SLIDER: Mining correlated motifs in protein-protein interaction networks

Peter Boyen, Frank Neven, Dries Van Dyck
Hasselt University, Transational University of Limburg
Agoralaan Building D, 3590 Diepenbeek, Belgium
{peter.boyen, frank.neven, dries.vandyck}@uhasselt.be

Aalt-Jan D.J. van Dijk, Roeland C.H.J. van Ham
Applied Bioinformatics - Plant Research International, Wageningen UR
Droevendaalsesteeg 1, Wageningen, The Netherlands

Abstract

*Correlated motif mining (*CMM*) is the problem to find overrepresented pairs of patterns, called motif pairs, in interacting protein sequences. Algorithmic solutions for* CMM *thereby provide a computational method for predicting binding sites for protein interaction. In this paper, we adopt a motif-driven approach where the support of candidate motif pairs is evaluated in the network. We experimentally establish the superiority of the χ^2-based support measure over other support measures. Furthermore, we obtain that* CMM *is an* NP*-hard problem for a large class of support measures (including χ^2) and reformulate the search for correlated motifs as a combinatorial optimization problem. We then present the method* SLIDER *which uses local search with a neigborhood function based on sliding motifs and employs the χ^2-based support measure. We show that* SLIDER *outperforms existing motif-driven* CMM *methods and scales to large protein-protein interaction networks.*

1. Introduction

Large-scale biological networks describing interactions between proteins are available now for several organisms [12]. Such data demonstrates how proteins function as part of an interaction network, but provide no insight into how interactions are encoded in protein sequences. In particular, it is unknown which part of the sequences correspond with physical interaction sites. Unfortunately, the discovery of these sites requires laborious and expensive biological experiments. In fact, it is estimated that it would take 20 years to determine all interactions types using current experimental techniques [2]. Therefore, several computational approaches have been proposed to locate binding sites by mining overrepresented pairs of patterns (called motifs) in the sequences of interacting proteins [7, 8, 9, 10, 13]. Correlated motif mining (CMM) is an approach to identify binding sites by looking for a consensus pattern in one set of proteins which interact with (almost) all proteins which contain another consensus pattern. If so, both patterns are likely to represent a part of the surface of the molecules which makes interactions possible through a physical binding. For instance, in Figure 1 the patterns $\{1, A\}$ and $\{2, B\}$ represent two such correlated motifs. In particular, there is an undirected edge between two protein sequences when the first one contains motif 1 (resp., 2) and the second one motif A (resp., B).

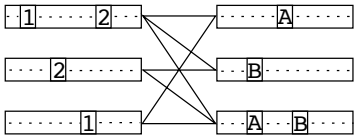

Figure 1. Compatible binding sites $1, A$ and $2, B$ as correlated motifs in sequences

These methods can be subdivided into two classes: (1) interaction-driven [8, 9, 10], and (2) motif-driven approaches [13, 7]. Interaction-driven methods mine for (quasi) bicliques, that is, disjoint subsets of vertices for which every vertex from one set is connected to (almost) all vertices of the other set. Such subgraphs exhibit a type of all-versus-all (or most-versus-most) interaction. A motif pair representing the corresponding interaction sites is then derived from the sequence carried by the vertices. The motif-driven approach, in contrast, starts from candidate motif pairs whose support is then evaluated in the network. Although both approaches have shown to produce biologically meaningful results, the second approach has several conceptual advantages over the first: (i) motif pairs are mined directly, not derived; (ii) *all* proteins containing one of the motifs, and not a subset, are taken into account; (iii) if the the interactions between two sets of proteins is a consequence of multiple compatible binding sites, such as $\{1, A\}$ and $\{2, B\}$ in Figure 1, the interaction-driven method neces-

sarily merges them into one motif pair; and, (iv) all interactions of proteins having both binding sites described by the motif pair are taken into account, i.e., the subsets containing each motif do not have to be disjoint.

In this article, we study different aspects of the motif-driven approach towards CMM for which currently only two techniques have been introduced and implemented. Unfortunately, both methods differ not only in the mining method but also in the used notion of support for correlated motifs. The first method by Tan et. al [13], called D-STAR, uses a χ^2-based scoring function to determine the support but the underlying mining method does not scale to networks containing more than 250 proteins. As contemporary biological networks contain upto thousands proteins (for instance the protein-protein interaction networks of yeast and human [4]), scalability is an increasingly important issue. The second method called MotifHeuristics employs a different, probabilistically motivated notion of support called p-score, is developed by Leung et al. [7] and does scale to larger networks. Although the authors argue in their paper that MotifHeuristics is superior to D-STAR, it remains unclear if the latter is due to the different support measure or the underlying mining method. Moreover, an in depth study of support measures *as such* has never been undertaken.

A first contribution of this paper is a thorough empirical study of the effectiveness of various notions of support for correlated motifs. We evaluate them in terms of precision and recall on artificial networks with implanted motifs at different noise levels. These experiments clearly show that the χ^2-based support measure is vastly superior in discovering highly interaction descriptive motif pairs.

As a second contribution, we formally prove that, under reasonable assumptions concerning the used notion of support, the complexity of the correlated motif mining problem is NP-hard and its associated decision problem is in NP. We therefore approach the problem as a combinatorial optimization problem.

More specifically, as the third and main contribution of this work, we present SLIDER, a local seach method in which the key component is its neighborhood function which views a motif as a window which slides over the amino acid sequences of the proteins. In contrast with more common neighborhood functions, it has a clear biological interpretation: it is based on the philosophy that if a motif overlaps with part of a binding site in a sequence it should be able to slide towards the binding site in a few steps. Although SLIDER can be used with an arbitrary support measure, we use the χ^2-based support measure, as the empirical study in the first contribution of this paper clearly indicates this is the best support measure known so far.

We validate SLIDER by showing that it outperforms all existing motif-driven approaches on retrieving implanted motif pairs from artificial networks. Furthermore our experiments show that SLIDER is able to tackle CMM on large protein-protein interaction networks.

Outline. In Section 2, we formally define CMM and in Section 3 we discuss support measures. In Section 4, we prove CMM to be NP-hard for a large class of support measures. In Section 5, we introduce the novel method SLIDER. In Section 6, we introduce our artificial and biological datasets on which our novel method SLIDER is assessed in Section 7. We discuss related work in Section 8 and conclude in Section 9.

