J Syst Sci Complex (2010) 23: 917–930

ON NETWORK-BASED KERNEL METHODS FOR PROTEIN-PROTEIN INTERACTIONS WITH APPLICATIONS IN PROTEIN FUNCTIONS PREDICTION*

Limin LI · Waiki CHING · Yatming CHAN · Hiroshi MAMITSUKA

DOI: 10.1007/s11424-010-0207-y
Received: 9 November 2009 / Revised: 17 January 2010

Abstract Predicting protein functions is an important issue in the post-genomic era. This paper studies several network-based kernels including local linear embedding (LLE) kernel method, diffusion kernel and laplacian kernel to uncover the relationship between proteins functions and protein-protein interactions (PPI). The author first construct kernels based on PPI networks, then apply support vector machine (SVM) techniques to classify proteins into different functional groups. The 5-fold cross validation is then applied to the selected 359 GO terms to compare the performance of different kernels and guilt-by-association methods including neighbor counting methods and Chi-square methods. Finally, the authors conduct predictions of functions of some unknown genes and verify the preciseness of our prediction in part by the information of other data source.

Key words Diffusion kernel, kernel method, Laplacian kernel, local linear embedding (LLE) kernel, protein function prediction, support vector machine.

1 Introduction

Assigning biological functions to an uncharacterized protein is an immediate challenge in the post-genomic era. To our best knowledge, even for the most well-studied organisms such as yeast, there are still about one-fourth of the proteins remain uncharacterized. Recently, different data sources and different methods have been proposed to predict protein functions including those based on protein-protein interaction (PPI), structure, sequence relationship,

Limin LI
Department of Mathematics, Xi'an Jiaotong University, Xi'an 710049*, China.*
Email: liminli@mail.xjtu.edu.cn.
Waiki CHING · Yatming CHAN
Advanced Modeling and Applied Computing Laboratory, Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong, China.
Email: wching@hkusua.hku.hk; ymchan@maths.hku.hk.
Hiroshi Mamitsuka
Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011*, Japan.* Email: mami@kuicr.kyoto-u.ac.jp.
*This research is supported in part by HKRGC Grant 7017/07P, HKU CRCG Grants, HKU strategic theme grant on computational sciences, HKU Hung Hing Ying Physical Science Research Grant, National Natural Science Foundation of China Grant No. 10971075 and Guangdong Provincial Natural Science Grant No. 9151063101000021.

gene expression data, see for instance [1–5]. The classical methods for learning the protein functions are based on sequence similarity tools such as FASTA and BLAST. In such methods, the query protein sequence is used as an input to find a significantly similar sequence whose function has been characterized.

High-throughout experimental techniques have generated a large amount of data which are useful for inferring the functional roles of proteins. Gene expression data is one of these useful data sources, and several function prediction methods have been proposed[6−7]. However, discrepancies of prediction may arise due to the corruptions of gene expression data. Occasionally, the microarrays contain bad probes or are even damaged, and some locations in the gene expression matrix are corrupted[8]. protein-protein interaction (PPI) plays a key role in many cellular processes. The distortion of protein interfaces may lead to the development of many diseases. The global picture of protein interactions in the cell provides a new way to understand the mechanisms of protein recognition at the molecular level. This newly available large-scale PPI data gives an opportunity to study protein functions in the context of a network. The PPI data can be represented as a network, with nodes representing proteins and edges representing the interactions between the nodes. Many methods have been proposed to elucidate protein functions using PPI data. One of the simplest methods is the guilty-by-association methods, i.e., the neighbor-counting method[9]. The method predicts for a given protein up to three functions that are most common among its neighbors. The Chi-square method[10], it computes the Chi-square scores of function assignment and assign the functions with several largest scores to a given protein. Vazquez, et al.[11], Karaoz[12] and Nabieva[13] applied graph algorithms such as cut-based approach and flow-based approach for functional analysis. In contrast to the local neighbor-counting methods, these methods take into account the full topology of the network. Deng, et al.[14] proposed Markov Random Field (MRF) method to predict yeast protein functions based on a PPI network. They assign functions to unknown proteins with a probability representing the confidence of the prediction. From the experimental results, MRF method shows 52% precision and recall and is much better than those simple guilty-by-association methods. Lanckriet, et al.[15] considered a support vector machine (SVM) approach for predicting protein functions using a diffusion kernel on a protein interaction network. The diffusion kernel provides means to incorporate all neighbors of proteins in the network. Lee, et al.[16] developed a novel Kernel Logistic Regression (KLR) method based on diffusion kernel for protein interaction networks and showed that the prediction accuracy is comparable to the protein function classifier based on the SVM, using a diffusion kernel.

The remainder of this paper is structured as follows. Section 2 gives an introduction to the kernel methods. In Section 3, numerical experiments are given to demonstrate the effectiveness of our proposed method. Finally concluding remarks are given in Section 4 to address further research issues.

2 The Kernel Methods

In this section, we first give a brief description of kernel methods and then we present three network-based kernels: Diffusion kernel, Laplacian kernel, and local linear embedding (LLE) kernel. After the kernel is generated, SVM method is then applied to each GO term to classify whether a new gene is in the GO term or not.

Kernel methods[16−17] attempted to express the correlations or similarities between pairs of points in the data space Ω in terms of a kernel function K: $\Omega \times \Omega \mapsto R$, and thereby implicitly construct a mapping $\phi : \Omega \mapsto H_K$ to a Hilbert space (feature space) H_K, in which the kernel can be represented as an inner product: $K(x, y) = (\phi(x), \phi(y))$. Besides expressing the known structure of the data space, the function or the kernel K must satisfy two mathematical

requirements: i) it must be symmetric, i.e., $K(x, y) = K(y, x)$ and ii) it should be positive semi-definite. In fact, effectiveness of a kernel-based method lies on the fact that it can implicitly map a data point to a higher dimensional feature space which can better captures the inherent structure of the data. The kernel K of a graph G with N nodes is an $N \times N$ real symmetric matrix such that and its element K_{ij} represents the similarity between Node i and Node j. We will make use of the graph-like structure of a PPI network to construct the global similarity for any pair of proteins in the network, and perform SVM classification based on the kernel.

