A New Feature Vector Based on Gene Ontology Terms for Protein-Protein Interaction Prediction

Sanghamitra Bandyopadhyay and Koushik Mallick

Abstract—Protein-protein interaction (PPI) plays a key role in understanding cellular mechanisms in different organisms. Many supervised classifiers like Random Forest (RF) and Support Vector Machine (SVM) have been used for intra or inter-species interaction prediction. For improving the prediction performance, in this paper we propose a novel set of features to represent a protein pair using their annotated Gene Ontology (GO) terms, including their ancestors. In our approach, a protein pair is treated as a document (bag of words), where the terms annotating the two proteins represent the words. Feature value of each word is calculated using information content of the corresponding term multiplied by a coefficient, which represents the weight of that term inside a document (i.e., a protein pair). We have tested the performance of the classifier using the proposed feature on different well known data sets of different species like *S. cerevisiae, H. Sapiens, E. Coli, and D. melanogaster*. We compare it with the other GO based feature representation technique, and demonstrate its competitive performance.

Index Terms—Protein interaction prediction, GO based feature, kernel methods for PPI prediction

1 Introduction

Understanding the protein-protein interaction network is a challenging problem in computational biology. It is an important task to understand the biological functions of gene products and to discover involvement of new proteins in different pathways. This will help in discovering the relationship between diseases and genes, and may result in the identification of new drug targets. Though various advanced high throughput experimental assays provide a lot of interactions, the total number of explored interactions are still very few compared to the whole proteome. It is believed that proteins exhibit their functions by interacting with other proteins and none of them work in isolation. So the total number of interactions is expected to be very large, although only a small fraction is known. Discovering new interactions through laboratory experiments is expensive and time consuming. For this reason computational methods have been applied to predict PPI. Various genomic and proteomic knowledge are used to build computational models for PPI prediction. Some state-of-the methods used the information derived from phylogenetic conservation and co-evolution [1], phylogenetic distance [2], gene fusion [3], co-localization of genes in a chromosome [4], gene expression profile [5], protein domain interaction [6], network topology parameters [7], [8], protein structures [9], [10] etc. Protein motif domain information are used in a probabilistic way to infer interactions by Gomez et al. [11] and Huang et al[6]. A fine review work can be found in [12].

In [13], [14], [15], [16] etc. PPI prediction has been viewed as a supervised classification problem with features prepared from various proteomic information. High confidence interacting protein pairs, taken from some well known interaction database, are used as positive class. However, identifying negative data is difficult since not much information is available about protein pairs that do not interact [17]. A general strategy to build negative protein pairs is to select random protein pairs excluding those that are known to interact [18]. For classification, Martin et al. [13] used SVM with pairwise kernel with protein sequence spectrum features. Features prepared from different types of sequence signatures proposed in [15], [19], [20] and [21], were used to make consensus decision using SVM classifier [16]. Ben-hur et al. [14] used combination of pairwise kernels, where multiple kernels were derived from different features obtained from various information sources namely, GO similarity, k-mer sequence signature, protein motif domain data from PFAM, Emotif database and BLAST similarity score of interacting proteins from other species etc. Finally they combined these derived kernels to train an SVM classifier. SVM classifier is also used for features derived from auto covariance of physio-chemical properties of amino acids in a protein sequence [21] and conjoined triad based features [15] from protein sequence for PPI prediction.

GO is a controlled and structured vocabulary of terms, that describe information about protein's localization within the cells (i.e., cellular component or CC), participation in biological processes (BP) and associated molecular functions (MF)[22]. GO terms are connected by a directed acyclic graph (DAG) where nodes of the graph are represented by terms and edges among nodes represent relations. Commonly used types of relations between GO terms are *is_a*, *part_of*, *regulates*. In the DAG, child terms represent more specific biological concepts while parent terms represent more general concepts. Generally interacting proteins often participate in similar BP and/or exhibit similar MF and/or are co-localized in similar CC [23], [24], and hence exhibit

• *The authors are with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India.*
E-mail: sanghami@isical.ac.in, koushikbuie@gmail.com.

Manuscript received 23 Sept. 2015; revised 21 Feb. 2016; accepted 7 Apr. 2016. Date of publication 20 Apr. 2016; date of current version 4 Aug. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2016.2555304

high GO semantic similarity [23]. Many approaches to measure GO semantic similarity exist e.g., [25], [26], [27], [23], [28] and [29]. Many of them have shown that high GO similarity among proteins generally indicates interactions between them, with GO similarity value being considered as a measure of confidence of interaction. Therefore, interaction based on semantic similarity score is basically unsupervised in nature [30].

Ben-hur et al. [14] used a non-sequence kernel derived from GO similarity to train an SVM classifier for PPI prediction (the term non-sequence indicates that the kernel does not use any sequence information). GO similarity score between two proteins was considered as input to the kernel function. In their experiments [31] they used non-sequence kernel as a mixture of kernels derived from GO similarity score, BLAST score of orthologous interacting protein pairs and mutual clustering coefficient derived from PPI network. GO based non-sequence kernel was expressed by using the following equation [14]:

$$K_{non\_seq}((p_1,p_2),(p_1',p_2')) = K'(S_{GO}(p_1,p_2),S_{GO}(p_1',p_2')), \quad (1)$$

where S_{GO} is a GO similarity score between two proteins. Here, for computing the kernel for two pairs of proteins (p_1,p_2) and (p_1',p_2'), individual values of GO similarity between $p_1(or,p_1')$ and $p_2(or,p_2')$ are used. The kernel does not explicitly compare the similarity between the two protein pairs with respect to their annotated GO terms space. Tastan et al.[32] have used GO similarity value as features to predict viral host protein interaction.

