Simple but effective methods for combining kernels in computational biology

Hiroaki Tanabe, Tu Bao Ho, Canh Hao Nguyen, Saori Kawasaki
School of Knowledge Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, JAPAN
Email: {htanabe, bao, canhhao, skawasa}@jaist.ac.jp

***Abstract*—Complex biological data generated from various experiments are stored in diverse data types in multiple datasets. By appropriately representing each biological dataset as a kernel matrix then combining them in solving problems, the kernel-based approach has become a spotlight in data integration and its application in bioinformatics and other fields as well. While linear combination of unweighed multiple kernels (UMK) is popular, there have been effort on multiple kernel learning (MKL) where optimal weights are learned by semi-definite programming or sequential minimal optimization (SMO-MKL). These methods provide high accuracy of biological prediction problems, but very complicated and hard to use, especially for non-experts in optimization. These methods are also usually of high computational cost and not suitable for large data sets.**

In this paper, we propose two simple but effective methods for determining weights for conic combination of multiple kernels. The former is to learn optimal weights formulated by our measure FSM for kernel matrix evaluation (feature space-based kernel matrix evaluation measure), denoted by FSM-MKL. The latter assigns a weight to each kernel that is proportional to the quality of the kernel, determining by direct cross validation, named proportionally weighted multiple kernels (PWMK). Experimental comparative evaluation of the four methods UMK, SMO-MKL, FSM-MKL and PWMK for the problem of protein-protein interactions shows that our proposed methods are simpler, more efficient but still effective. They achieved performances almost as high as that of MKL and higher than that of UMK.

I. Introduction

PPIs are elementary in biological activities and processes. As for the foreknowledge of the interactions, it is an important means to assist scientists to examine a point that genetic phenomenon leads to a physiologic expression phenomenon of special protein structures in a special cell process. Conventionally, there are several experimental approaches to detect PPIs, such as yeast two hybrid [1], mass spectrometry [2], protein chips [3], etc. In particular, yeast two hybrid methods are considered a comprehensive protein-protein network analysis in *Saccharomyces cerevisiae* [4], [5]. Computational methods for detecting PPIs, derived from a number of machine learning techniques for various types of available biological data, allow studying more widely and deeply about PPIs. For example, Matthews et al. [6] investigated the extent to which a protein interaction map generated in one species can be used to predict interactions in another species under the interacting orthologs. Deng et al. [7] proposed Bayesian network models for motifs and domains to predict PPIs.

So far, PPIs computational prediction has been based on genomic sequence analysis. These approaches apply techniques to obtain sequence information for biological patterns. Recently, it is desirable to handle multiple data sources and heterogeneous sets of biological data, e.g., protein sequences, gene expression, protein structures, gene ontology annotation, etc. to predict PPIs. Different data sources contain different and independent features in proteins. If those features can be appropriately combined as one data source, it is possible to enhance the PPIs prediction. However, those data are stored in multiple databases in various types, including gene expression expressed as numerical value vector and time-series, protein-protein interactions as graph, amino acid sequence as alphabet, etc. Therefore, it was necessary to work with a common feature format.

One solution to the problem is using inductive logic programming (ILP) which has been recognized as appropriate for exploiting multiple biological data types [8]. Nguyen and Ho [9] applied the ILP to predict PPIs using multiple data sources, including sequences, structures, and textual data. Additionally, kernel methods were suggested as an alternative solution to the problem [10], [11], [12]. Kernel methods were shown to enable the combination of these heterogeneous data into a common format. These methods represent various data by means of a kernel function, which generally defines similarity between pairs of objects, called kernel matrix. When there are n proteins, the kernel matrix is an $n \times n$ similarity matrix. Yamanishi et al. [13] used the kernel similarity matrix between protein networks, and inferred protein function networks with kernel canonical correlation analysis from multiple data sources.

So far, the best popular kernel method is SVM algorithm [12], which can map the original data objects into a high dimensional space by replacing the inner product in the input space. SVM is a binary supervised classification method having a solid theoretical foundation and performs the classification task more accurately than most other algorithms in many applications. SVM can be applied to classify various protein function categories, e.g. ribosomal proteins, membrane proteins, etc.

Pavlidis et al. [14] combined kernel matrices generated from microarray gene expression data and phylogenetic profiles, and classified gene functions in discriminant of SVM. Ben-Hur et al. [15] predicted PPIs using pairwise kernel and simple linear combination with sequence kernels. Vert et al. [16] proposed metric learning pairwise kernel for biological network inference. In those works, they used simple kernel linear combination by do not weighting kernels (or equal weights of 1 for all data sources). On the contrary, Lanckriet et al. [17] formualte a multiple kernel learning (MKL) problem which optimizes kernel weights by training a SVM classifier using with semi-definite programming. Bach et al. [18] applied sequential minimal optimization (SMO) techniques for MKL by solving non differentiable problem in the cost function. The MKL formulation based on a *semi-infinite linear problem (SILP)* was proposed by Sonnenberg et al. [19]. One common point among these methods is that they are all expensive for learning the weights in addition to learning the SVM classifier itself.

