Exploiting Complex Protein Domain Networks for Protein Function Annotation

Bishnu Sarker, David W. Rtichie, and Sabeur Aridhi

University of Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
bishnu.sarker@inria.fr

Abstract. Huge numbers of protein sequences are now available in public databases. In order to exploit more fully this valuable biological data, these sequences need to be annotated with functional properties such as Enzyme Commission (EC) numbers and Gene Ontology terms. The UniProt Knowledgebase (UniProtKB) is currently the largest and most comprehensive resource for protein sequence and annotation data. In the March 2018 release of UniProtKB, some 556,000 sequences have been manually curated but over 111 million sequences still lack functional annotations. The ability to annotate automatically these unannotated sequences would represent a major advance for the field of bioinformatics. Here, we present a novel network-based approach called *GrAPFI* for the automatic functional annotation of protein sequences. The underlying assumption of *GrAPFI* is that proteins may be related to each other by the protein domains, families, and super-families that they share. Several protein domain databases exist such as InterPro, Pfam, SMART, CDD, Gene3D, and Prosite, for example. Our approach uses Interpro domains, because the InterPro database contains information from several other major protein family and domain databases. Our results show that *GrAPFI* achieves better EC number annotation performance than several other previously described approaches.

Keywords: complex protein domain networks, protein function annotation, label propagation, GrAPFI, bioinformatics

1 Introduction

Understanding protein function is one of the keys to understanding life at the molecular level, and is central to understanding human disease processes and drug discovery efforts. In this age of rapid and affordable amino-acid sequencing technologies, the number of protein sequences accumulating in databases is rising at an increasing rate. This presents many challenges for biologists and computer scientists alike. In order to make sense of this huge quantity of data, these sequences should be annotated with functional properties. The UniProt knowledgebase (UniProtKB) consists of two components: (i) the UniProtKB/Swiss-Prot database which contains protein sequences with reliable information that has been reviewed by expert bio-curators and (ii) the UniProtKB/TrEMBL database that stores unannotated sequences [5]. Thus, for all proteins in UniProtKB we have the primary amino-acid sequence as well as some further information such as structural domain definitions, which may have been identified from 3D protein structures or predicted from families of similar sequences.

The UniProt curators annotate UniProtKB/TrEMBL sequences using two complementary systems. The first, called UniRule, uses a large list of "if-then" rules. These rules have been generated manually, which is both a laborious and time consuming process. The rules in UniRule are generally very reliable but their coverage is low [10]. The second system is called Statistical Automatic Annotation System (SAAS), and was developed to support the labour-intensive UniRule system [15]. Automatic annotation rules are generated in SAAS using the annotations of the Swiss-Prot sequences and the decision tree algorithm. Other approaches exist for automatic protein function annotation. In particular, several approaches for predicting Enzyme Commission (EC) numbers that exploit protein structural similarities have been described in [8, 30, 22]. Many sequence similarity based approaches have also been described [26, 16, 24, 31]. Additionally, machine learning methods have also been used extensively [13, 22, 19, 12, 20, 23, 18, 28, 29].

Recently, the notion of network science has attracted great attention across many scientific communities. Network science has become a multi-disciplinary area of research due to its ability to describe complex systems. It has found applications in many real world scenarios from banking and the internet to modeling the human brain. There have been a number of works that use network science and neighborhood based techniques such as [27, 34, 11, 4, 21] where protein-protein interaction (PPI) networks are exploited for the purpose of functional annotation of proteins, mainly using terms from the Gene Ontology. One of the interesting features of biological networks is that they often require specialist biological knowledge to fully understand and exploit the network.

The following methods are widely used EC prediction methods that use combined approaches based on machine learning, sequence encoding, functional domain similarity and structural similarity. In DEEPre [18], a technique for feature extraction and classifier training is described for enzyme function prediction. DEEPre uses multiple algorithms involving PSI-Blast [1], HMMER [9], Convolutional and Recurrent Neural Network and sequence encoding using position specific scoring matrix (PSSM) to perform dimensionality uniformization, feature selection and classification model training simultaneously. Deep Neural models has shown tremendous performances over traditional machine learning models in classification task. However, being a "black box" model and having millions of parameters to be adjusted in training, it cannot provide an explanation of its predictions.

EzyPred [28] is a three-level EC number predictor, which predicts whether an input protein sequence is an enzyme, and if so, its main EC class and subclass. EzyPred exploits functional and evolutionary information of protein using pseudo amino acid composition [3] and functional encoding. Based on two features, EzyPred applied an improved version of K-Nearest Neighbor Classifier called OET-KNN: Optimized Evidence-Theoretic K-Nearest Neighbor. EzyPred is reported to be one of the more successful EC number prediction methods. However, it can only predict the first two digits of a four-digit EC number. Thus, its predictions are not very specific.