2 Correlated motif mining problem

We model a protein-protein interaction (PPI) network by an undirected labeled graph $G = (V, E, \lambda)$ in which the vertices V correspond to the proteins, the edges E to the interactions and the labels of the vertices to the amino acid sequence of the proteins. Hence, the label function λ maps each vertex $v \in V$ to a string $\lambda(v)$ over the alphabet $\Sigma = \{\mathtt{A}, \ldots, \mathtt{Z}\} \setminus \{\mathtt{B}, \mathtt{J}, \mathtt{O}, \mathtt{U}, \mathtt{X}, \mathtt{Z}\}$. We ignore self-interactions. Although biologically relevant, it is sometimes undesirable to take self-interactions into account as in some cases they are not easily or not at all detectable. For example, when yeast two-hybrid is used, a homodimeric interaction can obviously only be recovered if the protein that forms it is present both as bait and as prey in the screen. In addition, it has been reported that in some cases yeast two-hybrid has an inherent lower efficiency to detect homodimeric interactions [11].

An (ℓ, d)-*motif* is a string of length ℓ over the alphabet $\Sigma \cup \{\mathtt{x}\}$ containing exactly d $\mathtt{x}$-characters. The character $\mathtt{x}$ is interpreted as a wildcard-symbol, i.e., it matches with any character of Σ. For instance, $\mathtt{GAQPRNMY}$ matches the $(8, 4)$-motif $\mathtt{GxxPxNxY}$.

A protein *contains* an (ℓ, d)-motif X if its amino acid sequence contains a substring of length ℓ that matches X. Note that motifs starting and ending with a wildcard character are redundant because, in practice, the amino acid sequences are much longer than the motifs.

Given an (ℓ, d)-motif X and a PPI-network $G = (V, E, \lambda)$, let $V_X = \{v \in V \mid \lambda(v) \text{ contains } X\}$, be the set of proteins in the network containing the motif X, and

$$E_{X,Y} = \{\{u, v\} \in E \mid u \in V_X \wedge v \in V_Y\}$$

be the set of interactions between proteins containing X and proteins containing Y. Hence, the subgraph $G_{X,Y}$ *selected* by a motif pair $\{X, Y\}$ is then

$$G_{X,Y} = (V_X \cup V_Y, E_{X,Y}, \lambda_{|V_X \cup V_Y})$$

with $\lambda_{|V_X \cup V_Y}$ the restriction of λ to $V_X \cup V_Y$. Note that V_X and V_Y can share proteins.

A *support measure* f is simply a function mapping a motif pair $\{X, Y\}$ and a graph G to a postive real number $f(\{X, Y\}, G)$. We refer to $f(\{X, Y\}, G)$ as the *support* of $\{X, Y\}$ in G. In Section 3 and 7.2 we discuss and compare several instantiations of support measures.

We next formulate the correlated motif pair mining problem in a PPI-network (Correlated Motif Mining, CMM):

- **Input:** a PPI-network $G = (V, E, \lambda)$, three natural numbers ℓ, d, k and a support measure f mapping a motif pair $\{X, Y\}$ and a graph G to a real positive number $f(\{X, Y\}, G)$.
- **Output:** the k (ℓ, d)-motif pairs $\{X_1, Y_1\}$, ..., $\{X_k, Y_k\}$ with highest support in G with respect to f.

3 Support measures

Support measures should reflect the power of a motif pair to describe interactions. Several considerations should be taken into account in deciding how to measure the descriptive power of a motif pair for a given PPI-network $G = (V, E, \lambda)$: (i) $E_{X,Y}$ should be significantly larger than expected given G, V_X and V_Y; and, (ii) V_X and V_Y should be large enough in order to minimize the likelihood that the appearance of the motif X respectively Y in the sequences of the proteins in V_X respectively V_Y is just by chance.

In other words, we want the motifs X and Y to truly represent an overrepresented consensus pattern in the sequences of the proteins in V_X respectively V_Y in order to increase the likelihood that they correspond to, or at least overlap with, a so called *binding site* – a part of the molecule on the surface that makes interactions between proteins from V_X and V_Y possible through a molecular lock-and-key mechanism.

Before we discuss support measures in detail, we need some more concepts from graph theory. A graph is called *complete* if any two vertices are connected by an edge. A complete subgraph on k vertices is a k-*clique*. A *bipartite graph* is a graph for which the vertex set can be partitioned into two disjoint sets B and W such that each edge connects a vertex of B with a vertex of W. It is called *balanced* if $|B| = |W|$ and *complete* if each vertex of B is connected to each vertex of W. A complete bipartite subgraph is called a *biclique*.

We call $\{X, Y\}$ a $(k_X, k_Y, k_{X,Y})$-motif pair for a PPI-network $G = (V, E, \lambda)$ if $|V_X| = k_X, |V_Y| = k_Y$ and $|V_X \cap V_Y| = k_{X,Y}$. We call it *complete* if all vertices from V_X are connected with all vertices from V_Y. Clearly, a complete $(k_X, k_Y, k_{X,Y})$-motif pair is an ideal candidate provided that k_X and k_Y are sufficiently large. In that case, $G_{X,Y}$ is a subgraph of G that is a combination of a biclique with parts $V_X \setminus V_Y$ and V_Y (or V_X and $V_Y \setminus V_X$) and a $k_{X,Y}$-clique ($V_X \cap V_Y$). If $k_{X,Y} = 0$, it is a pure biclique and if $k_{X,Y} = k_X = k_Y$ it is a pure clique. Figure 2 shows an example. We point out that the methods in [8, 9, 10] search for

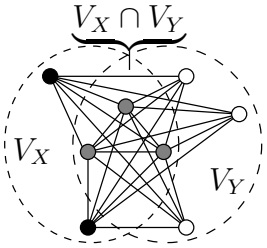

Figure 2. An example of a network selected by a complete (5,6,3)-motif pair.

(quasi-)bicliques where $V_X \cap V_Y$ is always the empty set. As such, the maximal number of edges any $(k_X, k_Y, k_{X,Y})$-motif pair can have in any PPI-network is

$$M(k_X, k_Y, k_{X,Y}) = \left(k_X k_Y - \binom{k_{X,Y}}{2} - k_{X,Y} \right) .$$

3.1 A χ^2-based support measure

Tan et al. [13] introduced the χ^2-score for statistical significance as a support measure for CMM:

$$f_{\chi^2}(\{X, Y\}, G) = \begin{cases} \frac{(|E_{X,Y}| - \overline{E_{X,Y}})^2}{\overline{E_{X,Y}}} & \text{if } |E_{X,Y}| > \overline{E_{X,Y}} \\ 0 & \text{if } |E_{X,Y}| \leq \overline{E_{X,Y}} \end{cases}$$

with $\overline{E_{X,Y}}$ the expected number of interactions between V_X and V_Y. The value $\overline{E_{X,Y}}$ is calculated by assuming a uniform *density* of edges. To that end, let $\mathrm{ed}(G)$ be the *edge density* of G, i.e., the proportion of edges it has of all its potential edges. Then,

$$\overline{E_{X,Y}} = \mathrm{ed}(G) M(|V_X|, |V_Y|, |V_X \cap V_Y|) ,$$

with $\mathrm{ed}(G) = |E| / \binom{|V|}{2}$.