To facilitate our discussion, we introduce the following notations. Let G be a PPI network of N proteins. Then one can represent the network G by its adjacency matrix $W = (w_{ij}) \in \mathbb{R}^{N\times N}$, where $w_{ij} = 1$ means there is an edge between Node i and Node j in the network, 0 otherwise there is no edge between them. We define $D = (d_{ij})$, where

$$d_{ii} = \sum_j w_{ij} \quad \text{and} \quad d_{ij} = 0 \text{ if } i \neq j.$$

The graph Laplacian is defined as $L = D - W$. We consider the feature for each protein determined by its neighborhood relationship with all the other proteins, then the trivial linear kernel can be defined as $K_{\text{linear}} = W^{\mathrm{T}}W$.

Diffusion Kernel Kondor and Lafferty[17] proposed a general method for establishing similarities among the nodes of a graph based on a random walk on a graph. This method efficiently accounts for all possible paths connecting two nodes, and for the lengths of those paths. Nodes that are connected by shorter paths or by many paths are considered to be more similar to each other. Let the eigenvalue decomposition of L be

$$L = U \cdot \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N) \cdot U^{-1}, \tag{1}$$

then the kernel generated is defined as

$$K = U \cdot \text{diag}(\mathrm{e}^{-\frac{\sigma^2}{2}\lambda_1}, \mathrm{e}^{-\frac{\sigma^2}{2}\lambda_2}, \cdots, \mathrm{e}^{-\frac{\sigma^2}{2}\lambda_N}) \cdot U^{-1} = \mathrm{e}^{-\frac{\sigma^2}{2}L}. \tag{2}$$

Here $\text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ is the diagonal matrix having diagonal elements $\lambda_1, \lambda_2, \cdots, \lambda_n$. The diffusion constant σ controls the rate of diffusion through the network. By varying the parameter σ, one can get different kernels. The diffusion kernel has been applied by Lanckriet, et al.[18] in protein-protein interaction network to predict protein functions.

Laplacian Kernel This kernel[19] is a kind of network-based kernel and is generated by the adjacency matrix W. The Laplacian kernel is defined as

$$K = L^{\dagger} = (D - W)^{\dagger}, \tag{3}$$

where $L^{\dagger}$ is the pseudo-inverse of the matrix L.

Local Linear Embedding Kernel The LLE is an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embeddings for high-dimensional inputs[20]. The input of LLE is N high-dimensional data points (m dimension), and output is the corresponding N low-dimensional data points (d-dimension). The three main steps in LLE are the followings:

i) Identify the neighbors of each data point $x_i \in R^m$. Denote by N_i the index set for the k-neighbors of x_i;

ii) Compute the weights that best linearly reconstruct x_i from its neighbors. This can be done by solving the minimization problem:

$$\min_{A=(a_{ij})\in R^{N\times N}} \left\{ \sum_{i=1}^{N} \left| x_i - \sum_{j\in N_i}^{k} a_{ij}x_j \right|^2 \right\}. \tag{4}$$

iii) Find the low-dimensional embedding vectors by solving

$$\min_{Y=[y_1,\cdots,y_N]\in R^{d\times N}} \left\{ \sum_{i=1}^{N} \left| y_i - \sum_{j\in N_i}^{k} a_{ij} y_j \right|^2 \right\} \tag{5}$$

with the constraints $\frac{1}{N} YY^{\mathrm{T}} = I$ and $Y\boldsymbol{e} = 0$, where $\boldsymbol{e}$ is the column vector with all ones. It has been shown that this problem can be solved by the eigenvalue problem of the matrix $M = (I - A)^{\mathrm{T}}(I - A)$, where A is the weight matrix obtained in Step ii). The optimal d-dimensional embedding Y can be obtained by the $(N - 1 - d)$th to $(N - 1)$th eigenvectors of M when its eigenvalues are in decreasing order.

In the LLE method, we first constructs for each data point a local geometric structure that is invariant to translations and orthogonal transformations in its neighborhood. We then project the data points into a low-dimensional space that best preserves those local geometries. In the case of a PPI network, we assume that each protein can be represented as a m-dimensional vector and all the points lie on a d-dimensional manifold with noise, where m and d are both unknown. For each point, all its neighbors in the PPI network will then be used to construct the local geometry based on the hypothesis that the weights for its different neighbors are same in its neighborhood, thus we can put the weight matrix A in Step ii) to be the normalized adjacency matrix $A = D^{-1}W$. After Step iii) of LLE, the intuitive way to do the classification is to perform SVM on some kernel defined by the LLE output Y to classify proteins into different functional group.