SVM-Prot is a support vector machine based classification method first described in 2004 and later updated in 2016 [19]. This approach is based on physico-chemical representations of protein sequences using various properties like AAC, polarity, hy-

drophobicity, surface tension, charge, normalized Van der Waals volume, polarizability, secondary structure, solvent accessibility, molecular weight, solubility, number of hydrogen bond donors in side chain and number of hydrogen bond acceptors in side chain. In the updated version of the SVM-Prot two more classifiers, K-Nearest Neighbor (KNN) and Probabilistic Neural Networks (PNN) were added for improved performance.

A Structure-based protein function annotation is proposed in COFACTOR [33]. COFACTOR uses a hybrid model combining information from structure and sequence homologies, as well as PPI networks, for the prediction of GO terms, EC numbers, and ligand-binding sites.

EFICAz (Enzyme Function Inference by Combined Approach) is an EC number prediction server first proposed in 2004 with latest release as EFFICAz2.5 [16]. This large-scale enzyme function inference combines predictions from four different methods optimized to achieve high prediction accuracy using: (i) recognition of functionally discriminating residues (FDRs) in enzyme families obtained by a Conservation-controlled HMM Iterative procedure for Enzyme Family classification (CHIEFc), (ii) pairwise sequence comparison using a family specific Sequence Identity Threshold, (iii) recognition of FDRs in Multiple Pfam enzyme families, and (iv) recognition of multiple Prosite patterns of high specificity.

In this paper, we present a novel protein-protein network (PPN) based approach called *GrAPFI* which combines the notion of domain similarity with a graph neighborhood inference technique for automatic EC number annotation. More specifically, the functional annotations of reviewed proteins in SwissProt are used to predict those of non-reviewed proteins in TrEMBL using label propagation on a complex network representation of the protein data. Our analysis shows that *GrAPFI* has better annotation performance than other state of the art techniques.

2 Methods

Our approach for automatic protein function annotation works as follows. First, it constructs a network representation of the protein database using the domain composition of the reviewed proteins. Then, given an non-reviewed protein, a label propagation algorithm is applied to the protein graph in order to infer appropriate annotations.

2.1 Protein-Protein Network Construction

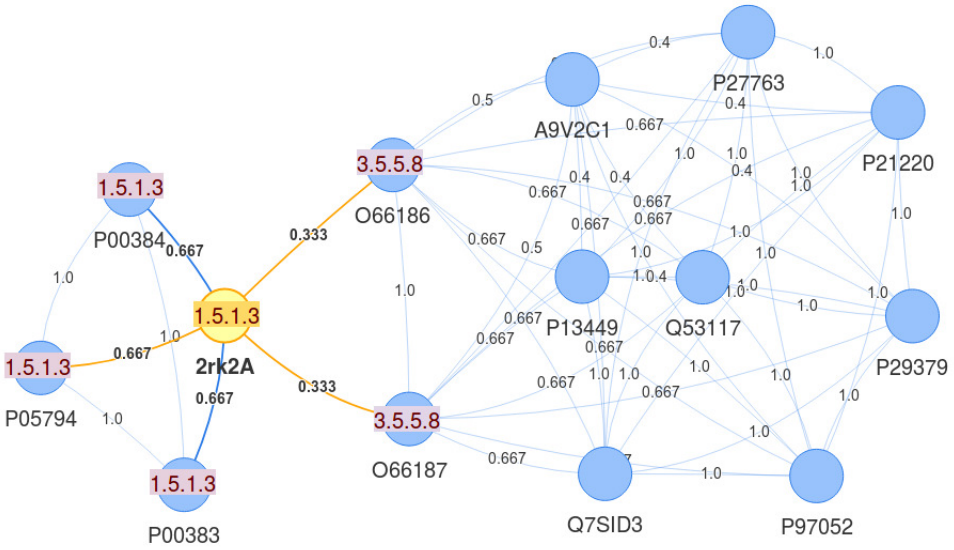
We present here a novel way of connecting the proteins using their constituent protein domains. Domains may be considered as natural building blocks of proteins. During evolution, protein domains have been duplicated, fused, and recombined in different ways to produce proteins with distinct structures and functions [17]. Here, each node of the network represents a protein while a link between two nodes means that the proteins exhibit a given minimum level of domain similarity. Thus, each node u is identified by a set of labels $L(u)$ (one or more annotations to propagate), has a set of neighbours $N(u)$, and for every neighbour v it has an associated weight $W_{u,v}$.

