Huang *et al. EURASIP Journal on Bioinformatics and Systems Biology* (2016) 2016:8
DOI 10.1186/s13637-016-0040-2

EURASIP Journal on Bioinformatics and Systems Biology
a SpringerOpen Journal

RESEARCH Open Access

Inference of protein-protein interaction networks from multiple heterogeneous data

Lei Huang[1], Li Liao[1*] and Cathy H. Wu[1,2]

Abstract

Protein-protein interaction (PPI) prediction is a central task in achieving a better understanding of cellular and intracellular processes. Because high-throughput experimental methods are both expensive and time-consuming, and are also known of suffering from the problems of incompleteness and noise, many computational methods have been developed, with varied degrees of success. However, the inference of PPI network from multiple heterogeneous data sources remains a great challenge. In this work, we developed a novel method based on approximate Bayesian computation and modified differential evolution sampling (ABC-DEP) and regularized laplacian (RL) kernel. The method enables inference of PPI networks from topological properties and multiple heterogeneous features including gene expression and Pfam domain profiles, in forms of weighted kernels. The optimal weights are obtained by ABC-DEP, and the kernel fusion built based on optimal weights serves as input to RL to infer missing or new edges in the PPI network. Detailed comparisons with control methods have been made, and the results show that the accuracy of PPI prediction measured by AUC is increased by up to 23 %, as compared to a baseline without using optimal weights. The method can provide insights into the relations between PPIs and various feature kernels and demonstrates strong capability of predicting faraway interactions that cannot be well detected by traditional RL method.

Keywords: Protein interaction network, Network inference, Interaction prediction, Differential evolution

1 Introduction

Uncovering protein-protein interaction (PPI) is crucial to having a better understanding of intracellular signaling pathways, modeling of protein complex structures and elucidating various biochemical processes. Although several high-throughput experimental methods, such as yeast two-hybrid system and mass spectrometry method, have been used to determine a larger number of protein interactions, these methods are known to be prone to having high false-positive rates, besides of their high cost. Therefore, efficient and accurate computational methods for PPI prediction are urgently needed.

Generally, current computational methods for PPI prediction can be classified into two categories: A) pair-wise biological similarity based methods and B) network level-based methods. For category A, computational approaches have been developed to predict if any given pair of proteins interact with each other, based on various properties such as sequence homology, gene co-expression and phylogenetic profiles [1–5]. Moreover, some previous work also demonstrated that three-dimensional structural information, when available, can be used to predict PPIs with accuracy superior to predictions based on non-structural evidence [6, 7]. However, with no first principles to tell deterministically yet if two given proteins interact or not, the pair-wise biological similarity based on various features and attributes can run out its predictive power, as often the signals may be too weak or noisy. Therefore, recently, many researches have been focused on integrating heterogeneous pair-wise features, e.g., genomic features, semantic similarities, in seek of better prediction accuracy [8–11]. It is biologically meaningful if we can disentangle the relations among various pair-wise biological similarities and PPIs, but it is still in early stage for the incomplete and noisy pair-wise similarity kernels.

To circumvent the limitations with using pair-wise biological similarity, efforts have also been made to investigate PPI prediction in the context of networks, which may provide extra information to resolve ambiguities incurred

*Correspondence: liliao@udel.edu
[1] Department of Computer and Information Sciences, University of Delaware, 18 Amstel Avenue, 19716 Newark, DE, USA
Full list of author information is available at the end of the article

at pairwise level. A network can be constructed from reliable pair-wise PPIs, with nodes representing proteins and edges representing interactions. Topological features, such as the number of neighbors, can be collected for nodes and then are used to measure the similarity for any given node pair to make PPI prediction for the corresponding proteins [12–15]. Inspired by the PageRank algorithm [16], variants of random walk-based methods have been proposed to go beyond these node centric topological features to get the whole network involved; the probability of interaction between given two proteins is measured in terms of how likely a random walk in the network starting at one node will reach the other node [17–19]. These methods are suitable for PPI prediction in cases when the task is to find all interacting partners for a particular protein, by using it as the start node for random walks. The computational cost increases from $O(N)$ to $O(N^2)$ for all-against-all PPI prediction. To overcome the limitation of single start-node random walk, many kernels on network for link prediction and semi-supervised classification have been systemically studied [20], which can measure the random-walk distance for all node pairs at once. Compared with the random walk methods, kernel methods are obviously more efficient and applicable to various network types. But, both the variants of random walk and random walk-based kernels cannot differentiate faraway interacting candidates well. Besides, instead of computing proximity measures between nodes from the network structure directly, Kuchaiev et al. and Cannistraci et al. proposed geometric de-noise methods that embed PPI network into a low-dimensional geometric space, in which protein pairs that are closer to each other represent good candidate interactions [1, 21].

Furthermore, when the network is represented as an adjacent matrix, the prediction problem can be transformed into a spectral analysis and matrix completion problem. For example, Symeonidis et al. [22] did link prediction for biological and social networks based on multiway spectral clustering. Wang et al. [23] and Krishna et al. [24] predicted PPI interactions through matrix factorization-based methods. By and large, the prediction task will be reduced to convex a optimization problem, and the performance depends on the objective function, which should be carefully designed to ensure fast convergence and avoidance of being stuck in the local optima.

The two kinds of methods, pair-wise biological similarity-based methods and network level-based methods, can be mutually beneficial. For example, weights can be assigned to edges in the network using pair-wise biological similarity scores. In Backstrom et al. [19], a supervised learning task is proposed to learn a function that assigns weighted strengths to edges in the network such that a random walker is more likely to visit the nodes to which new links will be created in the future. The matrix factorization-based methods proposed by Wang et al. [23] and Krishna et al. [24] also included multi-modal biological sources to enhance the prediction performance. In these methods, however, only the pair-wise features for the existing edges in the network will be utilized, even though from a PPI prediction perspective, what is particularly useful is to incorporate pair-wise features for node pairs that are not currently linked by a direct edge but will if a new edge (PPI) is predicted. Therefore, it would be of great interest if we can infer PPI network directly from multi-modal biological features kernels that involve all node pairs. In Yamanishi et al. [25], a method is developed to infer protein networks from multiple types of genomic data based on a variant of kernel canonical correlation analysis (CCA). In that work, all genomic kernels are simply added together, with no weights to regulate these heterogeneous and potentially noisy data sources for their contribution towards PPI prediction. Also, it seems that the partial network needed for supervised learning based on kernel CCA needs to be sufficiently large, e.g., a leave-one-out cross validation is used, to attain good performance.