Instead of using the GO similarity value of protein pairs directly as features, Maetschke et al. [30] used GO terms in a binary feature vector for supervised learning. Random Forest and Bayesian classifier were used in their work. Here if a protein pair (p_1 and p_2) has an experimental evidence about their interaction then another protein pair (p_1' and p_2'), having similar GO term annotation profile as (p_1 and p_2), may also interact. Learning was done with feature vectors prepared from GO terms annotating the two proteins, as well as their ancestors. For the purpose of feature representation, various combinations of ancestor terms were considered, viz., all ancestor terms of an annotated pair, terms that are common from the lowest common ancestor up to the root of an annotated pair, terms up to lowest common ancestor of annotated term pairs of a protein pair etc. Finally, the feature vector was expressed as a combined binary vector, based on presence or absence of terms selected by any of the above mentioned combination methods of induced terms for a protein pair (i.e., if the kth term is present in a term set of a protein pair, then the kth position of the feature vector is 1 otherwise 0). So if a particular proteome has a total of n unique GO term annotations (including all ancestor terms), then the dimension of the feature vector is n. An intuition behind this representation is to train a classifier using associated GO terms of interacting and non-interacting protein pairs. However, the simple 0-1 based representation of feature has a major problem. Here all the terms annotating a protein is represented by 1 regardless of where the terms are located. Note that terms appearing in different levels in a GO graph have different information content (IC) values [33]. IC is defined as the negative logarithm of probability of occurrence of a GO term in a proteome. It is defined as follows [33].

$$IC(t) = -\log(p(t)), \quad (2)$$

$$p(t) = \frac{annotation(t) + \sum_{d\in descendant(t)}(annotation(d))}{\sum_{c\in descendant(root)} annotation(c)}, \quad (3)$$

where $p(t)$ is the probability of occurrence of a term t, $annotation(t)$ is the number of direct annotations of the term t in the considered proteome, and $\sum_{d\in descendant(t)}(annotation(d))$ is the number of indirect annotations by the descendant in the GO graph of the term t. The summation of the former two counts are divided by the total number of the annotated terms in the proteome. GO terms with higher IC values indicate more specific biological concepts; representing all the terms by the same value of 1 may be misleading. Another important issue is that a term may be annotated directly or indirectly by any descendant of that term. Therefore, a good representation of features based on GO terms should consider the specificity of that term as well as the annotation depth of the indirectly annotated terms. In the approach proposed in this article, all annotated GO terms, including their ancestors, are treated as a bag of words as applied in document classification [34]. The IC value of each indirectly annotated GO term is multiplied by a coefficient which decreases with its distance from the directly annotated term. This coefficient will differentiate the value of a GO term on the basis of whether the term is directly or indirectly annotated by a protein pair.

The article is organized as follows: In Section 2, a discussion about the feature vector construction method is presented in detail. The different datasets used in this article are described in Section 3. For performance evaluation of the new feature, Gaussian kernel SVM is used in this article. The classifier performance is evaluated using Area Under the Curve of Receiver Operating Characteristics (AUC-ROC) and precision etc. These results are discussed at Section 4. Section 5 concludes the article.

2 A New Feature Vector Using GO Terms

Annotated GO terms of a protein are extracted from the GO annotation database *www.geneontology.org*. Let a protein P be annotated by n_p number of GO terms, $termset(P) = (t_1,t_2\ldots,t_{n_p})$ and let $ancs(t_i) = (t_i,t_i^1,t_i^2\ldots t_i^{root})$ be all ancestor terms of a term t_i. The actual term set of the protein P is a union of the ancestors of all terms in $termset(P)$, expressed as $termset_{all}(P) = \bigcup_{\forall i\in termset(P)} ancs(t_i)$. A common approach to represent GO term based feature in vector space model is by using the $termset_{all}(P)$ to prepare a binary encoded vector of GO terms on the basis of presence or absence of a term. The following section discusses the adopted approach for assigning weights to the members of the feature vector.

2.1 Computation of Weights of GO Terms

In this work, two types of weights for each GO term present in the protein feature are considered. The first one is a global significance of the term, which is represented by the IC value of the term. The second is a local weight of that

term i.e., topological weight of that term generated from the annotation list of the protein. Let t_i be an annotated term where $t_i \in termset(P)$ for a protein P. Then all terms in $ancs(t_i)$ are assigned their corresponding IC values as their global weights. The local weight of a term is computed as follows: Let $SP_{i,k} = (t_i, t_i^1, t_i^2 \ldots, t_i^{k-1}, t_i^k)$ be the shortest path from t_i to an ancestor t_i^k. Then the coefficient value c_i^k for t_i^k is computed as:

$$c_i^k = d_{t_i,t_i^1} * d_{t_i^1,t_i^2} \ldots d_{t_i^{k-1},t_i^k}, \tag{4}$$

where,

$$d_{i,j} = \begin{cases} 0, & \text{if } t_i \text{ and } t_j \text{ are not neighbour in } SP_{i,k} \\ \alpha_1, & \text{if } Rel(t_i,t_j) = \text{''is\_a''} \\ \alpha_2, & \text{if } Rel(t_i,t_j) = \text{''part\_of''} \\ 0, & \text{otherwise.} \end{cases}$$

In this article, the ranges of α_1 and α_2 can be written as $0 \leq \alpha_2 \leq \alpha_1 \leq 1$ in a combined way. This coefficient value is 1 for t_i which is a directly annotated term. The coefficient c_i^k is multiplied with the IC value of t_i^k, for $k = 1, 2, \ldots, root$. This computation helps to make the feature values of directly and indirectly annotated terms different with indirectly annotated terms necessarily having smaller values. It is noted that a term in the set $ancs(t_i)$, will be multiplied with a coefficient which decreases with increasing length of the shortest path from t_i. In this way the coefficient values of all ancestor terms in the set $ancs(t_i)$ are calculated. The weight of a term t_a^i is calculated by the following Eqn. 5:

$$w'_{t_a^i} = IC(t_a^i) * c_{t_a^i}. \tag{5}$$

The above weighted formulation is repeated for every term in $termset(P)$. If a term t_a is indirectly annotated by multiple m descendant terms $(t_{a_1}, t_{a_2}, \ldots t_{a_m}) \in termset_{all}(P)$, the coefficient value of the term t_a in the feature of protein P is the summation of individual coefficients ($c_{t_{a_i}}$) generated by each of those descendant annotated terms of the term t_a for protein P. The final weight w_{t_a} for the term t_a is computed as:

$$w_{t_a} = w'_{t_{a_1}} + w'_{t_{a_2}} \cdots + w'_{t_{a_n}} = IC(t_a) * \sum_{i=1}^{m} c_{t_{a_i}}. \tag{6}$$

Finally the weight value of individual terms are placed in an n dimensional feature vector for a protein P, where n is the number of unique GO terms in all proteins of a proteome. Representation of the feature vector for a pair of proteins (P_1, P_2) from individual features of P_1 and P_2 is discussed in Section 2.1.

A protein is considered as a bag of words (GO terms). Notably, this feature representation is motivated by tf-idf (term frequency-inverse document frequency) representation used for document classification [35]. In this regard, the coefficient (shown in Eqn. 4) that represents the weighted frequency of a term with respect to annotations in a protein, is analogous to tf whereas the IC value ($IC(t_a^i)$) is the specificity of a term (i.e., word) with respect to overall annotations in a proteome. This is analogous to idf.