To illustrate the construction of the PPN, let us consider five proteins with symbolic names $P1$,$P2$, $P3$, $P4$ and $P5$. Let us assume that these proteins are composed of domains $D1 = (d1, d2, d3, d4)$, $D2 = (d1, d3, d5)$, $D3 = (d1, d2, d10)$, $D4 = (d5, d6, d1)$, and $D5 = (d4, d1, d10, d40, d7, d9, d12, d52, d100)$, respectively. It is then evident that proteins $P1$ and $P2$ contain two domains in common namely $d1$ and $d3$. Therefore, proteins $P1$ and $P2$ may be linked and the number of shared domains may serve as a link weight such as $W_{P1,P2} = |(d1, d2, d3, d4) \cap (d1, d3, d5)| = |(d1, d3)| = 2$. In a similar way, proteins $P1$ and $P5$ may be linked with a link weight of $|(d1, d2, d3, d4) \cap (d4, d1, d10, d40, d7, d9, d12, d52, d100)| = |(d1, d4)| = 2$. In the both cases, the link weight is 2. However, the link weight computed in this way does not reflect the true strength of the relationship among the proteins. More specifically, in the first case there are total of $|(d1, d2, d3, d4) \cup (d1, d3, d5)| = |(d1, d2, d3, d4, d5)| = 5$ different domains among the two proteins, of which two are shared. In the second case, there are $|(d1, d2, d3, d4) \cup (d4, d1, d10, d40, d7, d9, d12, d52, d100)| = 11$ different domains of which two are again shared. Although two domains are shared in each case, P1 is intuitively more aligned with P2 than P5. Therefore, instead of using the above raw similarity score, we instead use the Jaccard index, or Jaccard similarity coefficient, to reflect better the similarity in composition. This is calculated as $\frac{|A \cap B|}{|A \cup B|}$, where A and B are the two sets of constituent domains. Using the Jaccard coefficient, the link weights for P1 and P2 are calculated as $W_{P1,P2} = \frac{|(d1,d2,d3,d4) \cap (d1,d3,d5)|}{|(d1,d2,d3,d4) \cup (d1,d3,d5)|} = \frac{|(d1,d3)|}{|(d1,d2,d3,d4,d5)|} = \frac{2}{5} = 0.4$. In other words, according to the Jaccard measure, protein P1 and P2 are 40% similar in their domain composition. Similarly for P1 and P5, the Jaccard link weight is calculated as $W_{P1,P5} = \frac{|(d1,d2,d3,d4) \cap (d4,d1,d10,d40,d7,d9,d12,d52,d100)|}{|(d1,d2,d3,d4) \cup (d4,d1,d10,d40,d7,d9,d12,d52,d100)|} = \frac{2}{11} = 0.18$. In this case, P1 and P5 are roughly 18% similar in their domain composition. After we have decided on the similarity function, the final PPN is built following two simple steps. In the first step, the data files that contain protein information are parsed to collect the constituent domains of each protein. If the training data contains only sequences, InterProScan [25, 14], a widely used protein domain identifier, is used to find the domains associated with each of the protein sequences.

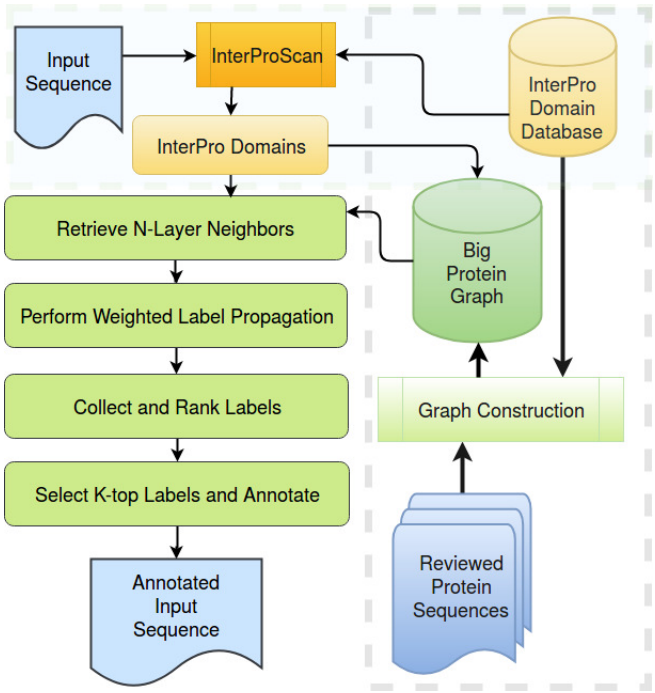
2.2 Label propagation for protein function annotation