In this paper we propose two other simple but effective methods for combining multiple kernels. The former is to learn optimal weights formulated in terms of our measure FSM for kernel matrix evaluation (feature space-based kernel matrix evaluation measure) [20], denoted by FSM-MKL. This method is very efficient as it scales quadratically in terms of the number of training data (as opposed to $O(n^3)$ of the previously proposed methods). It is suitable for large data sets. The latter is to assign a weight to each kernel that is proportional to the quality of the kernel, determining by direct cross validation, called proportionally weighted multiple kernels (PWMK). This method is efficient as it does not learn the weights but assigns directly based on some SVM training results on the data.

Experimental comparative evaluation of four methods UMK, MKL, FSM-MKL and PWMK for the problem of protein-protein predictions has been carried out. It shown that simple combinations of multiple kernels is possible and effective, especially PWMK can nearly achieve high performance as MKL and higher than that of UMK.

II. Kernel methods

Kernel methods in general, and SVM in particular, are increasingly used to solve various problems in computational biology, and now considered as state-of-the-art in various domains, have just became a part of the mainstream in machine learning and empirical inference recently. Most kernel methods must satisfy some mathematical conditions. They can only process square matrices, which are symmetric positive definite. This means that if k is an $n \times n$ matrix of pairwise comparisons, it should satisfy $k_{ij} = k_{ji}$ for any $1 \leq i, j \leq n$ and $c^\top kc \geq 0$ for any $c \in R^n$.

A function $k : X \times X \rightarrow R$ is called a *positive definite kernel*, if it is symmetric, that is $k(x, x') = k(x', x)$ for any two objects $x, x' \in X$, and positive definite, that is,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) > 0$$

for any $n > 0$, any choice of n objects $x_1, ..., x_n \in X$, and any choice of real numbers $c_1, ..., c_n \in R$. Additionally, all positive semi-definite and symmetric matrix are kernel matrices. Conversely, we only focus on whether kernel matrix is semi-definite or not.

There are many kernel functions, so how to select a good kernel function in an application is also a critical issue. However, normally, there are several popular kernel functions, such as linear kernel, RBF kernel, polynomial kernel and hyperbolic tangent kernel. In addition, for sequence in nonvector, Leslie et al. [22] proposed spectrum kernel which compares any two sequences by considering the number of these k-mers that two sequences share. For these popular kernel functions, we therefore can choose them in applications by heuristic selection. For example, for gene expression expressed as vectors and time-series, the RBF kernel will be used; for protein localization expressed as binary data, the linear kernel will be used; for sequences the spectrum kernel will be used, etc.

A. Pairwise kernel

The standard kernel obtained from each kernel function is called genomic kernel, which means a similarity between single objects. While it is possible to use the single similarity matrix for protein function classification, the task of yeast interaction prediction also needs to consider similarity between pairs of proteins.

Ben-Hur et al. [15] addressed pairwise kernels representing a relationship between pairs of objects. The method maps an embedding of single objects onto an embedding of pairs of objects, otherwise, converting a kernel defined on single objects into pairwise kernel. The mapping of protein implies that two pairs of proteins are similar when each protein in a pair is similar to the corresponding protein in the other. For example, if protein x_1 is similar to protein x_2, and x_3 is similar to x_4, then the pairs (x_1, x_3) and (x_2, x_4) are similar. The pairwise kernel is formulated as follows

$$\begin{aligned} &K_p((x_1, x_2), (x_3, x_4)) \\ &= K_g(x_1, x_3)K_g(x_2, x_4) + K_g(x_1, x_4)K_g(x_2, x_3) \end{aligned}$$

where K_p is the pairwise kernel, K_g is the genomic kernel.

B. Support Vector Machine (SVM)

SVM is a computational algorithm that learns by examples to assign labels to objects. SVMs have also been successfully applied to an increasingly wide variety of biological applications, e.g. classifying objects as diverse as proteins and DNA sequences, microarray expression profiles. The method is defined over a vector space where the classification problem is to find the decision surface that best separates the data points of the two classes.

In the case of linearly separable data, the decision surface is a hyperplane that maximized the margin between the two classes. This hyperplane can be written as $wx + b = 0$, where x is a data point and vector w and constant b are learned from the training set. Let $y_i\{+1.-1\}$ be the classification label for input vector x. Finding the hyperplane can be translated into the following optimization problem.

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i((\mathbf{w}, \mathbf{x}_i) + b) \geq 1 - \xi_i, \ i = 1, ..., n \\ & \xi_i \geq 0, \ i = 1, ..., n \end{aligned}$$

This problem is called soft margin optimization, which is able to deal with errors in the data by allowing a few anomalous points to fall on the wrong side of the separating hyperplane. Generally, this problem has a trade-off between maximizing geometric margin and minimizing some measure of classification error on the training set. We discuss about parameter C to regulate the false positives and false negatives containing training label in SVM at the experiment section.

