Made available by Hasselt University Library in https://documentserver.uhasselt.be

SLIDER: A Generic Metaheuristic for the Discovery of Correlated Motifs in Protein-Protein Interaction Networks

Non Peer-reviewed author version

BOYEN, Peter; VAN DYCK, Dries; NEVEN, Frank; van Ham, Roeland C. H. J. & van Dijk, Aalt D. J. (2011) SLIDER: A Generic Metaheuristic for the Discovery of Correlated Motifs in Protein-Protein Interaction Networks. In: IEEE-ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, 8(5). p. 1344-1357.

DOI: 10.1109/TCBB.2011.17
Handle: http://hdl.handle.net/1942/12110

SLIDER: a generic metaheuristic for the discovery of correlated motifs in protein-protein interaction networks

Peter Boyen, Dries Van Dyck, Frank Neven, Roeland C.H.J. van Ham, and Aalt D.J. van Dijk

***Abstract*—Correlated motif mining (CMM) is the problem of finding overrepresented pairs of patterns, called motifs, in sequences of interacting proteins. 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 Chi-square-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 Chi-square) and reformulate the search for correlated motifs as a combinatorial optimization problem. We then present the generic metaheuristic SLIDER which uses steepest ascent with a neighborhood function based on sliding motifs and employs the Chi-square-based support measure. We show that SLIDER outperforms existing motif-driven CMM methods and scales to large protein-protein interaction networks.**

The SLIDER-implementation and the data used in the experiments are available on http://bioinformatics.uhasselt.be.

***Index Terms*—Graphs and networks, Biology and genetics**

I. Introduction

LARGE-SCALE biological networks describing interactions between proteins are available for several organisms [16]. Such data demonstrate 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 at the present rate of protein structure determination, it would take 20 years to determine all interaction types using current experimental techniques [2]. Moreover, even if this would be accomplished, one would still have to deal with predicting for a given interacting sequence to what interaction type it adheres. 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 [11]–[14], [17]. 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 Fig. 1 the patterns $\{1, \mathtt{A}\}$ and $\{2, \mathtt{B}\}$ represent two such correlated motifs. In particular, there is an undirected edge between two protein sequences when the first one contains motif 1, and the second one motif A, and similarly for motif 2 and B. Currently, despite the development of several algorithms (see below) it is unclear which fraction of interfaces can be described by such correlated motifs. However, results from existing approaches clearly indicate that correlated motifs do contain information about interfaces [11]–[14], [17].

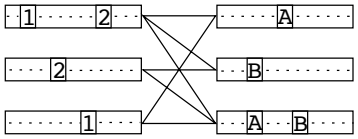


Fig. 1. Compatible binding sites $1, \mathtt{A}$ and $2, \mathtt{B}$ as correlated motifs in sequences.

These methods can be subdivided into two classes: (i) interaction-driven [12]–[14], and (ii) motif-driven approaches [11], [17]. Interaction-driven methods mine for (quasi-)bicliques, i.e., subsets of vertices for which (almost) 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; and, (iii) if the interactions between two sets of proteins are a consequence of multiple compatible binding sites, such as $\{1, \mathtt{A}\}$ and $\{2, \mathtt{B}\}$ in Fig. 1, the interaction-driven method necessarily merges them into one motif pair.

In this article, we study 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. [17], called D-STAR, uses a χ^2-based scoring function to determine the support, but the underlying mining method

P. Boyen, F. Neven and D. Van Dyck are with Hasselt University and Transnational University of Limburg.
E-mail: {peter.boyen, frank.neven, dries.vandyck}@uhasselt.be
A. van Dijk and R. van Ham are with Applied Bioinformatics - Plant Research International, Wageningen UR.

The present article is the extended full version of [5].

does not scale to networks containing more than 250 proteins. As contemporary biological networks contain up to thousands of proteins (see Section VI), scalability is an increasingly important issue. The second method, called MotifHeuristics, employs a different, probabilistically motivated notion of support called p-score. This method is developed by Leung et al. [11] 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, 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 NP-complete. 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 generic metaheuristic containing two steepest ascent[1] methods, the key components of which are their neighborhood functions, based on viewing a motif as a window that slides over the amino acid sequence of one of the proteins. In contrast with more common neighborhood functions, they have a clear biological interpretation: they are 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. So both neighborhood functions want to find neighboring motifs that could be close to each other on actual proteins. The difference is that one considers as neighbors of a motif all motifs which could theoretically be near it on any protein whereas the other only takes motifs actually nearby on a single selected protein. 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 its methods outperform 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.

This article expands upon an earlier conference paper [5] by the following additional elements:

1) We present a more thorough treatment of the complexity of CMM. In particular, we formally prove, under reasonable assumptions concerning the used notion of support, the NP-hardness of CMM for biclique-maximal measures, which includes χ^2, by proving that even a simplified version of the associated decision problem is NP-complete.
2) We introduce a new version of SLIDER which uses an improved neighborhood function to explore the search space and yields significantly better results.
3) We present a more thorough experimental validation of both variants of SLIDER using more data and more experiments. In particular, we present an in-depth assessment of the biological relevance of our results by looking if the motif hits overlap with known interaction sites in protein structure data. This allows us to assess not only the biological usefulness of our methods but also of the (ℓ, d)-CMM model itself, by considering the best motif pairs found by brute force — as far as we know, the latter has never been done before on a genome-wide scale.
4) We discuss in depth our choice for steepest ascent over more advanced metaheuristics.