After building the PPN from the reviewed proteins, the network is ready to be used for the function annotation of new protein sequences. A simple label propagation algorithm was designed to perform the annotation task. Given the constituent domains of the input protein sequence, all of its neighboring proteins and their annotations are retrieved from the PPN. Once the neighbors have been obtained, their labels are then weighted with link weights. The details of our label propagation algorithm are described in Algorithm 1. Overall, for a given input sequence, the annotation is calculated according to the flow diagram shown in Figure 1(b).

Figure 1(a) shows an example of a protein from the Protein Data Bank (PDB), entry 2rk2A, which is annotated using the network built from the Uniprot/SwissProt database. In this example, only 1-layer neighbours are considered for label propagation. It is evident from the network that 2rk2A has five neighbours with two distinct labels (EC 1.5.1.3 and 3.5.58). However, based on neighbor count and link weights, the EC



(a)



(b)

Fig. 1. An example Protein Domain Network is shown in 1(a) to annotate 2rk2A enzyme 1(b) shows the Annotation work-flow of GrAPFI. The shaded portion of the work-flow is pre-computated using the reviewd proteins for UniprotKB/Swissprot and InterPro Domain Database.

Algorithm 1 Label Propagation in a PPN

Input: A PPN $G = (V, E)$ and an unknown protein u with domain list d and minimum domain similarity θ
Output: EC Annotations
procedure LABELPROPAGATION
    $Annotations \leftarrow \emptyset$
    $N' \leftarrow FilterNeighbors(N(u), \theta)$
    for each neighbor $v \in N'$ **do**
        $EC_v \leftarrow CollectAnnotations(v)$
        for each ec in EC_v **do**
            if $ec \in Annotations$ **then**
                $Annotations_{ec} \leftarrow Annotations_{ec} + W_{u,v}$
            else
                $Annotations_{ec} \leftarrow W_{u,v}$
            end if
        end for
    end for
    Rank the *Annotations*
    Select the top ranked function and assign it to the protein u
    end Procedure

function $FilterNeighbors\ (N(u), \theta))$
    $N' \leftarrow \emptyset$
    for each $v \in N(u)$ **do**
        if $W_{u,v} >= \theta$ **then**
            $N' \leftarrow N' \cup \{v\}$
        end if
    end for
    return N'

label 1.5.1.3 has a greater weight than 3.5.5.8. Therefore, 2RK2A is annotated with EC 1.5.1.3. If desired, node neighbour may be selected in other ways to reflect the requirements of the problem at hand.

3 Experiments

EC annotations use a four digit numbering system with a hierarchical structure [6]. The first level (digit) describes one of six main enzyme classes: (i) oxidoreductases, (ii) transferases, (iii) hydrolases, (iv) lyases, (v) isomerases and (vi) ligases. The second digit describes a more specific sub-class of the top-level enzyme class. Similarly, the third digit typically indicates a specific enzyme type, while the fourth digit typically denotes a specific enzyme substrate. Here, we consider the assignment of an EC number to be correct if the first three digits of the assigned EC number match the first three digits of the ground-truth annotation from SwissProt.

To evaluate *GrAPFI*, we used Leave One Out (LOO) cross validation on four major taxonomic species from the March 2018 Release of UniProt/SwissProt, namely Viruses, Archaea, Eukaryota, and Bacteria. Additionally, we performed an accuracy test using a previously described benchmark dataset [33] to compare the performance of *GrAPFI* with [18] and [19]. The datasets were downloaded from the UniProtKB portal and filtered to retrieve only those proteins that are annotated with at least one EC number. This benchmark dataset contains 318 protein enzymes from the PDB[1]. Removing benchmark proteins that are not annotated with any EC annotation, gave 297 enzymes. We used InterProScan [14] to find the domain composition of these enzymes, and we then applied label propagation on a pre-built network of reviewed proteins having over 260,000 nodes and several million edges. Table 1 summarizes the network properties of the PPNs of the four selected taxonomies.

Dataset	# Nodes	# Edges	Average Degree	# Domains	Total EC
Viruses	3208	478447	298.28	1031	150
Archaea	10619	1168710	220.12	2499	727
Eukaryota	55042	30753219	1117.45	6744	2832
Bacteria	193429	409837148	4237.6	6480	2902

Table 1. A brief summary of the PPNs built for the Viruses, Archaea, Eukaryota and Bacteria datasets.