C. Normalized kernel

The normalization of the vectors of the input space can be considered as basic type of preprocessing. But the normalization in input space has loss of normalization or a scale problem for significant vector algorithm. For the problem, normalization in the feature space presents a solution. Assume $K(\mathbf{x}, \mathbf{y})$ is the kernel function representing a dot product in the feature space, then normalization in the feature space can be defined a new kernel function $K_{nor}(\mathbf{x}, \mathbf{y})$ as follows:

$$K_{nor}(\mathbf{x}, \mathbf{y}) = \frac{K(\mathbf{x}, \mathbf{y})}{\sqrt{K(\mathbf{x}, \mathbf{x}) K(\mathbf{y}, \mathbf{y})}} \in R$$

This normalization also means avoiding attributes in larger numeric ranges dominating those in smaller ranges. In this work, we also preprocess these normalization before combining several kernels generated from pairwise kernel function.

III. Kernel combination

The kernel methods make it possible to convert heterogeneous data into a common format by representing it as a similarity matrix in the feature space and combining kernel matrices. What to focus on here is how to combine several kernel matrices into a kernel. The kernel formalism also allows several kernels to be generated from various types of data to be combined.

A. Unweighed Multiple Kernel Method (UMK)

The basic combination method is the addition of operation. This operation is based on positive semi-definiteness. For example, given several kernel functions $\{K_1, ..., K_i\}$ and the embedding $\phi_1, ..., \phi_i$, we can represent that $K = K_1 + ... + K_k = \Sigma_{i=1}^{k} K_i$. This is expressed as linear combination or *unweighted multiple kernels method* (UMK), which satisfy semi-definite condition. In the past previous works, this combination method was employed in kernel integration, and shown to be useful and effective in SVM.

Because it is a simple linear sum of kernels, the weight is equally assigned for kernel matrix. We can regard this as a simple method for not considering the weight as well as the quality of each kernel. On the other hand, we can consider convex combinations of K kernels as

$$K = \sum_{i=1}^{k} \mu_i K_i$$

with $\mu_i \geq 0$ and $\Sigma_{i=1}^{k} \mu = 1$, where the sets of K_i are kernel matrices, e.g., linear, RBF, polynomial, spectrum kernels, etc., each computed on a dataset. This combination also preserves the kernel properties.

B. Multiple Kernel Learning (MKL)

In practice, it is desirable to build classifiers based on the combinations of the above multiple kernels. Lanckriet et al. [24] considered conic combinations of such kernel matrices in the SVM and showed that the optimization of the coefficients of the combination can be reduced to a convex optimization problem known as quadratically-constrained quadratic program (QCQP).

The key characteristic of MKL is to find a suitable set of weights for combinations, i.e., to conduct a grid search over all possible weightings and select the weights that minimize the error. The MKL problem can explain that dual formulation of 1-norm soft margin optimization problem in SVM: converting kernel of dual formulation into multiple kernel $\Sigma_k \mu_k K_k$, in short,

$$\min_{\Sigma_j \mu_j K_j \succeq 0} \left(\max_{\alpha, \alpha^\top y = 0} \quad \alpha^\top e - \frac{1}{2} \alpha^\top \left(\Sigma_j \mu_j K_j \right) \alpha \right)$$

This problem is reformulated into semi-definite programming. However, the MKL based on semi-definite programming (SDP-MKL) is only used for problem with small number of data points and kernels. For the problem, Bach et al. (2004) suggested an algorithm based on sequential minimization optimization (SMO-MKL).

In this framework, the problem of data representation is transferred to the choice of μ. The coefficients α, b and μ are obtained by solving the dual of the following optimization problem,

$$\begin{aligned} \min \quad & \frac{1}{2} \big(\sum_{j=1^m} \mu_j \|w_j\|_2 \big)^2 + C \sum_{i=1}^{n} \xi_i \\ & \mathbf{w} \in R^k = R^{k_1} \times ... \times R^{k_m}, \ \xi \in R_+^n, \ b \in R \\ \text{subject to} \quad & y_i(\Sigma_j w_j^\top x_{ji} + b) \geq 1 - \xi_i, \ \forall i \in 1, .., n \end{aligned}$$

In addition, this problem is transformed as dual problem,

$$\begin{aligned} \max \quad & \gamma \\ & \gamma \in R, \alpha \in R^n \\ \text{subject to} \quad & \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K_k(\mathbf{x_i}, \mathbf{x_j}) - \sum_{i=1}^{n} \leq \gamma \\ & \forall k = 1, ..., K \\ & 0 \leq \alpha \leq C, \ \alpha_i^\top y_i = 0 \end{aligned}$$

We can find the optimal weights μ_i by solving this dual problem.

IV. The proposed methods

A. FSM-based Multiple Kernel Learning (FSM-MKL)

The difficulty of learning weights for kernels mainly comes from the fact that one needs an optimization procedure to optimize the margin of the weighted combination of kernels. The optimization procedure is expensive as it involves a large number of variables (kernel evaluations and weights). One of the ways to avoid this difficulty is to use some surrogate measures. The measures should reflect how good a kernel is compared to others in terms of generalization error. It also should be calculated efficiently, in $O(n^2)$ time complexity.