Outline. In Section II, we formally define CMM and in Section III we discuss support measures. In Section IV, we prove CMM to be NP-hard for a large class of support measures. In Section V, we introduce the generic SLIDER metaheuristic. In Section VI, we introduce our artificial and biological datasets on which the effectiveness of our methods is assessed in Section VII. We discuss related work in Section VIII and conclude in Section IX.

For the reader's convenience, a table of major notation has been added to the Supplementary Material.

II. Correlated motif mining

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 sequences of the proteins. Hence, the label function λ maps each vertex $v \in V$ to a string $\lambda(v)$ over the alphabet $\Sigma = \{\mathtt{A}, \ldots, \mathtt{Z}\} \setminus \{\mathtt{B}, \mathtt{J}, \mathtt{O}, \mathtt{U}, \mathtt{X}, \mathtt{Z}\}$.

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, GAQPRNMY matches the $(8, 4)$-motif 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 $V_X = \{v \in V \mid v \text{ contains } X\}$,

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

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

[1] In contrast with [5], we use the term steepest ascent instead of local search here because the latter is also used to refer to all metaheuristics which are based on the notion of a neighborhood function.

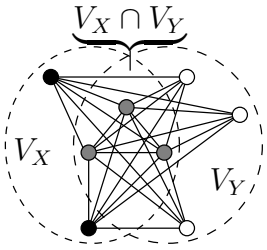


Fig. 2. An example of a network selected by a complete (5,6,3)-motif pair.

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 a function mapping a motif pair $\{X, Y\}$ and a graph G to a positive real number $f(\{X, Y\}, G)$. We refer to $f(\{X, Y\}, G)$ as the *support* of $\{X, Y\}$ in G. In Section III and VII-B we discuss and compare several instances of support measures.

We next formulate the correlated motif mining problem in a PPI-network (CMM):

- **Input:** a PPI-network $G = (V, E, \lambda)$, $\ell, d, k \in \mathbb{N}$ 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:** $\{X_1, Y_1\}, \ldots, \{X_k, Y_k\}$, the k (ℓ, d)-motif pairs with highest support in G with respect to f.

III. 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 (Y) in the sequences of the proteins in V_X (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 site on the surface of the molecule 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 *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*. The *edge density* $\mathrm{ed}(G)$ of a graph $G = (V, E)$ is the proportion of edges it has of all its potential edges: $\mathrm{ed}(G) = |E| / \binom{|V|}{2}$.

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. Fig. 2 shows an example. As such, the maximal number of edges any $(k_X, k_Y, k_{X,Y})$-motif pair can have in any PPI-network is

$$E^{\max}_{k_X, k_Y, k_{X,Y}} = \left(k_X k_Y - \binom{k_{X,Y}}{2} - k_{X,Y} \right) .$$

A. A χ^2-based support measure

Tan et al. [17] 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:

$$\overline{E_{X,Y}} = \mathrm{ed}(G) E^{\max}_{|V_X|, |V_Y|, |V_X \cap V_Y|} .$$

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

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

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.

B. p-score: a probabilistic support measure

The p-score is a measure introduced by Leung et al. [11] 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 at least $|E_{X,Y}|$ 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 $N(V_X) = \{u \mid \exists v \in V_X : \{u, v\} \in E\}$, 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 [11] for details)

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

where

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

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:

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

C. 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 implicitly makes 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, i.e., becomes more interesting. In addition, 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.

IV. 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.

For technical reasons, we restrict ourselves to support measures which abide by three reasonable conditions. 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*[2] 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) $f(M_{k_X+1,k_Y,k_{X,Y}},G) > f(M_{k_X,k_Y,k_{X,Y}},G)$.

Informally, the first condition says that the support can be computed efficiently, which is crucial for scalability reasons. The second condition states that if the subnetworks selected by 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. Hence, the last two conditions formalize the intuition that a good support measure prefers motif pairs which select large, dense subnetworks. On the other hand, the last two conditions also induce some bias as they implicitly assume that the support only depends on V_X, V_Y, $E_{X,Y}$ and/or its relation to the PPI-network as a whole.

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

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

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 biclique-maximal compliant support measure f.

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

We will now describe a reduction R which transforms an unlabeled graph $G = (V,E)$, with $V = \{v_1,\ldots,v_n\}$, into a labeled graph $R(G) = G' = (V,E,\lambda)$. Afterwards, we will show this reduction can be used to prove the NP-completeness of D-CMM for biclique-maximal measures.

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, ignoring the cases with V_X or V_Y empty, and thus $E_{X,Y}$ empty, every motif pair in G' is necessarily a $(1,1,k')$-, $(1,k,k')$-, $k' \in \{0,1\}$, or a (k,k,k')-motif pair, $0 \leq k' \leq k$. Moreover, for an (n,k)-motif X containing only 0's and

[2]The notion of compliance we use here is looser than the notion we used in [5] which also demanded that the support must increase if there are more proteins in $V_X \cap V_Y$.

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 $|E_{X,Y}| = E^{\max}_{|V_X|,|V_Y|,0}$, then $(V_X \cup V_Y, E_{X,Y})$ is a balanced complete bipartite graph.

Given a graph G and a natural number k, deciding whether G contains a biclique such that both parts are of size k, is called the *balanced complete bipartite subgraph problem* (BCBS). BCBS is known to be NP-complete [7]. We will now show that we can decide BCBS on G by deciding D-CMM on $R(G) = G'$ for a compliant, biclique-maximal support measure.