Since the low dimension d is difficult to determine, we use the following alternative way to perform the SVM classification. Let $\lambda_{\max}$ be the largest eigenvalue of M, then the LLE kernel is defined as

$$K_{\mathrm{LLE}} = \lambda_{\max} I - M. \tag{6}$$

Here I is the identity matrix. It is easy to prove that the leading eigenvector of K_{LLE} is $\boldsymbol{e}$, and the second eigenvector up to the $(d + 1)$th eigenvector provide the d-dimensional LLE embedding Y. Let $K_{\mathrm{LLE}} = U\Lambda U^{\mathrm{T}}$ where $U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N]$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ with $\lambda_1 \geq \cdots \geq \lambda_N$ then $Y = [\boldsymbol{u}_2, \boldsymbol{u}_3, \cdots, \boldsymbol{u}_N]^{\mathrm{T}}$. Here we used this LLE kernel to perform SVM and to classify the proteins into different functions. In fact, there is a close relationship between this kernel and a Y-based kernel. We define a low dimensional kernel matrix based on low dimension embedding Y as $K_{\mathrm{Low}} = Y^{\mathrm{T}}\Lambda_d Y \in R^{N\times N}$ where $\Lambda_d = \mathrm{diag}(\lambda_2, \lambda_3, \cdots, \lambda_{d+1})$. It is easy to prove that

$$K_{\mathrm{LLE}} - K_{\mathrm{Low}} = \lambda_{\max}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}} + \sum_{i=d+2}^{N} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^{\mathrm{T}}. \tag{7}$$

This means when d is large enough, there is only a difference of a constant matrix with all same elements between LLE kernel K_{LLE} and Y-based kernel K_{Low}.

3 Experimental Results

3.1 Data Source

We use GO data taken from [21] in our numerical experiment. The gene association data is taken from SGD in Feb. 2008. The PPI data is downloaded from MIPS database, which contains a manually curated yeast protein interaction dataset[22] collected by curators from the literature.

3.2 The Gene Ontology

The Gene Ontology (GO) is a framework consisting of controlled vocabularies describing three aspects of gene product functions: i) molecular function, ii) biological process, and iii) cellular component. Each aspect of the functions is called an ontology. Each ontology is a directed acyclic graph (DAG), where the GO terms are represented as nodes in the graph and are arranged hierarchically from a general one to a specific one. Here functional annotation of protein is defined by GO biological process. The hierarchical structure of GO indicates that if a gene is assigned to one term, then the gene will be assigned to all ancestors of this term indirectly. In the following discussion, we assume that the genes associated to a node include all the indirect genes associated to this node. It should be noted that a gene can be in more than one GO class. For each GO term T, all proteins that are annotated with T are labeled as positive, while all proteins that are not annotated with T are labeled as negative. Generally speaking, for each GO term, the number of negative proteins far exceeds the number of positive proteins. In this case, to test and compare the efficiency of different method, we randomly select a subset of negative proteins so that the number of positives and negatives are equal. Thus for each GO term, after labeling the training set, one can use SVM technique to generate a SVM classifier, which will be used to classify the unknown proteins into positive or negative classes.

3.3 The Prediction Performance

We first extracte a subnetwork of the whole PPI network to make sure every protein in the subnetwork has been annotated by GO. The number of the nodes in the subnetwork used in the cross validation is 3187. We generated different kinds of graph kernels for these 3187 proteins. We note that we did not use all the GO Terms to check the classification performance because for most of GO Terms, there are too few positive genes (less than 30) and for some GO Terms, there are too many positive genes (more than 1000). We removed all these GO terms, and 359 GO terms are left. We then evaluated the classification performance by 5-fold cross validation using kernel methods with different kernels including linear kernel, LLE kernel, diffusion kernel and Laplacian kernel and guilt-by-association methods including neighbor counting method and Chi-square method. For the diffusion kernel, we chose the diffusion constant σ to be $0.5, 1, 2$ and 3. For each GO class, a classifier can be constructed by training the proteins in training data set. Then this classifier will be used to classify the proteins in the test data set into either positive or negative group. For each method, we calculate 359 AUCs for all the 359 GO Terms and an AUC for the multiple classification[15].

Figure 1 shows the results of cross validation on all the proteins in PPI, and Figure 2 gives the results for the balanced protein sets. The left of Figure 1 and Figure 2 show the ROC curves of different methods, and the right are the AUCs of 359 GO Terms. Table 1 reports the AUCs of different methods. From Figure 1, Figure 2, and Table 1, one can see that for any specific kernel, the unbalanced method is generally better than the current balanced method. This provides a direction for future research of the problem. One can also see that the kernel methods are better than both the Chi-square method and neighbor counting method. This implies that neighborhood relationships provide limited information for the functions of unknown proteins. Moreover, although the different kernel methods performs similarly, Laplacian kernel and diffusion kernel with diffusion constant 1 are a little better than other kernels. This implies that for the network of PPI, Laplacian kernel and diffusion kernel are good choices to mingling the network structure.

3.4 Prediction Results in Yeast Genome

In previous subsection, we have shown the effectiveness of the kernel type methods. Here,

in particular, we employ one of them, namely the LLE kernel on PPI network to predict the functions for the yeast genes which are uncharacterized by GO. We will show even the LLE kernel method can give nice prediction results. Note that some of them have been classified to some functional categories (FunCat[23]) in MIPS, which makes it possible to validate our prediction result after we manually created a mapping from GO Term to FunCat classes. We first extract the largest connected subnetwork of PPI, which includes 3396 known proteins and 645 unknown proteins. For each of the 359 GO terms, after building a classifier using the labeled proteins, we can determine whether a unlabeled protein is annotated with this GO function or not. Table 1 lists the predictions for the unknown proteins in GO. For each GO term in the first column, the predicted genes associated to it are listed in column 2, with the bracket classifying the genes to be three classes: 1) obviously supported by MIPS; 2) not obviously supported by MIPS; 3) unclassified in MIPS. From the table we can see that most of the predictions can be supported by the MIPS Comprehensive Yeast Genome Database (CYGD). For each of the 645 unknown proteins, the file 'Predicted_function_645_LLE.txt', which can be downloaded from [24] lists all its predicted functions. Among these 645 uncharacterized SGD genes, 6 genes cannot be found in FunCat and 92 genes are also classified as unknown in FunCat. The function of the remaining 547 unknown genes have been annotated in FunCat.