In this paper, we propose a new method based on ABC-DEP sampling method and regularized Laplacian (RL) kernel to infer PPI networks from multiple hetergeneous data. The method uses both topological features and various genomic kernels, which are weighted to form a kernel fusion. The weights are optimized using ABC-DEP sampling [26]. Unlike data fusion with genomic kernels for binary classification [27], the combined kernel in our case will be used instead to create a regularized Laplacian kernel [20, 28] for PPI prediction. We demonstrate how the method circumvents the issue of unbalanced data faced by many machine-learning methods in bioinformatics. One main advantage of our method is that only a small partial network is needed for training in order to make the inference at the whole network level. Moreover, the results show that our method works particularly well with detecting interactions between nodes that are far apart in the network, which has been a difficult task for other methods. Tested on Yeast PPI data and compared to two control methods, traditional regularized Laplacian kernel method and regularized Laplacian kernel based on equally weighted kernels, our method shows a significant improvement of over 20 % increase in performance measured by ROC score.

2 Methods and data

2.1 Problem definition

Formally, a PPI network can be represented as a graph $G = (V,E)$ with V nodes (proteins) and E edges (interactions). G is defined by the adjacency matrix A with $V \times V$ dimension:

$$A_{i,j} = \begin{cases} 1, if\ (i,j) \in E \\ 0, if\ (i,j) \notin E \end{cases} \quad (1)$$

where i and j are two nodes in the nodes set V, and (i,j) represents an edge between i and j, $(i,j) \in E$. The graph is called *connected* if there is a path of edges to connect any two nodes in the graph. For supervised learning, we divide the network into three parts: connected training network $G_{tn} = (V, E_{tn})$, validation set $G_{vn} = (V_{vn}, E_{vn})$, and testing set $G_{tt} = (V_{tt}, E_{tt})$. For G_{tn}, it consists of a minimum spanning tree, augmented with a small set of randomly selected edges. Because all edges are equally weighted, each time a minimum spanning tree is newly built, it will be different from a previous one. And G_{vn} and G_{tt} are two non-overlapping subsets of edges randomly chosen from the edges that are not in G_{tn}.

A kernel is a symmetric positive definite matrix K, whose elements are defined as a real-valued function $K(x, y)$ satisfying $K(x, y) = K(y, x)$ for any two proteins x and y in the data set. Intuitively, the kernel for a given dataset can be regarded as a measure of similarity between protein pairs with respect to the biological properties, from which kernel function takes its value. Treated as an adjacency matrix, a kernel can also be thought of as a complete network in which all the proteins are connected by weighted edges. Kernel fusion is a way to integrate multiple kernels from different data sources by a linear combination. For our task, this combination is made of the connected training network and various feature kernels $K_i, i = 1, 2, 3 \ldots n$ by optimized weights $W_i, i = 0, 1, 2, 3 \ldots n$, which formally is defined by Eq. (2)

$$K_{fusion} = W_0 G_{tn} + \sum_{i=1}^{n} W_i K_i \quad (2)$$

Note that the training network is incomplete, i.e., with many edges taken away and reserved as testing examples. Therefore, our inferring task is to predict or recover the interactions in the testing set G_{tt} based on the kernel fusion.

2.2 How to infer PPI network?

Once the kernel fusion is obtained, it will be used to make PPI inference, in the spirit of random walk. However, instead of directly doing random walk, we apply regularized Laplacian (RL) kernel to the kernel fusion, which allows for PPI inference at the whole network level. The regularized Laplacian kernel [28, 29] is also called the normalized random walk with restart kernel in Mantrach et al. [30] because of the underlying relations to the random walk with restart model [17, 31]. Formally, it is defined as Eq. (3)

$$RL = \sum_{k=0}^{\infty} \alpha^k (-L)^k = (I + \alpha * L)^{-1} \quad (3)$$

where $L = D - A$ is the Laplacian matrix made of the adjacency matrix A and the degree matrix D; and $0 < \alpha < \rho(L)^{-1}$ where $\rho(L)$ is the spectral radius of L. Here, we use kernel fusion in place of the adjacent matrix, so that various feature kernels in Eq. (2) are incorporated in influencing the random walk with restart on the weighted networks [19]. With the regularized Laplacian matrix, no random walk is actually needed to measure how "close" two nodes are and then use that closeness to infer if the two corresponding proteins interact. Rather, RL_K is the inferred matrix, and is interpreted as a probability matrix P in which $P_{i,j}$ indicates the probability of an interaction for protein i and j. Algorithm 1 shows the general steps to infer PPI network from a optimal kernel fusion. Figure 1 contains a toy example to show the process of inference, where both the kernel fusion and the regularized Laplacian are shown as heatmap. The lighter a cell is, the more likely the corresponding proteins. However, to ensure good inference, it is important to learn optimal weights for G_{tn} and various K_i to build kernel fusion K_{fusion}. Otherwise, given the multiple heterogeneous kernels from different data sources, the kernel fusion without optimized weights is likely to generate erroneous inference on PPI.