For efficient computation of the weights of terms, the following steps are followed as specified in the Algorithm 1.

Algorithm 1. Steps to compute the weight of features for proteins

1: For every GO term t in the Gene Ontology Omnibus Database make an adjacency list graph structure ($AL_{GO}(t)$) for that term, with its immediate descendants.
2: Extract the annotated term set of each characterized protein (i.e., $termset_{BP}(P_i)$ for protein P_i) for BP ontology from an annotation dataset. The same is done for CC and MF ontology. Build sperate list of uniquely annotated terms l_{BP} for BP ontology from the $termset_{BP}(P_i)$. The Same is done for other ontologies (l_{CC}, l_{MF}) individually.
3: Extract the ancestor subgraph upto root term for every term $t_i \in l_{BP}$ starting from $AL_{GO}(t_i)$. Compute the shortest path from t_i to all ancestor nodes in that subgraph and calculate coefficients of those terms using the Eqn. 4. Keep the termset of the ancestor subgraph with their coefficient values for the term t_i in a list named $Ancs_{BP}(t_i)$. Repeat this step for all other ontology term lists (l_{CC}, l_{MF}).
4: For all terms $t \in Ancs_{BP}(t_i)$ calculate the weight by using the coefficient value from Step 3 and Eqn. 6 and do the same for other lists ($Ancs_{CC}(t_i)$, $Ancs_{MF}(t_i)$).

2.2 Feature Construction of Protein Pairs

For feature representation of a protein pair (P_1, P_2) one of the two approaches is usually followed: (i) combination $(F(P_1) \oplus F(P_2))$, or (ii) concatenation $[F(P_1), F(P_2)]$. Here $F(P)$ is the feature vector corresponding to protein P. Here $\oplus$ operation will add element by element the feature values of the two proteins. Let length of $F(P)$ be n. Then combination and concatenation operations produce feature vectors of length n and $2n$ respectively. For the prediction of PPI using GO features, a protein pair feature representation using concatenation is not preferred because with this approach $[F(P_1), F(P_2)]$ and $[F(P_2), F(P_1)]$ do not have a same representation in feature space. Consequently different kernel value is computed for different ordering of concatenation of the protein feature during SVM optimization. In contrast, feature using combination approach is independent of the ordering of a protein pair. Generally two interacting proteins share similar types of GO terms in the ontology (i.e., generally they participate in similar biological process and molecular function, and/or may be co-localized in the same cellular component). This is the key intuition for representing a pair of proteins as a document with a bag of words with the combination approach. The combination approach of representing features of protein pairs is used in this article. Once the features of a protein pair is determined, an SVM classifier is trained on PPI data. The direct pairwise kernel K of two combined vectors $(F(P_1) \oplus F(P_2))$ and $(F(P_1') \oplus F(P_2'))$ is given by the following equation:

$$K(F(P_1) \oplus F(P_2), F(P_1') \oplus F(P_2')) = k'(F(P_1), F(P_1')) + k'(F(P_1), F(P_2')) + k'(F(P_2), F(P_1')) + k'(F(P_2), F(P_2')), \tag{7}$$

where $k'(a,b) = a * b^T$, is a linear kernel. Note that this kernel is symmetric in nature, and operates on the feature of

protein pairs directly. This becomes possible because of the combination approach adopted. This kernel can be used on the top of a polynomial or Gaussian kernel to map the data into more complex non-linear space.

2.3 Selection of Induced GO Terms

In this article we consider two different categories of induced GO term based features that were found to perform well in [30]. For each category, all the features are assigned certain weights. The first category of feature comprises all annotated GO terms and their ancestor terms for a protein pair. Weights are assigned to all these terms using Eqn. 6. This feature is mentioned as weighted all ancestors or WAA in this article.

Instead of considering all annotated GO terms in the feature vector, the second category includes all induced terms up to the lowest common ancestor (ULCA) [30]. The induced GO terms are selected by the following method. For each pair of annotated GO terms $((t_i, t_j) : t_i \in termset(P_1), t_j \in termset(P_2))$ for a protein pair (P_1, P_2), the terms located up to the lowest common ancestor (LCA) of (P_1, P_2), including t_i and t_j, are chosen. Finally the union of the ULCA induced term sets resulting from all the term pairs $((t_i, t_j) : t_i \in termset(P_1), t_j \in termset(P_2))$ is used for characterizing the protein pair (P_1, P_2). As in the case for WAA, weights are assigned to all these ULCA induced terms using Eqn. 6, where instead of all descendants, only the ULCA induced descendants are considered. This feature is called Weighted ULCA or WULCA in this article.

3 Datasets

3.1 Gene Ontology Data

We have constructed the GO graph from Gene Ontology omnibus data collected from *www.geneontology.org* [36]. Two types of relations among GO terms are considered in this article, *is_a* and *part_of*. We have collected GO annotation dataset for different species from Uniprot database and gene ontology omnibus data. Ancestors of annotated GO terms are retrieved from the GO graph.

3.2 PPI Datasets for Different Species

PPI datasets for various species were collected from existing databases. Only those proteins that have GO annotation with respect to all the three GO aspects (BP, MF and CC) are considered. Descriptions of the datasets are provided in the following paragraphs.

Ben-hur et al. [14] datasets: Three yeast PPI datasets from [14] have been used in this article. The first one, referred to as BIND, was collected from BIND database (10517 positive interactions and same number of negative interactions). The second, referred to as Reliable-BIND, consisted of 750 reliable interactions filtered from the latter database. Finally, the third one, referred to as DIP-MIPS, was collected from DIP/MIPS database (4,837 positive interactions and around 10,000 negative interactions). In all these three datasets, randomly selected protein pairs with no known interaction were used as the negative set.

Park's [16] dataset: The yeast PPI data in [16] has been used in this article. The dataset was prepared from DIP core interaction dataset. It had a total of 3,734 positive interactions and around 3,00,000 randomly selected negative interaction. So positive to negative data ratio was about 1:100.

Meatschke et al. [30] datasets: The PPI datasets of S. Cerevisiae (SC), H. Sapiens (HS), E. Coli (EC) and D. Melanogaster (DM) as used in [30] have been taken in this article. These data sets were collected from the STRING [37] database. There are 15,238, 3,490, 1,167 and 321 positive interactions for yeast, human, E. Coli and D. Melanogaster, respectively. For negative data, of the same size as the positive data, they have used random protein pairs.

