5

Mining Protein Interaction Groups

Lusheng Wang
Department of Computer Science
City University of Hong Kong
Hong Kong

1. Introduction

Proteins with interactions carry out most biological functions within living cells such as gene expression, enzymatic reactions, signal transduction, inter-cellular communications and immunoreactions. As the interactions are mediated by short sequence of residues among the long stretches of interacting sequences, these interacting residues or so-called interaction (binding) sites are at the central spot of proteome research. Although many imaging wet-lab techniques like X-ray crystallography, nuclear magnetic resonance spectroscopy, electron microscopy and mass spectrometry have been developed to determine protein interaction sites, the solved amount of protein interaction sites constitute only a tiny proportion among the whole population due to high cost and low throughput. Computational methods are still considered as the major approaches for the deep understanding of protein binding sites, especially for their subtle 3-dimensional structure properties that are not accessible by experimental methods.

The classical graph concept—maximal biclique subgraph (also known as maximal complete bipartite subgraph)—has been emerged recently for bioinformatics research closely related to topological structures of protein interaction networks and biomolecular binding sites. For example, Thomas *et al.* introduced complementary domains in (Thomas et al., 2003), and they showed that the complementary domains can form near complete bipartite subgraphs in PPI networks. A lock-and-key model has been proposed by Morrison *et al.* which is also based on the concept of maximal complete bipartite subgraphs (Morrison et al., 2006). Very recently, Andreopoulos *et al.* used clusters in PPI networks for identifying locally significant protein mediators (Andreopoulos et al., 2007). Their idea is to cluster common-friend proteins, which are in fact complete-bipartite proteins, based on their similarity to their direct neighborhoods in PPI networks. Other computational methods studying bipartite structures of PPI networks include (Bu et al., 2003; Hishigaki et al., 2001) which are focused on protein function prediction.

To identify motif pairs at protein interaction sites, Li *et al.* introduced a novel method with the core idea related to the concept of complete bipartite subgraphs from PPI networks (Li et al., 2006). The first step of the algorithm in (Li et al., 2006) finds large subnetworks with all-versus-all interactions (complete bipartite subgraphs) between a pair of protein groups. As the proteins within these protein groups have similar protein interactions and may share the same interaction sites, the second step of Li's algorithm is to compute conserved motifs

(possible interaction sites) by multiple sequence alignments within each protein group. Thus, those conserved motifs can be paired with motifs identified from other protein groups to model protein interaction sites. One of the novel aspects of the algorithm in (Li et al., 2006) is that it combines two types of data: the PPI data and the associated sequence data for modeling binding motif pairs.

Each protein in the above PPI networks is represented by a vertex and every interaction between two proteins is represented by an edge. Discovering complete bipartite subgraphs in PPI networks can thus be formulated as the following biclique problem: Given a graph, the biclique problem is to find a subgraph which is bipartite and complete. The objective is to maximize the number of vertices or edges in the bipartite complete subgraph. We note that the maximum vertex biclique problem is polynomial time solvable (Yannakakis, 1981). This problem is also equivalent to the maximum independent set problem on bipartite graphs which is known to be solvable by a minimum cut algorithm. However, the maximum vertex balanced biclique problem is NP-hard (Garey & Johnson, 1979). The maximum edge biclique problem is proved to be NP-hard as well (Peeters, 2003).

In this paper, we consider incompleteness of biological data, as the interaction data of PPI networks is usually not fully available. On the other hand, within an interacting protein group pair, some proteins in one group may only interact with a proportion of the proteins in the other group. Therefore, many subgraphs formed by interacting protein group pairs are not perfect bicliques. They are more often near complete bipartite subgraphs. Therefore, methods of finding bicliques may miss many useful interacting protein group pairs. To deal with this problem, we use quasi-bicliques instead of bicliques to find interacting protein group pairs. With the quasi-biclique, even though some interactions are missing in a protein interaction subnetwork, we can still find the two interacting protein groups. In this paper, we introduce and investigate the maximum vertex quasi-biclique problem. We show that the problem is NP-hard. We also propose approximation and heuristic algorithms for finding large quasi-bicliques in PPI networks. The applications for finding protein-protein binding sites are illustrated.

2. Bicliques and quasi-bicliques

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where each vertex represents a protein and there is an edge connecting two vertices if the two proteins have an interaction. Since $\mathcal{G}$ is an undirected graph, any edge $(u, v) \in \mathcal{E}$ implies $(v, u) \in \mathcal{E}$. For a selected edge (u, v) in $\mathcal{G}$, in order to find the two groups of proteins having the similar pairs of binding sites, we translate the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ into a bipartite graph. Let $X = \{x|(x, v) \in \mathcal{E}\}$, $Y_1 = \{y|(u, y) \in \mathcal{E} \& y \notin X\}$ and $Y_2 = \{w|(u, w) \in \mathcal{E} \& w \in X\}$. For a vertex $w \in Y_2$, w is incident to both u and v in $\mathcal{G}$. Thus both X and Y_2 contain w. We keep w in X and replace w in Y_2 with a new virtual vertex $\overline{w}$. After replacing all vertices w in Y_2 with $\overline{w}$, we get a new vertex set $\overline{Y_2}$. Let $Y = Y_1 \cup \overline{Y_2}$ and $E = \{(x, y)|(x, y) \in \mathcal{E} \& x \in X \& y \in Y_1\} \cup \{(x, \overline{w})|(x, w) \in \mathcal{E} \& x \in X \& \overline{w} \in \overline{Y_2}\}$. In this way, we have a bipartite graph $G = (X \cup Y, E)$. A biclique in G corresponds to two subsets of vertices, say, subset A and subset B, in $\mathcal{G}$. In $\mathcal{G}$, every vertex in A is adjacent to all the vertices in B, and every vertex in B is adjacent to all the vertices in A. Moreover, $A \cap B$ may not be empty. In this case, for any vertex $w \in A \cap B$, $(w, w) \in \mathcal{E}$. This is the case, where the protein has a self-loop. Self-loops are very common in practice. When a self-loop appears, one protein

molecule interacts with another identical protein molecule. For example, two identical protein subunits can assemble together to form a homodimeric protein.

In the following, we focus on the bipartite graph $G = (X \cup Y, E)$. For a vertex $x \in X$ and a vertex set $Y' \subseteq Y$, the degree of x in Y' is the number of vertices in Y' that are adjacent to x, denoted by $d(x, Y') = |\{y | y \in Y' \& (x, y) \in E\}|$. Similarly, for a vertex $y \in Y$ and $X' \subseteq X$, we use $d(y, X')$ to denote $|\{x | x \in X' \& (x, y) \in E\}|$. Now, we are ready to define the *δ-quasi-biclique*.

Definition 1. *For a bipartite graph $G = (X \cup Y, E)$ and a parameter $0 < \delta \leq \frac{1}{2}$, G is called a δ-quasi-biclique if for each $x \in X$, $d(x, Y) \geq (1 - \delta)|Y|$ and for each $y \in Y$, $d(y, X) \geq (1 - \delta)|X|$.*

Similarly, a δ-quasi-biclique in G corresponds to two subsets of vertices, say, subset A and subset B, in $\mathcal{G}$. In $\mathcal{G}$, every vertex in A is adjacent to at least $(1 - \delta)|B|$ vertices in B, and every vertex in B is adjacent to at least $(1 - \delta)|A|$ vertices in A. Moreover, according to the translation and the definition, $A \cap B$ may not be empty. Again, if a protein appears in both sides of a δ-quasi-biclique and there is an edge between the two corresponding vertices, the protein has a self-loop. In our experiments, we observe that about 22% of the δ-quasi-bicliques produced by our program contain self-loop proteins.

In many applications, due to various reasons, some edges in a clique/biclique may be missing and a clique/biclique becomes a quasi-clique/quasi-biclique. Thus, finding quasi-cliques/quasi-bicliques is more important in practice. Here we show that large quasi-bicliques may not contain any large bicliques.