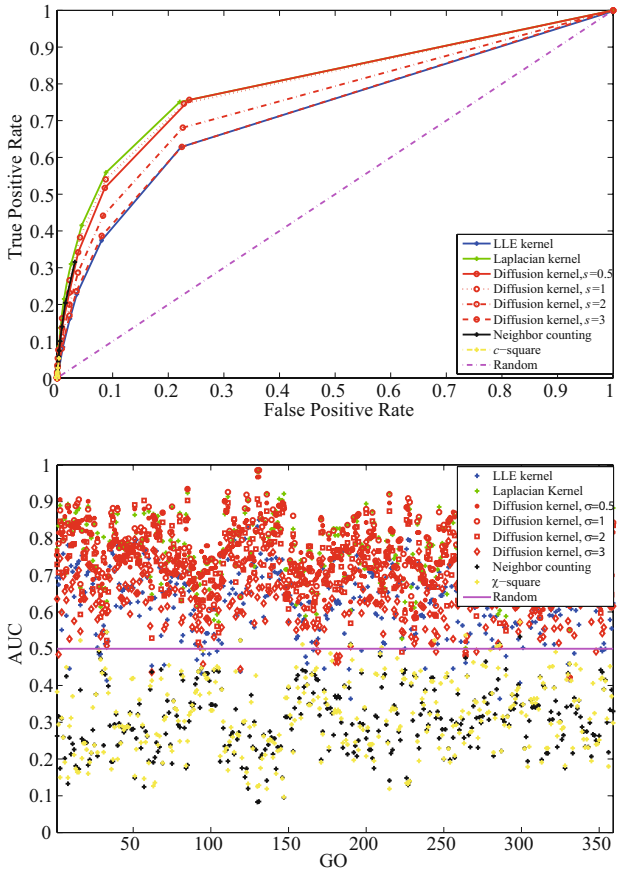

Figure 1 Prediction results using unbalanced methods. Left: ROC curves; Right: AUCs for different GO Terms

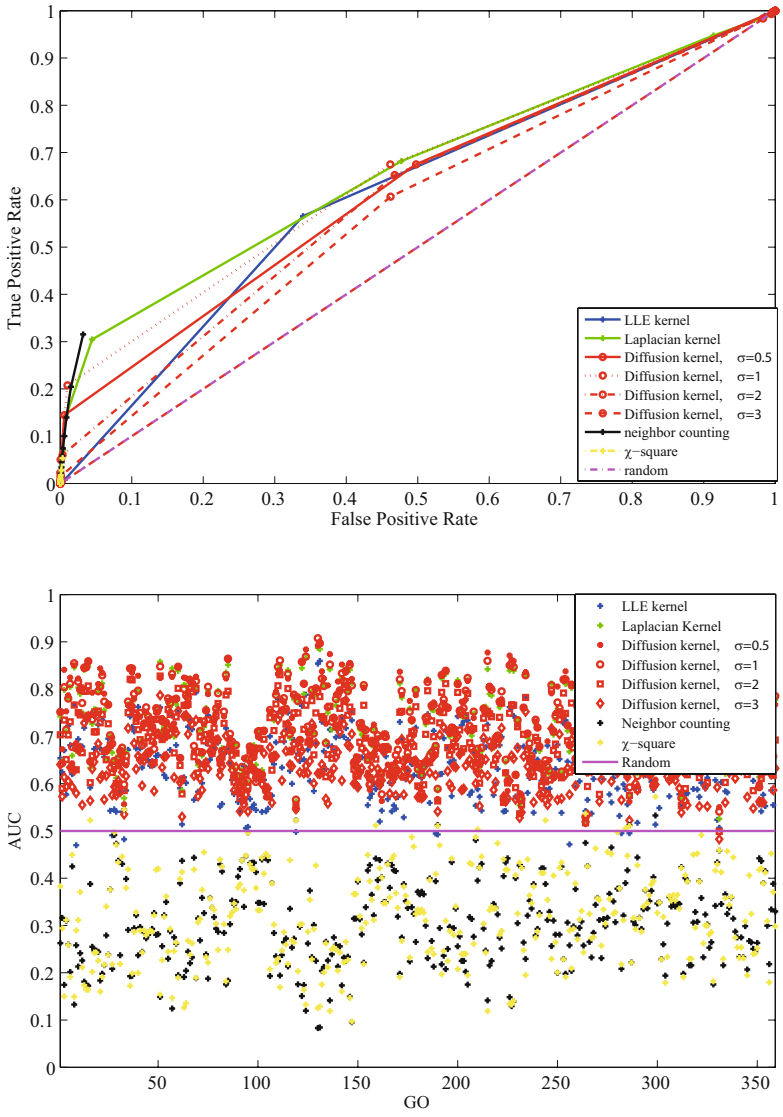

Figure 2 Prediction results using balanced methods. Left: ROC curves; Right: AUCs for different GO Terms

Table 1 Comparison of balanced and unbalanced methods

AUC	Balanced	Unbalanced
LLE	0.6353	0.7705
Laplacian	0.6770	0.8370
Diffusion,0.5	0.6591	0.8268
Diffusion,1	0.6778	0.8316
Diffusion,2	0.6562	0.7987
Diffusion,3	0.6124	0.7600
Neighbor counting	0.2449	
Chi-square	0.2578	