3.1 Evaluation Metrics

The performance of the method is measured using precision, recall, and F-Measure. Additionally, the coverage is also reported. To summarize the prediction result for cross-validation, the average precision and average recall is computed as $Pr_{avg} = \frac{1}{M} \sum_{\forall p \in P} Pr_p$

[1] Protein Data Bank, https://www.rcsb.org/

and $Re_{avg} = \frac{1}{M}\sum_{\forall p \in P} Re_p$, respectively. Here, M is the number of proteins that are predicted with at least one EC number, P is the set of Proteins to be tested using LOO cross validation and $Pr_p = \frac{k_p}{m_p}$ and $Re_p = \frac{k_p}{n_p}$ are the precision and recall computed for the protein p. Here n_p is the set of known functions for the protein p, m_p is the set of predicted functions and k_p is the overlap between the two sets. The F-Measure is the harmonic mean of the precision and recall and is computed as $F_{Measure} = \frac{2 \times Pr_{avg} \times Re_{avg}}{Pr_{avg} + Re_{avg}}$. Coverage indicates the number of proteins that a method can annotate in a particular test case. Coverage is represented relative to the number of proteins in the test case. Here, coverage is calculated as $Coverage = \frac{M}{N}$, where M is the number of proteins for which at least one EC is predicted and N is the total number of proteins in the Test set. A coverage of 90% means that 90% of the proteins in the Test set are annotated (rightly or wrongly).

3.2 GrAPFI Performance Analysis

We applied LOO cross validation to validate the results. In LOO cross validation, one node is held back for testing while the other nodes constitute the network. Then, the test node is annotated using label propagation. The results are presented in Figures-2(d) to 2(o) considering different domain similarity thresholds (10% to 50%). For each of the four datasets, the precision, recall, and F-measure are presented for the first second and third EC digit.

Precision and Recall For every protein, precision is measured by comparing the predicted EC number annotation with the actual annotation. To measure the precision and recall for each annotation, four different cases are handled.

1 There is only one EC annotation in ground truth and prediction. In such a case, both the precision and recall is 1.0 if they agree on the first n digits for n-digit EC annotation. Otherwise they are 0.
2 There is more than one EC annotation in the ground truth but the number of predicted EC number is one. In such a case, the EC numbers having the same first n digits are considered one single EC and are compared with the prediction. For example, let us suppose the predicted EC is [2.6.1.78] against the ground truth ECs: [2.6.1.11 and 2.6.1.81]. In this case the precision and recall is 1.0, as both the ground truth and the prediction agree on the first three digits. However, if ground truth ECs are [2.6.1.11, 2.6.1.81, 2.6.7.13] and the predicted EC is [2.6.1.78] then the precision is 1.0 (because the predicted EC has a matching ground truth EC) and the recall is 0.5 (because the prediction has missed one ground truth EC).
3 There is only one ground truth EC but more than one predicted EC. For example, let us suppose the predicted ECs are [2.6.1.11, 2.6.1.81] and the ground truth EC is [2.6.1.78]. In this case, the precision is 1.0 and the recall is 1.0 as both the ground truth and the prediction agree on the first three digits. To extend this example, let us suppose the predicted ECs are [2.6.1.11, 2.6.1.81, 2.6.7.13] and the ground truth EC is [2.6.1.78]. Here, the precision is 0.5 (because one out of 2 predicted ECs has an agreeing ground truth EC) and recall is 1.0 (because the ground truth EC is correctly predicted).