Theorem 1. *Let $G = (X \cup Y, E)$ be a random graph with $|X| = |Y| = n$, where for each pair of vertices $x \in X$ and $y \in Y$, (x, y) is chosen, randomly and independently, to be an edge in E with probability $\frac{2}{3}$. When $n \to \infty$, with high probability, G is a $\frac{1}{2}$-quasi-biclique, and G does not contain any biclique $G' = (X' \cup Y', E')$ with $|X'| \geq 2 \log n$ and $|Y'| \geq 2 \log n$.*

In the biological context, Theorem 1 indicates that it is possible that some large interacting protein groups cannot be obtained by simply finding a maximal biclique if a few (interaction) edges are missing. As large interacting protein groups are more useful, according to this theorem, we have to develop new computational algorithms to extract from PPI networks large interacting protein groups which form quasi-bicliques.

In terms of false positive edges, both quasi-biclique and biclique can handle spurious edges very well. If very few spurious edges are added, in most cases, an irrelative protein will not be included in the quasi-bicliques or biclique unless $(1 - \delta)|A|$ spurious edges are simultaneously added to the protein that has no interaction with any of the proteins in A, where A is one of the two interaction groups.

The maximum vertex quasi-biclique problem is defined as follows.

Definition 2. *Given a bipartite graph $G = (X \cup Y, E)$ and $0 < \delta \leq \frac{1}{2}$, the maximum vertex δ-quasi-biclique problem is to find $X' \subseteq X$ and $Y' \subseteq Y$ such that the $X' \cup Y'$ induced subgraph is a δ-quasi-biclique and $|X'| + |Y'|$ is maximized.*

The maximum vertex biclique problem, where $\delta = 0$, can be solved in polynomial time (Yannakakis, 1981). Here we show that the maximum vertex δ-quasi-biclique problem

when $\delta > 0$ is NP-hard. The reduction is from *X3C* (Exact Cover by 3-Sets), which is known to be NP-hard (Karp, 1972).

Theorem 2. *For any constant integers $p > 0$ and $q > 0$ such that $0 < \frac{p}{q} \leq \frac{1}{2}$, the maximum vertex $\frac{p}{q}$-quasi-biclique problem is NP-hard.*

3. A polynomial time approximation scheme

The following lemma that is originally from (Li et al., 2002) will be repeatedly used in our proofs.

Lemma 1. *Let $X_1, X_2, \ldots, X_n$ be n independent random 0-1 variables, where X_i takes 1 with probability p_i, $0 < p_i < 1$. Let $X = \sum_{i=1}^{n} X_i$, and $\mu = E[X]$. Then for any $0 < \epsilon \leq 1$,*

$$\mathbf{Pr}(X > \mu + \epsilon\, n) < exp(-\frac{1}{3}n\epsilon^2),$$
$$\mathbf{Pr}(X < \mu - \epsilon\, n) \leq exp(-\frac{1}{2}n\epsilon^2).$$

The Main Ideas and Techniques: The problem can be formulated as a quadratic programming problem. We use a random sampling technique and a randomized rounding method to get a good approximate solution for the quadratic programming problem under the conditions that $|X_{opt}| = \Omega(|X|)$ and $|Y_{opt}| = \Omega(|Y|)$. The random sampling technique involves to randomly select $r1 = \Omega(\log |X_{opt}|))$ vertices from X_{opt} when X_{opt} is not known. This can be done when $|X_{opt}| = \Omega(|X|)$ and $|Y_{opt}| = \Omega(|Y|)$.

In order to make sure that $|X_{opt}| = \Omega(|X|)$ and $|Y_{opt}| = \Omega(|Y|)$, we design a combinatorial approach to find a subset $X' \subseteq X$ and a subset $Y' \subseteq Y$ such that $|X'| = \Omega(|X_{opt}| + |Y_{opt}|)$, $|X' \cap X_{opt}| \geq (1-\epsilon)|X_{opt}$, $|Y'| = \Omega(|X_{opt}| + |Y_{opt}|)$ and $|Y' \cap Y_{opt}| \geq (1-\epsilon)|Y_{opt}|$. See Lemma 2. Thus, we can work on a bipartite graph induced by X' and Y'. Without loss of generality, we can assume that $|Y_{opt}| \geq |X_{opt}|$. Now, two subcases arise: Case 1: $|X_{opt}| \leq \epsilon|Y_{opt}|$, and Case 2: $|X_{opt}| > \epsilon|Y_{opt}|$. For case 1, we can use linear programming approach and a brute-force approach to solve the problem. For case 2, we can use the quadratic programming approach to solve the problem.

Let $G = (X \cup Y, E)$ be the input bipartite graph. Let $X_{opt} \subseteq X$ and $Y_{opt} \subseteq Y$ be the optimal biclique for the maximum quasi-biclique problem. Without loss of generality, we can assume that

Assumption 1: $|Y_{opt}| \geq |X_{opt}|$.

The basic idea of our algorithm is to (1) formulate the problem into a quadratic programming problem and (2) use a random sampling approach to approximately solve the problem. In order to make the random sampling approach work, we have to make sure that

$$|X_{opt}| = \Omega(|X|) \tag{1}$$

and

$$|Y_{opt}| = \Omega(|Y|). \tag{2}$$

However, for any input bipartite graph $G = (X \cup Y, E)$, there is no guarantee that (1) and (2) hold. Here we propose a method to find a subset X' of X and Y' of Y such that for any $t > 0$, $|X_{opt}| = \Omega(|X'|)$, $|X_{opt} \cap X'| \geq \frac{t-1}{t}|X_{opt}|$, $|Y_{opt}| = \Omega(|Y'|)$, and $|Y_{opt} \cap Y'| \geq \frac{t-1}{t}|Y_{opt}|$. If we can obtain this kind of X' and Y', then we can work on the induced bipartite graph $G' = (X' \cup Y', E')$, where $E' = \{(u,v)|u \in X', v \in Y' \text{ and } (u,v) \in E\}$. Obviously, any good approximate solution of G' is also a good approximate solution of G.

Let x_i be a vertex in the bipartite graph $G = (X \cup Y, E)$. Define $D(x_i, Y)$ to be the set of vertices in Y that are incident to x_i. The following lemma tells us how to obtain X' and Y'.

Lemma 2. *For any $t > 0$, there exist k vertices $x_1, x_2, \ldots, x_k$ in X for $k = \lceil \delta t \rceil$ such that $|\bigcup_{i=1}^{k} D(x_i, Y)| \leq k(|Y_{opt}| + |X_{opt}|)$ and $|Y_{opt} \cap \bigcup_{i=1}^{k} D(x_i, Y)| \geq \frac{t-1}{t}|Y_{opt}|$. Similarly, there exists k vertices $y_1, y_2, \ldots, y_k$ in Y for $k = \lceil \delta t - 1 \rceil$ such that $|\bigcup_{i=1}^{k} D(y_i, X)| \leq k(|Y_{opt}| + |X_{opt}|)$ and $|X_{opt} \cap \bigcup_{i=1}^{k} D(y_i, X)| \geq \frac{t-1}{t}|X_{opt}|$.*

Though we do not know which k vertices in X we should choose, we can try all possible size k subsets of X in $O(|X|^k)$ time for constant k. The value of k is $\lceil \delta t \rceil$ and is determined by t later. Thus, from now on, we assume that the k vertices $x_1, x_2, \ldots, x_k$ are known. Let $X' = \bigcup_{i=1}^{k} D(y_i, X)$ and $Y' = \bigcup_{i=1}^{k} D(x_i, Y)$. We will focus on finding a quasi-biclique in the sub-graph $G' = (X' \cup Y', E')$ of G induced by X' and Y'.

Let $X'_{opt} \subseteq X'$ and $Y'_{opt} \subseteq Y'$ be a quasi-$(\delta + \frac{1}{t})$-biclique with maximum number of vertices in G'. From Lemma 2, $|X'_{opt}| + |Y'_{opt}| \geq (1 - \frac{1}{t})(|X_{opt}| + |Y_{opt}|)$ since $X' \cap X_{opt}$ and $Y' \cap Y_{opt}$ also form a quasi-$\delta + \frac{1}{t}$-biclique of size $(1 - \frac{1}{t})(|X_{opt}| + |Y_{opt}|)$. From now on, we will try to find a good approximate solution for X'_{opt} and Y'_{opt}.

If $|X'_{opt}|$ and $|Y'_{opt}|$ are approximately the same, then we have $|X'_{opt}| = \Omega(|X'|)$ and $|Y'_{opt}| = \Omega(|Y'|)$. That is, (1) and (2) hold for graph G'. Therefore, we can use quadratic programming approach to solve the problem. Nevertheless, there is no guarantee that $|X'_{opt}|$ and $|Y'_{opt}|$ are approximately the same. For any $\epsilon > 0$, we consider two cases.