Since the support measure is compliant, we know that a complete $(k_X, k_Y, k_{X,Y})$-motif pair will always have higher support than any other $(k_X, k_Y, k_{X,Y})$-motif pair. Let $M_{k_X,k_Y,k_{X,Y}}$ be a complete $(k_X, k_Y, k_{X,Y})$-(n, k)-motif pair for G', $k_{X,Y} \leq \min(k_X, k_Y)$ and $k \geq 2$. We know that, by construction of G', $k_X, k_Y \in \{1, k\}$. As f is compliant and biclique-maximal it holds that:

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

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

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

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

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

which is maximal for $k_{X,Y} = 0$. On the other hand, remark that f_p is not compliant because $f_p(\{X, Y\}, G)$ depends on the neighborhood of the selected subnetwork $G_{X,Y}$ in G (G_X and G_Y).

V. Our methods

Since the decision problem associated with CMM is in NP, we can efficiently check if a motif pair has higher support than an other which makes it possible to tackle CMM 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 *metaheuristics* are known to yield good solutions to a wide variety of combinatorial optimization problems.

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

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

Fig. 3. The general steepest ascent algorithm with abstract neighbor function applied to CMM (SA-CMM).

One such metaheuristic is *steepest ascent* [1]. Steepest ascent algorithms move from the current point to the best neighboring point in the space of candidate solutions until a locally optimal solution is found, i.e., a solution that maximizes f in its neighborhood. Hence, to apply steepest ascent one needs to define a neighborhood function which returns the neighbor points of each point in the search space. The neighborhood function is a key component of the steepest ascent method and has to be chosen carefully and fine-tuned for the problem at hand. The initial points from where steepest ascent is started are randomly chosen. In Section VIII, we discuss other metaheuristics and explain the choice for steepest ascent.

The main idea behind our steepest ascent algorithm for CMM is illustrated by the pseudo-code in Fig. 3. To be able to specify the difference between our two methods, we use an abstract neighborhood function N. For reasons of clarity, we use 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 as many runs as can be completed in a given time frame, and store the results sorted by support.

The method randomMotifPair() picks (i) a random interaction $\{u, v\}$, (ii) a random position p_u in $\lambda(u)$ and p_v in $\lambda(v)$, (iii) a random motif X by first picking d random positions in $[p_u + 1, p_u + \ell - 1]$ as the wildcard positions and taking the remaining positions as the non-wildcard positions, and; (iv) a random motif Y from $\lambda(v)$ in the same way.

In order to apply steepest ascent to CMM, we need to define a neighborhood function 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 $G_{X,Y}$. The main idea behind a steepest ascent algorithm is to gradually improve a candidate solution until it becomes (locally) optimal. Consequently, it is desirable that the subnetwork $G_{X',Y'}$ selected by a neighbor $\{X', Y'\} \in N(\{X, Y\})$ is also "close" to $G_{X,Y}$ in the sense that at least some proteins and interactions are shared between $G_{X,Y}$ and $G_{X',Y'}$. That is, we would like that the candidate solution in the *dual* search space of selected subnetworks also improves gradually in order to avoid that the algorithm jumps around in the network selecting completely different networks in each step. Suppose for instance that $\{X, Y\}$ is a motif pair which describes (a part

Fig. 4. Two neighboring (by N^{mot}) $(6,3)$-motifs seen as sliding windows on a sequence.

of) compatible binding sites in most proteins in V_X and V_Y. If at some point in the algorithm we reach a motif pair $\{X', Y'\}$ for which a significant fraction of the motif hits of X' and Y' overlap with the desired motifs X and Y, it would be undesirable that X' is changed into a motif which has almost no motif hits in V_X.

A straightforward way to ensure that some proteins are kept, is by only considering motifs of the form $\{X, Y'\}$ or $\{X', Y\}$ as candidate neighbors, such that either V_X or V_Y remains in $G_{X',Y'}$. The neighbor functions we will define in the next sections share the principle that one motif remains fixed and that the neighborhood of the pair is defined in terms of a neighbor function N on the motifs, more formally: $\{X', Y'\} \in N(\{X, Y\})$ if $X' \in N(X) \wedge Y' = Y$ or $Y' \in N(Y) \wedge X' = X$.

Hence, to ensure that $G_{X',Y'}$ is also likely to share interactions with $G_{X,Y}$, it suffices to define the neighborhood function N on motifs in such a way that V_X shares proteins with $V_{X'}$ for most of the motifs $X' \in N(X)$.

On the other hand, it is also desirable that N is powerful enough to move from any $\{X, Y\}$ to any other $\{X', Y'\}$ in a reasonable number of steps, while keeping $N(\{X, Y\})$ small enough to keep evaluating all neighbors of $\{X, Y\}$ tractable.

A. M-SLIDER: *Sliding over motifs*

In this subsection, we formally introduce a first neighborhood function N^{mot} on motifs which will be the basis for M-SLIDER (short for motif-SLIDER) – the method introduced in [5]. N^{mot} is based on the observation that 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 characters of X. Hence, any motif X' obtained by closing one hole and creating a new one (not too far from the other ones so as to respect the window size ℓ) will select the same protein we are sliding the window over. In this way, the motif window can slide to the left or right if the new hole is punched before the first or after the last original character. We will call any motif X' a neighbor of X, if it can be obtained from it, by replacing one non-wildcard character with a wildcard and then adding a new non-wildcard character making it an (ℓ, d)-motif again. We can see in Fig. 4 that moving from `RTxTxx` to `KxxTxT`, by closing the hole over the `R` and opening a new one over the `K`, shifts the window to the left. The motif `RTxxxA` is also a neighbor, but does not select the same protein.