If we also use the edge density of the selected subnetwork $\mathrm{ed}(G_{X,Y}) = |E_{X,Y}| / M(|V_X|, |V_Y|, |V_X \cap V_Y|)$ we can rewrite the χ^2-support of $\{X, Y\}$ for which $|E_{X,Y}| > \overline{E_{X,Y}}$ as

$$f_{\chi^2}(\{X, Y\}, G) = M(|V_X|, |V_Y|, |V_X \cap V_Y|) \frac{(\mathrm{ed}(G_{X,Y}) - \mathrm{ed}(G))^2}{\mathrm{ed}(G)} .$$

As $\mathrm{ed}(G)$ is a constant for a fixed PPI-network, we clearly see in this form that f_{χ^2} uses two criteria to determine the support of a motif pair $\{X, Y\}$:

1. the difference in edge density of $G_{X,Y}$ and G, which rewards a larger $E_{X,Y}$ than expected; and
2. the (potential) size of $G_{X,Y}$ in terms of the number of edges, which rewards larger V_X and V_Y.

3.2 p-score: a probabilistic support measure

The p-score is a measure introduced by Leung et al. [7] to evaluate the statistical significance of a motif pair $\{X, Y\}$ in a PPI-network $G = (V, E, \lambda)$ by estimating the conditional probability that there are $|E_{X,Y}|$ or more interactions between V_X and V_Y given the number of interactions involving V_X and assuming a uniform distribution of interactions over all interaction partners. Motif pairs for which this probability is small are considered to be statistically significant.

More formally, given a motif pair $\{X, Y\}$ and a PPI-network $G = (V, E, \lambda)$, let $N(V_X) = \{v \mid \exists x \in V_X : \{v, x\} \in E\}$, i.e., the set of all vertices connected with a vertex from V_X, and $E_X = \{\{u, v\} \in E \mid u \in V_X\}$, the set of interactions involving vertices from V_X.

The probability p_X that there are $|E_{X,Y}|$ interactions between V_X and V_Y given $V_X, V_Y, N(V_X)$ and E_X is estimated by (see [7] for details)

$$p_X = \sum_{i=|E_{X,Y}|}^{E_{X,Y}^{\max}} \frac{\binom{i-1}{|N(V_X)\cap V_Y|-1}\binom{|E_X|-i-1}{|N(V_X)\setminus V_Y|-1}}{\binom{|E_X|-1}{|N(V_X)|-1}}$$

where

$$E_{X,Y}^{\max} = \min(|E_X| - |N(V_X) \setminus V_Y|, |V_X||N(V_X) \cap V_Y|)$$

represents the maximal possible size of $E_{X,Y}$. The idea is that p_X is a good estimator for the conditional probability of $|E_{X,Y}|$ or more interactions between V_X and V_Y given $V_X, N(V_X), E_X, V_Y, N(V_Y)$ and E_Y if $|E_{X,Y}|/\overline{E_{Y\to X}}$ is small, with

$$\overline{E_{Y\to X}} = (|E_Y|/|N(V_Y)|)|N(V_Y) \cap V_X|$$

the expected number of interactions between V_Y and $N(V_Y) \cap V_X$ given $V_Y, N(V_Y), E_Y$ and V_X. Of course, similar formulas can be obtained for p_Y and $\overline{E_{X\to Y}}$ and the p-score based support measure f_p uses the best of both estimators:

$$f_p(\{X, Y\}, G) = \begin{cases} 1 - p_X & \text{if } \overline{E_{Y\to X}} \geq \overline{E_{X\to Y}} \\ 1 - p_Y & \text{if } \overline{E_{Y\to X}} < \overline{E_{X\to Y}} \end{cases}$$

3.3 Comparison of f_{χ^2} and f_p

Comparing f_p with f_{χ^2}, a major difference is that f_{χ^2} bases its support on the whole network G, while f_p-support is based on the statistical significance of a motif pair $\{X, Y\}$ in two subnetworks of the whole PPI-network: $G_X = (V_X \cup N(V_X) \cup V_Y), E_X)$ and $G_Y = (V_Y \cup N(V_Y) \cup V_X), E_Y)$. Moreover, besides the typical edge distribution assumption, f_p makes implicitly the following additional assumptions:

1. V_X and V_Y are disjoint;
2. every interaction from E_X (E_Y) can be described using X (Y), thus to calculate the support of $\{X, Y\}$ each protein is assumed to have only one binding site.

Finally, we stress a design flaw in the definition of f_p: the approximation p_X becomes less precise when $|E_{X,Y}|/\overline{E_{X\to Y}}$ becomes larger. But the latter happens precisely when the selected subgraph contains more edges than expected, that is, becomes more interesting. In addition to the just mentioned weaknesses, our experiments in Section 7.2 confirm that f_p is inferior to f_{χ^2} in recovering implanted correlated motifs at different noise levels.

4 Complexity of CMM

We will prove that CMM is NP-hard when f_{χ^2} is used as support measure. However, in order to make the result as broadly applicable as possible, we will prove the NP-hardness of CMM for a whole class of support measures and show at the end of the section that f_{χ^2} is a member of that class.

We restrict ourselves to support measures which abide to three reasonable conditions. Informally, the first condition says that the support can be computed efficiently, the second that if the topology of the selected subnetworks of two motif pairs differ only in the number of edges, the one which covers more interactions has higher support. Finally, the last condition states that the support of a complete motif pair increases with the size of the selected subnetwork.

To that end, let $G = (V, E, \lambda)$ be any PPI-network and let $M_{k_X,k_Y,k_{X,Y}}$ be a complete $(k_X, k_Y, k_{X,Y})$-motif pair for G, $k_{X,Y} \leq \min(k_X, k_Y)$. We call a support measure f *compliant* if the following conditions hold for f:

1. f is polynomial time computable in the size of G,
2. for any two $(k_X, k_Y, k_{X,Y})$-motif pairs $\{X, Y\}$, $\{X', Y'\}$ in G:

$$f(\{X, Y\}, G) = 0 \vee \Big(f(\{X, Y\}, G) > f(\{X', Y'\}, G) \iff |E_{X,Y}| > |E_{X',Y'}|\Big).$$

3. for $0 < i \leq k_X - k_{X,Y}$ and $0 < j \leq k_{X,Y}$:

$$f(M_{k_X,k_Y,k_{X,Y}}, G) > f(M_{k_X-i,k_Y,k_{X,Y}}, G) \wedge f(M_{k_X,k_Y,k_{X,Y}}, G) > f(M_{k_X-j,k_Y,k_{X,Y}-j}, G).$$

Remark that, because the f_p-support of a motif pair $\{X, Y\}$ in a PPI-network G depends also on the neigborhood of the selected subnetwork $G_{X,Y}$ in G (G_X and G_Y), it will not always abide to the last two conditions.