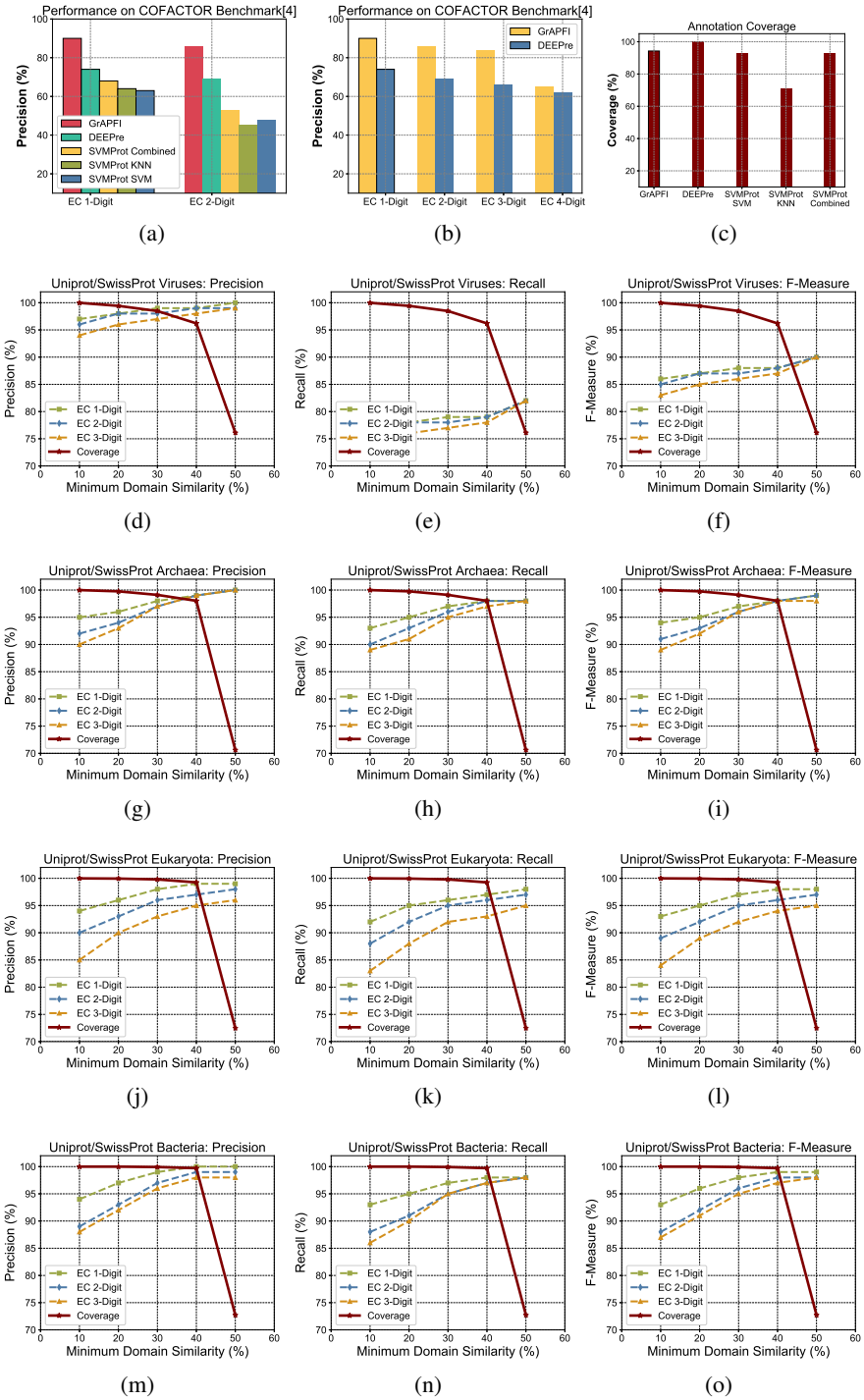


Fig. 2. Figures 2(a) to 2(c) show a comparison of GrAPFI with other methods. Only the precision is reported, as recall equals precision here. Figures 2(d), 2(e) and 2(f) show the precision, recall, and f-measure for the Viruses dataset for 1-digit, 2-digit, and 3-digit EC number predictions using different similarity thresholds. Similarly, Figures 2(g) to 2(i) show the corresponding results for Archaea, Figures 2(j) to 2(l) for Eukaryota, and Figures 2(m) to 2(o) for Bacteria.

4 There are more than one ground truth and more than one prediction ESc. For example, let us suppose the predicted ECs are [2.6.1.11, 3.6.1.81] and the ground truth ECs are [2.6.1.11, 3.4.4.81]. In this case, the precision is 0.5 and the recall is 0.5 because one out of two predicted ECs has an agreeing ground truth EC.

Performance measures are based on the top-most annotations. As shown in Figures 2(a) to 2(o), *GrAPFI* achieves high precision in most cases. However, a relatively lower recall is observed for the Viruses dataset (see Figure 2(e)). High precision values reflect the precise annotation capability of *GrAPFI*.

We note that *GrAPFI* is tested on different domain similarity thresholds that serves the purpose of filtering the collected neighbors. In fact, in a network of millions of nodes, it is common to have a large number of neighbors sharing domains with a node. It is also true that all of the neighbors do not contribute equally. In this context, a filtering approach is adapted to reduce the neighbors keeping those with higher affiliation with test node. The domain similarity of 10% considers neighbors with distant relationship, whereas a 50% domain similarity represents a closer relationship among the neighbors. As the higher domain similarity threshold filters out a large number of neighbors, cases may arise where the test protein is left without any neighbors, and hence has no annotations to propagate. The coverage curve (red lines) explains the phenomenon in Figures 2(d) to 2(o).