Next, we formally define N^{mot}. 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 $X' \in N^{\text{mot}}(X)$ if X and X' have the same length and $\text{trim}(Y) = \text{trim}(Y')$ where Y is obtained from X by changing one non-wildcard character into a wildcard, and similarly for Y' and X'. In Fig. 4, X equals `RTxTxx` while X' equals `KxxTxT`. Now, $X' \in N^{\text{mot}}(X)$ as X (X'), can be transformed into $Y =$ `xTxTxx` ($Y' =$ `xxxTxT`) by changing one non-wildcard character into a wildcard and Y equals Y' after stripping leading and trailing wildcards.

Remember that when applying N^{mot} to pairs of motifs, one of the motifs remains fixed. From our experiments we observed that fixing one motif at each step greatly improves the effectiveness.

It is fairly easy to show that N^{mot} allows to reach any $\{X', Y'\}$ in at most $2(\ell - d)$ steps and $|N^{\text{mot}}(\{X, Y\})| = \Theta(\ell^2)$ which keeps evaluating all neighbors tractable for the typical values for ℓ and d. Moreover, at least $2d(\ell - d)$ neighbors will select a subnetwork that shares at least one interaction with $G_{X,Y}$ (see Supplementary Material).

Definition 1: We define the method M-SLIDER as steepest ascent with
(i) neighborhood function N^{mot}; and,
(ii) support measure f_{χ^2}.

It can be formally shown that, if we assume that the number of steps can be bound by a small constant as observed in our experiments (for instance, in our experiments the number of steps never exceeded 15), M-SLIDER runs in time $O\left(\ell^2\ (|V|^2 + \ell|V|\lambda_{\max})\right)$, with $\lambda_{\max} = \max_{v \in V} |\lambda(v)|$. Remark that the former is almost linear in the size of G, when $|G| = |V|^2$. However, using a theoretical maximum number of steps $|V|^5$, we obtain the bound $O\left(|V|^5\ \ell^2\ (|V|^2 + \ell|V|\lambda_{\max})\right)$ (proof in Supplementary Material).

B. SEQ-SLIDER: *Sliding over sequences*

Although a significant number of the neighbors of a motif pair $\{X, Y\}$ under N^{mot} are expected to select a subnetwork $G_{X',Y'}$ that is also "close" in the network in the sense that $G_{X,Y}$ and $G_{X',Y'}$ share interactions, this property is not guaranteed for any neighbor. For that reason, we also designed a second neighborhood function N^{seq} which focusses on this aspect, but does not guarantee that all other motif pairs can be reached by moving from one neighbor to the other. The N^{seq} neighborhood function forms the basis of our second SLIDER variant SEQ-SLIDER.

N^{seq}_u defines the neighborhood of a motif X on the sequence level by considering all (ℓ, d)-motifs that match a region around the motif hits of X in the sequence of one particular protein $u \in V_X$. The idea is that, in each run, after picking a random pair $\{X, Y\}$ that describes some interaction $\{u, v\}$, we only consider motif pairs based on the region around the motifs hits of X in $\lambda(u)$ and of Y in $\lambda(v)$, i.e.,

$$N^{\text{seq}}_{u,v}(\{X, Y\}) = \{\{X', Y\} \mid X' \in N^{\text{seq}}_u(X)\} \cup \{\{X, Y'\} \mid Y' \in N^{\text{seq}}_v(Y)\}.$$

In that way, $N^{\text{seq}}_{u,v}$ guarantees that the subnetwork $G_{X',Y'}$ selected by any neighbor $\{X', Y'\}$ of a motif pair $\{X, Y\}$ will always contain $\{u, v\}$.

More formally, for an (ℓ, d)-motif X and a protein u, denote by $pos(X, u)$ the set of positions of substrings in $\lambda(u)$ that match X. An (ℓ, d)-motif $X' \in N_u^{\text{seq}}(X)$ if there exist positions $p \in pos(X, u)$ and $p' \in pos(X', u)$ such that $|p - p'| \leq \delta$, where δ is some small distance bound (we use $\delta = \lceil \ell/3 \rceil$). Hence, $N_{u,v}^{\text{seq}}(\{X, Y\})$ defines the neighborhood of $\{X, Y\}$ relative to $u \in V_X$ and $v \in V_Y$.

For instance, the two motifs in Fig. 4 are also neighbors under N^{seq} as they both have matches in the sequence within the distance bound. The motif `KYxTxx` is an example of a motif that would be a neighbor under N^{seq} but not under N^{mot} as it differs more than one non-wildcard character from the original. In fact, it even does not share any amino acids at all with the original motif. The motif `RTxxxA` on the other hand is a neighbor using N^{mot} but not using N^{seq} as it does not have any matches within a δ-region of a match of the original motif.

Thus, for a sufficiently high number of runs, we are likely to have considered a local optimum under $N_{u,v}^{\text{seq}}$ for each $\{u, v\}$ in E, which gives SEQ-SLIDER a bias towards a set of complementary best motif pairs in the sense that all of them together are likely to cover more interactions than the set of best motif pairs returned by M-SLIDER.

From a theoretical point of view however, SEQ-SLIDER has some disadvantages compared to M-SLIDER: it cannot reach every motif pair from an arbitrary motif pair and evaluating all neighbors of a motif pair can be expensive as the number of neighbors $|N^{\text{seq}}(\{X, Y\})|$ can become as large as $\binom{\ell}{d}(2\delta + 1)(|pos(X, u)| + |pos(Y, v)|)$ (see Supplementary Material), which can become prohibitive for larger values of ℓ and d. Nevertheless, as we will see in the experimental section, SEQ-SLIDER obtains significantly better results than M-SLIDER in the same time frame for the typically small values of ℓ and d.