We call a support measure f *biclique-maximal* if:

$$f(M_{k,k,0}, G) > f(M_{k,k,k'}, G), \quad 0 < k' \leq k$$

and *clique-maximal* if:

$$f(M_{k,k,k}, G) > f(M_{k,k,k'}, G), \quad 0 \leq k' < k\ .$$

We will now show that CMM is NP-hard by proving that even a simplified version of the associated decision (D) problem is already NP-complete. Let D-CMM be the problem to decide whether for a given PPI-network $G = (V, E, \lambda)$, natural numbers ℓ, d, a real number t and a support measure f, there exists an (ℓ, d)-motif pair $\{X, Y\}$ for which $f(\{X, Y\}, G) \geq t$.

Theorem 1 D-CMM *is* NP*-complete for any clique- or biclique-maximal compliant support measure* f.

Proof: D-CMM is obviously in NP: a motif pair M for which $f(M, G) \geq t$ can serve as polynomial time verifiable certificate.

Given a graph $G = (V, E)$ and a natural number k, deciding whether G contains a k-clique is called the *clique* problem. Similarily, deciding whether G contains a biclique such that both parts are of size k, is called the *balanced complete bipartite subgraph problem* (BCBS). Both problems are known to be NP-complete and BCBS even when restricted to bipartite graphs [5].

We will show that D-CMM is NP-complete for biclique- respectively clique-maximal support measures by reducing BCBS respectively CLIQUE to D-CMM.

So, given a graph $G = (V, E)$, with $V = \{v_1, \ldots, v_n\}$, the reduction transforms G into a labeled graph $G' = (V, E, \lambda)$. For convenience, we will use the alphabet $\Sigma = \{0, 1\}$ and label the vertices of G' as follows: $\lambda(v_i) = w_1^i \ldots w_n^i$, with $w_i^i = 1$ and $w_j^i = 0$, for $j \neq i$.

The labels of the vertices are chosen in such a way that for any (n, k)-motif X, $|V_X| \in \{0, 1, k\}$. Indeed, we can discriminate the following cases:

1. if X contains at least two 1's then $V_X = \emptyset$;
2. if X contains a 1 at position i and all other non-wildcard symbols are 0 then $V_X = \{v_i\}$; and,
3. if X contains only wildcard symbols and 0's then $v_i \in V_X$ if the symbol at position i is a wildcard symbol.

As such, every motif pair in G' is necessarily a $(1, k, k')$-, $k' \in \{0, 1\}$, or a (k, k, k')-motif pair, $0 \leq k' \leq k$. Moreover, for an (n, k)-motif X containing only 0's and wildcard symbols, v_i will be in V_X if and only if position i of X is a wildcard symbol. In other words, for any subset $W \subseteq V$ of size k, we can choose an X such that $V_X = W$. Consequently, if $\{X, Y\}$ is a motif pair for which $|V_X| = |V_Y|$, $V_X \cap V_Y = \emptyset$ and $|E_{X,Y}| = M(|V_X|, |V_Y|, 0)$, then $(V_X \cup V_Y, E_{X,Y})$ is a balanced complete bipartite graph. Similarily, if $V_X = V_Y$ and $|E_{X,Y}| = M(|V_X|, |V_X|, |V_X|)$ then $(V_X, E_{X,Y})$ is a k-clique.

Let $M_{k_X,k_Y,k_{X,Y}}$ be a complete $(k_X, k_Y, k_{X,Y})$-motif pair for G', $k_{X,Y} \leq \min(k_X, k_Y)$ and $k \geq 2$.. We know that f is compliant. If f is biclique-maximal it holds that:

$$f(M_{k,k,0}, G') > f(M_{1,k,0}, G') \\ \wedge f(M_{k,k,0}, G') > f(M_{k,k,k}, G') > f(M_{1,k,1}, G')$$

and if f is clique-maximal we have:

$$f(M_{k,k,k}, G') > f(M_{1,k,1}, G') \\ \wedge f(M_{k,k,k}, G') > f(M_{k,k,0}, G') > f(M_{1,k,0}, G')\ .$$

Thus, G contains a balanced complete bipartite subgraph with both parts of size k, if and only if there exists an (n, k)-motif pair $\{X, Y\}$ for which

$$f(\{X, Y\}, G') \geq t = f(M_{k,k,0}, G')$$

with f a biclique-maximal support measure. By the same reasoning, G contains a k-clique if there exists an (n, k)-motif pair $\{X, Y\}$ for which

$$f(\{X, Y\}, G') \geq t = f(M_{k,k,k}, G')$$

with f a clique-maximal support measure.

The proof is complete by noting that the transformation of G into G' and the calculation of t can be done in polynomial time. □

It is easy to see that f_{χ^2} is compliant and biclique-maximal. Indeed, for fixed k, the support for a complete $(k, k, k_{X,Y})$ motif pair $\{X, Y\}$ in PPI-network G is

$$M(k, k, k_{X,Y}) \frac{(1 - \mathrm{ed}(G))^2}{\mathrm{ed}(G)}\ ,$$

which is maximal for $k_{X,Y} = 0$.

5 SLIDER

Since the decision problem associated with CMMis in NP, CMM can be tackled efficiently as a search problem in the space of all possible (ℓ, d)-motif pairs. If we add the assumption that similar motifs can be expected to get similar support, it has the typical form of a *combinatorial optimization problem*. In combinatorial optimization, the objective is to find a point in a discrete search space which maximizes a user-provided function f. A number of heuristic algorithms called *meta-heuristics* are known to yield good solutions to a wide variety of combinatorial optimization problems.

One such meta-heuristic is *local search* [1]. Local search algorithms move from the current point to a neighboring point in the space of candidate solutions until a local optimal solution is found, i.e., a solution that maximizes f in its neighborhood. Hence, to apply local search one needs to define a neighborhood function which returns the neighbor points of each point in the search space. The neigborhood function is a key component of the local search method and has to be chosen carefully and fine-tuned for the problem at hand. The initial points from where local search is started are typically a combination of randomly and heuristically chosen points. In the related works section, we discuss other meta-heuristics and explain the choice for local search.