Case 1: $|X'_{opt}| < \epsilon |Y'_{opt}|$. In this case, the number of vertices in Y'_{opt} will dominate the size of the whole quasi-biclique. If we select a vertex $x \in X'_{opt}$, then x and $D(x, Y')$ form a biclique of size at least $1 + (1 - \delta)|d(x, Y')| \geq 1 + (1 - \delta)|Y'_{opt}|$. When the value of δ is big with respect to ϵ, we do not have the desired quasi-biclique. If we try to add more vertices from Y', we have to guarantee that for every selected vertex y in Y', y is incident to at least $(1 - \delta)|X'|$ selected vertices in X'. This is impossible if x is the only selected vertex from X'. Therefore, we have to consider to add more vertices from both X' and Y'. It is clear that the task here is non-trivial.

In the following lemma, we will show that there exists a subset of r vertices (for some constant r) $X_r \subseteq X'$ and a subset $Y''_{opt} \subseteq Y'_{opt}$ such that X_r and Y''_{opt} form a quasi-$(\delta + \epsilon'')$-biclique with $|Y''_{opt}| \geq (1 - \epsilon'')|Y_{opt}|$ for some $\epsilon'' > 0$. Here r and ϵ'' are closely related.

Lemma 3. *Let $\frac{1}{t} = \epsilon'$. There exists a subset X'_r of X'_{opt} containing $r = \frac{2}{\epsilon'^2}\log(\frac{1}{\epsilon'})$ elements and a subset Y''_{opt} of Y'_{opt} with $|Y''_{opt}| \geq (1 - \frac{r(r-1)}{2|X_{opt}|} - 2\epsilon')|Y'_{opt}|$ such that X'_r and Y''_{opt} form a quasi-$(\delta + \frac{r(r-1)}{2|X'_{opt}|} + 2\epsilon')$-biclique.*

Based on Lemma 3, we can design an algorithm that finds a quasi-$(\delta + 4\epsilon')$-biclique with size at least $(1 - 4\epsilon' - \epsilon)(|X'_{opt}| + |Y'_{opt}|)$. Let $G' = (X' \cup Y', E')$ be the sub-graph obtained from Lemma 2. For any $\epsilon' > 0$, define $r = \frac{2}{\epsilon'^2} \log(\frac{1}{\epsilon'})$.

Case 1.1. $|X'_{opt}| \geq \frac{r(r-1)}{\epsilon'}$: When $|X'_{opt}| \geq \frac{r(r-1)}{\epsilon'}$, $\frac{r(r-1)}{2|X'_{opt}|} \leq \epsilon'$. Thus, there exist a quasi-$(\delta + 3\epsilon')$-biclique $X_r \subset X'$ and Y''_{opt} as described in Lemma 3.

We select r vertices from X'. For each subset $X_r \subseteq X'$ of r vertices $\{v_1, v_2, \ldots, v_r\}$, we define the following integer linear programming. Let $c_{i,j}$ be a constant, where $c_{i,j} = 1$ if $(v_i, u_j) \in E'$; and $c_{i,j} = 0$ if $(v_i, u_j) \notin E'$. Let y_i be a 0/1 variable, where $y_i = 1$ indicates that the vertex u_i in Y' is selected in the quasi-biclique and $y_i = 0$ otherwise.

$$y_i(\sum_{j=1}^{r} c_{i,j}) \geq (1 - \delta - \frac{1}{t} - \epsilon')r \tag{3}$$

$$\sum_{i=1}^{|Y'|} y_i c_{i,j} \geq (1 - \delta - 3\epsilon')|Y'_{opt}| \text{ for } j = 1, 2, \ldots, r, \tag{4}$$

Here we do not know $|Y'_{opt}|$. However, we can guess the value of $|Y'_{opt}|$ by trying $|Y'_{opt}| = 1, 2, \ldots, |Y'|$. The integer programming problem formulated by (3) and (4) has no objective function and we just want a feasible solution to fit (3) and (4). The integer programming problem is hard to solve. However, we can obtain a fractional solution $\bar{y_i}$ for (3) and (4) with $0 \leq \bar{y_i} \leq 1$ in polynomial time. After obtaining the fractional solution $\bar{y_i}$, we randomly set y_i to be 1 with probability $\bar{y_i}$.

Lemma 4. *Assume that* $\frac{1}{2}(1 - \delta - 3\epsilon')|Y'_{opt}|\epsilon'^2 \geq 2 \log r$ *and* $\frac{1}{t} = \epsilon'$. *With probability at least* $1 - \frac{1}{r}$, *we can get a pair of subsets* $X_A \subseteq X'$ *and* $Y_A \subseteq Y'$ *(an integer solution) by randomized rounding according to the probability* $\bar{y_i}$ *such that* X_A *and* Y_A *form a quasi-*$(\delta + 4\epsilon')$*-biclique with* $|X_A| + |Y_A| \geq (1 - \delta - 4\epsilon')|Y'_{opt}|$.

A standard method in (Li et al., 2002) can give a de-randomized algorithm.

When $\frac{1}{2}(1 - \delta - 3\epsilon')|Y'_{opt}|)\epsilon'^2 < 2 \log r$, we can enumerate all possible subsets of size $(1 - \delta - 3\epsilon')|Y'_{opt}|$ in Y' in polynomial time to get the desired solution.

Case 1.2. $|X'_{opt}| < \frac{r(r-1)}{\epsilon'}$: In this case, X'_{opt} and Y'_{opt} form the desired quasi-δ-biclique. Instead of selecting r vertices in X', we select $|X'_{opt}|$ vertices in X'. Though we do not know the value of $|x'_{opt}|$, we can guess the value for $|x'_{opt}| = 1, 2, \ldots, \frac{r(r-1)}{\epsilon'}$. We also solve the integer linear programming (3) and (4) in the same way as in Case 1.1. The algorithm for Case 1 is given in Fig. 1.

Theorem 3. *Assume* $|X'_{opt}| \leq \epsilon|Y'_{opt}|$. *We set* $\frac{1}{t} = \epsilon'$ *in the algorithm. With probability at least* $1 - \frac{1}{r}$, *Algorithm 1 finds a quasi-*$(\delta + 4\epsilon')$*-biclique* $X_A \subseteq X$ *and* $Y_A \subseteq Y$ *with* $|X_A| + |Y_A| \geq (1 - \delta - 4\epsilon')(|X_{opt}| + |Y_{opt}|)(1 - \epsilon')/(1 + \epsilon)$ *in time* $O((|X||Y|)^{\lceil \delta t \rceil}[|X||Y||Y'|^{\frac{4 \log r}{\epsilon'^2}} + |X'|^{\frac{r(r-1)}{\epsilon'}} \frac{r(r-1)}{\epsilon'}(|X| + |Y|)^3)])$.

Algorithm 1: Algorithm for Solving Case 1: $|X'_{opt}| \leq \epsilon |Y'_{opt}|$.

Input: a bipartite graph $G = (X \cup Y, E)$, a real number $0 \leq \delta \leq 0.5$, a number $t > 0$, a number $\epsilon > 0$, and a number $\epsilon' > 0$.

0. Let $k = \lceil \delta t \rceil$.
1. **for** any $v_1, v_2, \ldots, v_k \in X$ and any $u_1, u_2, \ldots, u_k \in Y$ **do**
2. Set $X' = \cup_{i=1}^{k} D(v_i, Y)$ and $Y' = \cup_{i=1}^{k} D(u_i, X)$.
3 $r == \frac{2}{\epsilon'^2} \log(\frac{1}{\epsilon'})$
4 Guess $|X'_{opt}|$ and $|Y'_{opt}|$ assuming $|X'_{opt}| \leq \epsilon |Y'_{opt}|$.
5 **if** $\frac{1}{2}(1 - \delta - 3\epsilon')|Y'_{opt}|\epsilon'^2 < 2 \log r$ **then** enumerate all possible subsets of size $(1 - \delta - 3\epsilon')|Y'_{opt}|$ in Y' in polynomial time to get the desired solution.
6 **if** $\frac{1}{2}(1 - \delta - 3\epsilon')|Y'_{opt}|)\epsilon'^2 > 2 \log r$) **then**
7 **for** $i = r, r + 1, \ldots \frac{r(r-1)}{\epsilon'}$ **do**
8 **for** every i-elements subset $X_i = \{x_1, x_2, \ldots, x_i\}$ **do**
9 give a fractinal solution $\bar{y}_i$ for (3) and (4).
10 randomly set $y_i = 1$ with probability $\bar{y}_i$.
8. Output a $\delta + \frac{1}{t} + 4\epsilon'$ quasi-biclique with the biggest $|X_A| + |Y_A|$.