Definition 2: We define the method SEQ-SLIDER as steepest ascent with
(i) neighborhood function $N_{u,v}^{\text{seq}}$ with $\delta = \lceil \ell/3 \rceil$; and,
(ii) support measure f_{χ^2}.
Let $\lambda_{\max} = \max_{v \in V} |\lambda(v)|$. We formally proved that SEQ-SLIDER runs in time $O\left(\delta\binom{\ell}{d}\lambda_{\max}\ (|V|^2 + \ell|V|\lambda_{\max})\right)$, if we again assume the number of steps is constant, and $O\left(|V|^5\ \delta\binom{\ell}{d}\lambda_{\max}\ (|V|^2 + \ell|V|\lambda_{\max})\right)$ otherwise (proof in Supplementary Material).

VI. 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 [6]. We then generate 50 random $(8, 3)$-motifs[3] and implant k instances of each motif in the sequences of randomly chosen proteins, with k chosen uniformly from $\mathbb{N}[3, 10]$. 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 and repeat this procedure until a chosen minimal edge density e is obtained — we used 0.1, 0.2 and 0.3. Consequently, the network obtained is perfect in the sense that there is an interaction $\{u, v\}$ if and only if a motif pair is present in $\lambda(u)$ and $\lambda(v)$. Because noise and missing data are 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 created versions of each network with added noise, by choosing a certain noise level a (from 0.05 to 0.3 in steps of 0.05) and switch the edge relation of each pair of vertices with probability a (remove the edge if they are connected and add one if not). We used 105 networks in total — 5 networks for each (e, a)-combination.

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, which is necessary to evaluate the results.

As a sanity check, we also constructed networks where only a small portion of interactions can be explained by a motif pair. Several tests were run which show that SLIDER performs similarly in that case (see Supplementary Material).

Biological data. To assess the effectiveness on larger networks, we ran our method and MotifHeuristics on the high-confidence PPI-network of yeast [6] consisting of 1 620 proteins and 9 060 interactions and on the human PPI-network [15] which has 8 872 proteins and 14 230 interactions — two of the largest and most complete interaction datasets available. The interactions in the human network are curated from the literature [15] and the interactions in the yeast dataset are determined using tandem-affinity purification followed by mass-spectrometry (TAP/MS) — a technique which is used to determine the proteins in a complex [6]. As a consequence, the interactions determined by TAP/MS contain both direct and indirect interactions. For that reason, it is expected that the human dataset contains less false positive but more false negative interactions in comparison with yeast. Hence, these two interaction datasets are ideal to assess our methods as they are

1) large, which allows us to test the scalability of our methods;
2) as complete as available at the moment, which allows to assess if the best scoring (ℓ, d)-motif pairs found by our methods and by a brute force method can describe the interactions given enough data;
3) complementary in terms of noise, which allows to assess how the descriptive power of the best scoring (ℓ, d)-motif pairs of our methods and a brute force method are affected by different kinds of noise (false positives vs. false negatives).

VII. Experiments

The brute force runs on yeast and human (which calculate support for each possible motif pair) were run on a computer cluster. All other experiments were run on a 3GHz Mac Pro

[3]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 Fig. 1 in [20]).

using 2GB of RAM and 8 cores. In the following, 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 (i) support measures to assign a support to a motif pair reflecting its power to describe interactions; and, (ii) neighborhood functions to find the motif pairs with highest support by exploring the space of all motif pairs. Furthermore, we compare both SLIDER variants with other motif-driven CMM-methods. To this end, we need a notion of precision[4] 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 the SLIDER variants on the yeast and human PPI-networks.

A. Precision for motif pairs

Before we define our notion of precision, we need a similarity measure to compare the found motif pairs against the implanted 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 $\{v, w\} \in E_{X,Y} \cap_{pos} E_{X',Y'}$ if there exists substrings s_v and s'_v in $\lambda(v)$ and s_w and s'_w in $\lambda(w)$ such that
(i) s_v matches with X and s_w with Y
(ii) s'_v matches with X' and s'_w with Y'
(iii) s_v and s'_v as well as s_w and s'_w are at the same position 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 $s(M_i, M_j) = 1$. 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 $s(M_i, M_T) = 1$. We note that, when $k = |T|$, the precision as defined above also corresponds to the usual notion of recall[5].

B. 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 descriptive 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 VI. We used a collection of networks G_e^a with edge density e and noise level a. We compare the support measures by looking at the precision of the best motif pairs obtained by a brute force method at rank m against the implanted motif pairs on G_e^a, where m equals the number of implanted motif pairs.

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 $f_v(\{X, Y\}, G) =$

$$\frac{|E_{X,Y}|}{E^{\max}_{|V_X|,|V_Y|,|V_X \cap V_Y|} + |V_X \cup V_Y|}\ .$$

f_v is the edge density corrected with an extra term in the denominator to prefer larger subnetworks ($E^{\max}_{|V_X|,|V_Y|,|V_X \cap V_Y|}$ grows quadratically in $|V_X \cup V_Y|$). 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 Fig. 5 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 noisy networks shows that f_{χ^2} is vastly more robust to noise — a crucial aspect since contemporary PPI-networks contain large amounts of both noise and missing data [19].

When we compare the results of the brute force runs on yeast for f_{χ^2} and f_p, we also notice that the 1 000 best scoring subnetworks for f_{χ^2}, have an average edge density of 97.2% and a *minimum* edge density of 64%, while those for f_p have an average edge density of 14.5% and a *maximum* edge density of 16.7%. The edge density for the latter is obviously much lower than desired.