Yu et al. [38] dataset: High confidence (HC) PPI dataset (15,408 positive data) of human comprising interactions present in both BioGRID and HPRD datasets was considered [38]. They have used two different types of negative datasets namely, random pair of proteins and a balanced random pair of proteins. In balanced random protein pair, the number of times a protein appears in negative dataset is same as that in the positive dataset. However, note that Park et al. [18] have shown that this kind of negative data selection fails to simulate the behavior of the global protein pair population which is better simulated by randomly selected negative data.

4 Results and Discussion

In this section the PPI prediction performance using the proposed weighted GO induced term features (WAA and WULCA) are compared with those of a recently developed approach containing binary features (AA and ULCA) [30]. SVM classifier with Gaussian kernel (C-SVC of libsvm [39]) is used for classification. Features generated from the three ontology (BP, CC and MF) are combined into a single vector in all the experiments.

Preparation of negative data is an important issue for PPI prediction. Various approaches proposed in the literature include protein pairs constructed from different cellular locations [14], involved in different biological processes [40], etc. However, this type of negative data selection criteria can make the dataset biased, specially when GO based features are used for PPI prediction. Most of the data sets considered in this article used randomly selected negative data. Only the data of Yu et al. [38] used balanced random negative dataset.

Performance of the different approaches are reported using the measures Area under the curve of ROC (AUC-ROC) and ROC50 (AUC-ROC50), sensitivity and specificity. AUC-ROC is a curve plotting the sensitivity versus (1-specificity) at different thresholds of the classifier prediction score. AUC-ROC measures the accuracy of the complete set of predictions. AUC-ROC50 is the area under the curve of ROC upto first 50 false positives. It measures the high confidence prediction performance of a classifier.

4.1 Estimation of Parameters and Classifier Details

Two parameters α_1 and α_2 are used for weighting of the is_a and $part\_of$ relations respectively in the Eqn. 4. More preference is given to the is_a relation than the $part\_of$ relation in this article. Because is_a relation is considered more direct than the $part\_of$ relation and former is also highly frequent than the latter. Smaller weight value of $part\_of$ relation will contribute lower weight values to the terms those

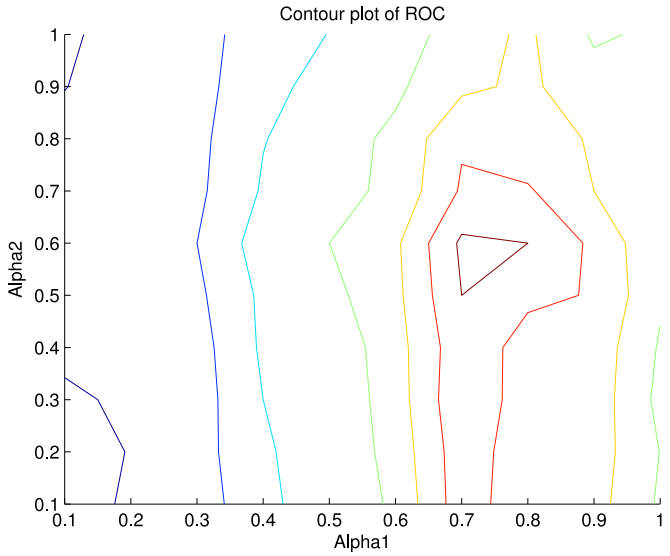

Fig. 1. Illustration of contour plot of different values of parameters α_1 and α_2 used for the weights of *is_a* and *part_of* relation. Cross validation AUC score was used to draw the contour plot for the dataset of S. Cerevisiae of Maetschke et al. [30]. Horizontal axis shows α_1, vertical axis shows α_2.

are linked by that relation and reverse will happen for an is_a relation link. For identifying the appropriate values of these parameters, we have run the grid search for two parameters α_1 and α_2 in the range of [0,1] with a step size 0.1. ROC of the classification using SVM is used to find the good value of parameters. Contour plot was plotted (See Fig. 1) in the range of parameters for the WULCA feature on the S. Cerevisiae dataset of Maetschke et al. [30]. It is found that the performance is best with $\alpha_1 \in$ [0.7,0.8] and $\alpha_2 \in$ [0.5,0.6]. For the other dataset and features it is observed that the same range of values of α_1 and α_2 performed best.

C-SVC of libsvm with Gaussian kernel is used as classifier. We have tested with many other kernels but we found that Gaussian kernel performs the same as, often better than, other kernels for all features and for all data sets. For the sake of brevity we have omitted results with other kernels. A critical issue for using SVM classifier is to select parameters for optimized performance, which is determined by cross validation. C is the regularization parameter, i.e., used as penalty for misclassification. Here we have kept the value of misclassification penalty for positive data high, because random negative dataset is less reliable than positives. In our all experiments the values of C for positive and negative data are considered with a ratio 9:1. We have found that the value of $C \leq 20$ is effective for data sets larger than 5,000 instances and positive to negative data ratio 1:1. Table 1 reports the C and γ values for the SVM.

TABLE 1
Parameter Value Used in Our Experiments Using SVM Classifier Obtained by Cross Validation Using Proposed Feature WULCA

Author name	Data set name	C	Gamma
Yu et al.	High confidence positive with random negative	20	0.02
Ben-hur	DIP-MIPS	47	0.3
	BIND	25	0.23
Park	Yeast PPI	10	0.3
Meatche et al.	S. Cerevisiae	10	0.1
	H. Sapiens	10	0.03

4.2 Result on Ben-hur et al.'s Dataset [14]

Table 2 reports the result on the data set from [14]. The second column shows the results obtained in the authors' own work [14], where they used five-fold cross validation (to keep parity all the other results were also obtained using five-fold cross validation). For BIND they have used GO features using non-sequence kernel whereas for reliable-BIND a mixture of non-sequence kernels prepared from GO features as well as blast score and MCC features are used in their [31] work. For DIP-MIPS they have used sequence spectrum count based features with pairwise kernel. As can be seen, the SVM classifier trained with proposed WULCA and WAA features performed the best (0.89) as compared to AA (0.83) and ULCA (0.88) features (see Table 2). The result reported in [14] is the lowest. In the case of the proposed features, the ROC50 score (0.65) for the WULCA feature is also the best with respect to the other features. For the case of reliable-BIND, an AUC-ROC score of 0.95 was reported in [14]. From the Table 2 we see that the proposed GO term based WULCA shows a marginal improvement (AUC 0.96). Note that in this experiment they [14] have used some other types of features in addition to GO similarity score feature in the non sequence kernel. However, the proposed weighted GO term based feature is found to be strong enough to be combined with a simple kernel to provide a superior performance. Again, both WAA and WULCA encoding of features performed better than AA and ULCA based approach, respectively. This indicates that the proposed weighted features may have an advantage over the unweighted ones.