We use FSM [20] to score different kernel combinations to select optimal weights. FSM measures the quality of a kernel matrix for a given classification task. FSM is defined to be *the ratio of the total within-class standard deviation in the direction between the class means to the distance between the class means*. Concretely,

$$FSM(K) = \frac{std_+ + std_-}{\|\phi_+ - \phi_-\|}$$

where std_+ and std_- are standard derivation of the positive and negative class, ϕ_+ and ϕ_- are class centers in the feature space, respectively. These quantities are efficiently computable from kernel matrices, which are readily available in a multiple kernel learning problem. This measure is selected over other siblings for the following reasons. First, it is optimally efficient in time. Second, it overcomes drawbacks of previously proposed measures as it is robust to data manipulation in feature space. It also implies some error bounds on training data. It has shown a high performance for kernel selection problems in various data sets.

The method using FSM is called FSM based Multiple Kernel Learning (FSM-MKL). It is formulated as follows:

$$\begin{aligned} \min \quad & FSM(\sum_{i=1}^{k} \mu_i K_i) \\ \text{subject to} \quad & \mu_i \in R, \ \mu_i \geq 0 \\ & \sum_{i=1}^{k} \mu_i = 1. \end{aligned}$$

The methods to solve this problem is very simple. We use a grid search over the simplex of μ_i in R^k. The fact that there is a small number of kernels to combine makes this method very efficient. This method scales quadratically (as opposed to cubic for other MKL methods) in the number of data, hence efficient enough for large kernels.

B. Proportionally Weighted Multiple Kernels (PWMK)

As mentioned above, MKL-based approach is difficult to understand and apply in several fields, in particular combining biological data. Our first interest is to develop a weighted method for kernel combination that is easy to understand and use but effective. Additionally, our second interest is to answer the question if combining all available datasets always gives higher performance than combining some of them. When combining weaker learners in ensemble learning, Zhou et al. [23] shown that when there is a number of available learners, ensembling many of them can be better than ensembling all of them. In the dual task of combining data by kernels, our assumption is also combining data from a subset of available datasets can be better than combining of the datasets.

In the following, we propose a method for data combination with kernels in considering together the two above issues.

The *quality* of a kernel (or of the corresponding dataset) in a given computation task is the performance estimated by a 10×10 cross validation when using that single kernel to perform the task. Particularly, for the prediction task the performance is usually measured by accuracy.

Firstly, the key idea of our weighted combination is to assign a weight to each kernel that is proportional to its quality. It ensures that the higher quality of the kernel the higher weight assigned to it in the combination, i.e. the higher contribution of the corresponding data to the combination. Secondly, each available kernel is individually compared its quality with a given threshold δ. And the kernel is included in the combination if and only if its quality is higher than the threshold. This allows us to combine only kernels that are considered good enough in term of quality. In fact, we distinguish threshold δ for two cases: one is choosing it equally to the minimum accuracy of kernels, the other is choosing it as $10 \times$ the tens digit of minimum accuracy in kernels. The former can directly exclude the low accuracy from combination by setting threshold into minimum accuracy. The later can decrease the effect of low accuracy kernel. The weight assigned to each kernel in the combination is determined as follows:

$$\mu_m = \frac{accuracy_{K_m} - \delta}{\Sigma_{i=1}^{k} accuracy_{K_i} - \delta \cdot k}$$

and our general formulas of *proportionally weighted multiple kernels* (PWMK) is

$$K = \sum_{m=1}^{k} \left(\frac{accuracy_{K_m} - \delta}{\sum_{i=1}^{k} accuracy_{K_i} - \delta \cdot k} \right) \times K_m$$

where k is a number of kernel matrices, K_m is m-th kernel matrix, $accuracy_{K_i}$ is the interactions prediction accuracy of kernel K_i when using SVM. The weights are standardized and we have $\Sigma_{m=1}^{k} \mu_m = 1$. In case that δ is the minimum accuracy, we have $k \geq 3$.

Here, we must consider that kernel matrices are positive semi-definite, i.e, eigenvalues of matrix are non-negative. In this formulation, the parameters μ_m correspond to eigenvalues and are constrained to be nonnegative. We confirm that the weights μ_m satisfy the condition $accuracy_{K_m} \geq \delta$, $\Sigma_{i=1}^{k} accuracy_{K_i} \geq \delta \cdot k$.

Fig. 1. (a) This figure is single kernel matrix generated from mRNA translation profile data. Color intensity of each dot represents similarity value: the darker the color, the higher the similarity value.

Fig. 2. (b) Pairwise kernel matrix are computed from pairwise kernel function in physical interactions. This matrix is generated from single kernel.

Thus, by assigning weight to each kernel and setting threshold, our method makes it possible to avoid the low performance kernel in kernel combination process and obtain higher performance. Moreover, it allows many people to use in several applications.

In experiments, we denote that PWMK(a) is the case when δ value is assigned as 10 × the tens digit of minimum accuracy, and PWMK(b) is the case when δ value is assigned as minimum accuracy.

This method is inherently different from multiple kernel learning ones. It assigns weight a priori, making the weight learning part trivial. Therefore, it scales linearly in the number of kernels (as opposed to higher order of all other learning methods).