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

C. Evaluation of neighborhood functions

We will now confirm that our neighborhood functions, which are based on a sliding window interpretation on the sequences, are superior to neighborhood 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^{mot}, denoted $N^{\text{mot}}_{\ominus}$, which forces the motif to slide left or right by only allowing to change the leftmost (rightmost) non-wildcard character into a wildcard and demanding that the new non-wildcard character is added

[4]The notion of precision we will use is similar to the notion of sensitivity of a binary classifier. Specificity, however, cannot be defined for a ranking problem such as CMM because there is no meaningful notion of true negative.

[5]Pathetic constructions are possible in which one motif pair is similar to multiple "ground truth" motif pairs, but these are extremely unlikely in practice.

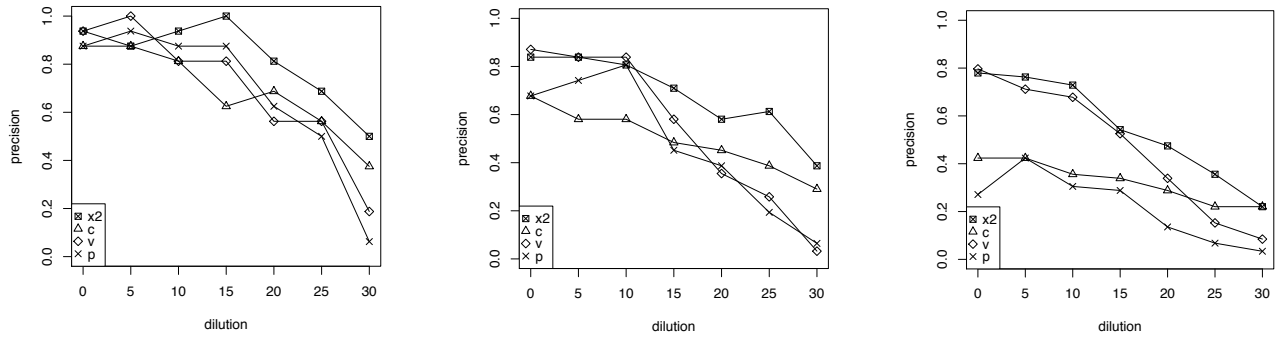


Fig. 5. Precision of support measures on artificial networks with implanted motif pairs and different edge densities (10%, 20%, 30%).

to the right (left) of the existing ones. The corresponding neighborhood functions on pairs of motifs are defined similarly: one motif is kept fixed, while the other is replaced by a neighbor. As a naive baseline, we also compare with the method Random, which evaluates random motif pairs using f_{χ^2}.

Fig. 6 displays the precision of SA-CMM with each of these neighborhood functions on five implanted networks of density 10% and their noisy versions. The displayed precision is averaged over 5 SA-CMM runs. Runs on the networks of density 20 and 30% give similar results (data not shown). As the speed of SA-CMM is highly dependent on the chosen neighborhood function, we provided each run the same amount of time (10 minutes). In this way, faster neighborhood functions like LCorSA can process more randomly chosen starting motif pairs than slower functions like N^{mot} and N^{seq} (cf. Fig. 7). As can be seen from Fig. 6, N^{seq}, and thereby SEQ-SLIDER, outperforms the other SA-CMM variants using other neighborhood functions, including M-SLIDER which is second.

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 noise level, indicating that in those cases the merit of a larger neighborhood is overshadowed by the time it costs to search it.

D. Comparison with existing methods

D-STAR. Tan et al. introduced the first motif-driven method for CMM: D-STAR [17]. 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 that differs at most $2d$ characters from s_X, and similarly for V_Y and 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 VII-A 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 Leung et al. [11], 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 steepest ascent 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 [11] and confirmed the correctness of the implementation by reproducing all results on the SH3-dataset from [11].

Comparison. Given that each method relies on different principles, it is not easy to compare them directly. Both SLIDER variants and MotifHeuristics share the principle that they start from a random motif pair which is improved by local search principles. One could be tempted to compare them by looking at the results which each method obtains using a fixed number of motif pair seeds but such a comparison would favor a method which considers a larger neighborhood in each step, that is, has an expensive neighbor function. Moreover, D-STAR is a deterministic method and as such is unable to improve its results by using more time. For those reasons, we believe that comparing the results obtained by each method within a given time frame yields the fairest comparison as time is the most important constraint for large PPI-networks and any method method requires time to produce results.

The graph in Fig. 8 depicts the precision of the various methods on the artificial network of density 10%, including Random as naive baseline. D-STAR took 5 minutes to finish. We let Random and both SLIDER variants run once for 10 and once for 20 minutes. In order to give our unoptimized

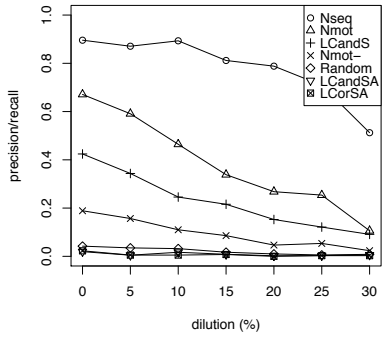


Fig. 6. Precision of SA-CMM with different neighborhood functions on artificial networks with implanted motifs.

Fig. 7. Average amount of randomly chosen initial motif pairs per run for each neighborhood function.