The main idea behind local search for CMM is illustrated by the pseudo-code in Algorithm 1. For reasons of clarity, we use an abstract neigborhood function N, an abstract support measure f and focus on the case in which only the best pair is returned ($k = 1$). In practice, we accumulate the best results found over several runs with $1\,000 \leq k \leq 10\,000$ in a single execution.

Algorithm 1 Local Search Algorithm (LSA) for CMM, with neighborhood function N and support measure f

Require: PPI-network $G = (V, E, \lambda)$, $\ell, d \in \mathbb{N}$, $d < \ell$
Ensure: $\{X^*, Y^*\}$ best correlated motif pair found in G

```
1:  {X*, Y*} ← randomOrHeuristicMotifPair()
2:  maxsup ← f({X*, Y*}, G)
3:  sup ← 0
4:  while maxsup > sup do
5:    {X, Y} ← {X*, Y*}
6:    sup ← maxsup
7:    for all {X', Y'} ∈ N({X, Y}) do {scan neighborhood}
8:      if f({X', Y'}, G) > maxsup then
9:        {X*, Y*} ← {X', Y'}
10:       maxsup ← f({X', Y'}, G)
```

Thus, in order to apply local search to CMM, we need to define a neighborhood function N which maps a motif pair $\{X, Y\}$ to its neighbors $N(\{X, Y\})$ in the space of all motif pairs. Consider a motif pair $\{X, Y\}$ and the selected subnetwork $G_{X,Y}$. Ideally, the subnetwork $G_{X',Y'}$ selected by a neighbor $\{X', Y'\} \in N(\{X, Y\})$ should also be "close" to $G_{X,Y}$ in the sense that at least some proteins and interactions should be shared between $G_{X,Y}$ and $G_{X',Y'}$.

To that end, we first define a neighbor function N^{slide} on motifs, which will the basis for a neighbor function on motif pairs. Looking for a match of an (ℓ, d)-motif X in a protein can be seen as sliding a window of length ℓ with $\ell - d$ holes over the sequence until the characters in the holes match the non-wildcard caracters of X. Hence, a motif X' obtained by closing a hole on a matching substring and creating a new one while respecting the window size ℓ, guarantees that

Figure 3. Two neighboring $(6, 3)$-motifs as sliding windows on a sequence. Moving from `RTxTxx` to `KxxTxT`, shifts the window to the left.

the same protein will contain X'. In this way, we can slide the motif window to the left or right by punching the new hole before the first or after the last original character, as illustrated in Figure 3 and formally defined next.

For a motif X, denote by $\text{trim}(X)$, the motif obtained from X by removing leading and trailing wildcards. That is, $\text{trim}(\texttt{xTxTxx}) = \texttt{TxT}$. A motif $X' \in N^{\text{slide}}(X)$ if X and X' have the same length and $\text{trim}(Y) = \text{trim}(Y')$ where Y (resp., Y') is obtained from X (resp., X') by changing one non-wildcard character into a wildcard. In Figure 3, X equals `RTxTxx` while X' equals `KxxTxT`. Now, $X' \in N^{\text{slide}}(X)$ as X (resp., X') can be transformed into $Y = \texttt{xTxTxx}$ (resp., $Y' = \texttt{xxxTxT}$) by changing one non-wildcard character into a wildcard and Y equals Y' after stripping leading and trailing wildcards. Next, we define N^{slide} for motif pairs. That is, $\{X', Y'\} \in N^{\text{slide}}(\{X, Y\})$ if $X' \in N^{\text{slide}}(X) \wedge Y' = Y$ or $Y' \in N^{\text{slide}}(Y) \wedge X' = X$. Note that when applying N^{slide} to pairs of motifs, one of the motifs remains fixed. Our experiments reported in Section 7.3, show that fixing one motif at each step greatly improves the effectiveness.

We are now ready to define our novel CMM-method SLIDER:

Definition 1 We define the method SLIDER as LSA with
(i) neighbor function N^{slide};
(ii) support measure f_{χ^2}; and,
(iii) random starting seeds.

6 Datasets

Artificial data. To evaluate the biological relevance of the different notions of support and the power of heuristic methods to retrieve the best motif pairs in terms of describing interactions, we created a number of artificial networks as follows. Each network is composed of 100 proteins which are randomly chosen out of the well-known yeast network [4]. We then generate 50 random $(8, 3)$-motifs[1] and implant 3 to 10 instances of each motif in the sequences of randomly chosen proteins. Then, we implant motif pairs by randomly selecting two implanted motifs X and Y and connecting each protein containing X with each protein containing Y

[1] Using entropy analysis, research has shown that the highest amount of structural information per sequence length can be found in subsequences of length 7 to 9 (see Figure 1 in [16]).

until a chosen minimal edge density is obtained – we used 0.1, 0.2 and 0.3. Consequently, the network obtained is perfect in the sense that each interaction is a direct consequence of an implanted motif pair. Because noise and missing data is an important factor in biological networks, we also evaluate the resistance to noise of both the support measures and heuristic methods. To that end, we also create "diluted" versions of each network, by choosing a certain dilution level a (from 0.05 to 0.3 in steps of 0.05) and flip the edge relation of each pair of vertices with probability a.

We restrict ourselves to networks of 100 proteins because this is more or less the maximum size for which we are still able to mine the motif pairs with highest support for each support measure by a brute force computation within a reasonable time frame.

Biological data. To assess the effectiveness of SLIDER on larger networks, we ran our method on the high-confidence protein-protein interaction network of yeast [4] consisting of 1620 nodes and 9060 interactions. It is very difficult to measure the biological significance of the found correlated motifs, because only very few of them are actually known. Therefore, we executed a brute force CMM-algorithm over the yeast network on a computer cluster, finding the best 1 000 correlated motifs according to f_{χ^2} and compared these to the results returned by SLIDER. The brute force computation occupied about 100 nodes in the cluster spanning a period of 2 weeks. Its purpose was to create a baseline for motif-drive CMM-algorithms as well as collecting the best correlated yeast motifs for biological analysis (which is still ongoing at this point).

7 Experiments

With the exception of the brute force run on yeast, all experiments were run on a 3GHz Mac Pro with 4GB of RAM and 8 cores. In the sequel, whenever a timing is mentioned and unless explicitly mentioned otherwise, the experiment was run using only 1 core. Nevertheless, we stress that our SLIDER-prototype, implemented in Java, can use as many processors as are available. In this section, we experimentally assess the effectiveness of (1) support measures to assign a support to a motif pair reflecting its power to describe interactions; and, (2) neighbor functions to find the motif pairs with highest support by exploring the space of all motif pairs. Furthermore, we compare SLIDER with other motif-driven CMM-methods. To this end, we need a notion of precision that compares an ordered set of motif pairs versus a set of motif pairs which is considered to be the "ground truth". Finally, we assess the effectiveness of SLIDER on the yeast network.