Table 2 A list of predictions for 645 uncharacterized proteins

GO Term	Uncharacterized Proteins
GO75: Cell cycle checkpoint	[YAL016W YBR136W YBR274W YCL024W YCL061C YDL028C YDL134C YDL188C YDR099W][YAL047C YBL051C]
GO82: G1/S transition of mitotic cell cycle	[YAL040C YBR160W YBR215W YCR008W YDL047W YDL132W YDL134C YDL188C YDR002W][YCR065W]
GO226: Microtubule cytoskeleton organization and biogenesis	[YAL020C YAL047C YBL034C YBL063W YDL028C YDR016C][YBL084C YCL029C YDL008W YDL064W YDR022C]
GO282: Ccellular bud site selection	[YBL007C YCL014W YCL024W YCR002C YCR009C YCR038C YCR047CYCR063W]
GO398: Nuclear mRNA splicing, via spliceosome	[YAL032C YBL026W YBR055C YBR065C YBR102C YBR119W YBR152W YBR188C YBR237W YDL030W YDL084W YDL087C YDL098C YDR088C]
GO902: Cell morphogenesis	[YAL041W YAR014C YBL007C YBL085W YBR040W YBR200W YCR002C YCR009C YCR088W YCR089W YDL223C YDL225W][YAR042W YCL014W YCL024W YCL027W YCR038C YCR047C YCR057C YCR063W YDR085C]
GO910: Cytokinesis	[YAR019C YBL061C YBR023C YBR038W YCL014W YCL024W YCR002C YCR057C YDL117W YDL225W][YBL007C YBL085W YBR143C YBR156C YCR009C YCR038C YCR047C YCR063W]
GO6281: DNA repair	[YAR007C YBR073W YBR087W YBR088C YBR098W YBR114W YBR195C YBR223C YBR272C YBR278W YCR014C YCR066W YCR092C YDL042C YDL059C YDL102W YDL105W YDL164C YDR004W YDR030C YDR076W YDR092W YDR097C][YBL088C YBR289W YDL020C YDL047W YDL084W][YDR078C]
GO6350: Transcription	[YAL001C YAL013W YAL021C YAL032C YAL043C YAR003W YBL008W YBL014C YBL021C YBL025W YBL052C YBL093C YBR020W YBR050C YBR060C YBR083W YBR112C YBR121C YBR123C YBR154C YBR182C YBR193C YBR198C YBR215W YBR240C YBR253W YBR275C YBR279W YCL055W YCL066W YCL067C YCR042C YCR052W YCR065W YCR081W YCR084C YCR093W YDL005C YDL020C YDL042C YDL080C YDL084W YDL106C YDL108W YDL140C YDL165W YDL170W YDR005C YDR009W YDR028C YDR043C YDR045C][YAL056W YAR014C YBR088C YBR095C YBR135W YBR160W YBR175W YBR195C YBR202W YBR278W YCL061C YDL017W YDL074C YDL076C YDL214C]
GO6352: Transcription initiation	[YAL001C YBR123C YBR198C YCR042C YDL108W]
GO6364: RNA processing	[YBL018C YBR167C YCL031C YCL054W YCL059C YCR018C YCR035C YCR057C YDL014W YDL031W YDL148C YDL166C YDL208W YDL213C YDR021W YDR087C][YAL025C YCR031C]
GO6405: RNA export from nucleus	[YBL079W YBR034C YCL011C YDL084W YDL088C YDL175C YDL207W YDR002W][YBR118W YCR073W-A]

GO Term	Uncharacterized Proteins
GO6412: Translation	[YAL003W YAL016W YAL035W YBL038W YBL080C YBR061C YBR079C YBR101C YBR118W YBR143C YCL037C YDL081C YDL134C YDL188C YDL219W YDL229W YDR091C][YAL005C YBL027W YBL072C YBL087C YBL092W YBR048W YBR084C-A YBR120C YBR121C YBR122C YBR146W YBR181C YBR282W YCR024C YCR031C YCR046C YCR077C YDL033C YDL061C YDL069C YDL075W YDL202W YDR012W YDR023W YDR025W]
GO6413: Translational initiation	[YAL035W YBR079C YDR091C][YCR077C]
GO6417: Regulation of translation	[YBR048W YCR077C YDL229W YDR025W]
GO6457: Protein folding	[YAL005C YBL075C YBR072W YBR169C YCL043C][YDL212W YDL229W][YAL053W]
GO6461: Protein complex assembly	[YBR037C YBR044C YBR081C YDL058W YDL239C YDR079W][YAR002W YBL023C YBL079W YBR060C YBR173C YBR202W YCL024W YDL088C YDR004W YDR076W]
GO6468: Protein amino acid phosphorylation	[YAL017W YAR019C YBL009W YBL016W YBL088C YBL105C YBR097W YBR136W YBR156C YBR160W YBR274W YCL024W YCR073C YDL017W YDL108W YDL159W]
GO6473: Protein amino acid acetylation	[YBL052C YBR081C YBR198C YCL010C YCR020C-A][YAL054C]
GO6486: Protein amino acid glycosylation	[YAL023C YBL020W YBL082C YBR015C YBR110W YBR205W YBR243C YDL055C YDL095W YDL232W]
GO6487: Protein amino acid N-linked glycosylation	[YBL020W YBL082C YBR110W YBR205W YBR243C YDL232W]
GO6508: Proteolysis	[YBL022C YBL084C YBR082C YBR170C YBR173C YBR201W YCL057W YDL008W YDL126C YDL132W YDL147W YDL190C YDR002W YDR069C][YBR105C YBR114W YBR290W YCL008C][YBR273C]
GO6512: Ubiquitin cycle	[BL084C YBR058C YBR082C YBR165W YDL008W YDL074C YDL132W YDL190C YDR059C YDR069C YDR092W][YBR114W YDL013W YDL165W]
GO6605: Protein targeting	[YAL005C YAL055W YAR002W YBL069W YBL075C YBL079W YBR017C YBR091C YBR097W YBR131W YBR164C YBR165W YBR171W YBR217W YBR283C YBR290W YCL008C YCL038C YDL065C YDL088C YDL113C YDL149W YDL217C YDR002W YDR086C][YBR105C]
GO6623: Protein targeting to vacuole	[YBR097W YBR105C YBR131W YBR164C YBR217W YBR290W YCL008C YCL038C YDL113C YDL149W]
GO6629: Lipid metabolic process	[YAL013W YBL039C YBR035C YBR036C YBR041W YBR042C YBR161W YBR265W YCL004W YCL026C-A YCR048W YDL015C YDL142C YDR062W YDR072C][YBL082C YBR004C YBR006W YBR109C YBR110W YCR044C YDL170W][]
GO6631: Fatty acid metabolic process	[YBR035C YBR041W YCL026C-A YDL015C][YBR006W YDL170W]
GO6644: Phospholipid metabolic process	[YBL039C YBR042C YCL004W YDR072C][YAL013W YBR004C YBR109C YCR044C]