Algorithm 1 *PPI Inference*

Input: $RL \leftarrow$ *Regularized Laplacian prediction kernel*
$G_{tn} \leftarrow$ *training network*
$G_{vn} \leftarrow$ *validation set*
$G_{tt} \leftarrow$ *testing set*
$K \leftarrow$ *feature kernels*

Output: Inferred network

1: $W^{opt} \leftarrow ABC\text{-}DEP(G_{tn}, G_{vn}, RL, K)$
2: $OPT\text{-}K \leftarrow W_0^{opt} G_{tn} + \sum_{i=1}^{n} W_i^{opt} K_i$ // *OPT-K* is the optimal kernel fusion based on optimal weights
3: $RL_{OPT\text{-}K} \leftarrow RL(OPT\text{-}K)$ // Apply RL model to the kernel fusion
4: Rank $RL_{OPT\text{-}K}$ and infer G_{tt}

2.3 ABC-DEP sampling method for learning weights

In this work, we revise the ABC-DEP sampling method [26] to optimize the weights for kernels in Eq. (2). ABC-DEP sampling method, based on approximate Bayesian computation with differential evolution and propagation, shows strong capability of accurately estimating parameters for multiple models at one time. The parameter optimization task here is relatively easier than that in [26] as there is only one RL-based prediction model. Specifically, given the connected training network G_{tn} and N feature kernels in Eq. (2), the length of the particle in

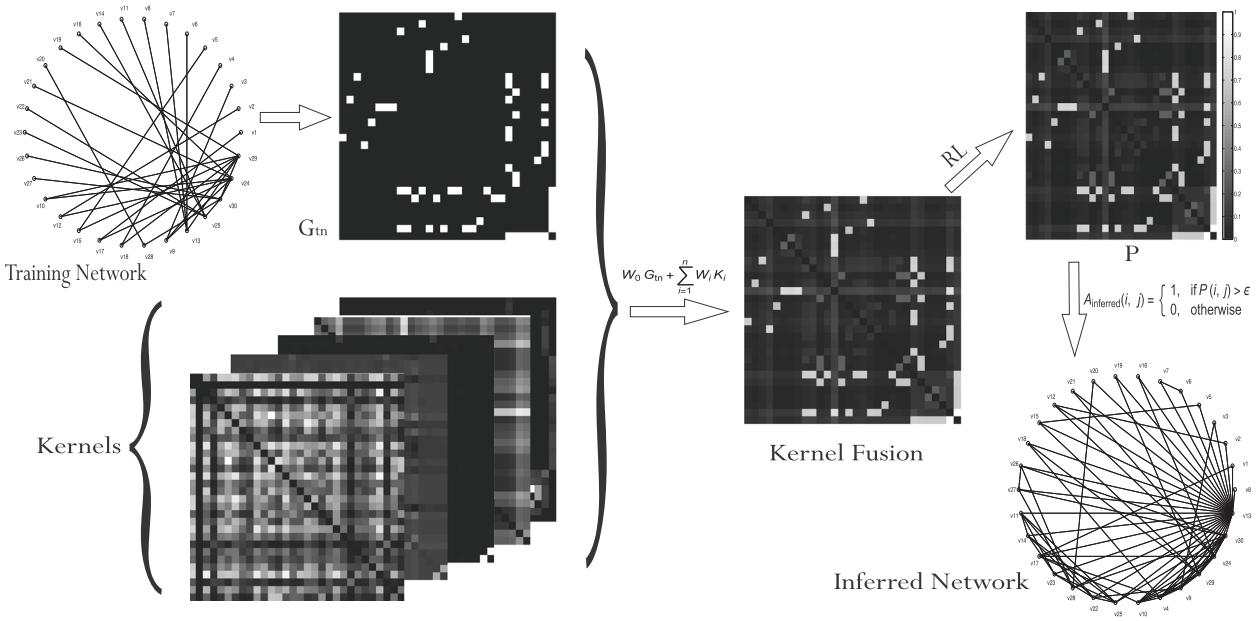

Fig. 1 An example to show the inference process. The example comprises of a small module in the DIP yeast PPI network, which consists of protein P25358 (ELO2, elongation of fatty acids protein 2) and its 1 ∼ 3 hops away neighbors. The kernel fusion and the regularized Laplacian are shown as heatmap. The *lighter* a cell is, the more likely the corresponding proteins interact

ABC-DEP would be $N + 1$, where particle can also be seen as a sample including the $N + 1$ weight values. As mentioned before, the PPI network is divided into three parts: the connected training network G_{tn}, validation set G_{vn} and testing set G_{tt}. To obtain the optimal particle(s), a population of particles with size N_p is intialized, and ABC-DEP sampling is run iteratively until a particle is found in the evolving population that maximizes the AUC of inferring training network G_{tn}, validation set G_{vn}. The validation set G_{vn} is used to avoid over-fitting as the algorithm converges. Algorithm 2 shows the detailed sampling process.

Algorithm 2 is the main structure in which a population of particles with equal importance is initialized and each particle consists of kernel weights randomly generated from a uniform prior. Given the particle population, Algorithm 3 samples through the parameter space for good particles and assigns them weights according to the predicting quality of their corresponding kernel fusion K_{fusion}. Note that, different from the ABC-DEP sampling method in [26] where the logarithm of the Boltzmann distribution is adopted, here, we accept or reject a new candidate particle based on Boltzmann distribution with simulated annealing method [32]. Through the evolution process, bad particles will be filtered out and good particles will be kept for the next generation. We repeat this process until the algorithm converges. The optimal particle is used to build kernel fusion K_{fusion} for PPI prediction.

Algorithm 2 *ABC-DEP*

Input: G_{tn}, G_{vn}, RL, K
$M \leftarrow$ *iteration times*
$N_p \leftarrow$ *particles*
Output: W^{opt}

while $t \leq M$ **do**
  if $t = 1$ **then**
    Initialize N_p particles, each particle contains weights $W_i, 0 < W_i < 1, i = 0, 2, 3 \ldots n$ for training network and n-1 feature kernels
    $P_t, I_t \leftarrow \{P^i, I^i\}_{i=1}^{N_p}$ // P^i is a particle, I^i is the weight or importance of P^i. P_t, I_t represents the t^{th} generation of particles and weights.
  else
    $\{P_t, I_t\}_{i=1}^{N_p} \leftarrow Sampling((P_{t-1}, I_{t-1}))$
  end if
  $(P_{t+1}, I_{t+1}) \leftarrow DEP(P_t, I_t, G_{tn}, G_{vn}, RL, K)$
  $t \leftarrow t + 1$
end while
$Normalize(P, I)$
$W^{opt} \leftarrow P^i$ *if* $I^i = max(I)$