For DIP-MIPS PPI data set, pairwise kernel SVM with sequence spectrum count was used in [14]. They prepared

TABLE 2
AUC-ROC Values for Ben-hur et al. [14] Data Sets

Dataset name	Ben-hur et al.[14] AUC/AUC50	AA AUC/AUC50	Proposed WAA AUC/AUC50	ULCA AUC/AUC50	Proposed WULCA AUC/AUC50
BIND	0.68/-	0.83/0.57	0.89/0.64	0.88/0.63	**0.89/0.65**
Reliable BIND	0.95/-	0.89/0.58	0.94/0.61	0.94/0.60	**0.96/0.62**
DIP-MIPS	0.87-0.97/0.08-0.46 (negative pair threshold range 0.5-0.04)	0.88/0.46	0.93/0.51	0.92/0.49	0.93/0.52

Results of DIP-MIPS dataset using GO features do not use any threshold based filtering of negative data.

TABLE 3
AUC-ROC Values for Park et al. [16] Data Sets with 1:10 Positive to Negative Ratio

Park AUC/AUC50	AA AUC/AUC50	WAA AUC/AUC50	ULCA AUC/AUC50	WULCA AUC/AUC50
0.85/-	0.85/0.54	0.926/0.58	0.917/0.58	**0.93/0.59**

negative interaction dataset by filtering with GO CC similarity threshold i.e., the similarity of each negative interaction pair is lower than the threshold. As is reported in the table, AUC-ROC score varied from 0.87 to 0.97 when thresholds of GO CC similarity was varied from 0.50 to 0.04. In the other approaches, no such thresholding of the negative dataset was used. Both of the proposed WAA and WULCA feature with Gaussian SVM kernel achieved an AUC-ROC score 0.93 but The proposed WULCA feature demonstrated superior performance with respect to WAA feature in terms of the ROC50 score. Again, it was observed that the weighted features always outperformed the unweighted ones. As expected, the results reported in [14] were somewhat better because of the negative data thresholding, since protein pairs taken from different cellular component are less likely to interact.

4.3 Result on Yeast Dataset of Park [16]

Here, the effectiveness of proposed weighted features are compared to other unweighted GO based features for PPI prediction using yeast dataset of Park [16] is discussed. In that work, the author combined the results of four SVMs, each using a different set of sequence based features, in a ensembling approach. Four-fold cross validation was used.

Table 3 shows the performance of the approach of Park [16] along with those of the other GO-based features using SVM as the underlying classifier. It is observed that the proposed WULCA feature (AUC-ROC = 0.93) again outperforms all others including the Park's approach. WAA feature (AUC-ROC = 0.926) performed very similar with the WULCA. Moreover, the ULCA inducer based feature (AUC-ROC = 0.917) outperforms the AA inducer based feature (AUC-ROC = 0.85) by a large margin. This was also observed in [30] where Random Forests classifier was used.

A further experimentation was carried out on the yeast dataset using the WULCA features by varying the positive to negative training data ratio. The results are shown in Table 4. As expected, with increasing proportion of negative data, the classifier becomes biased toward the negatives, with more positive data getting misclassified as negatives. Consequently, the specificity (true negative rate) increases while sensitivity (true positive rate) decreases. However, AUC scores remain more or less stable, showing very little change.

TABLE 5
AUC-ROC Values for Meastache et al. [30] Data Sets of S. Cerevisiae, H. Sapiens, E. Coli, and D. Melanogaster

Species	Proposed WULCA AUC/ AUC50	Proposed WAA AUC/ AUC50	ULCA AUC/ AUC50
S. Cerevisiae	**0.95 / 0.64**	0.95/0.63	0.92 / 0.630
H. Sapiens	0.93 /0.60	**0.95/0.63**	0.90 / 0.57
E. Coli	**0.96 / 0.60**	0.96/0.59	0.93 / 0.58
D. Melanogaster	**0.86 / 0.52**	0.85/0.51	0.82 / 0.50

4.4 Result on Maetschke et al.'s Dataset [30]

Here the performance of the proposed as well as existing features for PPI prediction on datasets of different species as mentioned in Section 3 are compared. Results for ULCA, proposed WAA and WULCA features only are included. Results of AA is not shown because their performance is consistently poor. Ten fold cross validation results are reported in Table 5. As can be seen, the proposed WAA feature provides best AUC and AUC50 scores for the dataset of H. Sapiens. For rest of the cases WULCA feature shows best performance and followed by the performance of WAA feature with a very close margin. For illustration purpose ROC plot for S. Cerevisiae dataset is shown in Fig. 2.

4.5 Result on Yu et al.'s Dataset [38]

In this dataset there are High confidence human protein interaction pairs and two types of negative dataset namely, random and balanced random as mentioned in Section 3. In their work [38], the authors used protein sequence based features by counting the amino acid trigrams, resulting in a set of 8,000 features. Tensor product pairwise kernel (TPPK8000) [13] is used with SVM classifier and trained with five-fold cross validation. With random negative dataset, AUC score of 0.82 was achieved with TPPK8000 kernel. Using SVM classifier with ULCA, proposed WAA and WULCA features are trained with five fold cross validation, the AUC-ROC scores obtained were 0.80, 0.82 and 0.83, respectively (see Table 6). The ROC plot is provided in Fig. 3. With balanced random negative dataset, the classifier achieved AUC-ROC scores of 0.64, 0.67 and 0.68 with ULCA, WAA and WULCA features respectively, whereas the TTPK8000 kernel provides AUC score of 0.60. From the results, we see that for both the datasets, WULCA feature performs the best, especially for the balanced random negative dataset.