Neighbor func.	seeds
N^{seq}	90K
N^{mot}	277K
LCandS	784K
$N^{\text{mot}}_{\ominus}$	1 986K
LCandSA	3 315K
LCorSA	3 643K
Random	15 924K

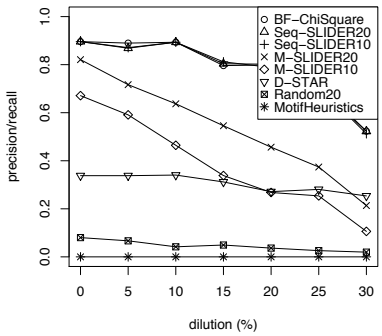


Fig. 8. Precision of SLIDER compared with that of D-STAR, MotifHeuristics and Random on artificial networks.

implementation of MotifHeuristics a fair chance, we allowed it to run for 175 minutes. 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 SEQ-SLIDER is vastly superior to all other methods — the precision obtained by both SEQ-SLIDER runs are so close to the precision obtained by brute force that they are almost indistinguishable in the figure. M-SLIDER is second as long as the network is not too noisy but loses to D-STAR as the networks become more noisy. It might be noteworthy that D-STAR finishes in more or less 5 minutes but, as mentioned earlier, its results cannot be improved by giving it more time. We also performed 5 minute M-SLIDER runs to make the comparison with D-STAR more fair and in that time frame M-SLIDER's precision is only better than D-STAR's for the original networks. On the other hand, if we give M-SLIDER more time it beats D-STAR on all noise levels. Somewhat surprisingly, Random performs better than MotifHeuristics. Calculating f_p-support for an enormous amount of neighbors takes so much time that our implementation of MotifHeuristics could handle only about 120 initial motif pairs in 175 minutes. Hence, this experiment indicates that even a random search using f_{χ^2} is a better approach to retrieve implanted motif pairs than a heuristic search using f_p.

Overall, both SLIDER variants are more effective and robust than its competitors although M-SLIDER needs more time to outclass D-STAR on these small networks.

We conclude the comparison by pointing out that both SLIDER variants obtain a precision$>$80% in 20 minutes on the original networks, which is quite fast in comparison with the 40 hours necessary to obtain the best motif pairs by brute force.

E. Biological validation

Next, we assess the effectiveness of SLIDER on two of the largest real-life PPI-networks: the yeast network and the human network.

Retrieving the best motif pairs. We will first assess if M-SLIDER and SEQ-SLIDER are still capable of retrieving the best motif pairs on networks of this size. As the motif pairs which describe the interactions in the real PPI-networks are not known, we use the 1 000 best scoring motif pairs obtained by a brute force algorithm as the "ground truth". Hence, the notion "precision" is a bit misleading here because the real motifs describing the interactions are unknown and might not even exist because of the limitations of the (ℓ, d)-motif model. Nevertheless, from a purely theoretical point of view, calculating precision against the best scoring motif pairs is a correct and objective merit to assess the capability of our methods to find the best motif pairs *according to the model*. Moreover, because in this setting we are guaranteed to compare against all best scoring motif pairs, we do not have to rely on the positional similarity measure and can compare the two sets of motif pairs directly.

To give an idea, the brute force computation for (8,3)-motif pairs on the yeast network occupied about 100 nodes in the cluster spanning a period of 2 weeks.

We ran M-SLIDER and SEQ-SLIDER for 20 minutes exploiting all 8 cores of the Mac Pro. The average precision of the 1 000 best results returned by M-SLIDER over 5 runs is 14%, that of SEQ-SLIDER is 74.2%. The number implies that SEQ-SLIDER succeeds in recovering 742 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 SEQ-SLIDER returns a ranked list, these 742 motif pairs occur at the top.

Biological relevance of best motif pairs. We will now assess the biological relevance of the results of the brute force algorithm, SEQ-SLIDER and MotifHeuristics on the yeast network and the human network. We used our own implementation of MotifHeuristics, but allowed it to run significantly longer. We did not assess D-STAR, because even though D-STAR terminated on our artificial networks within 5 minutes, the method does not scale to larger networks. In particular, Leung et al. [11] 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 a month without result. We took protein structures from the protein databank (PDB) [3] and selected only those that could be mapped to proteins in the human and yeast networks (using `pdb_homologs.tab` from `yeastgenome.org` for yeast

and the GTOP database [10] for human), with blast e-value $<$ 1E-10. We discarded any structures where no two separate chains of the structure could be mapped to two interacting proteins in one of the networks, or where one or both of those proteins didn't contain a motif from the result. Subsequently, we used NACCESS [9] to calculate relative solvent accessibility (RSA) of each residue in the PDB structures. The higher RSA, the more at the surface a residue is. Protein sequences were aligned with PDB protein sequences, and in this way the solvent accessibility of residues covered by a correlated motif was obtained (see example in Fig. 9). This was done two times for each residue: once in the structure of the complex (two chains bound to each other) and once in the free protein chain. The solvent accessibility of these residues in the single proteins was compared with that in the protein complex structure. Residues which have a smaller accessibility in the complex, are considered to be at the interaction site. For example, for the residues listed in Fig. 9, the first, second, fourth and eighth residue, respectively `R`, `D`, `P` and `F`, have accessibility 35.6, 39.2, 33.3 and 7.5 in the single chain, but only 1.2, 18.0, 6.0 and 0 in the complex, which implies that that they are indeed at the interaction site.

Unfortunately, because of the limited available structure information, none of the proteins of the human network survived both the PDB-mapping and motif-filtering phase for (8,3)-motifs obtained by SEQ-SLIDER. The number of proteins remaining for yeast is also extremely small, as can be seen from Fig. 10. For that reason, we ran the brute force method and SEQ-SLIDER using (the less informative) (8,5)-motifs where we used all 8 cores of our machine for an hour and 15 minutes (for an equivalent of 10 hours of computation on a single core) for both the yeast and human network to increase the number of motif hits for which RSA values can be obtained. Each of these results gave us 1 000 motif pairs ranked by their χ^2-support. We ran our own implementation of MotifHeuristics for the equivalent of a month of computation time.