7.1 Precision for motif pairs

Before we define our notion of precision, we need a similarity measure on motif pairs. We define the similarity between an (ℓ, d)-motif pair $\{X, Y\}$ and $\{X', Y'\}$ in a PPI-network $G = (V, E, \lambda)$ as

$$s(\{X, Y\}, \{X', Y'\}, G) = \frac{|E_{X,Y} \cap_{pos} E_{X',Y'}|}{|E_{X,Y} \cup E_{X',Y'}|}$$

where $\{v, w\} \in E_{X,Y \cap_{pos} X',Y'}$ if there exists substrings s_v and s'_v in $\lambda(v)$ and substrings s_w and s'_w in $\lambda(w)$ such that s_v (resp., s_w) matches with X (resp., Y), s'_v (resp. s'_w) matches with X' (resp., Y'), and, s_v and s'_v as well as s_w and s'_w overlap in at least $\lceil \ell/3 \rceil$ positions in $\lambda(v)$ respectively $\lambda(w)$.

Let $S = \{M_1, \ldots, M_n\}$ be a list of motif pairs, then we reduce S by deleting for every j from 1 to n, every M_i for $i > j$ such that $s(M_i, M_j) \geq 0.9$. We denote the reduced version of S by S^*.

Let T be a set of "ground truth" (ℓ, d)-motif pairs and let $S = \{M_1, \ldots, M_n\}$ be a list of (ℓ, d)-motif pairs to be compared against T. We define the precision of S against T at rank k as the fraction of motif pairs M_i in S^*, $1 \leq i \leq k$ for which there exists a motif pair M_T in T^* such that $s(M_i, M_T) \geq 0.9$. We note that, when $k = |T^*|$, the precision as defined above also corresponds to the usual notion of recall.

7.2 Evaluation of support measures

We start by assessing the effectiveness of support measures in assigning a support to a motif pair reflecting its power to describe interactions. Since the most describing motif pairs in real PPI-networks are unknown, we measure the ability of a support measure to assign the highest support to motif pairs on artificial networks with implanted motifs, as described in Section 6. We used a collection of networks G_e^a with edge density $e\%$ and dilution level $a\%$. We compare the support measures by looking at the precision of implanted motif pairs on G_e^a at rank m, where m equals the number of implanted motif pairs. Remark that, in this setting, recall and precision are the same.

In order to make sure that the f_{χ^2} and f_p assign a meaningful support, we also implemented two naive support measures f_c and f_v. The f_c-support in a PPI-network $G = (V, E)$ is simply the number of interactions covered: $f_c(\{X, Y\}, G) = |E_{X,Y}|$ and

$$f_v(\{X, Y\}, G) = \frac{|E_{X,Y}|}{M(|V_X|, |V_Y|, |V_X \cap V_Y|) + |V_X \cup V_Y|}.$$

Both measures are naive in the sense that they are independent of the interaction distribution in G. It is straightforward to show that both measures are compliant, thus meeting the basic requirements of a support measure. Moreover, they are biclique-maximal.

A visual inspection of the graphs in Figure 4 already indicates that f_{χ^2} globally outperforms the other support measures in selecting motif pairs describing actual interactions. Indeed, at every data point, the precision of f_{χ^2} is the best

value or very close to the best value of the four support measures considered. Moreover, comparing precision on diluted networks shows that f_{χ^2} is vastly more robust to noise — a crucial aspect since contemporary PPI-networks still contain large amounts of both noise and missing data [15].

Thus, we can conlude this experimental section that f_{χ^2} is superior to all other support measures considered on all merits.

7.3 Evaluation of neighborhood functions

We will now confirm that our neighborhood function which is based on a sliding window interpretation on the sequences is superior to the standard neighbor functions which simply define small perturbations to explore the search space.

In particular, we define the following perturbations: letter change (LC, replace one non-wildcard character by another); swap adjacent (SA, swap an adjacent wildcard and non-wildcard character); and, swap (S, swap an arbitrary wildcard and non-wildcard character). We denote neighborhood functions combining these perturbations by concatenating their abbreviations with boolean operators. For instance, LCandSA denotes the neighborhood function which requires a letter change *and* a swap adjacent perturbation. Finally, we consider a simple version of N^{slide}, denoted SimpleSlide, which only allows to change non-wildcard characters into wildcard ones at opposing ends of the motif. The corresponding neighborhood functions on pairs of motifs are defined similarly to N^{slide}: one motif is kept fixed, while the other is replaced by a neighbor.

Figure 6 displays the precision of LSA with each of these neighborhood functions on the implanted network of density 10%. The displayed precision is averaged over 5 LSA runs. As the speed of LSA is highly dependent on the chosen neighbor function, we provided each run the same amount of time (10 minutes). In this way, faster neighborhood functions like LCorSA can process more seeds than slower functions like N^{slide} (cf. Figure 5). As can be seen from Figure 6, Slide, and thereby SLIDER, outperforms the other neighbor functions.

For the sake of completeness, we also experimented with neighborhood functions on motif pairs where both motifs can be replaced with a neighboring one (in contrast to the previous neighborhood functions where one is kept fixed). Unfortunately, the precision was never larger than 10%, independent of the level of dilution.

7.4 Comparison with existing methods

D-STAR. Tan et al. introduced the first motif-driven method for CMM: D-STAR [13]. In contrast with our approach, D-STAR uses (ℓ, d)-motifs in the *mismatch model*. In the mismatch model, an (ℓ, d)-motif is simply a string s of length ℓ and an amino acid sequence is said to contain the (ℓ, d)-motif s if it contains a substring of length ℓ that differs in at most d characters from s. D-STAR is based on the observation that two strings s_1 and s_2 which both differ at most d characters from s, differ at most in $2d$ characters from each other. Strictly spoken, D-STAR does not deliver (ℓ, d)-motifs. Instead it returns two strings s_X and s_Y, and two sets of proteins V_X and V_Y together with the indices of the substring of the amino acid sequence of each protein in V_X (respectively V_Y) that differs at most $2d$ characters from s_X (respectively s_Y). To construct the $\{V_X, V_Y\}$-pairs, D-STAR considers for each interaction $\{v, w\}$, each substring of length ℓ in $\lambda(v)$ and $\lambda(w)$ as the initial strings s_X and s_Y, determines V_X and V_Y, and evaluates $\{V_X, V_Y\}$ using f_{χ^2}. As the similarity in Section 7.1 is defined in terms of positions of substrings, we can directly use the returned subsets V_X and V_Y to compare with implanted motifs. Every run of D-STAR on the same network produced the same result, consequently the running time of D-STAR cannot be parameterized. We used the D-STAR implementation freely available on the web.