V. Experimental comparative valuation

In this evaluation, we regard kernel matrices generated from heterogeneous datasets as that of decision function within SVM, and verify the performance obtained by solving the discriminant problem. Basically, the sum of kernels is very effective in SVM, especially, MKL is better than other combination methods [24]. But MKL is hard to understand and be adapted for applications in several fields. By contrast, our methods is an easy and applicable method.

The objective is to evaluate the performance of the four methods: UMK, MKL, FSM-MKL and PWMK by comparing their performance in interaction prediction accuracy in different situations, and to verify the effectiveness of PWMK.

Interacting proteins often function in the same biological process. By considering this characteristic, we assume that two proteins acting in the same biological process are more likely to interact than two proteins involved in different processes. In short, we assume that if one protein involves a function together with the other, they interact with each other. Such similarity of protein function can be expressed on kernel similarity matrix which represents relations among all proteins as inner products.

Hence, we chose several datasets that have the feature of protein function in experimental results under several environments.

A. Experimental design

The experiment is designed in the following steps.

1) Preprocess each type of data to deal with missing values and extract feature elements from the raw data.
2) Select common open reading frames (ORFs) in all datasets.
3) Construct the kernel matrices using kernel function from the datasets as shown Fig. 1.
4) Compute kernel matrices from pairwise kernel functions using the single kernel as shown Fig. 2.
5) Construct the normalized kernel from pairwise kernel matrices.
6) Combine kernels for each of UMK, MKL, FSM-MKL and PWMK.
7) Classify yeast interactions using SVM to evaluate the performance of the combination methods by 10 × 10-folds cross validation.

In addition, we performed two types of experiments:

- An experiment to evaluate the difference between data types and four methods of data combination. This experiment is used for investigating physical interactions between proteins.
- An experiment to evaluate the difference of training labels, such as genetic interactions.

B. Data for experiments

The experiments were carried out using two types of training data: physical interactions and genetic interactions in yeast. Additionally, as a training data, we selected four types of datasets on gene expression, protein localization, mRNA translation profile, and genetic sequence. In this section, we present a dataset and the interactions data in more details.

1) Biological interactions data: We selected physical interactions and genetic interactions in *Saccharomyses cerevisiae* as explained above, and 2000 interactions abstracted positive data records from Munich Information Center for Protein Sequences (MIPS) (2007/12/19). Direct physical interactions among yeast proteins are mapped by systematic two-hybrid and mass spectrometric characterization of protein complexes. Genetic interactions are also mapped by crossing mutations in different query genes into a set of viable gene yeast deletion mutants and scoring the double mutant progeny for fitness defects. These interactions data were used as training labels in SVM. We also randomly selected interactions 2000 negative data records from all interaction datasets in BioGRID (about 30000 data) except interaction datasets in MIPS.

2) Expression data: We employed the genomic expression dataset detected from DNA microarrays, which have been used to efficiently and quantitatively measure all genetic dynamic behaviors [25]. The data obtained from microarrays make it possible to know how expression affects a life cycle. It corresponds to 173 experiments for 6140 ORFs. We preprocessed missing values using mean in each column and selected RBF kernel which parameter σ set 10 by cross validation.

3) Protein localization: Protein cellular localizations are known to determine the environments in which proteins operate. As such, subcellular localization influences protein function by controlling access to and availability of all types of molecular interaction partners. The comprehensive knowledge of the location of proteins within cellular environments plays a significant role in characterizing the cellular function of hypothetical and newly discovered proteins.

Therefore, the analysis of protein subcellular localization is important to elucidate protein functions. So, we selected localization information obtained from budding yeast localization which has 23 intracellular locations [26] for each 6234 ORFs. We also did preprocessed for this because this also contains some missing values. In addition, the kernel function chosen is linear kernel.

4) mRNA translation profile: While gene expression involves many steps, including the transcription and translation, the analysis of mRNA levels leads to more detailed information, such as the changes in mRNA levels and transit times at each step in response to mutation and changes in growth conditions in these steps. Accordingly, we selected mRNA translation profile datasets, which represent 14×3 fractions that were analyzed by quantitative microarray analysis across sucrose gradient as vector in feature elements for 6180 ORFs [27]. The kernel function chosen is RBF kernel and parameter σ set 25 by cross validation.

5) Sequence: We focused on sequences that have more information about many biological functions. The genetic and amino acid sequences of proteins can be now available. This information is already known to prompt the resourceful bioinformaticians to find ways to predict interactions based on sequence information of the proteins. Additionally, it offers valuable clues to understanding the workings of more-advanced organisms. So, we selected genetic acid sequence and abstracted 5598 gene from Kyoto Encyclopedia of Genes and Genomes (KEGG). The kernel function chosen is spectrum kernel and the k-mers is 5.

C. Reliability of interactions

In Deng et al. [28], the reliability of interaction data in MIPS e evaluated with three measures: the distribution of gene expression correlation coefficients, the reliability based on gene expression correlation coefficients, and the accuracy of protein function predictions. In addition, they carried out different experiments for PPIs with added noises including both false positive and false negative cases which cause errors in PPI classification. In the dataset contains anomalous data, hard margin SVM cannot perform well. The reason is that hard-margin SVM does not allow their data to fall on the wrong side of the separating hyperplane. As the solution of the problem, SVM algorithm can be modified by using the soft margin version, which is what we use here using a hyperparameter C to trade off a margin and margin errors. The parameter C in our experiments is determined by 5 times cross validation. The best parameters are as follows: for physical interactions: $C = 12$ and for genetic interactions: $C = 9$.