To compare *GrAPFI* with other methods, we ran each method on a benchmark dataset from COFACTOR [33]. We have collected the results of DEEPre, SVMProt (SVM, KNN and Combined) and measured the precision. When comparing the performance, we considered only the top predicted annotation. Because most of these methods predict only the first two EC digits, Figure-2(a) presents performance results for EC number annotation at the 1-digit and 2-digit level. On the other hand, because DEEPre can predict four-digit EC numbers, Figure-2(b) presents a comparison between DEEPre and *GrAPFI* for the full four-digit EC annotation predictions. In all the cases, *GrAPFI* gives better performance measures than the others tested here. The coverage is reported in Figure-2(c). This figure shows that DEEPre has slightly better coverage than our *GrAPFI*. However, *GrAPFI* is more precise than DEEPre in predicting EC number annotation.

4 Conclusion

We have presented *GrAPFI*, a novel network based approach for automatic protein functional annotation. The method first constructs a network representation of the UniProtKB protein database and then applies a label propagation method to the network in order to propagate annotations from reviewed proteins to non-reviewed ones. The experimental results on the UniProtKB/SwissProt data and a selected benchmark dataset show that *GrAPFI* provides high quality EC number annotations of protein sequence data.

In the future, we aim to customize *GrAPFI* for large scale annotations on large dataset such as UniProtKB/TrEMBL. The algorithm is already designed in such a way that both the Graph Construction and Label Propagation steps can be done in parallel and in a distributed way. Our next aim is to implement a distributed version of *GrAPFI*

using a parallel/distributed framework such as Hadoop MapReduce [7], BLADYG [2] and Spark [32].

Acknowledgements

This work was partially supported by the CNRS-INRIA/FAPs project "TempoGraphs" (PRC2243). Bishnu Sarker is a doctoral student funded by an INRIA CORDI-S contract.

References

1. Altschul, S.F., Madden, T.L., Schffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Research **25**(17), 3389–3402 (1997). DOI 10.1093/nar/25.17.3389
2. Aridhi, S., Montresor, A., Velegrakis, Y.: Bladyg: A graph processing framework for large dynamic graphs. Big Data Research **9**, 9 – 17 (2017)
3. Chou, K.C.: Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. Current Proteomics **6**(4), 262–274 (2009)
4. Chua, H.N., Sung, W.K., Wong, L.: Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. Bioinformatics **22**(13), 1623–1630 (2006)
5. Consortium, T.U.: Uniprot: a hub for protein information. Nucleic Acids Research **43**(D204-D212) (2015). DOI 10.1093/nar/gku989. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4384041/
6. Cornish-Bowden, A.: Current iubmb recommendations on enzyme nomenclature and kinetics. Perspectives in Science **1**(1-6), 74–87 (2014)
7. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM **51**(1), 107–113 (2008)
8. Dobson, P.D., Doig, A.J.: Predicting enzyme class from protein structure without alignments. Journal of molecular biology **345**(1), 187–199 (2005)
9. Finn, R.D., Clements, J., Eddy, S.R.: Hmmer web server: interactive sequence similarity searching. Nucleic Acids Research **39**(2), W29–W37 (2011). DOI 10.1093/nar/gkr367. URL http://dx.doi.org/10.1093/nar/gkr367
10. Gattiker, A., Michoud, K., Rivoire, C., Auchincloss, A.H., Coudert, E., Lima, T., Kersey, P., Pagni, M., Sigrist, C.J., Lachaize, C., Veuthey, A.L., Gasteiger, E., Bairoch, A.: Automated annotation of microbial proteomes in SWISS-PROT. Computational Biology and Chemistry **27**(1), 49–58 (2003). DOI 10.1016/s1476-9271(02)00094-4
11. Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., Takagi, T.: Assessment of prediction accuracy of protein function from protein–protein interaction data. Yeast **18**(6), 523–531 (2001)
12. Huang, W.L., Chen, H.M., Hwang, S.F., Ho, S.Y.: Accurate prediction of enzyme subfamily class using an adaptive fuzzy k-nearest neighbor method. Biosystems **90**(2), 405–413 (2007)
13. des Jardins, M., Karp, P.D., Krummenacker, M., Lee, T.J., Ouzounis, C.A.: Prediction of enzyme classification from protein sequence without the use of sequence similarity. In: Proc Int Conf Intell Syst Mol Biol, vol. 5, pp. 92–99 (1997)
14. Jones, P., Binns, D., Chang, H.Y., Fraser, M., Li, W., McAnulla, C., McWilliam, H., Maslen, J., Mitchell, A., Nuka, G., et al.: Interproscan 5: genome-scale protein function classification. Bioinformatics **30**(9), 1236–1240 (2014)