MotifHeuristics. Another method, called MotifHeuristics, proposed by [7], derives (ℓ, d)-motifs directly within the wildcard model and introduced the probabilistically motivated f_p-support. Although the authors do not describe it as such, MotifHeuristics can be seen as a local search method in which the neighbors of a motif-pair $\{X, Y\}$ are all motif pairs $\{X, Y'\}$ at odd steps and all motif pairs $\{X', Y\}$ at even steps. Because we could not obtain an implementation of MotifHeuristics, we implemented our own version based on the algorithmic description in [7].

Comparison. The graph in Figure 7 depicts the precision of the various methods on the artificial network of density 10%. As a naive baseline, we ran the method Random, evaluating random motif pairs using f_{χ^2}. D-STAR took 5 minutes to finish. We gave Random and SLIDER 10 minutes of computation time. In order to give our unoptimized implementation of MotifHeuristics a fair chance, we allowed it to run 30 times longer than SLIDER (that is 5 hours). The underlying reason why MotifHeuristics takes such a long time is that for every search step a number of supports has to be calculated which approaches the total number of motifs. The graph makes it quite apparent that the succes rates of both D-STAR and MotifHeuristics are smaller than or equal to that of SLIDER. Moreover, while the succes rate of SLIDER is consistent with the level of dilution, this is not the case for D-STAR and MotifHeuristics. Overall, SLIDER is more effective and more robust than its competitors. All algorithms perform significantly better than random search.

When we double the execution time of SLIDER to 20 minutes, the precision increases significantly. The latter execution time is still minor in comparison with the brute force computation which takes about 40 hours.

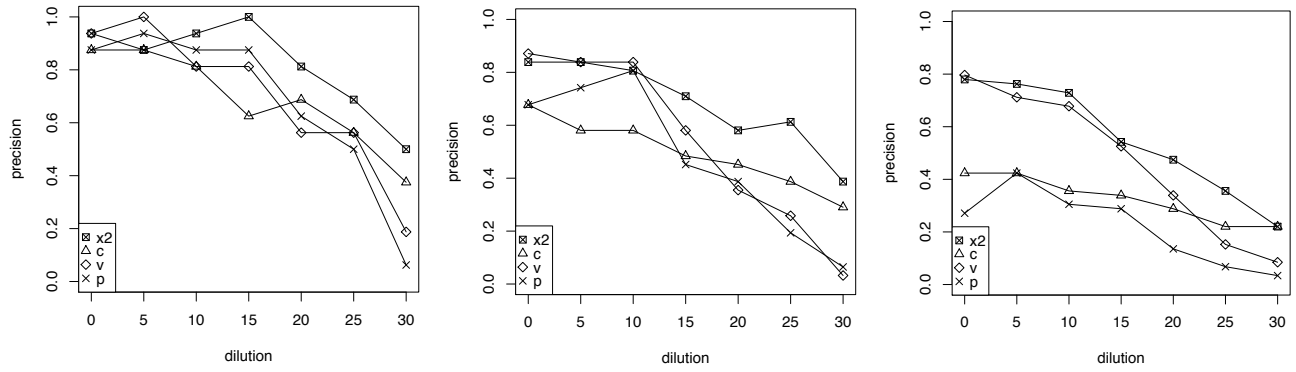

Figure 4. Precision of support measures on artificial networks with implanted motif pairs and different edge densities (10%, 20%, 30%).

Neighbor func.	seeds
Slide	355K
LCandS	1107K
SimpleSlide	3143K
LCandSA	3867K
LCorSA	4222K

Figure 5. Total amount of starting seeds for each neighbor functions.

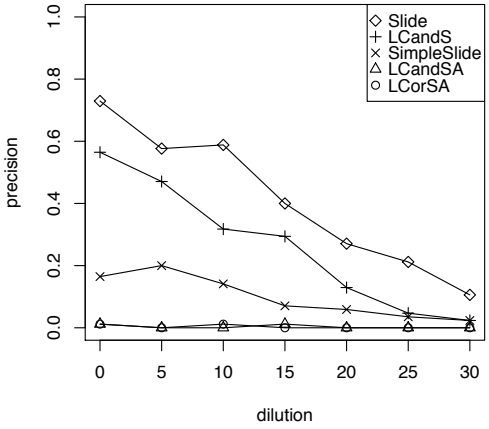

Figure 6. Precision of LSA with different neighborhood functions on artificial networks with implanted motifs.

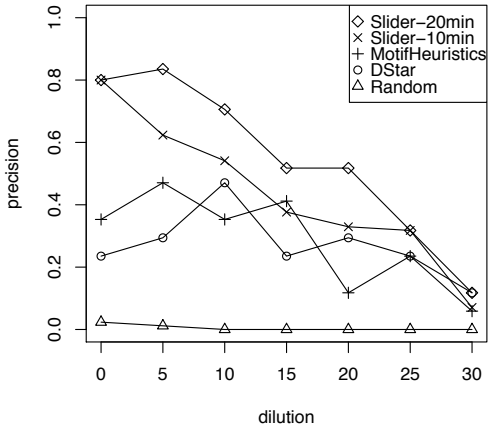

Figure 7. Precision of SLIDER compared with that of D-STAR, MotifHeuristics and Random.

7.5 Biological validation

Next, we assess the effectiveness of SLIDER on the yeast network. We did not try MotifHeuristics as it already takes a long time on networks of modest size (cf. Section 7.4). Furthermore, although D-STAR terminated on our artificial networks within 5 minutes, the method does not scale to larger networks. In particular, Leung et al. [7] mention an experiment where they executed D-STAR on the yeast network and it did not finish in 5 days, we ourselves have run D-STAR on this network for 48 hours without result.

We ran SLIDER for 20 minutes exploiting all 8 cores of the Mac Pro. The average precision of the 1 000 best results returned by SLIDER over 5 runs, while taking the 1 000 best motifs returned by the brute force computation as a baseline, is 16%. We point out that the name precision is misleading in this context as we do not compare with implanted motifs. The number implies that SLIDER succeeds in recovering no less than 160 of the 1000 best correlated motifs out of a search space of 6×10^{15} (8,3)-motif pairs after only a run of 20 minutes which is quite satisfactory. As SLIDER returns a ranked list, these 160 motif pairs occur at the top. Moreover, these found correlated motifs occurred uniformly within the baseline set. The latter is confirmed by the histograms in Figures 8 and 9. They show that the frequency of the sizes of the subnetworks selected by the returned motif pairs are similar to those of the overall best 1 000 motif pairs. We mention that after 10 minutes, already a precision of 11% was obtained.