GO Term	Uncharacterized Proteins
GO6796: Phosphate metabolic process	[YAL016W YAL017W YAR019C YAR071W YBL009W YBL016W YBL056W YBL088C YBL105C YBR093C YBR097W YBR136W YBR160W YBR274W YBR276C YCL024W YCR073C YDL006W YDL017W YDL047W YDL108W YDL134C YDL159W YDL188C YDL230W YDL236W YDR075W][Q0045 Q0085 Q0275 YBR039W YBR156C YDL067C YDL181W][YBL046W]
GO6810: Transport	[Q0085 YAL002W YAL005C YAL014C YAL026C YAL030W YAL042W YAL055W YAR002C-A YAR002W YBL007C YBL020W YBL037W YBL040C YBL042C YBL047C YBL050W YBL075C YBL079W YBL102W YBL106C YBR008C YBR017C YBR021W YBR024W YBR034C YBR037C YBR039W YBR041W YBR043C YBR059C YBR068C YBR069C YBR080C YBR091C YBR097W YBR102C YBR105C YBR106W YBR109C YBR131W YBR164C YBR165W YBR171W YBR180W YBR192W YBR207W YBR214W YBR217W YBR241C YBR254C YBR283C YBR288C YBR290W YBR291C YBR293W YBR296C YCL008C YCL011C YCL025C YCL034W YCL038C YCL069W YCR009C YCR011C YCR028C YCR053W YCR067C YCR075C YCR098C YDL054C YDL058W YDL065C YDL084W YDL088C YDL113C YDL126C YDL137W YDL145C YDL149W YDL161W YDL175C YDL181W YDL192W YDL195W YDL198C YDL207W YDL210W YDL212W YDL217C YDL226C YDL231C YDL247W YDR002W YDR027C YDR059C YDR069C YDR080W YDR086C YDR091C][YAL047C YAR042W YBL069W YBR020W YBR118W YCR073W-A YCR094W YDL100C YDL166C YDL193W][YAL053W YCR099C YDL099W YDR003W YDR084C]
GO6812: Cation transport	[Q0085 YBR024W YBR037C YBR039W YBR207W YBR290W][YDL181W]
GO6839: Mitochondrial transport	[YBR091C YBR291C YDL217C][YAL005C]
GO6887: Exocytosis	[YBL106C YBR102C][YAR042W]
GO6888: ER to Golgi vesicle-mediated transport	[YAL042W YAR002C-A YBL040C YBL050W YBR080C YBR254C YCR067C YDL058W YDL137W YDL145C YDL192W YDL195W YDL212W YDL226C][YDL099W]
GO6897: Endocytosis	[YAL030W YBL007C YBL047C YBR059C YBR109C YBR207W YBR214W YCL034W YCR009C YCR028C YCR053W YDL161W YDL231C YDR059C YDR069C][YAL026C YAR042W YCR094W]
GO6914: Autophagy	[YBR128C YBR131W YBR217W YCL038C YCR068W YDL113C YDL149W][YBR077C YBR109C YDR022C]

GO Term	Uncharacterized Proteins
GO6950: Response to stress	[YAL005C YAL028W YAR007C YBL022C YBL051C YBL061C YBL075C YBL088C YBR006W YBR072W YBR073W YBR082C YBR093C YBR114W YBR126C YBR136W YBR173C YBR216C YBR244W YBR274W YCL032W YCL033C YCL035C YCL051W YCR009C YCR021C YCR073C YCR092C YDL006W YDL013W YDL022W YDL059C YDL100C YDL106C YDL166C YDL190C YDL235C YDR001C YDR004W YDR059C YDR074W YDR098C][YBL047C YBL056W YBR037C YBR087W YBR088C YBR098W YBR182C YBR195C YBR223C YBR228W YBR272C YBR278W YBR289W YCL016C YCR014C YCR066W YDL020C YDL042C YDL047W YDL084W YDL102W YDL105W YDL164C YDR030C YDR076W YDR092W YDR097C YDR099W][YBR046C YDR078C]
GO7010: Cytoskeleton organization and biogenesis	[YAL016W YAL020C YAL047C YBL007C YBL034C YBL063W YBL105C YBR059C YBR109C YBR172C YBR234C YCL034W YCR088W YDL028C YDL029W YDL047W YDL134C YDL135C YDL161W YDL188C YDR016C][YAR014C YBL084C YBR211C YCL024W YCL029C YDL008W YDL064W YDL226C YDR022C]
GO7049: Cell cycle	[YAL009W YAL016W YAL021C YAL040C YAL041W YAL047C YAR007C YAR019C YBL009W YBL016W YBL051C YBL063W YBL084C YBL097W YBR017C YBR038W YBR073W YBR087W YBR109C YBR133C YBR135W YBR136W YBR160W YBR186W YBR198C YBR202W YBR215W YBR274W YBR276C YCL016C YCL024W YCL029C YCL055W YCL061C YCR008W YCR033W YCR042C YCR052W YCR065W YCR086W YCR092C YCR093W YCR094W YDL008W YDL020C YDL028C YDL047W YDL056W YDL064W YDL126C YDL132W YDL134C YDL154W YDL155W YDL165W YDL179W YDL188C YDL226C YDR002W YDR004W YDR016C YDR076W YDR099W][YBL023C YBL056W YBR088C YBR098W YDL164C YDR097C][YBR250W YDL139C]
GO7067: Mitosis	[YAL016W YAL041W YAL047C YAR019C YBL063W YBL084C YBL097W YCL016C YCL029C YDL008W YDL028C YDL134C YDL188C][YBL009W YBR088C]
GO7126: Meiosis	[YAL009W YBL009W YBR073W YBR136W YBR160W YBR186W YCL055W YCR033W YCR086W YDL154W YDR004W YDR076W][YAR007C YBR098W YCR092C YDR097C][YBR250W]
GO7127: Meiosis I	[YBR073W YBR136W YCR086W YDL154W YDR004W YDR076W][YAR007C YBR098W YCR092C YDR097C]
GO7131: Meiotic recombination	[YAR007C YBR073W YBR098W YBR136W YDL154W YDR004W YDR076W][YCR092C YDR097C]
GO7163: Establishment and/or maintenance of cell polarity	[YAL041W YBL007C YBL085W YBR200W YCL014W YCL024W YCR002C YCR009C YCR038C YCR047C YCR057C YCR063W YCR088W YDL225W][YAR042W]