D. Experimental setup

We did preprocessing of data, extraction of feature elements of k-mers from sequences collected from KEGG, construction of single kernels, pairwise kernels and normalized kernels, data combination using those methods. For MKL, we employed MKL toolbox (SMO-MKL) described by Bach et al. [18]. This tool is too memory consuming that we can only run with very small datasets. Hence, we randomly extracted 500 × 500 (SMO-MKL500) and 1000 × 1000 matrices (SMO-MKL1000) from each kernel matrix generated from combination methods. We used LIBSVM for SVM implementation.

VI. Results and Discussion

This section shows the performance of different kernel combination methods by means of accuracies of SVM classifiers trained on the combined kernels.

A. Physical interaction classification

The experimental results using physical interactions are represented in Table 1. Good performances are not seen with some types of data. Especially, protein localization kernel yields a low accuracy of 54.83%. Combining several kernels using SMO-MKL, FSM-MKL and PWMK provides a better

TABLE I
PHYSICAL INTERACTION

Kernel matrix	μ_{exp}	μ_{loc}	μ_{seq}	μ_{pro}	*Accuracy(%)*
Expression	1	-	-	-	63.99±0.24
Localization	-	1	-	-	54.83±0.19
Sequence	-	-	1	-	64.91±0.42
Profile	-	-	-	1	61.46±0.37
UMK	-	-	1	1	66.00±0.29
SMO/MKL(500)	-	-	0.737	0.263	66.43±0.25
SMO/MKL(1000)	-	-	0.654	0.346	66.85±0.24
FSM-MKL	-	-	0.731	0.269	**66.19±0.24**
PWMK(a)	-	-	0.770	0.230	**66.77±0.48**
UMK	1	-	-	1	65.31±0.40
SMO/MKL(500)	0.658	-	-	0.342	65.68±0.35
SMO/MKL(1000)	0.550	-	-	0.450	65.53±0.35
FSM-MKL	0.788	-	-	0.212	**65.44±0.29**
PWMK(a)	0.732	-	-	0.268	**65.90±0.27**
UMK	1	1	-	-	60.95±0.49
SMO/MKL(500)	0.930	0.070	-	-	64.74±0.31
SMO/MKL(1000)	0.930	0.064	-	-	64.43±0.41
FSM-MKL	0.753	0.247	-	-	**63.59±0.30**
PWMK(a)	0.743	0.257	-	-	**63.63±0.24**
UMK	1	-	1	-	66.95±0.36
SMO/MKL(500)	0.454	-	0.546	-	67.70±0.28
SMO/MKL(1000)	0.410	-	0.590	-	67.70±0.31
FSM-MKL	0.521	-	0.479	-	**66.89±0.22**
PWMK(a)	0.449	-	0.551	-	**67.28±0.38**
UMK	-	1	1	-	62.57±0.31
SMO/MKL(500)	-	0.052	0.948	-	65.85±0.37
SMO/MKL(1000)	-	0.051	0.949	-	65.87±0.30
FSM-MKL	-	0.241	0.759	-	**64.56±0.24**
PWMK(a)	-	0.250	0.750	-	**65.11±0.27**
UMK	-	1	-	1	59.28±0.32
SMO/MKL(500)	-	0.100	-	0.900	61.95±0.30
SMO/MKL(1000)	-	0.073	-	0.927	61.95±0.25
FSM-MKL	-	0.486	-	0.514	**59.46±0.33**
PWMK(a)	-	0.302	-	0.698	**60.85±0.32**
UMK	-	1	1	1	63.93±0.31
SMO/MKL(500)	-	0.049	0.677	0.274	66.80±0.28
SMO/MKL(1000)	-	0.049	0.620	0.331	66.77±0.23
FSM-MKL	-	0.168	0.457	0.375	**65.13±0.34**
PWMK(a)	-	0.155	0.478	0.367	**66.17±0.32**
PWMK(b)	-	0	0.603	0.397	**66.89±0.26**
UMK	1	-	1	1	66.63±0.41
SMO/MKL(500)	0.380	-	0.439	0.181	67.89±0.24
SMO/MKL(1000)	0.331	-	0.423	0.246	67.87±0.30
FSM-MKL	0.402	-	0.333	0.265	**66.52±0.32**
PWMK(a)	0.385	-	0.474	0.141	**67.65±0.32**
PWMK(b)	0.423	-	0.577	0	**67.75±0.39**
UMK	1	1	-	1	62.95±0.36
SMO/MKL(500)	0.621	0.063	-	0.316	65.15±0.28
SMO/MKL(1000)	0.537	0.047	-	0.416	65.06±0.15
FSM-MKL	0.491	0.162	-	0.347	**64.49±0.24**
PWMK(a)	0.462	0.160	-	0.378	**64.67±0.36**
PWMK(b)	0.580	0	-	0.420	**65.50±0.35**
UMK	1	1	1	-	64.39±0.29
SMO/MKL(500)	0.443	0.053	0.504	-	67.19±0.18
SMO/MKL(1000)	0.398	0.042	0.560	-	67.30±0.17
FSM-MKL	0.325	0.107	0.568	-	**66.91±0.24**
PWMK(a)	0.415	0.143	0.442	-	**66.81±0.35**
PWMK(b)	0.476	0	0.344	-	**66.97±0.28**
UMK	1	1	1	1	66.03±0.22
SMO/MKL(500)	0.373	0.051	0.407	0.169	67.58±0.20
SMO/MKL(1000)	0.351	0.042	0.411	0.196	67.46±0.27
FSM-MKL	0.241	0.083	0.424	0.252	**66.45±0.25**
PWMK(a)	0.310	0.107	0.330	0.253	**67.20±0.34**
PWMK(b)	0.354	0	0.390	0.256	**67.64±0.36**