Fig. 1. The algorithm for solving Case 1.

Case 2: $|X'_{opt}| \geq \epsilon |Y'_{opt}|$. In this case, we have $|X'_{opt}| = \Omega(|X'|)$ and $|Y'_{opt}| = \Omega(|Y'|)$. We will use a quadratic programming approach to solve the problem. We can formulate the quasi-biclique problem for the bipartite graph $G' = (X' \cup Y', E')$ into the following quadratic programming problem.

Quadratic programming formulation:

Let x_i and y_j be 0/1 variables, where $x_i = 1$ indicates that vertex v_i in X' is in the quasi-biclique and $y_j = 1$ indicates that vertex u_j in Y' is in the quasi-biclique. Define $e_{i,j} = 1$ if $(v_i, u_j) \in E'$ and $e_{i,j} = 0$ otherwise. Let c_1 and c_2 be two integers representing the sizes of X'_{opt} and Y'_{opt}, respectively. We can guess the values of c_1 and c_2 in polynomial time though we do not know c_1 and c_2. We have the following inequalities:

$$y_i(\sum_{j=1}^{|X'|} e_{i,j} x_j) \geq (1 - \delta - \frac{1}{t}) y_i c_1 \text{ for } i = 1, 2, \ldots, |Y'| \tag{5}$$

$$x_i(\sum_{j=1}^{|Y'|} e_{i,j} y_j) \geq (1 - \delta - \frac{1}{t}) x_i c_2 \text{ for } i = 1, 2, \ldots, |X'| \tag{6}$$

$$\sum_{i=1}^{|Y'|} y_i = c_1, \tag{7}$$

$$\sum_{i=1}^{|X'|} x_i = c_2. \tag{8}$$

(5) and (6) indicate that $x_i > 0$ and $y_i > 0$ imply that $\sum_{j=1}^{|X'|} e_{i,j} x_j \geq (1 - \delta - \frac{1}{t}) c_1$ and $\sum_{j=1}^{|Y'|} e_{i,j} y_j \geq (1 - \delta - \frac{1}{t}) c_2$, respectively.

Let $\hat{x}_i$ and $\hat{y}_j$ be the 0/1 integer solution for the quadratic programming problem (5)-(8). Let $\hat{r}_i = \sum_{j=1}^{|X'|} e_{i,j}\hat{x}_j$ and $\hat{s}_i = \sum_{j=1}^{|Y'|} e_{i,j}\hat{y}_j$. To deal with the quadratic programming problem, the key idea here is to estimate the values of $\hat{r}_i$ and $\hat{s}_j$. If we know the values of $\hat{r}_i$ and $\hat{y}_j$, then (5) and (6) become

$$y_i\hat{r}_i \geq y_i c_1(1-\delta-\frac{1}{t}) \text{ for } i = 1, 2, \ldots, |Y'| \tag{9}$$

$$x_i\hat{s}_i \geq x_i c_2(1-\delta-\frac{1}{t}) \text{ for } i = 1, 2, \ldots, |X'|, \tag{10}$$

where $\hat{r}_i$ and $\hat{s}_i$ in (9) and (10) are constants and the quadratic inequalities become linear inequalities.

Estimating $\hat{r}_i$ and $\hat{s}_i$.

The approach for giving a good estimation of $\hat{r}_i$ and $\hat{s}_i$ is to randomly and independently select a subset $B_{X'}$ of $O(\log(|X'_{opt}|))$ vertices and a subset $B_{Y'}$ of $O(\log(|Y'_{opt}|))$ vertices in X'_{opt} and Y'_{opt}, respectively. Let $c_1 = |X'_{opt}|$ and $c_2 = |Y'_{opt}|$. We do not know c_1 and c_2, but we can guess them in $O(|X'| \times |Y'|)$ time. Then we can use $\frac{c_1}{k}\sum_{v_j \in B_{X'}} e_{i,j}$ and $\frac{c_2}{k}\sum_{u_j \in B_{Y'}} e_{i,j}$ to estimate $\hat{r}_i$ and $\hat{s}_i$, respectively. Since we do not know X'_{opt} and Y'_{opt}, it is not easy to randomly and independently select vertices from X'_{opt} and Y'_{opt}. We develop a method to randomly select $p \times \log|Y'|$ vertices in Y'_{opt} from Y' when Y'_{opt} is not known. Here p is a constant to be determined later.

Finding $p\log|Y'|$ vertices in Y'_{opt} when Y'_{opt} is not known

Let $|Y'| = c|Y'_{opt}|$. The idea here is to randomly and independently select a subset B of $(c+1) \times p \times \log|Y'|$ vertices from Y' and enumerate all size $p \times \log|Y'|$ subsets of B in time $C^{p\log|Y'|}_{p(c+1)\log|Y'|} \leq O(|Y'|^{p(c+1)})$. We can show that with high probability, we can get a set of $p\log|Y'|$ vertices randomly and independently selected from Y'_{opt}.

Lemma 5. *With probability at least* $1 - |Y'|^{-\frac{p}{2c^2(c+1)}}$, *$B$ contains a size $p\log|Y'|$ subset of Y'_{opt}.*

Proof. Let us consider the probability that B contains less than $p\log|Y'|$ vertices in Y'_{opt}. Let b be the expected number of vertices in B that are also in Y'_{opt}. Recall that $|Y'| = c|Y'_{opt}|$. If we randomly select a vertex in Y', the probability that the vertex is in Y'_{opt} is $\frac{1}{c}$. Let μ be the expected number of vertices in B that are in Y'_{opt}. We have $\mu = \frac{|B|}{c} = \frac{1}{c}\lceil(c+1)p\log|Y'|\rceil$. Let $X_1, X_2, \ldots, X_{|B|}$ be $|B|$ independent random 0/1 variables, where $X_i = 1$ with probability $\frac{1}{c}$ indicating that the selected vertex is in Y_{opt}. Thus,

$$b = \sum_{i=1}^{|B|} X_i \tag{11}$$

and

$$\mu = E(\sum_{i=1}^{|B|} X_i) = \frac{1}{c}\lceil (c+1)p\log|Y'|\rceil. \tag{12}$$

Since we selected $(c+1)p\log|Y'|$ vertices,

$$|B| = \lceil (c+1)(p\log|Y'|)\rceil. \tag{13}$$

Based on Lemma 1, we have

$$\begin{aligned}
\mathbf{Pr}(b < p\log|Y'|) &\leq \mathbf{Pr}(b < (\frac{1}{c} - \frac{1}{c(c+1)})\lceil (c+1)(p\log|Y'|)\rceil) \\
&= \mathbf{Pr}(\sum_{i=1}^{|B|} X_i < \mu - \frac{1}{c(c+1)}|B|) \quad \text{(From (11), (12) and (13))} \\
&\leq exp(-\frac{1}{2c^2(c+1)^2}|B|) \\
&\leq exp(-\frac{1}{2c^2(c+1)^2}(c+1)(p\log|Y'|)) \\
&= exp(-\frac{p\log|Y'|}{2c^2(c+1)}) = |Y'|^{-\frac{p}{2c^2(c+1)}}.
\end{aligned}$$

Therefore, with probability at most $|Y'|^{-\frac{p}{2c^2(c+1)}}$, B does not contain any size $p\log|Y'|$ subset of Y'_{opt}. This completes the proof. □

Let $B_{X'}$ and $B_{Y'}$ be the sets of randomly and independently selected vertices in X'_{opt} and Y'_{opt}. Let $|B_{X'}| = p_1\log|X'|$ and $|B_{Y'}| = p_2\log|Y'|$. We define $\bar{r}_i = \sum_{v_j\in B_{X'}} e_{i,j}$ and $\bar{s}_i = \sum_{u_j\in B_{Y'}} e_{i,j}$. The following lemma shows that $\frac{c_1}{|B_{X'}|}\bar{r}_i$ and $\frac{c_2}{|B_{Y'}|}\bar{s}_i$ are good approximations of $\hat{r}_i$ and $\hat{s}_i$.