GO Term	Uncharacterized Proteins
GO7165: Signal transduction	[YAL041W YAL056W YBL016W YBL056W YBL085W YBL105C YBR077C YBR140C YBR195C YBR203W YCL032W YCR038C YCR073C YDL035C YDL047W YDL135C YDL138W YDL159W YDL194W YDL235C YDR006C YDR085C YDR099W][YAL055W YBL047C YBL051C YBR097W YBR136W YBR274W YDL006W]
GO7264: Small GTPase mediated signal transduction	[YAL041W YAL056W YBL085W YBR140C YBR195C YCR038C YDL135C YDR099W]
GO8033: tRNA processing	[YAL043C YAR008W YBL018C YBR167C YCR073W-A YDL006W][YAL020C YBL024W YBR061C YCL017C YDL033C YDL036C YDL112W]
GO8104: Protein localization	[YAL005C YAL055W YAR002W YBL040C YBL069W YBL075C YBL079W YBR017C YBR091C YBR097W YBR131W YBR162W-A YBR164C YBR165W YBR171W YBR217W YBR283C YBR290W YCL008C YCL038C YDL065C YDL088C YDL113C YDL149W YDL217C YDR002W YDR086C][YAL026C YAR007C YBR105C YDL126C YDR080W][YDL139C]
GO8380: RNA splicing	[YAL032C YBL026W YBR055C YBR065C YBR102C YBR119W YBR152W YBR188C YBR237W YDL030W YDL084W YDL087C YDL098C YDR088C][YAR008W YCR063W YDL006W]
GO8654: Phospholipid biosynthetic process	[YBL039C YBR042C YCL004W][YBR004C YBR109C YCR044C]
GO9060: Aerobic respiration	[Q0045 Q0275 YBL045C YBL080C YBR185C YDR079W][YBR243C]
GO15031: Protein transport	[YAL026C YBL079W YBR091C YBR165W YBR171W YBR283C YDL088C YDL126C YDL217C YDR080W YDR086C][YAL005C YAL055W YAR002W YBL069W YBL075C YBR017C YBR097W YBR105C YBR131W YBR164C YBR217W YBR290W YCL008C YCL038C YDL065C YDL113C YDL149W YDR002W]
GO16311: Dephosphorylation	[YAL016W YBL056W YBR276C YDL006W YDL047W YDL134C YDL188C YDL230W YDL236W YDR075W][YBL046W]
GO16567: Protein ubiquitination	[YBL084C YBR082C YBR165W YDL008W YDL074C YDL132W YDR059C YDR092W][YBR114W YDL013W YDL165W]
GO16568: Chromatin modification	[YAL011W YAR003W YBL052C YBR060C YBR081C YBR175W YBR195C YBR198C YBR231C YBR275C YBR278W YBR289W YCL061C YCR033W YCR052W YDL002C YDL042C YDL074C YDL076C YDR073W][YAL013W YAL054C YBL088C YBR088C YBR095C YBR112C YBR136W YBR279W YCL010C YDL017W YDL084W YDL236W]
GO16573: Histone acetylation	[YBL052C YBR081C YBR198C YCL010C][YAL054C]

GO Term	Uncharacterized Proteins
GO19236: Response to pheromone	[YAL041W YBL016W YBR040W YBR200W YCL027W YCL032W YCL055W YCR002C YCR089W YCR093W YDL159W YDL165W YDL223C YDR085C][YAR031W YDL214C]
GO30001: Metal ion transport	[YBR024W YBR037C YBR207W YBR290W]
GO30003: Cellular cation homeostasis	[YBR036C YBR127C YCL017C YCR008W YCR044C YDL120W YDL198C][YDR098C]
GO30010: Establishment of cell polarity	[YAL041W YBL007C YBL085W YBR200W YCL014W YCL024W YCR002C YCR009C YCR038C YCR047C YCR057C YCR063W YCR088W YDL225W]
GO30029: Actin filament-based process	[YAL016W YBL007C YBL105C YBR059C YBR234C YCL034W YCR088W YDL029W YDL047W YDL134C YDL135C YDL161W YDL188C][YDL226C]
GO30036: Actin cytoskeleton organization and biogenesis	[YAL016W YBL007C YBL105C YBR059C YBR234C YCL034W YCR088W YDL029W YDL047W YDL134C YDL135C YDL161W YDL188C][YDL226C]
GO32197: Transposition, RNA-mediated	[YAR009C YBL100W-A YBR012W-B YCL019W YCL020W][YBR010W YBR279W YCR073C YDL074C YDR017C][YCL074W]
GO42255: Ribosome assembly	[YBR048W YCL031C YCR031C YDL014W YDL031W YDR025W YDR060W][YAL026C]