TABLE II
GENETIC INTERACTION

Kernel matrix	μ_{exp}	μ_{loc}	μ_{seq}	μ_{pro}	*Accuracy(%)*
Expression	1	-	-	-	79.47±0.25
Localization	-	1	-	-	54.60±0.28
Sequence	-	-	1	-	80.27±0.22
Profile	-	-	-	1	74.93±0.33
UMK	1	1	1	1	81.97±0.17
SMO/MKL(500)	0.349	0.039	0.479	0.133	84.12±0.17
SMO/MKL(1000)	0.328	0.036	0.504	0.132	84.06±0.22
FSM-MKL	0.231	0.070	0.266	0.433	**82.99±0.27**
PWMK (a)	0.330	0.052	0.339	0.279	**83.65±0.14**
PWMK (b)	0.351	0	0.362	0.287	**83.80±0.12**

performance than individual ones. This shows the merit of MKL methods.

Here, it shows a weakness for UMK for the decline of performance in some kernel combinations. As shown in Table 1, localization kernel makes the kernels combination performance worse in UMK. In contrast, SMO-MKL, FSM-MKL and PWMK avoid the effect by assigning a small weight to the localization kernel, and provides higher performance than using each kernel. We regard the decline of performance as the effect of localization kernel, and from the accuracy of localization kernel, we consider that localization kernel is irrelevant. Lewis et al. [21] investigated performance of UMK and SDP-MKL in various situations, such as missing and noisy data, and suggested that for many applications UMK is sufficient but MKL is effective under condition including missing and noisy data. Indeed, in our experiments, once again confirm their hypothesis. When combining kernels, having more kernels does not necessarily increase performance.

Experimental results show that the sequence-based spectrum kernel is more informative than the other kernels. The spectrum kernel yields that the FSM-MKL, SMO-MKL and PWMK assign a larger weight to the spectrum kernel than to other kernels. Accordingly, excluding the spectrum kernel in kernel combination causes a decline in performance.

The results also show that specific combinations of (expression + profile + sequence) and (expression + sequence) performed better than all data combinations. These are combined kernels of high accuracy. It is shown that the performance depends on which kernels are selected. A conclusion can be drawn is that a high performance requires combinations of high quality kernels, excluding the low quality ones.

B. Genetic interaction classification

In the results shown in Table 2, the MKL also gains the highest accuracy and our proposed methods achieved almost the same performance for MKL. The accuracy achieved high values compared to physical interactions. Deng et al. [28] showed that genetic interactions in MIPS also provide high reliability. Here, we confirm that genetic data for genetic interactions is also more reliable to predict.

In these two experiments, finally, we obtained that SMO-MKL has the highest performance in the combination methods, closely followed by PWMK and FSM-MKL. These methods clearly give higher performances than UMK does. This also shows that our methods, despite the fact that they are simple and efficient, still give a comparable performance to other expensive methods. It is noteworthy that FSM-MKL and PWMK are very efficient in terms of computational time and memory usage.

VII. Conclusion

In this paper, we proposed FSM-MKL and PWMK methods for kernel combination and applied MKL, FSM-MKL and PWMK to SVM binary classification for predicting physical genetic PPIs. Our methods have advantages of simplicity and computational efficiency in comparison with previously developed methods. Our methods improve performance of the classification accuracy in comparison with the others using a single kernel at a time. Especially, PWMK provides a similar performance with MKL, followed by FSM-MKL. We also demonstrated a superiority of models constructed by MKL, FSM-MKL and PWMK in the PPI prediction problem. We also experimentally shown that combining all does not always give the best performance. Instead, only high quality kernels should be used for combination to achieve the highest result. The proposed approaches still need improvements at the point of finding better ways to avoid the effects of low quality kernels. Nevertheless, compared with competing methods to weight-based kernel combination, we see that our methods can achieve effective prediction performance for kernel combination problem in an economical budget of time and memory.

References

[1] S. Fields and O-K. Song. (1989), A novel genetic system to detect protein-protein interaction, *Nature*, Vol. 340, pp. 245-246.

[2] Y. Ho, A Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennet, and K. Boutilier. (2002), Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry, *Nature*, Vol. 415, pp. 180-183.