TABLE 4
AUC-ROC on the Dataset Used by Park et al. [16] Tested With Different Positive to Negative Ratio

Data Ratio	WULCA			WAA			ULCA		
Positive:negative	AUC/std	SN	SP	AUC/std	SN	SP	AUC/std	SN	SP
1:1	0.930/.02	0.83	0.85	0.926/.02	0.82	0.85	0.917/.01	0.8	0.82
1:5	0.932/.03	0.82	0.89	0.928/.01	0.83	0.89	0.909/.03	0.79	0.85
1:10	.931/ .01	0.78	0.95	0.926/.02	0.76	0.94	0.903/.01	0.76	0.89
1:100	0.928 /.02	0.76	0.98	0.919/.03	0.74	0.97	0.894/.02	0.75	0.95

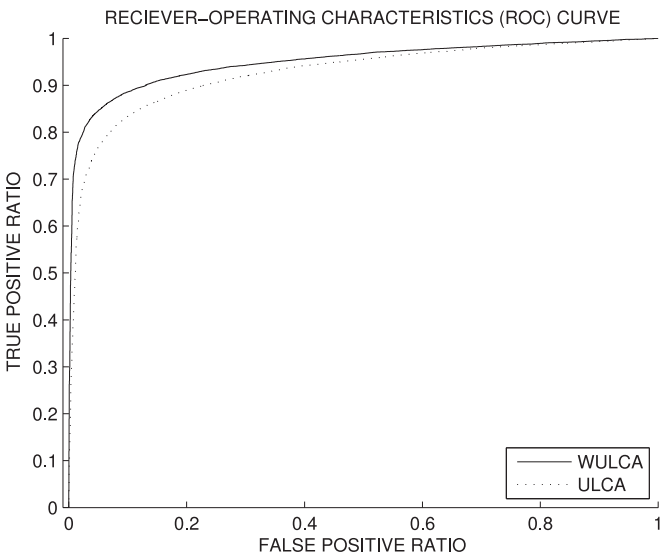

Fig. 2. ROC plot using SVM classifier with WULCA and ULCA features on S. Cerevisiae data set of Maetschke et al. [30].

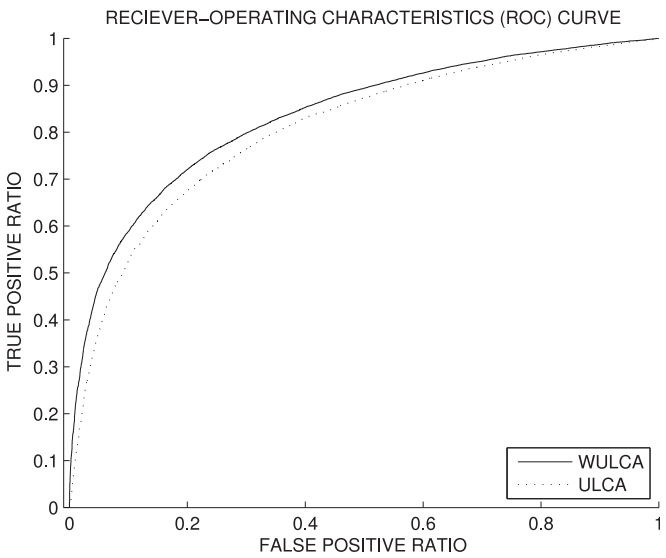

Fig. 3. ROC plot using SVM classifier with WULCA and ULCA features on human PPI data set of Yu et al. [38] with random negative data.

4.6 Discussion

In this work, a new GO-based feature representation method has been proposed for supervised PPI prediction. Each protein pair is represented by a weighted feature vector, where the weights are derived from the term specificities and the topological structure of the GO graph. An SVM classifier is trained on biologically validated interacting protein pairs, while using a randomly generated negative set. The trained classifier is used to predict protein protein interactions. From the results of the various experiments, it is found that the proposed weighted feature vector representation (WULCA) performs better than the unweighted version (ULCA) which is statistically significant (See Table 7). In Table 2, the result of BIND dataset represents the GO-based features which are trained by non-sequence GO kernel [14] as mentioned in the Section 1. The proposed feature WULCA achieved best performance over the non-sequence GO kernel and other features. For the reliable-BIND dataset, a combination of different non-sequence kernels (mixture of kernels prepared from GO similarity and sequence homology score) were used. Note that, GO based WULCA feature (0.96) are more discriminative than non-sequence kernels (0.95). As mentioned earlier, several authors [16], [31], [41] have reported PPI prediction using sequence based features. From the results shown in Section 4 (see results obtained on DIP-MIPS, Park and High confidence positive datasets in Table 2, Table 3 and Table 6 respectively) it can be concluded that the performance of weighted GO based features is better than the sequence signature based features for PPI prediction. From the analysis of Ben-hur et al. [14] it is revealed that sequence based features improves the performance of a classifier for PPI prediction if it is combined with GO based features. Both the features have large number of dimensions. Computation of kernel matrix for the full dataset using these features is computationally expensive. In this context GO term based feature is computationally simple and performance is better.

All three combined GO ontology terms produce around 6,800 and 17,000 features for yeast and human, respectively. With this large dimensionality and small number of instances, the use of SVM as the underlying classifier is proposed in this article. Note that here we could also have used the RF-classifier, but in such cases, it has a greater chance of over fitting. The result of RF classifier in the Table 8 demonstrated that proposed weighted GO feature performed better than the unweighted version. For understanding the effect of over fitting, we have evaluated the performance of both RF and SVM classifiers on the training set itself (training error). In this experiment, for the S. Cerevisiae dataset shown in Table 8, RF classifier (AUC-ROC = 0.99) demonstrated a higher accuracy with respect to the SVM classifier (AUC-ROC = 0.97) with WULCA feature. During the 10 fold cross validation testing RF classifier (AUC-ROC = 0.935) performed inferior compared to SVM (AUC-ROC = 0.95) for the WULCA feature. Similar behavior was observed for other GO based features. These observations exhibit that the RF classifier is prone to overfitting for PPI

TABLE 6
AUC-ROC Values for Yu et al. [38] Data Sets

High confidence positive	Yu. et al (TPPK8000)	ULCA	WAA	WULCA
Random negative	0.82	0.80	0.82	**0.83**
Balanced Random negative	0.60	0.64	0.67	**0.68**

TABLE 7
McNemar's Test P-values of Significant Performance Improvement of Feature WULCA Compared to ULCA on Some of the Datasets Used in this Article

Author name	Data set name	WULCA Vs ULCA
Yu et al.	High confidence positive with random negative	2.4×10^{-9}
Ben-hur	DIP-MIPS	1.9×10^{-4}
	BIND	1.3×10^{-5}
Park	Yeast PPI	4.7×10^{-18}
Maetschke et al.	S. Cerevisiae	2.3×10^{-21}
	H. Sapiens	5.4×10^{-22}

TABLE 8
AUC-ROC Values Using the Random Forest Classifier

Dataset	AA	WAA	WULCA	ULCA
S.Cerevisiae Meatchske et al. [30]	0.88	0.93	**0.935**	0.90
Human HC positive with random negative Yu et al. [38]	0.74	0.798	**0.80**	0.77

prediction with GO based feature vectors. Again, with a huge number of instances and unbalanced positive to negative data ratio (often observed in the domain of PPI prediction), the performance of RF classifier becomes unreliable and very much time consuming as well. We have seen that SVM performs better than the RF classifier for the above mentioned cases. However, selection of parameter values is an important issue with the SVM classifier.