2.4 Data and kernels

We use yeast PPI networks downloaded from DIP database (Release 20150101) [33] to test our algorithm. Notably, some interactions without Uniprotkb ID have been filtered out in order to do name mapping and make

Algorithm 3 *DEP*

Input: $G_{tn}, G_{vn}, RL, K, N_p$

Output: P, I

for $i = 1$ to N_p **do**
  Randomly select P^f, P^j, P^k where $i \neq j \neq k \neq f$
  // P^i is the target particle, P^f, P^j and P^k are three randomly selected particles. $P^i.\theta, P^j.\theta, P^k.\theta$ and $P^f.\theta$ represent particles' parameter vectors that consist of weights for feature kernels.
  if $P^i.\theta = P^j.\theta = P^k.\theta = P^f.\theta$ **then**
    $Z^i \leftarrow Propagation(P^i)$
  else
    $Z^i \leftarrow DifferentialEvolution(P^i, P^j, P^k, P^f)$
  end if
end for
for $i = 1$ to N_p **do**
  $r'_{G_{tn}}, r'_{G_{vn}} = Inference(RL, Z^i, K, G_{tn}, G_{vn})$
  $r' = r'_{G_{tn}} + r'_{G_{vn}}$ // In the *Inference* function, particle Z^i is used to weight kernels in K to get kernel fusion K_{fusion}. $r'_{G_{tn}}, r'_{G_{vn}}$ represent results (AUCs) of recovering G_{tn} and G_{vn} based on K_{fusion} respectively
  $r_{G_{tn}}, r_{G_{vn}} \leftarrow Inference(RL, P^i, K, G_{tn}, G_{vn})$.
  $r \leftarrow r_{G_{tn}} + r_{G_{vn}}$
  if $rand(0,1) < e^{\frac{r'-r}{T(t)}}$ **then**
    $P^i \leftarrow Z^i, I^i \leftarrow I^i * \frac{\beta}{\alpha - r'}$
  else
    $P^i \leftarrow P^i, I^i \leftarrow I^i * \frac{\beta}{\alpha - r}$ // β and α are two positive parameters that can be used to update particles' importances and adjust converging speed.
  end if
end for
$Normalize(P, I)$

use of genomic similarity kernels [27]. As a result, the PPI network contains 5093 proteins and 22,423 interactions, from which the largest connected component is used to serve as golden standard network. It consists of 5030 proteins and 22,394 interactions. Only tens of proteins and interactions are not included in the largest connected component, which makes the golden standard data almost as complete as the original network. As mentioned before, the golden standard PPI network is divided into three parts that are connected training network G_{tn}, validation set G_{vn} and testing set G_{tt}, where training network G_{tn} is included in the kernel fusion, validation set G_{vn} is used to find optimal weights for feature kernels and testing set G_{tt} is used to evaluate the inference capability of our method.

Six feature kernels are obtained from http://noble.gs.washington.edu/proj/sdp-svm/ for this study and the following list is about the detailed information of these kernels.

G_{tn}: G_{tn} is the connected training network that provides connectivity information. It can also be thought of as a base network to do the inference.
$K_{Jaccard}$ [34]: This kernel measure the similarity of protein pairs i, j in term of $\frac{neigbors(i) \cap neighbors(j)}{neighbors(i) \cup neighbors(j)}$.
K_{SN}: It measures the total number of neighbors of protein i and j, $K_{SN} = neighbors(i) + neighbors(j)$.
K_B [27]: It is a sequence-based kernel matrix that is generated using the BLAST [35].
K_E [27]: This is a gene co-expression kernel matrix constructed entirely from microarray gene expression measurements.
K_{Pfam} [27]: This is a generalization of the previous pairwise comparison-based matrices in which the pairwise comparison scores are replaced by expectation values derived from hidden Markov models (HMMs) in the Pfam database [36].

These kernels are positive semi-definite. Please refer to [27] for detailed analysis (or proof). Moreover, Eq. (2) is guaranteed to be positive semi-definite, because basic algebraic operations such as addition, multiplication, and exponentiation preserve the key property of positive semi-definiteness [37]. Finally, all these kernels are normalized to the scale of $(0, 1)$ in order to avoid bias.

3 Results and discussion

3.1 Inferring PPI network

To show how well our method can infer PPI network from the kernel fusion, we make the task challenging by dividing the golden standard yeast PPI network into the following three parts: the connected training network G_{tn} has 5030 nodes and 5394 edges, the validation set G_{vn} has 1000 edges, and the testing set G_{tt} has 16,000 edges. This means that we need to infer and recover a large number of testing edges based on the kernel fusion and a small validation set. Firstly, we check the converging process of finding the optimal weights that used to combine feature kernels, which is shown by the Fig. 2. It clearly shows that when the AUC of predicting the training network G_{tn} reaches to 1 quickly, but the AUC of predicting the validation set G_{vn} is still in an upward trend. So G_{tn} alone cannot guarantee the optimality of the weights when the algorithm converges, which is the reason the validation set G_{vn} is used. After several iterations, the ABC-DEP algorithm is converged when both AUCs have become steady.