15. Kretschmann, E., Fleischmann, W., Apweiler, R.: Automatic rule generation for protein annotation with the c4.5 data mining algorithm applied on swiss-prot. Bioinformatics **17 10**, 920–6 (2001)
16. Kumar, N., Skolnick, J.: Eficaz2. 5: application of a high-precision enzyme function predictor to 396 proteomes. Bioinformatics **28**(20), 2687–2688 (2012)
17. Kummerfeld, S.K., Teichmann, S.A.: Protein domain organisation: adding order. BMC Bioinformatics **10**(1), 39 (2009)
18. Li, Y., Wang, S., Umarov, R., Xie, B., Fan, M., Li, L., Gao, X.: Deepre: sequence-based enzyme ec number prediction by deep learning. Bioinformatics **34**(5), 760–769 (2018). DOI 10.1093/bioinformatics/btx680
19. Li, Y.H., Xu, J.Y., Tao, L., Li, X.F., Li, S., Zeng, X., Chen, S.Y., Zhang, P., Qin, C., Zhang, C., et al.: Svm-prot 2016: a web-server for machine learning prediction of protein functional families from sequence irrespective of similarity. PloS one **11**(8) (2016)
20. Lu, L., Qian, Z., Cai, Y.D., Li, Y.: Ecs: an automatic enzyme classifier based on functional domain composition. Computational biology and chemistry **31**(3), 226–232 (2007)
21. 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**(suppl_1), i302–i310 (2005)
22. Nagao Chioko, N.N., Kenji, M.: Prediction of detailed enzyme functions and identification of specificity determining residues by random forests. PloS one **9**(1) (2014)
23. Nasibov, E., Kandemir-Cavas, C.: Efficiency analysis of knn and minimum distance-based classifiers in enzyme family prediction. Computational biology and chemistry **33**(6), 461–464 (2009)
24. Quester, S., Schomburg, D.: Enzymedetector: an integrated enzyme function prediction tool and database. BMC bioinformatics **12**(1), 376 (2011)
25. Quevillon, E., Silventoinen, V., Pillai, S., Harte, N., Mulder, N., Apweiler, R., Lopez, R.: Interproscan: protein domains identifier. Nucleic acids research **33**(suppl_2), W116–W120 (2005)
26. Rahman, S.A., Cuesta, S.M., Furnham, N., Holliday, G.L., Thornton, J.M.: Ec-blast: a tool to automatically search and compare enzyme reactions. Nature methods **11**(2), 171 (2014)
27. Schwikowski, B., Uetz, P., Fields, S.: A network of protein–protein interactions in yeast. Nature biotechnology **18**(12), 1257 (2000)
28. Shen, H.B., Chou, K.C.: Ezypred: a top–down approach for predicting enzyme functional classes and subclasses. Biochemical and biophysical research communications **364**(1), 53–59 (2007)
29. Volpato, V., Adelfio, A., Pollastri, G.: Accurate prediction of protein enzymatic class by n-to-1 neural networks. BMC bioinformatics **14**(1), S11 (2013)
30. Yang, J., Yan, R., Roy, A., Xu, D., Poisson, J., Zhang, Y.: The i-tasser suite: protein structure and function prediction. Nature methods **12**(1), 7 (2015)
31. Yu, C., Zavaljevski, N., Desai, V., Reifman, J.: Genome-wide enzyme annotation with precision control: Catalytic families (catfam) databases. Proteins: Structure, Function, and Bioinformatics **74**(2), 449–460 (2009)
32. Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J., et al.: Apache spark: a unified engine for big data processing. Communications of the ACM **59**(11), 56–65 (2016)
33. Zhang, C., Freddolino, P.L., Zhang, Y.: Cofactor: improved protein function prediction by combining structure, sequence and proteinprotein interaction information. Nucleic Acids Research **45**(1), 291–299 (2017)
34. Zhao, B., Hu, S., Li, X., Zhang, F., Tian, Q., Ni, W.: An efficient method for protein function annotation based on multilayer protein networks. Human genomics **10**(1), 33 (2016)