A reduced GO term set (i.e., GO-slim term set) can be used for reducing the high dimensionality of GO features. Notably, according to the analysis of Maetschke et al. [30], reduced GO term set affects the performance of a classifier. A naive classifier with less overhead in determining its parameter values or linear SVM (viz., *liblin*) [42] can be easily trained. This is the advantage of using a reduced dimensional data. Since there is millions of protein pairs in the dataset, therefore it is difficult to use the standard feature extraction methods (like PCA or KPCA) that require 'eigen decomposition'. Hence, feature extraction method from such large data sets can be developed in future. For the purpose of the analysis of the performance of reduced dimension GO feature, we have trained the classifier using the different combination of features prepared from the three ontologies (BP, CC and MF). From the result shown in the Table 9 it is revealed that features from BP ontology alone achieved best accuracy among the features from other ontology. Best performance was achieved with the merged features of all three ontologies. Merged features of BP and CC have shown the nearest performance to the best one. Some GO terms are annotated by automatic electronic inference (known as IEA evidence code). An analysis is done to realize the effect of performance of the classifier for the case of the exclusion of IEA annotated terms in the feature vector. It can be seen from the results of Table 10, that the performance of SVM classifier is not affected much by the exclusion of IEA annotated GO terms.

TABLE 9
AUC-ROC Values Using Different Combination of Features of Different Sub-Ontologies (BP, CC and MF) of the S. Cerevisiae Dataset [30] Using SVM Classifier and 10 Fold Cross Validation

Combination of ontologies	WAA	WULCA	ULCA
BP	0.936	0.94	0.90
CC	0.927	0.928	0.88
MF	0.83	0.84	0.78
(BP, CC)	0.942	0.947	0.912
(BP, MF)	0.927	0.93	0.90
(MF, CC)	0.90	0.91	0.88
(BP, CC, MF)	**0.95**	**0.95**	**0.92**

TABLE 10
Results Simulates the Effect of IEA Annotated GO Term Inclusion (IEA+) and Exclusion (IEA-) in the Feature Vector

Data set	IEA+			IEA-		
	WAA	WULCA	ULCA	WAA	WULCA	ULCA
Human HC positive with random negative Yu et al.[38]	0.82	0.83	0.80	0.817	0.826	0.792
S.Cerevisiae Meatchske et al. [30]	0.95	0.95	0.92	0.946	0.943	0.917

AUC-ROC value is reported for SVM classifier.

5 Conclusion

In this work we have proposed a new feature representation technique based on annotated GO terms of a protein pair. A supervised classifier (SVM) is used for the purpose of predicting novel PPIs. Unsupervised approaches for PPI prediction use GO semantic similarity as confidence value, which is not efficient compared to supervised. Features from GO terms have been developed earlier with the help of various inducer term sets including the ancestor terms. However, usually a binary (0-1) representation based on the presence or absence of terms has been used. In this work we have considered weighted values of feature instead of binary values, where the weight is proportional to the global annotation statistics and topological position of terms in a GO subgraph. Results demonstrate that the proposed inducer based weighted feature (WULCA) performs better than the simple binary inducer based approach for all the benchmark PPI datasets. Computation of WAA feature vector is easy compared to WULCA feature and the performance is also competitive. From the analysis in the article it can be concluded that GO term based features have better performance than sequence based spectrum count features.

In future, all types of existing relations in a GO graph may be considered which will add more terms in the feature vector and will thereby increase performance. This feature can also be applied for different types of classification problems where GO semantic similarity is used, namely in viral host protein interaction prediction, drug target prediction and discovery of functional similarity of protein motifs.

References

[1] F. Pazos, M. Helmer-Citterich, G. Ausiello, and A. Valencia, "Correlated mutations contain information about protein-protein interaction," *J. Molecular Biol.*, vol. 271, no. 4, pp. 511–523, 1997.
[2] R. A. Craig and L. Liao, "Phylogenetic tree information aids supervised learning for predicting protein-protein interaction based on distance matrices," *BMC Bioinf.*, vol. 8, no. 1, p. 1, 2007.
[3] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis, "Protein interaction maps for complete genomes based on gene fusion events," *Nature*, vol. 402, no. 6757, pp. 86–90, 1999.
[4] T. Dandekar, B. Snel, M. Huynen, and P. Bork, "Conservation of gene order: A fingerprint of proteins that physically interact," *Trends Biochem. Sci.*, vol. 23, no. 9, pp. 324–328, 1998.
[5] C. Von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork, "Comparative assessment of large-scale data sets of protein–protein interactions," *Nature*, vol. 417, no. 6887, pp. 399–403, 2002.
[6] C. Huang, F. Morcos, S. P. Kanaan, S. Wuchty, D. Z. Chen, and J. A. Izaguirre, "Predicting protein-protein interactions from protein domains using a set cover approach," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 4, no. 1, pp. 78–87, Jan./Mar. 2007.