[3] H. Zhu, M. Bilgin, R. Bangham, D. Hall, A. Casamayor, P. Bertone, N. Lan, R. Jansen, S. Bidlingmaier, and T. Houfek. (2001), Global analysis of protein actives using proteome chips, *Science*, Vol. 293, pp. 2101-2108.

[4] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakai. (2001), Comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci*, USA, Vol. 98, pp. 4569-4574.

[5] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. (2000), A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, Vol. 403, pp. 623-627.

[6] L. R. Matthews, P. Vaglio, and J. Reboul. (2001), Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or 'interologs', *Genome Res*, Vol. 30(1), pp. 31-34.

[7] M. Deng, S. Mehta, F. Sun and T. Chen. (2002), Inferring domain-domain interactions from protein-protein interactions, *Genome Research*, Vol. 12(10), pp. 1540-1548.

[8] D. Page and M. Craven. (2003), Biological Applications of Multi-Relational Data Mining, Appears in *SIGKDD Explorations*, special issue on Multi-Relational Data Mining

[9] T. P. Nguyen and T. B. Ho. (2007), Combining domain fusions and domain-domain interactions to predict protein-protein interactions, *The 7th International Workshop on Data Mining in Bioinformatics* (*BIOKDD '07*), *ACM SIGKDD '07*, San Jose, CA, USA, pp. 27-34, ACM Press.

[10] N. Cristianini and J. Shawe-Taylor. (2000), An Introduction to Support Vector Machines. Cambridge University Press, Cambridge, MA. MIT press.

[11] B. Schölkoph and A. Smola. (2002), Learning with Kernels, MIT Press, Cambridge, MA.

[12] V. N. Vapnik. (1998), Statistical Learning Theory, John Wiley & Sons.

[13] Y. Yamanishi, J. P. Vert, and M. Kanehisa. (2004), Protein network inference from multiple genomic data: a supervised approach, *Bioinformatics*, Vol. 2, pp. 363-370.

[14] P. Pavlidis, J. Weston, J. Cai, and W. S. Noble. (2002), Learning gene functional classifications from multiple data types, *Journal of computational biology*, Vol. 9, pp. 401-411.

[15] A. Ben-Hur and W. S. Noble. (2005), Kernel methods for predicting protein-protein interactions, *Bioinformatics* (*Proceedings of the Intelligent Systems for Molecular Biology Conference*). Vol. 21, Suppl 1, pp. 38-46.

[16] J. P. Vert, J. Qiu, and W. S. Noble. (2007), A new pairwise kernel for biological network inference with support vector machines, *BMC Bioinformatics*, Vol. 8 Sppl 10, pp. 8-18.

[17] G. R. G. Lanckriet, N. Cristianini, L. EL. Ghaoui, P. Bartlett, and M. I. Jordan. (2004), Learning the kernel matrix with semi-definite programming, *Journal of Machine Learning Research*, Vol. 5, pp. 27-72.

[18] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. (2004), Multiple kernel learning, conic duality, and the SMO algorithm, In C E Brodley, editor, *Twenty-first international conference on Machine learning*. ACM.

[19] S. Sonnenbrug, G. Rötsch, C. Schäfer, and B. Schölkopf. (2006), Large scale multiple kernel learning, *Journal of Machine Learning Research*, Vol. 1, pp. 1-18.

[20] C. H. Nguyen and T. B. Ho. (2007) Kernel Matrix Evaluation, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 987-992.

[21] D. P. Lewis, T. Jebra, and W. S. Noble. (2006), Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure, *Bioinformatics* Vol. 22(22), pp. 2753-2760.

[22] C. Leslie, E. Eskin, and W. S. Noble. (2002), The spectrum kernel: A string kernel for SVM protein classification. *Pacific Symposium on Biocomputing*, Vol. 7, pp. 566-575.

[23] Z. H. Zhou, J. X. Wu, Y. Jiang, S. F. Chen. (2001), Genetic Algorithm based Selective Neural Network Ensemble, *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 797-802.

[24] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. (2004), Kernel-based data fusion and its application to protein function prediction in yeast. In R B Altman, A K Dunker, L Hunter, T A Jung, T E Klein(Eds.), *Proceedings of the Pacific Symposium on Biocomputing*, pp. 300-311. World Scientific.

[25] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, Vol. 11, pp. 4241-4257.

[26] W. K. Huh, F. V. Falve, L. C. Gerke, A. S. Carroll, R. W. Howson, J. S. Weissman, and E. K. O'Shea. (2003), Global analysis of protein localization in budding yeast, *Nature*, Vol. 425, pp. 686-691.

[27] Y. Arava, Y. Wang, J. D. Storey, C. L. Liu, P. O. Brown, D. Herschlag. (2003). Genome-wide analysis of mRNA translation profiles in Saccharomyces cerevisiae, *Proc Natl Acad Sci*, USA, Vol. 100, pp. 3889-3894.

[28] M. Deng, F. Sun, and T. Chen. (2003), Assessment of the reliability of protein-protein interactions and protein function prediction, *Pac Symp Biocomput*, pp. 140-151.