Lemma 6. *With probability at least* $1 - 2|Y'||X'|^{-\frac{\epsilon^2}{3}p_1} - 2|X'||Y'|^{-\frac{\epsilon^2}{3}p_2}$, *for any* $i = 1, 2, \ldots, |X'|$ *and* $j = 1, 2, \ldots, |Y'|$,

$$(1-\epsilon)\hat{r}_i \leq \frac{c_1}{|B_{X'}|}\bar{r}_i \leq (1+\epsilon)\hat{r}_i$$

and

$$(1-\epsilon)\hat{s}_j \leq \frac{c_2}{|B_{Y'}|}\bar{s}_j \leq (1+\epsilon)\hat{s}_j.$$

Now, we set $r_i = \frac{c_1}{|B_{X'}|}\bar{r}_i$ and $s_i = \frac{c_2}{|B_{Y'}|}\bar{s}_j$. We consider the following linear programming problem.

$$y_i r_i \geq y_i c_1(1-\epsilon)(1-\delta) \text{ for } i = 1, 2, \ldots, m, \tag{14}$$

$$x_i s_i \geq x_i c_2(1-\epsilon)(1-\delta) \text{ for } i = 1, 2, \ldots, m, \tag{15}$$

$$\sum_{i=1}^{|Y'|} y_i = c_1, \tag{16}$$

$$\sum_{i=1}^{|X'|} x_i = c_2 \tag{17}$$

$$\sum_{j=1}^{|X'|} e_{i,j} x_j \geq \frac{r_i}{1+\epsilon} \tag{18}$$

$$\sum_{j=1}^{|Y'|} e_{i,j} y_j \geq \frac{s_i}{1+\epsilon}. \tag{19}$$

The term $(1-\epsilon)$ in (14) and (15) ensures that the quadratic programming problem has a solution when the estimated values of r_i and s_i are smaller than $\hat{r_i}$ and $\hat{s_i}$. Similarly, the term $(1+\epsilon)$ in (18) and (19) ensures that the quadratic programming problem has a solution when the estimated values of r_i and s_i are bigger than $\hat{r_i}$ and $\hat{s_i}$.

Randomized rounding

Let x'_i and y'_j be a fractional solution for (14) -(19). In order to get a 0/1 solution, we randomly set x_i and y_j to be 1 using the fractional solution as the probability. That is, we randomly set x_i and y_j to be 1's with probability x'_i and y'_i, respectively. (Otherwise, x_i and y_j will be 0.)

Lemma 7. *With probability* $1 - 2exp(-\frac{1}{3}|X'|\epsilon^2) - 2exp(-\frac{1}{3}|Y'|\epsilon^2) - |Y'|exp(-\frac{1}{2}|X'|\epsilon^2) - |X'|exp(-\frac{1}{2}|Y'|\epsilon^2)$, *we can find a subset* $\hat{X} \subseteq X'$ *and a subset* $\hat{Y} \subseteq Y'$ *with* $(1-\epsilon)c_1 \leq |X'| \leq (1+\epsilon)c_1$ *and* $(1-\epsilon)c_2 \leq |Y'| \leq (1+\epsilon)c_2$ *such that for any* $x \in \hat{X}$, $d(x, Y') \geq (1-\delta-4\epsilon)|\hat{Y}|$ *and for any* $y \in \hat{Y}$, $d(y, X) \geq (1-\delta-4\epsilon)|\hat{X}|$.

The complete algorithm for Case 2 is given in Fig. 2. Let $k = \lceil \delta t \rceil$ as defined in Lemma 2. Here c_x , c_y are set to be $k(1+\frac{1}{\epsilon})$ and $2k$, respectively. $p_1 = p_2 = \frac{5}{\epsilon^2}$.

Theorem 4. *With probability at least* $1 - o(1)$, *Algorithm 2 finds a quasi-*$(\delta + 4\epsilon + \frac{1}{t})$*-biclique of size* $(1 - \frac{1}{t} - \epsilon)(|X_{opt}| + |Y_{opt}|)$ *in* $O((k \times \frac{1}{\epsilon^2}|X||Y|)^{\lceil \delta t \rceil}(|X|^{\frac{5}{\epsilon^2}k(1+\frac{1}{\epsilon})} + |Y|^{\frac{5}{\epsilon^2}2k})(|X| + |Y|^3))$ *time.*

We can derandomize the algorithm to get a polynomial time deterministic algorithm. Step 3 can be derandomized by using the standard method. For instance, instead of randomly and independently choosing $p_1 \log(|X'|)$ and $p_2 \log(|Y'|)$ vertices from X' and Y', we can pick the vertices encountered on a random walk of the same length on a constant degree expander. Obviously, the number of such random walks on a constant degree expander is polynomial. Thus, by enumerating all random walks of length $p_1 \log(|X'|)$ and $p_2 \log(|Y'|)$, we have a polynomial time deterministic algorithm.

Algorithm 2: Algorithm for Soving Case 2: $|X'_{opt}| > \epsilon|Y'_{opt}|$.

Input: a bipartite graph $G = (X \cup Y, E)$, a real number $0 \leq \delta \leq 0.5$, a number $t > 0$ and a number $\epsilon > 0$.

0. Let $k = \lceil \delta t \rceil$, $p_1 = p_2 = \frac{5}{\epsilon^2}$, $c_x = k(1 + \frac{1}{\epsilon})$ and $c_y = 2k$.
1. **for** any $v_1, v_2, \ldots, v_k \in X$ and any $u_1, u_2, \ldots, u_k \in Y$ **do**
2. Set $X' = \cup_{i=1}^{k} D(v_i, Y)$ and $Y' = \cup_{i=1}^{k} D(u_i, X)$.
3 Randomly and independently select a set $S_{X'}$ of $(c_x + 1)p_1 \log |X'|$ vertices in X' and a set $S_{Y'}$ of $(c_y + 1)p_2 \log |Y'|$ vertices in Y'.
4 **for** any size $p_1 \log |X'|$ subsset $B_{X'}$ of $S_{X'}$ and size $p_2 \log |X'|$ subset $B_{Y'}$ of $S_{Y'}$ **do**
   (a) $\bar{r_i} = \frac{c_1}{|B_{X'}|} \sum_{v_i \in |B|_{X'}} e_{i,j}$
   (b) $\bar{s_i} = \frac{c_2}{|B_{Y'}|} \sum_{u_i \in |B|_{Y'}} e_{i,j}$
   (c) Get a fractional solution x'_i and y'_i for $x_i \in X'$ and $y_i \in Y'$ of (11)-(16)
   (d) do randomrized rouding according to x'_i and y'_i
   (e) $X_A = \{v_i | x_i = 1\}$ and $Y_A = \{u_i | y_i = 1\}$
5. Output a $\delta + \frac{1}{t} + 4\epsilon$ quasi-biclique with the biggest $|X_A| + |Y_A|$.

Fig. 2. The algorithm for Case 2.

Step 4 (d) can be derandomized by using Raghavan's conditional probabilities method (Raghavan, 1988). From Case 1 and Case 2, we can immediately obtain the following theorem.

Theorem 5. *There exists a polynomial time approximation scheme that outputs a quasi-biclique $X_A \subseteq X$ and $Y_A \subseteq Y$ with $|X_A| + |Y_A| \geq (1 - \epsilon)(|X_{opt}| + |Y_{opt}|)$ such that any vertex $x \in X_A$ is incident to at least $(1 - \delta - \epsilon)|Y_A|$ vertices in Y_A and any vertex $y \in Y_A$ is incident to at least $(1 - \delta - \epsilon)|X_A|$ vertices in X_A for any $\epsilon > 0$, where X_{opt} and Y_{opt} form the optimal solution.*

4. The heuristic algorithm

In practice, we need to find large quasi-bicliques in PPI networks. Here, we propose a heuristic algorithm to find large quasi-bicliques. Consider a PPI network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Our heuristic algorithm has two steps. First, we construct a bipartite graph from the graph $\mathcal{G}$ based on a pair of interacting proteins (u, v). Using the method described at the beginning of Section 2, we can get a bipartite graph $G = (X \cup Y, E)$. Second, we find quasi-bicliques in G. The bipartite graph G contains all proteins that have interactions with u or v. So we can find large quasi-bicliques containing u and v in the bipartite graph.