8 Related Work

Local search is not only the oldest but also the simplest among the known meta-heuristics for combinatorial optimization [3]. As the success of all meta-heuristics depends largely on the effectiveness of the neigborhood function, we choose in this paper to stick to the simplest meta-heuristic, put the main focus on fine-tuning the neighborhood function and leave the exploration of more powerful heuristics to future work.

At first sight the present work seems highly related to the mining of frequent patterns in sequences (as for instance in

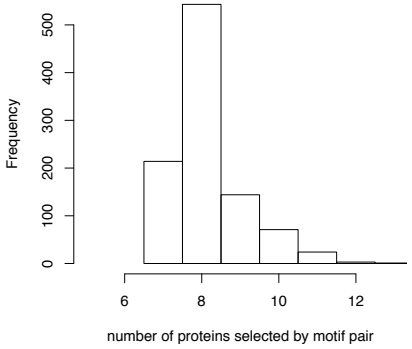

Figure 8. Frequency of the sizes of the subnetworks selected by the 1 000 best motif pairs as returned by a brute force run.

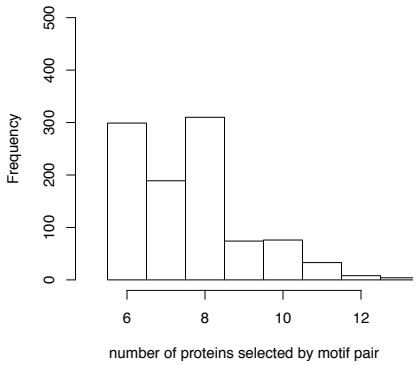

Figure 9. Frequency of the sizes of the subnetworks selected by the 1 000 best motif pairs as returned by SLIDER**.**

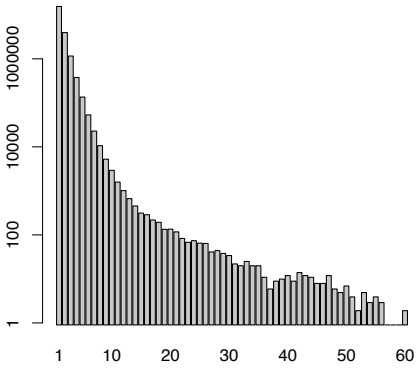

Figure 10. Distribution of selectivity of (8,3)-motifs in yeast (x-axis: number of selected proteins, y-axis: number of (8,3)-motifs)

[6]). It is therefore tempting to think about a method which first mines frequent motifs from protein sequences which are then paired together in a second step serving as candidates for high scoring correlated motifs. An examination of the 1 000 top correlated motifs in yeast, however, reveals that each of the participating motifs occur only in 3 to 10 motifs, whereas highly frequent motifs in yeast occur in up to 60 proteins as can be seen from the histogram in Figure 10. Therefore, mining correlated motifs is very different from mining frequent motifs.

9 Conclusion

This work lays the foundation of motif-driven CMM in establishing an adequate support-measure and determining the complexity of the general problem. The novel method SLIDER based on the sliding window neighbor function outperforms existing motif-driven CMM algorithms and shows a very promising behavior on real-world PPI-networks. Of course, there is still room for improvement. There are several directions for future work: address more advanced metaheuristics and investigate candidate generation for motif pairs. A detailed comparison with interaction-driven approaches should be done [8, 9, 10], although this would require a new type of artificial networks. Maybe ideas from both paradigms can be successfully combined into a hybrid method. Furthermore, we only considered the very simple model of (ℓ, d)-motifs. Although more expressive models exist (e.g., position weight matrix or Hidden Markov Models), (ℓ, d)-motifs are very common in the field of bioinformatics. Moreover, Van Dijk et al [14], already showed how motifs generated by D-STAR can be used to predict transcription factor interaction on small networks. Using SLIDER rather than D-STAR, the same methodology can be applied to larger networks. Nevertheless, it would be worthwhile to investigate more expressive motifs.

Finally, we mention that we could not confirm the claimed superiority in [7] of MotifHeuristics over D-STAR. In fact, our results clearly show that f_p is inferior to f_{χ^2} in recovering implanted motifs. These tests should be repeated on real world data, but as long as only few biological correlated motifs are known this is not possible.

References

[1] E. Aarts and J. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, 1997.

[2] P. Aloy and R. B. Russell. Ten thousand interactions for the molecular biologist. *Nat Biotechnol.*, 22:1317–1321, 2004.

[3] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35(3):268–308, 2003.

[4] S. Collins et al. Towards a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. *Mol Cell Proteomics.*, 2007.

[5] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 1979.

[6] K. Gouda, M. Hassaan, and M. J. Zaki. Prism: A primal-encoding approach for frequent sequence mining. In *ICDM*, pages 487–492, 2007.

[7] H. Leung et al. Finding linear motif pairs from protein interaction networks: A probabilistic approach. In *Computational Systems Bioinformatics (CSB)*, pg. 111–120, 2006.

[8] H. Li, J. Li, and L. Wong. Discovering motif pairs at interaction sites from protein sequences on a proteome-wide scale. *Bioinformatics*, 22(8):989–996, 2006.

[9] J. Li, G. Liu, H. Li, and L. Wong. Maximal biclique subgraphs and closed pattern pairs of the adjacency matrix: A one-to-one correspondence and mining algorithms. *IEEE Trans. Knowl. Data Eng.*, 19(12):1625–1637, 2007.

[10] J. Li, K. Sim, G. Liu, and L. Wong. Maximal quasi-bicliques with balanced noise tolerance: Concepts and co-clustering applications. In *SDM*, pages 72–83. SIAM, 2008.

[11] J. Newman et al. A computationally directed screen identifying interacting coiled coils from Saccharomyces cerevisiae. *Proc Natl Acad Sci U S A*, 97(24):13203–8, 2000.

[12] M. Stumpf et al. Estimating the size of the human interactome. *Proc Natl Acad Sci U S A*, 105(19):6959–64, 2008.

[13] Tan et al. A correlated motif approach for finding short linear motifs from protein interaction networks. *BMC Bioinformatics*, 7:502+, November 2006.

[14] A. D. J. van Dijk et al. Predicting and understanding transcription factor interactions based on sequence level determinants of combinatorial control. *Bioinformatics*, 24(1):26–33, 2008.

[15] C. von Mering et al. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417:399–403, 2002.

[16] M. Šikić, S. Tomić, and K. Vlahoviček. Prediction of protein-protein interaction sites in sequences and 3d structures by random forests. *PLoS Comput Biol*, 5(1):e1000278+, 2009.