With the optimal weights obtained from ABC-DEP sampling, we build the kernel fusion K_{fusion} by Eq. (2). PPI network inference is made with RL kernel Eq. (3). The performance of inference is evaluated by how well the testing set G_{tt} is recovered. Specifically, all node pairs are ranked in decreasing order by their edge weights in the RL matrix, and edges in the testing set G_{tt} are then

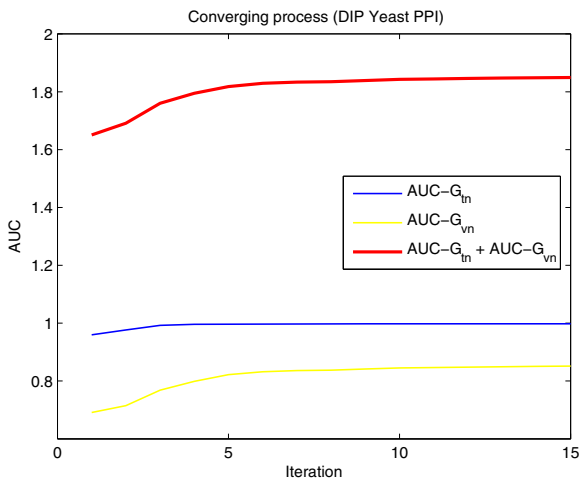

Fig. 2 The converging process of ABC-DEP sampling used to obtain optimal weights

labeled as positive and node pairs with no edges in G are labeled as negative. A ROC curve is plotted for true positive vs. false positives, by running down the ranked list of node pairs. Figure 3 shows the ROC curves and AUCs for three PPI network inferences: $RL_{OPT\text{-}K}$, $RL_{G_{tn}}$, and $RL_{EW\text{-}K}$, where $RL_{OPT\text{-}K}$ indicates the RL-based PPI inference is from kernel fusion that built by optimal weights, $RL_{G_{tn}}$ indicates RL-based PPI inference is solely from the training network G_{tn}, and $RL_{EW\text{-}K}$ represents RL-based PPI inference is from kernel fusion built by equal weights, e.g., $W_i = 1, i = 0, 1 \ldots n$. Additionally, $G_{set} \sim n$ indicates that there is n number of edges in the set G_{set}, e.g., $G_{tn} \sim 5394$ means the connected training network G_{tn} contains 5394 edges. As shown by Fig. 3, the PPI reference

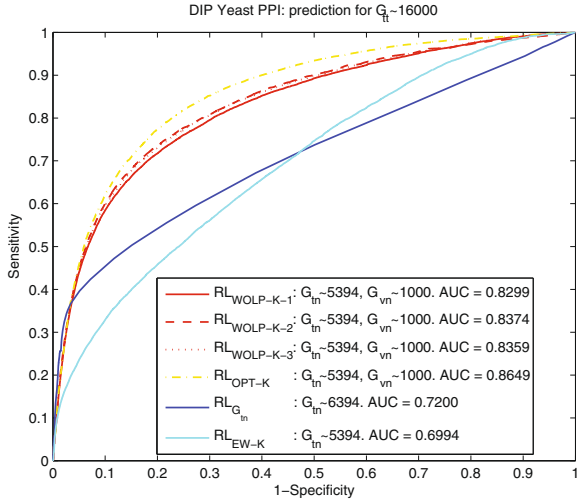

Fig. 3 ROC curves of predicting $G_{tt} \sim 16,000$ by $RL_{G_{tn}\sim 5394}$, $RL_{OPT\text{-}K}$, $RL_{EW\text{-}K}$, and $RL_{WOLP\text{-}K\text{-}i}$

$RL_{OPT\text{-}K}$ based on our method significantly outperforms the other two control methods, with a 20 % increase over $RL_{G_{tn}}$ and a 23.6 % over $RL_{EW\text{-}K}$ in terms of AUC. It is noted that the AUC of PPI inference $RL_{EW\text{-}K}$ based on the equally weighted built kernel fusion is even worse than that of $RL_{G_{tn}}$ based on a really small training network. It means there should be a lot of noises if we just naively combine different feature kernels to do PPI prediction. Our method provides an effective way to make good uses of various features for improving PPI prediction performance.

In Fig. 3, we also compared with another method, WOLP, which uses linear programming to optimize the weights W_i for the various kernel features [38]. It can be seen that WOLP, with AUC at about 0.83, also performs signigicantly better than the baseline, indicating that the method is effective in weighting various features to improve PPI inference. Note that although reference [38] has "random walk" in its title, the method WOLP does not do sampling; instead, the weights for kernel features are optimized by linear programming, constrained with the transition matrix from the training network for any would-be random walk over the PPI network when kernel features are incorporated. As such, WOLP is more computationally efficient but with a trade-off of slightly worse performance as compared to ABC-DEP, which has the best AUC, 0.86, in this study.

3.2 Effects of the training data

Usually, given a golden standard data, we need to retrain the prediction model for different divisions of training sets and testing sets. However, if optimal weights have been found for building kernel fusion, our PPI network inference method enable us to train the model once, and do prediction or inference for different testing sets. To demonstrate that, we keep the two PPI inferences $RL_{OPT\text{-}K}$ and $RL_{EW\text{-}K}$ obtained before (in last section) unchanged and evaluate the prediction ability for different testing sets. We also examine how performance is affected by sizes of various sets. Specifically, while the size of training network G_{tn} for $RL_{G_{tn}}$ increases, sizes of $RL_{OPT\text{-}K}$ and $RL_{EW\text{-}K}$ are kept unchanged. Therefore, we design several experiments by dividing the golden standard network into G^i_{tn} and G^i_{tt}, $i = 1, \ldots, n$, and building PPI inference $RL_{G^i_{tn}}$ to predict G^i_{tt} for every time. To make comparison, we also use $RL_{OPT\text{-}K}$ and $RL_{EW\text{-}K}$ to predict G^i_{tt}. Figure 4 shows the ROC curves of predicting $G_{tt} \sim 15000$ by $RL_{G_{tn}\sim 7394}$, $RL_{OPT\text{-}K}$ and $RL_{EW\text{-}K}$. Figures 5, 6 and 7 show similar results but just for different G_{tn} and G_{tt} sets. As shown by the Figs. 4, 5, 6, and 7, $RL_{OPT\text{-}K}$ trained on only 5394 golden standard edges still performs better than the control methods that employ significantly more golden standard edges.