In the algorithm for finding quasi-bicliques in G, we have two parameters δ and τ, which control the quality and sizes of the quasi-bicliques. We use a greedy method to get the seeds for finding large quasi-bicliques in G. At the beginning, we set $X' = \phi$ and $Y' = Y$. In each step, we find a vertex with the maximum degree in $X - X'$. The vertex is added into the biclique vertex set X', and we eliminate all vertices y in Y' such that $d(y, X') < (1 - \delta)|X'|$. We will continue this process until the size of Y' is less than τ. At each step, we get a seed for finding large quasi-bicliques.

The seeds may miss some possible vertices in the quasi-bicliques. We can extend the seeds to find larger quasi-bicliques. Let $X'' = X'$ and $Y'' = Y'$ be a pair of seed vertex sets. In the first step, we can find a vertex x in $X - X''$ with the largest degree $d(x, Y'')$ in $X - X''$. If

$d(x, Y'') \geq (1-\delta)|Y''|$, we add the vertex x to X''. In the second step, we can find a vertex y in $Y - Y''$ with the largest $d(y, X'')$ in $Y - Y''$. If $d(y, X'') \geq (1-\delta)|X''|$, we add the vertex y to Y''. We repeat the above two steps until no vertex can be added. The whole algorithm is shown in Fig. 3. We can also exchange the two vertex sets X and Y to find more quasi-bicliques using the algorithm.

Let n be the number of vertices in the bipartite graph G. In the greedy algorithm, the time complexity of Steps $3-5$ and Step 10 is $O(n)$, and the time complexity of Steps $6-9$ is $O(n^2)$. So the time complexity of Steps $3-10$ is dominated by $O(n^2)$. Since Steps $3-10$ is repeated $O(n)$ times, the time complexity of the whole algorithm is $O(n^3)$.

The Greedy Algorithm

Input A bipartite graph $(X \cup Y, E)$ and two parameters δ and τ.

Output A set of δ-quasi-bicliques $(X' \cup Y', E')$ with $|X'| \geq \tau$ and $|Y'| \geq \tau$.

1. Let $X' = \phi$ and $Y' = Y$.
2. **while** $|Y'| \geq \tau$ and $X' \neq X$ **do**
3. Find the vertex $x \in X - X'$ with the maximum degree $d(x, Y')$.
4. Add x into X', $X' = X' \cup \{x\}$, and delete from Y' all vertices $y \in Y'$ such that $d(y, X') < (1-\delta)|X'|$.
5. $X'' = X'$ and $Y'' = Y'$.
6. **repeat**
7. Find the vertex $x \in X - X''$ with the maximum degree $d(x, Y'')$. If $d(x, Y'') \geq (1-\delta)|Y''|$, add x to X'', $X'' = X'' \cup \{x\}$.
8. Find the vertex $y \in Y - Y''$ with the maximum degree $d(y, X'')$. If $d(y, X'') \geq (1-\delta)|X''|$, add y to Y'', $Y'' = Y'' \cup \{y\}$.
9. **until** no vertex is added in the steps 7 and 8.
10. **if** $|X''| \geq \tau$, $|Y''| \geq \tau$, for each $x \in X''$, $d(x, Y'') \geq (1-\delta)|Y''|$, for each $y \in Y''$, $d(y, X'') \geq (1-\delta)|X''|$, output $(X'' \cup Y'')$ as a quasi-biclique.

Fig. 3. The greedy algorithm.

5. Finding motifs from the multiple sequence alignment of computed δ-bicliques.

We implemented the heuristic algorithm described in the last section in JAVA. The software is called PPIExtend. In the implementation, we added a new parameter α to speed up the algorithm. In Step 3, instead of selecting one vertex with the best degree, we can select the best α vertices in $X - X'$ and add all the α vertices into X' in Step 4. As shown in the last step of the algorithm, some vertices in X'' may be adjacent to less than $(1-\delta)|Y''|$ vertices in Y'', but the average degree of the vertices in X'' is no less than $(1-\delta)|Y''|$. Similarly, some vertices in Y'' may be adjacent to less than $(1-\delta)|X''|$ vertices in $|X''|$, but the average degree of the vertices in Y'' is no less than $(1-\delta)|X''|$. In our experiments, these quasi-bicliques are still output to get more useful quasi-bicliques.

Our algorithm for PPIExtend consists of two steps: (i) find interacting protein group pairs (quasi-bicliques) using the greedy algorithm, (ii) find conserved motifs from multiple sequence alignments for each of the protein groups. (We use the existing multiple sequence alignment software PROTOMAT (Pietrokovski, 1996).)

The motifs found by PROTOMAT can be viewed as a *block*, that is a conserved region in a multiple sequence alignment of the proteins in a group. For each biclique X and Y obtained by the greedy algorithm, we use S_X and S_Y to denote the sets of motifs obtained by the multiple sequence alignments of protein sequences in X and Y, respectively. Any pair of motifs (m_1, m_2) with $m_1 \in S_X$ and $m_2 \in S_Y$ is a candidate protein-protein interaction motif pair. Thus, our algorithm can also output lots of motif pairs as candidate protein-protein interaction motif pairs.

We look at the numbers of motifs found by the programs PPIExtend and FPClose* that are also in the two block databases, BLOCKS (Pietrokovski, 1996) and PRINTS (Attwood & Beck, 1994). The LAMA program (Pietrokovski, 1996) is used to find the local optimal alignment of two blocks (the motif output by PPIExtend/FPClose* and a block in the databases), where the Z-score is computed to measure the alignments. The default threshold of Z-score was used in the experiments. The results are reported in Table 1. From this table, we can see that our method has more mappings to BLOCKS and PRINTS than FPClose* (Li et al., 2006; Grahne & Zhu, 2003).

	BLOCKS		PRINTS		BOTH	
	blocks	domains	blocks	domains	blocks	domains
FPClose*	6408/24294	3128/4944	2174/11170	1093/1850	24.1%	62.1%
PPIExtend	9325/29767	4191/6149	2423/11435	1160/1900	28.5%	66.4%

Table 1. The mappings between the motifs and the two databases: BLOCKS and PRINTS. FPClose* uses BLOCKS 14.0 and PRINTS 37.0. Our PPIExtend method uses BLOCKS 14.3 and PRINTS 38.0. Each entry a/b means the motifs are mapped to a blocks(domains) in all b blocks(domains) in the databases.

	BLOCKS	PRINTS	Pfam	*i*Pfam
Version	14.3	38.0	20.0	20.0
Number of domains	6149	1900	8296	2883
Number of entries	29767	11435	8296	3019

Table 2. Databases used in the experiments.

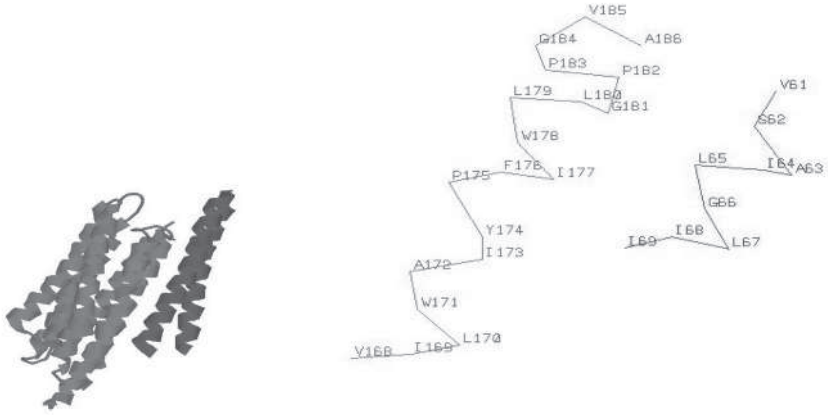
We look at the numbers of motif pairs found by the two programs PPIExtend and FPClose* that can be mapped into domain-domain interaction pairs in the domain-domain interaction database *i*Pfam (Finn et al., 2005). The versions of the databases are shown in Table 2. The *i*Pfam database is built on top of the Pfam database (Sonnhammer et al., 1997) which stores the information of protein domain-domain interactions. To examine whether the motif pairs found by PPIExtend and FPClose* can match some pairs of interacting domains in *i*Pfam, we map our motif pairs to domain pairs in *i*Pfam through the integrated protein family database InterPro (Apweiler et al., 2001) which integrates a number of databases. In fact, we strictly follow the procedure as suggested in (Li et al., 2006). (1) We map our motifs to domains (protein groups) in the database BLOCKS or PRINTS; (2) we map a protein group of BLOCKS to a protein group of InterPro based on the one-to-one mapping between an entry of BLOCKS

and an entry of InterPro; (Note that both PRINTS and Pfam are member databases of InterPro, and the mapping between PRINTS and Pfam is clear.) (3) we use existing cross-links between protein groups of InterPro and domains of Pfam to determine the crosslinks between the motifs found by PPIExtend/FPClose* and Pfam domains. In this way, we can map our motif pairs into domain pairs with Pfam domain entries. Note that the mapping between motif pairs and domain pairs is not one-to-one.