4 Concluding Remarks

In this paper, we propose network-based kernel methods to predict protein functions. Five-fold cross validation is then applied to compare different kernels. The results indicate that unbalanced methods are better than balanced methods, and Laplacian and diffusion kernels performs best among all the kernels. In our future research, we will consider different integration of the different data sources such as sequence, structure, expression data, and PPI network with different kernel methods.

5 Acknowledgments

The preliminary version of this paper has been presented in the OSB2009 conference and published in the corresponding conference proceedings[25]. The authors would like to thank the anonymous referees for their helpful comments and suggestions.

References

[1] W. Kim, C. Krumpelman, and E. Marcotte, Inferring mouse gene functions from genomic-scale data using a combined functional network/classification strategy, *Genome Biology*, 2008, **9**(Suppl 1): S5.

[2] E. Marcotte, M. Pellegrini, M. Thompson, et al., A combined algorithm for genome-wide prediction of protein function, *Nature*, 1999, **402**: 83–86.

[3] E. Marcotte, M. Pellegrini, N. H. Ricq, et al., Detecting protein function and protein-protein interactions from genome sequences, *Science*, 1999, **285**: 751–753.

[4] J. Watson, R. Laskowski, and J. Thornton, Predicting protein function from sequence and structural data, *Current Opinion in Structural Biology*, 2005, **15**: 275–284.

[5] X. Zhao, Y. Wang, L. Chen, and K. Aihara, Gene function prediction using labeled and unlabeled data, *BMC Bioinformatics*, 2008, **9**: 57.
[6] M. Brown, W. Grundy, D. Lin, et al., Knowledge-based analysis of microarray gene expression data by using support vector machines, *Proc. Natl. Acad. Sci.*, 2000, **97**: 262–267.
[7] M. Eisen, P. Spellman, P. Brown, and D. Bostein, Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci.*, 1998, **95**: 14863–14868.
[8] W. Ching, L. Li, N. Tsing, et al., A weighted local least squares imputation method for missing value estimation in microarray gene expression data, *International Journal of Data Mining and Bioinformatics*, 2010, **4**(3): 331–347.
[9] B. Schwikowski, P. Uetz, and S. Fields, A network of protein protein interactions in yeast, *Nat. Biotechnol*, 2000, **18**: 1257–1261.
[10] H. Hishigaki, K. Nakai, T. Ono, et al., Assessment of prediction accuracy of protein function from protein-protein interaction data, *Yeast*, 2001, **18**: 523–531.
[11] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani, Global protein function prediction from proteinCprotein interaction networks, *Nat. Biotechnol.*, 2003, **21**: 697–700.
[12] U. Karaoz, T. Murali, S. Letovsky, et al., Whole-genome annotation by using evidence integration in functional-linkage networks, *Proc. Natl. Acad. Sci.*, 2004, **101**: 2888–2893.
[13] E. Nabieva, K. Jim, A. Agarwal, et al., Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps, *Bioinformatics*, 2005, **21**(Suppl 1): 302–310.
[14] M. Deng, Z. Tu, F. Sun, and T. Chen, Mapping gene ontology to proteins based on protein-protein interaction data, *Bioinformatics*, 2003, **20**: 895–902.
[15] J. David and J. Robert, A simple generalisation of the area under the ROC curve for multiple class classification problems, *Machine Learning*, 2001, **45**: 171–186.
[16] H. Lee, Z. Tu, M. Sun, et al., Diffusion Kernel-based logistic regression models for protein function prediction, *OMICS, a Journal of Integrative Biology*, 2006, **1**(10): 40–55.
[17] R. Kondor and J. Lafferty, Diffusion kernels on graphs and other discrete input spaces, *Proc Int Conf Machine Learning*, 2002: 315–322.
[18] R. Lanckriet, M. Deng, M. Cristianini, et al., Kernel-based data fusion and its application to protein function prediction in yeast, *Proceedings of the Pacific Symposium on Biocomputing*, 2004, January 3–8, 300–311.
[19] J. Ham, D. Lee, S. Mika, and B. Scholkopf, A kernel view of the dimensionality reduction of manifolds, *Proceedings of the Twenty-First International Conference on Machine Learning*, (AAAI Press, Menlo Park, CA), 2004: 47–54.
[20] R. Sam and S. Lawrence, Nonlinear dimensionality reduction by locally linear embedding, *Science*, 2000, **290**: 2323–2326.
[21] http://www.geneontology.org/ontology/gene_ontology.obo.
[22] U. Guldener, M. Munsterkotter, M. Oesterheld, et al., MPact: The MIPS protein interaction resource on yeast, *Nucleic Acids Res.*, 2006, **34**: 436–441.
[23] A. Ruepp, et al., The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes, *Nucl. Acids. Res.*, 2004, **32**: 5539–5545.
[24] http : //hkumath.hku.hk/ ∼ wkc/Predicted_function_645_LLE.txt.
[25] W. Ching, L. Li, Y. Chan, and H. Mamitsika, A Study of network-based kernel methods on protein-protein interaction for protein functions prediction, *The Third International Symposium on Optimization and Systems Biology* (*OSB* 2009)*, Lecture Notes in Operations Research*, Series 11, 2009, **11**: 25–32.