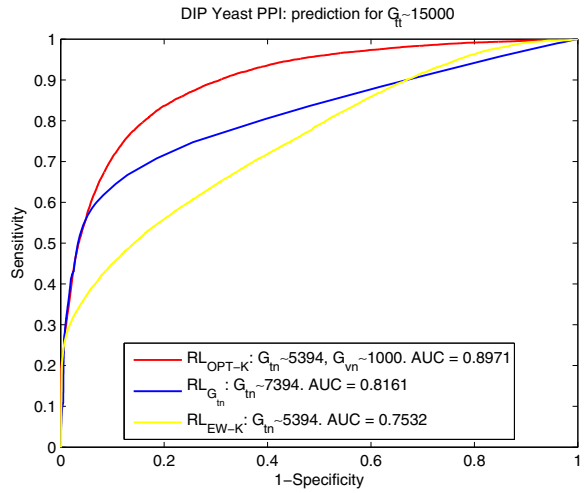

Fig. 4 ROC curves of predicting $G_{tt} \sim 15,000$ by $RL_{G_{tn}\sim 7394}$, $RL_{OPT\text{-}K}$, and $RL_{EW\text{-}K}$

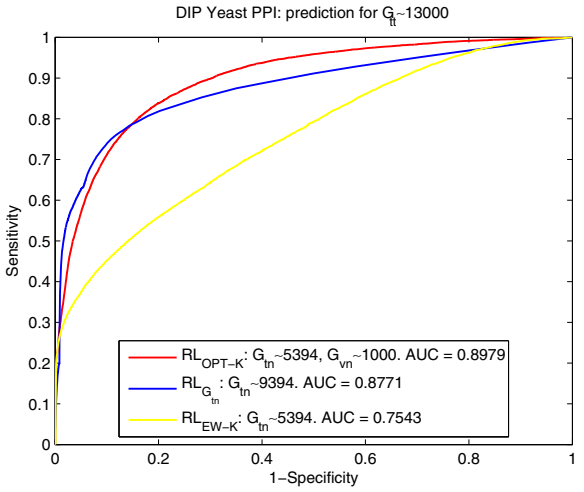

Fig. 6 ROC curves of predicting $G_{tt} \sim 13,000$ by $RL_{G_{tn}\sim 9394}$, $RL_{OPT\text{-}K}$, and $RL_{EW\text{-}K}$

3.3 Detection of interacting pairs far apart in the network

It is known that the basic idea of using random walk or random walk based kernels [17–20] for PPI prediction is that good interacting candidates usually are not faraway from the start node, e.g., only 2,3 edges away in the network. Consequently, for some existing network-level link prediction methods, testing nodes have been chosen to be within a certain distance range, which largely contributes to their good performance reported. In reality, however, a method that is capable and good at detecting interacting pairs far apart in the network can be even more useful, such as in uncovering cross talk between pathways that are not nearby in the PPI network.

To investigate how our proposed method performs at detecting faraway interactions, we still use $RL_{G_{tn}\sim 6394}$, $RL_{OPT\text{-}K}$, and $RL_{EW\text{-}K}$ for inferring PPIs, but we select node pairs (i,j) that satisfy $dist(i,j) > 3$ *given* $G_{tn} \sim 6394$ from G_{tt} as new testing set and name it $G_{tt}^{(dist(i,j)>3)}$. Figure 8 shows that $RL_{OPT\text{-}K}$ has not only a significant margin over the control methods in detecting long-distance PPIs but also maintains a high ROC score of 0.8438 comparable to that of all PPIs. In contrast, $RL_{G_{tn}\sim 6394}$ performs poorly and worse than $RL_{EW\text{-}K}$, which means the traditional RL kernel based on adjacent training network alone cannot detect faraway interactions well.

3.4 Analysis of weights and efficiency

As the method incorporates multiple heterogeneous data, it can be insightful to inspect the final optimal weights. In

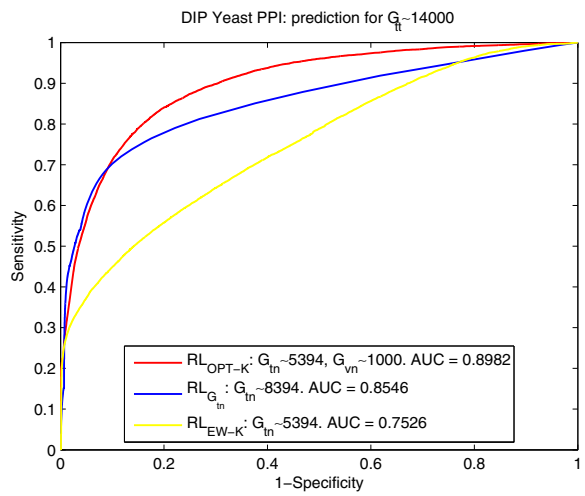

Fig. 5 ROC curves of predicting $G_{tt} \sim 14,000$ by $RL_{G_{tn}\sim 8394}$, $RL_{OPT\text{-}K}$, and $RL_{EW\text{-}K}$

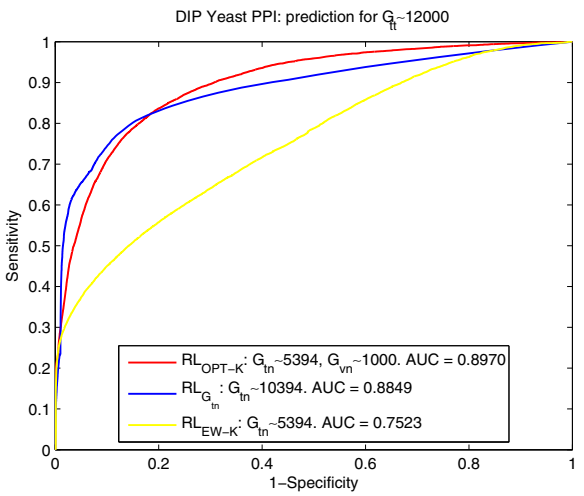

Fig. 7 ROC curves of predicting $G_{tt} \sim 12,000$ by $RL_{G_{tn}\sim 10394}$, $RL_{OPT\text{-}K}$, and $RL_{EW\text{-}K}$