We observed that the motif pairs found by PPIExtend can map to 81 distinct domain pairs in *i*Pfam. However, only 18 domain pairs were reported in (Li et al., 2006). This is a significant improvement and the main reason is the use of quasi-bicliques. In the 81 domain pairs, 48 pairs are domain-domain interactions on one protein (self-loops) and 33 pairs are domain-domain interactions on different proteins. Although the self-loops form a large portion, we still find many other domain-domain interactions that are not self-loops.

6. Protein interaction sites: a case study

In this section, we present detailed information about binding motif pairs that can be mapped to interacting domain pairs. The first motif pair is derived from a protein group pair in which the left protein group contains 7 proteins and the right protein group contains 10 proteins. There are 66 interactions between the two groups of proteins. Using the hypergeometric probability model, the p-value of the protein group pair is less than 1.57×10^{-191}. PROTOMAT finds two left blocks and two right blocks in this protein group pair. The second left block contains 20 positions and the first right block contains 12 positions. By the mapping method, the positions $1-19$ of the second left block can be aligned with the positions $9-27$ of block IPB001425B in BLOCKS, and the positions $4-12$ of the first right block can be aligned with the positions $1-9$ of block IPB003660A in BLOCKS. Block IPB001425B is in the Bac_rhodopsin domain, and block IPB003660A is in the HAMP domain. See Table 3 for more details. Our binding motif pair can map into the domain pair (PF00672, PF01036) in *i*Pfam. *i*Pfam shows that the HAMP domain interacts with the Bac_rhodopsin domain in protein complexes such as lh2s. 1h2s is the complex of *Natronobacterium pharaonis* sensory rho-dopsin II (sRII) with receptor-binding domain of HtrII. The X-ray structure of 1h2s was obtained at 1.93 Å resolution (Gordeliy et al., 2002) and it provided an atomic picture of the first step of the signal transduction. The interactions in the sRII-HtrII complex have been intensively investigated to find the signal relay mechanism from the receptor to the transducer (Bergo et al., 2005; Inoue et al., 2007; Sudo et al., 2007). The 3D structure of the interactions is shown in Fig. 4(a) and 4(b), which are generated by Protein Explorer (Martz, 2002). The shortest residue-residue distance between the two motifs in a pair is also interesting. In protein complex 1h2s, there are two chains: chain A (1h2s_A) and chain B (1h2s_B). The left motif is located at positions $168-186$ of 1h2s_A, and the right motif is located at positions $61-69$ of 1h2s_B (Table 3). We downloaded the coordinate information of 1h2s from http://www.ebi.ac.uk/msd-srv/msdlite/atlas/summary/1h2s.html, and computed the residue-residue distances between the two motifs. The shortest residue-residue distance is 4.07 Å between atom 1346 of residue 177 in 1h2s_A and atom 2018 of residue 69 in protein 1h2s_B (Fig. 4(b)). The average shortest residue-residue distance is 9.17Å. From these



(a) The 3D structure of 1h2s (asymmetric unit).

(b) The backbone structure of the two motifs in 1h2s.

Fig. 4. (a) The 3D structure (best viewed in color) of the interactions between the Bac_rhodopsin domain and the HAMP domain in 1h2s. The left part is chain A and contains the Bac_rhodopsin domain. The right part is chain B and contains the HAMP domain. (b) The backbone structure of the interactions between segment [168V,186A] in 1h2s_A and segment [61V,69I] in 1h2s_B.

calculation and information, we may conclude that the positions $1-19$ of the second left block and the positions $4-12$ of the first right block are possibly interaction sites.

7. Prediction of binding sites

After obtaining candidate domains (conserved regions) in multiple sequence alignment, we can further verify if a pairs of predicted domains really interact with each other by using some tools for protein binding site prediction. Here we briefly introduce a method originally in (Guo & Wang, 2011). This method assumes that the 3D structures of the two given proteins are known.

Given two complete protein structures, the task is to find the binding sites between the two proteins. The method contains three steps. Firstly, we do local sequence alignment at the atom level to get the alignments of conserved regions. Those alignments of conserved regions may contain some gaps. Secondly, among the conserved regions obtained in Step 1, we use the 3D structure information to identify the surface segments. Finally, for any pair of the surface segments identified in Step 2, we compute a rigid transformation to compare the similarity of the two substructures in 3D space and output the qualified pairs as binding sites. When computing the rigid transformations, we treat each protein as a molecule with some volume and introduce a method to ensure that the two whole protein 3D structures have no overlap under such a rigid transformation in 3D space. The software package is available at http://sites.google.com/site/guofeics/bsfinder.

```
AC  l18493xB;
    distance from previous block=(4,396)
DE  none BL  IIK motif=[6,0,17] motomat=[1,1,-10]
    width=20 seqs=7
 DIP:8095N  ( 206) VIGILIISYTKATCDMLAGK
 DIP:4973N  ( 536) MILILIAQFWVAIAPIGEGK
 DIP:5150N  ( 417) LIKDEINNDKKDNADDKYIK
 DIP:5371N  ( 384) IILALIVTILWFMLRGNTAK
 DIP:676N   ( 402) VIVAWIFFVVSFVTTSSVGK
 ...
 pdb 1h2s_A ( 168) VILWAIYPFIWLLGPPGVA
 Bac_rhodopsin:    VVLWLAYPVVWLLGPEGIG

AC  r18493xA;
    distance from previous block=(7,177)
DE  none BL  LLL motif=[6,0,17] motomat=[1,1,-10]
    width=12 seqs=8
 DIP:7371N  (  10) LALIILYLSIPL
 DIP:8128N  (  35) LSLRFLALIFDL
 DIP:4176N  ( 106) LVLTSLSLTLLL
 DIP:7280N  (  11) LSLFLPPVAVFL
 DIP:5331N  ( 178) LSFFVLCGLARL
 ...
 pdb 1h2s_B (  61)    VSAILGLII
 HAMP:                IALLLALLL
```

Table 3. Left block l18493xB aligning with the Bac_rhodopsin domain and right block r18493xA aligning with the HAMP domain. For brevity, only 5 sequences in each of the two blocks are shown. In line Bac_rhodopsin and line HAMP, each letter is the amino acid with the highest frequency in the corresponding column in the multiple alignment. Pdb 1h2s_A and pdb 1h2s_B are chain A and chain B in protein complex 1h2s, respectively.

8. Conclusion

We have proposed algorithms for finding the maximum vertex quasi-biclique problem. We illustrate the applications of the proposed algorithms for finding protein-protein binding sites. The general approach contains three steps: (1) find quasi-bicliques from PPI networks; (2) do multiple sequence alignment for each of the groups in the quasi-biclique and identify possible domains on the protein sequences. (3) use other methods, e.g., the one in (Guo & Wang, 2011), to further confirm the binding sites.

9. References

Yannakakis, M. (1981). Node deletion problems on bipartite graphs. *SIAM Journal on Computing*, Vol. 10, 1981, 310–327.

Thomas, A.; Cannings, R.; Monk, N. A. M. & Cannings, C. (2003). On the structure of protein-protein interaction networks. *Biochemical Society Transanctions*, Vol. 31, Dec 2003, 1491–1496.

Morrison, J. L.; Breitling, R.; Higham, D. J. & Gilbert, D. R. (2006). A lock-and-key model for protein-protein interactions. *Bioinformatics*, Vol. 22, No. 16, Aug 2006, 2012–2019.