[7] A. Birlutiu, F. d'Alche Buc, and T. Heskes, "A bayesian framework for combining protein and network topology information for predicting protein-protein interactions," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 22, no. 3, pp. 538–550, May/Jun. 2015.
[8] S. Wuchty, "Topology and weights in a protein domain interaction network–a novel way to predict protein interactions," *BMC Genomics*, vol. 7, no. 1, p. 122, 2006.
[9] R. Singh, J. Xu, and B. Berger, "Struct2net: Integrating structure into protein-protein interaction prediction," in *Proc. Pacific Symp. Biocomput.*, 2006, pp. 403–414.
[10] R. Hosur, J. Xu, J. Bienkowska, and B. Berger, "IWRAP: An interface threading approach with application to prediction of cancer-related protein–protein interactions," *J. Molecular Biol.*, vol. 405, no. 5, pp. 1295–1310, 2011.
[11] S. M. Gomez, W. S. Noble, and A. Rzhetsky, "Learning to predict protein–protein interactions from protein sequences," *Bioinf.*, vol. 19, no. 15, pp. 1875–1881, 2003.
[12] S. Pitre, M. Alamgir, J. R. Green, M. Dumontier, F. Dehne, and A. Golshani, "Computational methods for predicting protein–protein interactions," *Proc. Int. Conf. Adv. Biochem. Eng./Biotechnol.*, 2008, pp. 247–267.
[13] S. Martin, D. Roe, and J.-L. Faulon, "Predicting protein–protein interactions using signature products," *Bioinf.*, vol. 21, no. 2, pp. 218–226, 2005.
[14] A. Ben-Hur and W. S. Noble, "Kernel methods for predicting protein-protein interactions," *Oxford Bioinf.*, vol. 21, no. 4, pp. 38–46, Mar. 2005.
[15] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein–protein interactions based only on sequences information," *Proc. Nat. Academy Sci.*, vol. 104, no. 11, pp. 4337–4341, 2007.
[16] Y. Park, "Critical assessment of sequence-based protein-protein interaction prediction methods that do not require homologous protein sequences," *BMC Bioinf.*, vol. 10, no. 1, p. 419, 2009.
[17] A. Ben-Hur and W. S. Noble, "Choosing negative examples for the prediction of protein-protein interactions," *BMC Bioinf.*, vol. 7, p. S2, 2006.
[18] Y. Park and E. M. Marcotte, "Revisiting the negative example sampling problem for predicting protein–protein interactions," *Bioinf.*, vol. 27, no. 21, pp. 3024–3028, 2011.
[19] S. Pitre, F. Dehne, A. Chan, J. Cheetham, A. Duong, A. Emili, M. Gebbia, J. Greenblatt, M. Jessulat, N. Krogan, et al., "Pipe: A protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs," *BMC Bioinf.*, vol. 7, no. 1, p. 365, 2006.
[20] S. Pitre, C. North, M. Alamgir, M. Jessulat, A. Chan, X. Luo, J. Green, M. Dumontier, F. Dehne, and A. Golshani, "Global investigation of protein–protein interactions in yeast saccharomyces cerevisiae using re-occurring short polypeptide sequences," *Nucleic Acids Res.*, vol. 36, no. 13, pp. 4286–4294, 2008.
[21] Y. Guo, L. Yu, Z. Wen, and M. Li, "Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences," *Nucleic Acids Res.*, vol. 36, no. 9, pp. 3025–3030, 2008.
[22] G. O. Consortium, et al., "The gene ontology (go) project in 2006," *Nucleic Acids Res.*, vol. 34, pp. D322–D326, 2006.
[23] X. Wu, L. Zhu, J. Guo, D.-Y. Zhang, and K. Lin, "Prediction of yeast protein–protein interaction network: Insights from the gene ontology and annotations," *Nucleic Acids Res.*, vol. 34, no. 7, pp. 2137–2150, 2006.
[24] J. P. Miller, R. S. Lo, A. Ben-Hur, C. Desmarais, I. Stagljar, W. S. Noble, and S. Fields, "Large-scale identification of yeast integral membrane protein interactions," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 102, no. 34, pp. 12 123–12 128, 2005.
[25] S. Bandyopadhyay and K. Mallick, "A new path based hybrid measure for gene ontology similarity," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 1, pp. 116–127, Jan./Feb. 2014.
[26] D. Lin, "An information-theoretic definition of similarity." in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 296–304.
[27] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," arXiv preprint cmp-lg/9709008, 1997.
[28] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, 1995, pp. 448–453.
[29] A. Schlicker, F. S. Domingues, J. Rahnenführer, and T. Lengauer, "A new measure for functional similarity of gene products based on gene ontology," *BMC Bioinf.*, vol. 7, no. 1, p. 302, 2006.
[30] S. R. Maetschke, M. Simonsen, M. J. Davis, and M. A. Ragan, "Gene ontology-driven inference of protein–protein interactions using inducers," *Bioinf.*, vol. 28, no. 1, pp. 69–75, 2012.
[31] A. Ben-Hur and W. S. Noble, "Kernel methods for predicting protein-protein interactions," *Bioinf.*, vol. 21, no. 1, pp. i38–i46, 2005.
[32] O. Tastan, Y. Qi, J. G. Carbonell, and J. Klein-Seetharaman, "Prediction of interactions between HIV-1 and human proteins by information integration." in *Proc. Pacific Symp. Biocomput.*, 2009, pp. 516–527.
[33] P. Resnik, "Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language," *J. Artif. Intell. Res.*, vol. 11, pp. 99–130, Feb. 1999.
[34] L. M. Manevitz and M. Yousef, "One-class SVMS for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, 2002.
[35] B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based framework for text categorization," *Procedia Eng.*, vol. 69, pp. 1356–1364, 2014.
[36] G. O. Consortium, et al., "Gene ontology annotations and resources," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D530–D535, 2013.
[37] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. Lin, P. Minguez, P. Bork, C. von Mering, et al., "String v9. 1: Protein-protein interaction networks, with increased coverage and integration," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D808–D815, 2013.
[38] J. Yu, M. Guo, C. J. Needham, Y. Huang, L. Cai, and D. R. Westhead, "Simple sequence-based kernels do not predict protein–protein interactions," *Bioinf.*, vol. 26, no. 20, pp. 2610–2614, 2010.
[39] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.
[40] M.-G. Shi, J.-F. Xia, X.-L. Li, and D.-S. Huang, "Predicting protein–protein interactions from sequence using correlation coefficient and high-quality interaction dataset," *Amino Acids*, vol. 38, no. 3, pp. 891–899, 2010.
[41] J. Yu, M. Guo, C. J. Needham, Y. Huang, L. Cai, and D. R. Westhead, "Simple sequence-based kernels do not predict protein-protein interactions," *Bioinf.*, vol. 26, no. 20, pp. 2610–4, 2010.
[42] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learning Res.*, vol. 9, pp. 1871–1874, 2008.

Sanghamitra Bandyopadhyay received the PhD degree in computer science in 1998 from the Indian Statistical Institute, Kolkata, India, where she currently serves as a professor and director. She received the prestigious S. S. Bhatnagar Award in 2010, Humboldt fellowship for experienced researchers, and the Senior Associateship of ICTP, Italy. She is a fellow of the Indian National Academy of Engineering and the National Academy of Science, India. She has co-authored six books and more than 250 research papers. Her research interests include pattern recognition, data mining, evolutionary computing, and bioinformatics. She is a senior member of the IEEE.

Koushik Mallick received the ME degree in computer science and engineering from Jadavpur University in 2009. He is currently working toward the PhD degree at Calcutta University while working at the Indian Statistical Institute, Kolkata, India. He is currently an assistant professor at RCC Institute of Information Technology, Kolkata, India.