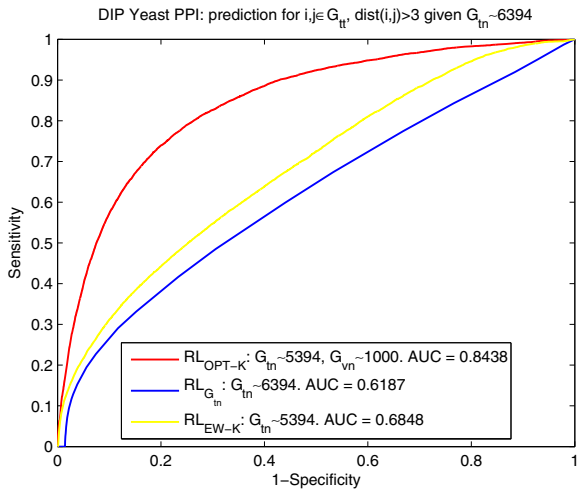

Fig. 8 ROC curves of predicting $G_{tt}^{(dist(i,j)>3)}$ by $RL_{G_{tn}\sim 6394}$, $RL_{OPT\text{-}K}$, and $RL_{EW\text{-}K}$

our case, the optimal weights are 0.8608, 0.1769, 0.9334, 0, 0.0311, 0.9837, respectively for feature kernels G_{tn}, $K_{Jaccard}$, K_{SN}, K_B, K_E, and K_{Pfam}. These weights indicate that K_{SN} and K_{Pfam} are the predominant contributors to PPI prediction. This observation is consistent with the intuition that proteins interact via interfaces made of conserved domains [39], and PPI interactions can be classified based on their domain families and domains from the same family tend to interact [40–42]. Although the true strength of our method lies in integrating multiple heterogeneous data for PPI network inference, the optimal weights can serve as a guidance to select most relevant features when time and resources are limited.

Lastly, despite of the common concern of time efficiency with methods based on evolutionary computing, the issue is mitigated in our case. In our experiments, only a small number of particles, 150 to be exact, is needed for the initial population for ABC-DEP sampling. Also, as shown in the Fig. 2, our ABC-DEP algorithm is quickly converged, within 10 iterations. Moreover, since the PPI inference from $RL_{OPT\text{-}K}$ is shown to be less sensitive to the size of training data, only 5394 gold standard edges, less than 25 % of the total number, are used. And, we do not need to retrain the model for different testing data, which is another time-saving property of our method.

4 Conclusions

In this work, we developed a novel supervised method that enables inference of PPI networks from topological and genomic feature kernels in an optimized integrative way. Tested on DIP yeast PPI network, the results show that our method exhibits competitive advantages over control methods in several ways. First, the proposed method achieved superior performance in PPI prediction, as measured by ROC score, over 20 % higher than the baseline, and this margin is maintained even when the control methods use a significantly larger training set. Second, we also demonstrated that by integrating topological and genomic features into regularized Laplacian kernel, the method avoids the short-range problem encountered by random-walk based methods—namely the inference becomes less reliable for nodes that are far from the start node of the random walk, and show obvious improvements on predicting faraway interactions. Lastly, our method can also provide insights into the relations between PPIs and various similarity features of protein pairs, thereby helping us make good use of these features. As more features with respect to proteins are collected from various -omics studies, they can be used to characterize protein pairs in terms of feature kernels from different perspectives. Thus, we believe that our method provides a useful framework in fusing various feature kernels from heterogeneous data to improve PPI prediction.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
LH designed the algorithm and experiments, and performed all the calculations and analyses. LL and CHW aided in interpretation of the data and preparation of the manuscript. LH wrote the manuscript; LL and CHW revised it. LL and CHW conceived of this study. All authors have read and approved this manuscript.

Acknowledgements
Funding: Delaware INBRE program, with grant from the National Institute of General Medical Sciences-NIGMS (P20 GM103446) from the National Institutes of Health.

Author details
[1]Department of Computer and Information Sciences, University of Delaware, 18 Amstel Avenue, 19716 Newark, DE, USA. [2]Center for Bioinformatics and Computational Biology, University of Delaware, 15 Innovation Way, 19711 Newark, DE, USA.

Received: 6 August 2015 Accepted: 9 February 2016
Published online: 19 February 2016

References

1. O Kuchaiev, M Rašajski, DJ Higham, N Pržulj, Geometric de-noising of protein-protein interaction networks. PLoS Comput. Biol. **5**(8), 1000454 (2009)
2. Y Murakami, K Mizuguchi, Homology-based prediction of interactions between proteins using averaged one-dependence estimators. BMC Bioinforma. **15**(1), 213 (2014)
3. L Salwinski, D Eisenberg, Computational methods of analysis of protein-protein interactions. Curr. Opin. Struct. Biol. **13**(3), 377–382 (2003)
4. R Craig, L Liao, Phylogenetic tree information aids supervised learning for predicting protein-protein interaction based on distance matrices. BMC Bioinforma. **8**(1), 6 (2007)
5. A Gonzalez, L Liao, Predicting domain-domain interaction based on domain profiles with feature selection and support vector machines. BMC Bioinforma. **11**(1), 537 (2010)
6. QC Zhang, D Petrey, L Deng, L Qiang, Y Shi, CA Thu, B Bisikirska, C Lefebvre, D Accili, T Hunter, T Maniatis, A Califano, B Honig, Structure-based prediction of protein-protein interactions on a genome-wide scale. Nature. **490**(7421), 556–560 (2012)