Andreopoulos, B.; An, A.; Wang, X.; Faloutsos, M. & Schroeder, M. (2007). Clustering by common friends finds locally significant proteins mediating modules. *Bioinformatics*, Vol. 23, No. 9, 2007, 1124–1131.

Bu, D.; Zhao, Y.; Cai, L.; Xue, H.; Zhu, X.; Lu, H.; Zhang, J.; Sun, S.; Ling, L.; Zhang, N.; Li, G. & Chen, R. (2003). Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research*, Vol. 31, No. 9, May 2003, 2443–2450.

Hishigaki, H.; Nakai, K.; Ono, T.; Tanigami, A. & Takagi, T. (2001). Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast*, Vol. 18, No. 6, Apr 2001, 523–531.

Li, H.; Li, J.; Wong, L. (2006). Discovering motif pairs at interaction sites from protein sequences on a proteome-wide scale. *Bioinformatics*, Vol. 22, No. 8, 2006, 989–996.

Garey, M. R. & Johnson, D.S. (1979). *Computers and Intractability, A guide to the theory of NP-completeness*. Freeman, San Francisco, 1979.

Peeters, R. (2003). The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, Vol. 131, No. 3, 2003, 651–654.

Karp, R. M. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations (R. E. Miller and J. W. Thatcher, eds.)*, 1972, 85–103.

Peleg, D.; Schechtman, G. & Wool, A. (1993). Approximating bounded 0-1 integer linear programs, *Proceedings of the 2nd Symposium on Theory of Computing and Systems.* IEEE Computer Society, 1993.

Pietrokovski, S. (1996). Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nucleic Acids Research*, Vol. 24, 1996, 3836–3845.

Attwood, T. K. & Beck, M. E. (1994). PRINTS-a protein motif fingerprint database. *Protein Engineering, Design and Selection*, Vol. 7, 1994, 841–848.

Finn, R. D.; Marshall, M.; & Bateman,A. (2005). iPfam: visualization of protein-protein interactions in PDB at domain and amino acid resolutions. *Bioinformatics*, Vol. 21, No. 3, Feb 2005, 410–412.

Grahne, G. & Zhu, J. (2003). Efficiently using prefix-trees in mining frequent itemsets. *Proceedings of the Workshop on Frequent Itemset Mining Implementataions (FIMI)*, 2003.

Sonnhammer, E. L.; Eddy, S. R. & Durbin, R. (1997). Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins: Structure, Function and Genetics*, Vol. 28, 1997, 405–420.

Apweiler, R.; Attwood, T. K.; Bairoch, A.; Bateman, A.; Birney, E.; Biswas, M.; Bucher, P.; Cerutti, L.; Corpet, F.; Croning, M. D.; Durbin, R.; Falquet, L.; Fleischmann, W.; Gouzy, J.; Hermjakob, H.; Hulo, N.; Jonassen, I.; Kahn, D.; Kanapin, A.; Karavidopoulou, Y.; Lopez, R.; Marx, B.; Mulder, N. J.; Oinn, T. M.; Pagni, M.; Servant, F.; Sigrist, C. J. & Zdobnov, E. M. (2001). The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Research*, Vol. 29, No. 1, Jan 2001, 37–40.

Gordeliy, V. I.; Labahn, J.; Moukhametzianov, R.; Efremov, R.; Granzin, J.; Schlesinger, R.; Büldt, G.; Savopol, T.; Scheidig, A. J.; Klare, J. P. & Engelhard, M. (2002). Molecular basis of transmembrane signalling by sensory rhodopsin II-transducer complex. *Nature*, Vol. 419, No. 6906, Oct 2002, 484–487.

Bergo, V. B.; Spudich, E. N.; Rothschild, K. J. & Spudich, J. L. (2005). Photoactivation perturbs the membrane-embedded contacts between sensory rhodopsin II and its transducer. *Journal of Biological Chemistry*, Vol. 280, No. 31, Aug 2005, 28 365–28 369.

Inoue, K.; Sasaki, J.; Spudich, J. L. & Terazima, M. (2007). Laser-induced transient grating analysis of dynamics of interaction between sensory rhodopsin II D75N and the HtrII transducer. *Biophysical Journal*, Vol. 92, No. 6, Mar 2007, 2028–2040.

Sudo, Y.; Furutani, Y.; Spudich, J. L. & Kandori, H. (2007). Early photocycle structural changes in a bacteriorhodopsin mutant engineered to transmit photosensory signals. *Journal of Biological Chemistry*, Vol. 282, No. 21, May 2007, 15 550–15 558.

Martz, E. (2002). Protein Explorer: easy yet powerful macromolecular visualization. *Trends in Biochemical Sciences*, Vol. 27, No. 3, 2002, 107–109.

Sivaraman, J.; Li, Y.; Banks, J.; Cane, D. E.; Matte, A. & Cygler, M. (2003). Crystal structure of Escherichia coli PdxA, an enzyme involved in the pyridoxal phosphate biosynthesis pathway. *Journal of Biological Chemistry*, Vol. 278, No. 44, Oct 2003, 43 682–43 690.

Sakai, A.; Kita, M. & Tani, Y. (2004). Recent progress of vitamin B6 biosynthesis. *Journal of Nutritional Science and Vitaminology (Tokyo)*, Vol. 50, No. 2, Apr 2004, 69–77.

Fitzpatrick, T. B.; Amrhein, N.; Kappes, B.; Macheroux, P.; Tews, I. & Raschle, T. (2007). Two independent routes of de novo vitamin B6 biosynthesis: not that different after all. *Biochemistry Journal*, Vol. 407, No. 1, Oct 2007, 1–13.

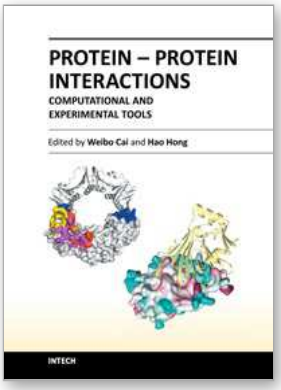
Guo, F. & Wang, L. (2011). Computing the Protein Binding Sites. *ISBRA*, 2011, 25-36.

Liu, X.; Li, J. & Wang, L. (2010). Modeling Protein Interacting Groups by Quasi-Bicliques: Complexity, Algorithm, and Application. *IEEE/ACM Trans. Comput. Biology Bioinform.*, Vol. 7, No. 2, 2010, 354-364.

Wang, L. (2011). Near Optimal Solutions for Maximum Quasi-bicliques, *Journal of Combinatorial Optimization*, on-line available at DOI 10.1007/s10878-011-9392-4.

Li, M.; Ma, B. & Wang, L. (2002). On the closest string and substring problems. *Journal of the ACM*, Vol. 49, No. 2, 2010, 157-171.

Raghavan, P. (1988). Probabilistic construction of deterministic algorithms: Approximate packing integer programs. *JCSS*, Vol. 37, No. 2, 2010, 130-143.



Protein-Protein Interactions - Computational and Experimental Tools

Edited by Dr. Weibo Cai

ISBN 978-953-51-0397-4
Hard cover, 472 pages
Publisher InTech
Published online 30, March, 2012
Published in print edition March, 2012

Proteins are indispensable players in virtually all biological events. The functions of proteins are coordinated through intricate regulatory networks of transient protein-protein interactions (PPIs). To predict and/or study PPIs, a wide variety of techniques have been developed over the last several decades. Many in vitro and in vivo assays have been implemented to explore the mechanism of these ubiquitous interactions. However, despite significant advances in these experimental approaches, many limitations exist such as false-positives/false-negatives, difficulty in obtaining crystal structures of proteins, challenges in the detection of transient PPI, among others. To overcome these limitations, many computational approaches have been developed which are becoming increasingly widely used to facilitate the investigation of PPIs. This book has gathered an ensemble of experts in the field, in 22 chapters, which have been broadly categorized into Computational Approaches, Experimental Approaches, and Others.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Lusheng Wang (2012). Mining Protein Interaction Groups, Protein-Protein Interactions - Computational and Experimental Tools, Dr. Weibo Cai (Ed.), ISBN: 978-953-51-0397-4, InTech, Available from: http://www.intechopen.com/books/protein-protein-interactions-computational-and-experimental-tools/mining-protein-interaction-groups

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821