MF	0.75	0.76	**0.85**	0.76	0.72	0.73
GO:0042805	MF	0.72	0.71	**0.8**	0.72	0.65	0.7

As can be seen from the results in Table 1 the proposed approach DeePin gives similar performance to TopFeat, which is the top performing existing method, when targeting BP GO terms. The DeePin performance is slightly better than TopFeat when the target terms are from the CC category. TopFeat significantly outperforms the proposed approach in the MF category. This is mainly due to the fact that very few MF targets are present in the final target term set. One drawback of the proposed approach lies in its need for higher number of positive and negative examples to train the deep learning models. As seen from the filtering of the data performed in this research this is

not always the case when it comes to more specific GO terms, which are much more significant in terms of computational function prediction. This is due to the incompleteness of the knowledge for all possible protein interactions and all possible functions a protein performs. However, with the rise in protein data generation in the future this problem will diminish. The current incompleteness of data can explain the "poor" performance the proposed approach has on MF terms, since the MF ontology is by far the "easiest" to predict [14]. With the data increase the proposed approach can only improve in performance since all steps are inherently sensitive to the amount of information presented.

4 Conclusion

In this paper a novel approach for computational function prediction using deep learning in protein interaction networks (PINs) i.e. DeePin is proposed. The approach is very simple since it requires a single algorithm/computation at each step of its pipeline. The PIN graph is translated in a vector space that is able to capture the topological/structural properties of the PIN and their dependencies in a single optimization problem, as opposed to extracting these features independently like in other competing methods. The quality of the prediction is enhanced by making an informed choice on the negative examples used in building the deep learning models. Finally, the deep models are very simple consisting of only six layers with total number of neurons of less than 1000. Experiments are performed using a highly reliable human protein interaction network. Results indicate that the proposed methodology can be very successful in determining protein function and its performance is comparable with state-of-the-art competing methods. The drawback of the proposed approach is its inherent sensitivity to the amount of data available. However, this may become an asset in the future, since big data is generated for proteins daily and the knowledge base for proteins, their interactions and functions, deepens.

References

1. Cao, M., et al.: Going the distance for protein function prediction: a new distance metric for protein interaction networks. PLoS ONE **8**, e76339 (2013)
2. Cao, S., Lu, W., Xu, Q.: Grarep: learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 891–900. ACM (2015)
3. Cesa-Bianchi, N., Re, M., Valentini, G.: Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. Mach. Learn. **88**, 209–241 (2012)
4. Consortium, G.O.: Expansion of the Gene Ontology knowledgebase and resources. Nucl. Acids Res. **45**, D331–D338 (2016)
5. Friedberg, I.: Automated protein function prediction—the genomic challenge. Brief. Bioinform. **7**, 225–242 (2006)
6. Fu, G., Wang, J., Yang, B., Yu, G.: NegGOA: negative GO annotations selection using ontology structure. Bioinformatics **32**, 2996–3004 (2016)

7. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864. ACM (2016)
8. Guan, Y., Myers, C.L., Hess, D.C., Barutcuoglu, Z., Caudy, A.A., Troyanskaya, O.G.: Predicting gene function in a hierarchical context with an ensemble of classifiers. Genome Biol. **9**, S3 (2008)
9. Hakes, L., Lovell, S.C., Oliver, S.G., Robertson, D.L.: Specificity in protein interactions and its relationship with sequence diversity and coevolution. Proc. Natl. Acad. Sci. **104**, 7999–8004 (2007)
10. Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., Takagi, T.: Assessment of prediction accuracy of protein function from protein–protein interaction data. Yeast **18**, 523–531 (2001)
11. Hu, H., Yan, X., Huang, Y., Han, J., Zhou, X.J.: Mining coherent dense subgraphs across massive biological networks for functional discovery. Bioinformatics **21**, i213–i221 (2005)
12. Hu, L., Huang, T., Shi, X., Lu, W.-C., Cai, Y.-D., Chou, K.-C.: Predicting functions of proteins in mouse based on weighted protein-protein interaction network and protein hybrid properties. PLoS ONE **6**, e14556 (2011)
13. Hulsman, M., Dimitrakopoulos, C., de Ridder, J.: Scale-space measures for graph topology link protein network architecture to function. Bioinformatics **30**, i237–i245 (2014)
14. Jiang, Y., et al.: An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biol. **17**, 184 (2016)
15. Li, Z., et al.: Large-scale identification of human protein function using topological features of interaction network. Sci. Rep. **6**, 37179 (2016)
16. McDermott, J., Bumgarner, R., Samudrala, R.: Functional annotation from predicted protein interaction networks. Bioinformatics **21**, 3217–3226 (2005)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Mostafavi, S., Morris, Q.: Using the gene ontology hierarchy when predicting gene function. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 419–427. AUAI Press (2009)
19. Mukhopadhyay, A., Ray, S., De, M.: Detecting protein complexes in a PPI network: a gene ontology based multi-objective evolutionary approach. Mol. BioSystems **8**, 3036–3048 (2012)
20. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., Singh, M.: Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. Bioinformatics **21**, i302–i310 (2005)
21. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710. ACM (2014)
22. Schaefer, M.H., Fontaine, J.-F., Vinayagam, A., Porras, P., Wanker, E.E., Andrade-Navarro, M.A.: HIPPIE: integrating protein interaction networks with experiment based quality scores. PLoS ONE **7**, e31826 (2012)
23. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
24. Trivodaliev, K., Bogojeska, A., Kocarev, L.: Exploring function prediction in protein interaction networks via clustering methods. PLoS ONE **9**, e99755 (2014)
25. Trivodaliev, K., Cingovska, I., Kalajdziski, S., Davcev, D.: Protein function prediction based on neighborhood profiles. In: Davcev, D., Gómez, J.M. (eds.) ICT Innovations 2009, pp. 125–134. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-10781-8_14

26. Trivodaliev, K., Kalajdziski, S., Ivanoska, I., Stojkoska, B.R., Kocarev, L.: SHOPIN: semantic homogeneity optimization in protein interaction networks. In: Advances in Protein Chemistry and Structural Biology, vol. 101, pp. 323–349. Elsevier (2015)
27. Valentini, G.: Hierarchical ensemble methods for protein function prediction. ISRN Bioinform. **2014**, 1–31 (2014)
28. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234. ACM (2016)
29. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: IJCAI, pp. 2111–2117 (2015)
30. Youngs, N., Penfold-Brown, D., Bonneau, R., Shasha, D.: Negative example selection for protein function prediction: the NoGO database. PLoS Comput. Biol. **10**, e1003644 (2014)
31. Youngs, N., Penfold-Brown, D., Drew, K., Shasha, D., Bonneau, R.: Parametric Bayesian priors and better choice of negative examples improve protein function prediction. Bioinformatics **29**, 1190–1198 (2013)
32. Zhang, Y., Lin, H., Yang, Z., Wang, J., Li, Y., Xu, B.: Protein complex prediction in large ontology attributed protein-protein interaction networks. IEEE/ACM Trans. Comput. Biol. Bioinform. **10**, 729–741 (2013)