7. R Singh, D Park, J Xu, R Hosur, B Berger, Struct2net: a web service to predict protein-protein interactions using a structure-based approach. Nucleic Acids Res. **38**(suppl 2), 508–515 (2010)
8. Y Deng, L Gao, B Wang, ppipre: predicting protein-protein interactions by combining heterogeneous features. BMC Syst. Biol. **7**(Suppl 2), 8 (2013)
9. J Sun, Y Sun, G Ding, Q Liu, C Wang, Y He, T Shi, Y Li, Z Zhao, Inpreppi: an integrated evaluation method based on genomic context for predicting protein-protein interactions in prokaryotic genomes. BMC Bioinforma. **8**(1), 414 (2007)
10. Y-R Cho, M Mina, Y Lu, N Kwon, P Guzzi, M-finder: uncovering functionally associated proteins from interactome data integrated with go annotations. Proteome Sci. **11**(Suppl 1), 3 (2013)
11. S-H Jung, W-H Jang, D-S Han, A computational model for predicting protein interactions based on multidomain collaboration. IEEE/ACM Trans. Comput. Biol. Bioinforma. **9**(4), 1081–1090 (2012)
12. H-H Chen, L Gou, XL Zhang, CL Giles, in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. Discovering missing links in networks using vertex similarity measures. SAC '12 (ACM, New York, 2012), pp. 138–143
13. L Lü, T Zhou, Link prediction in complex networks: a survey. Physica A. **390**(6), 11501170 (2011)
14. C Lei, J Ruan, A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity. Bioinformatics. **29**(3), 355–364 (2013)
15. N Pržulj, Protein-protein interactions: making sense of networks via graph-theoretic modeling. BioEssays. **33**(2), 115–123 (2011)
16. L Page, S Brin, R Motwani, T Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*. (Stanford InfoLab, Stanford, CA, USA, 1999). Previous number = SIDL-WP-1999-0120, http://ilpubs.stanford.edu:8090/422/
17. H Tong, C Faloutsos, J-Y Pan, Random walk with restart: fast solutions and applications. Knowl. Inf. Syst. **14**(3), 327–346 (2008). doi:10.1007/s10115-007-0094-2
18. R-H Li, JX Yu, J Liu, in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. Link Prediction: The Power of Maximal Entropy Random Walk (ACM, New York, NY, USA, 2011), pp. 1147–1156. http://doi.acm.org/10.1145/2063576.2063741
19. L Backstrom, J Leskovec, in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. Supervised random walks: Predicting and recommending links in social networks. WSDM '11 (ACM, New York, 2011), pp. 635–644
20. F Fouss, K Francoisse, L Yen, A Pirotte, M Saerens, An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. Neural Netw. **31**(0), 53–72 (2012)
21. CV Cannistraci, G Alanis-Lobato, T Ravasi, Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. Bioinformatics. **29**(13), 199–209 (2013)
22. P Symeonidis, N Iakovidou, N Mantas, Y Manolopoulos, From biological to social networks: link prediction based on multi-way spectral clustering. Data Knowl. Eng. **87**(0), 226–242 (2013)
23. H Wang, H Huang, C Ding, F Nie, Predicting protein–protein interactions from multimodal biological data sources via nonnegative matrix tri-factorization. J. Comput. Biol. **20**(4), 344–358 (2013). doi:10.1089/cmb.2012.0273
24. AK Menon, C Elkan, in *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*. Link prediction via matrix factorization. ECML PKDD'11 (Springer, Berlin, 2011), pp. 437–452
25. Y Yamanishi, J-P Vert, M Kanehisa, Protein network inference from multiple genomic data: a supervised approach. Bioinformatics. **20**(suppl 1), 363–370 (2004)
26. L Huang, L Liao, CH Wu, Evolutionary model selection and parameter estimation for protein-protein interaction network based on differential evolution algorithm. IEEE/ACM Trans. Comput. Biol. Bioinforma. **12**(3), 622–631 (2015)
27. GRG Lanckriet, T De Bie, N Cristianini, MI Jordan, WS Noble, A statistical framework for genomic data fusion. Bioinformatics. **20**(16), 2626–2635 (2004)
28. T Ito, M Shimbo, T Kudo, Y Matsumoto, in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. Application of kernels to link analysis. KDD '05 (ACM, New York, 2005), pp. 586–592
29. AJ Smola, R Kondor, ed. by B Schölkopf, MK Warmuth. Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings, vol. 2777 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2003), pp. 144–158. doi:10.1007/978-3-540-45167-9_12
30. A Mantrach, N van Zeebroeck, P Francq, M Shimbo, H Bersini, M Saerens, Semi-supervised classification and betweenness computation on large, sparse, directed graphs. Pattern Recogn. **44**(6), 1212–1224 (2011)
31. J-Y Pan, H-J Yang, C Faloutsos, P Duygulu, in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Automatic multimedia cross-modal correlation discovery. KDD '04 (ACM, New York, 2004), pp. 653–658
32. S Kirkpatrick, CD Gelatt, MP Vecchi, Optimization by simulated annealing. Science. **220**(4598), 671–680 (1983)
33. L Salwinski, CS Miller, AJ Smith, FK Pettit, JU Bowie, D Eisenberg, The database of interacting proteins: 2004 update. Nucleic Acids Res. **32**(90001), 449–451 (2004)
34. P Jaccard, Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin del la Société Vaudoise des Sciences Naturelles. **37**, 547–579 (1901)
35. SF Altschul, W Gish, W Miller, EW Myers, DJ Lipman, Basic local alignment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)
36. ELL Sonnhammer, SR Eddy, R Durbin, Pfam: A comprehensive database of protein domain families based on seed alignments. Proteins Struct. Funct. Bioinforma. **28**(3), 405–420 (1997)
37. C Berg, JPR Christensen, P Ressel, *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*, 1st edn., vol. 100. (Springer-Verlag New York, New York, 1984). doi:10.1007/978-1-4612-1128-0
38. L Huang, L Liao, CH Wu, in *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference On*. Protein-protein interaction network inference from multiple kernels with optimization based on random walk by linear programming, (2015), pp. 201–207. doi:10.1109/BIBM.2015.7359681
39. M Deng, S Mehta, F Sun, T Chen, Inferring domain-domain interactions from protein-protein interactions. Genome Res. **12**(10), 1540–1548 (2002)
40. Z Itzhaki, E Akiva, Y Altuvia, H Margalit, Evolutionary conservation of domain-domain interactions. Genome Biol. **7**(12), 125 (2006)
41. J Park, M Lappe, SA Teichmann, Mapping protein family interactions: intramolecular and intermolecular protein family interaction repertoires in the {PDB} and yeast1. J. Mol. Biol. **307**(3), 929–938 (2001)
42. D Betel, R Isserlin, CWV Hogue, Analysis of domain correlations in yeast protein complexes. Bioinformatics. **20**(suppl 1), 55–62 (2004)