In order to see if the current (real) motif pair interface coverage is statistically significant, we prepared 100 sets of random motif pair occurrences in the sequences from the interaction network and analyzed how many of them have more motif pair interface coverage than the real data. These datasets were generated from the original result set by choosing a random new position for each motif hit in the sequence in which it appears. Results of this comparison are shown in Fig. 10.

Both for the yeast and human network we have significantly more overlap than random with the interface. Notably, for the human network only 2 out of the 100 random sets have at least 45% of their motifs overlapping with the interface (as observed for the SEQ-SLIDER motifs). In this run, the average of the percentage of motif hits overlapping with the interface is 36.5 for the random motif hits and the standard deviation 4.5. The fact that SEQ-SLIDER has more overlap with the interaction site than brute force can be explained by the more complementary nature of the SEQ-SLIDER motif pairs – their motif hits cover more regions in the sequences (see Supplementary Material).

We also ran MotifHeuristics on the large-scale networks. As the method did not return a single motif pair after ten hours, we allowed it to run for a full month, still producing less motif pairs than SEQ-SLIDER in a ten-hour run. We restrict the comparison to the same number of found motif pairs. SEQ-SLIDER still finds a larger overlap with the interface (See Fig.10.

Using an additional cutoff for the interface (i.e. not only requiring change in RSA upon complexation but also that RSA in free protein is above a cutoff) does not change much in analysis (data not shown).

Conclusion. We find significant overlap of motif hits with interface residues for SEQ-SLIDER, on both the yeast and human results. That being said, the results on human are remarkably better than those for yeast. Our experimental results seem to suggest that the model itself is better in describing the interactions in the human network than the interactions in the yeast network. A possible explanation for the skewness in these results is that the (ℓ, d) with χ^2-support model suffers more from false positives caused by indirect interactions, which are prominently present in the yeast network, than from false negatives, which are assumed to be common in the human network, as explained above.

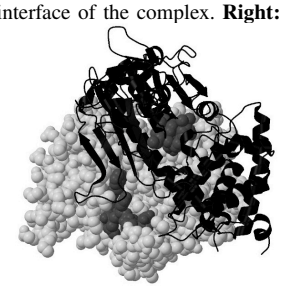
It might be worth pointing out that, as far as we know, this is the first effort to assess if CMM is able to produce biologically meaningful results from genome-wide PPI-networks.

VIII. Related Work

Steepest ascent is not only the oldest, but also the simplest among the known metaheuristics for combinatorial optimization [4]. Several others exist that would avoid getting stuck in local optima and move on to a better, global optimum. We tried simulated annealing with several parameters for its starting temperature, annealing schedule and acceptance function and found no improvement upon our steepest ascent algorithm, we even found it to generate worse results. The more advanced metaheuristics improve upon steepest ascent by escaping a local optimum by taking a few steps in a direction that decreases the support to gain access to a region from where a better local optimum can be reached. We checked if such a search path is feasible for the neighborhood function N^{seq}. As N^{seq} always takes its motifs from two proteins, we can visualize the search space (for one starting seed) as a 2D plane. Each point (x, y) on this plane represents the best support out of all possible motif pairs $\{X, Y\}$ where X starts at position x and Y at position y. We have visualized these search spaces for several interacting motif pairs in the yeast network and found that the local maxima are too far away from each other to be reached by such an approach. We also observed that the search space contains several positions where all neighbors have the same support. Steepest ascent would immediately stop at these points, where simulated annealing would continue to walk around randomly until it has moved its allotted steps. Hence, it appears that the search landscape of CMM is not suitable for these more advanced metaheuristics.

At first sight the present work seems highly related to the mining of frequent patterns in sequences (as for instance in [8]). It is therefore tempting to think about a method which

Fig. 9. **Left:** Mapping a motif hit of `RDxxxxNx` (rank 7, SEQ-SLIDER) in protein 18 010 of the human network to PDB 1Y8Q, chain C. The residues in bold are at the interaction site according to the RSA values. Its partner motif `GxGxxGxx` also occurs at the interface of the complex. **Right:** Two interacting chains C and D of PDB 1Y8Q in white and black and the two motif hits in gray.



Position in protein	**321**	**322**	323	**324**	325	326	327	**328**
Residue	**R**	**D**	P	**P**	H	N	N	**F**
Position in PDB	**322**	**323**	324	**325**	326	327	328	**329**
Residue	**R**	**D**	P	**P**	H	N	N	**F**
RSA (single chain)	**35.60**	**39.20**	8.90	**33.30**	21.00	1.20	0.80	**7.50**
RSA (complex)	**1.20**	**18.00**	8.90	**6.00**	21.00	1.20	0.80	**0.00**

Fig. 10. Occurrences at surface and at interaction site compared to random sampling.

Network (Nprot / Nint)	Yeast HC (1620/9060)			Human (8872/34230)	
Method	Brute force	Brute force	SEQ-SLIDER	Brute Force	SEQ-SLIDER
Parameters	χ^2, (8,3)	χ^2, (8,5)	χ^2, (8,5), 600min	χ^2, (8,5)	χ^2, (8,5), 600min
Proteins[a]	252	949	949	229	229
Motif hits[b]	48	5 335	1 157	188	137
At interaction site	13 (27%)	2 103 (39%)	335 (29%)	61 (32%)	62 (45%)
Random $\geq$ at interaction site[c]	37%	48%	12%	23%	2%

Method	SEQ-SLIDER	MotifHeuristics	SEQ-SLIDER	MotifHeuristics
Parameters	χ^2, (8,5), 600min	p, (8,5), 1 month	χ^2, (8,5), 600min	p, (8,5), 1 month
Results used	400	400	24	24
Proteins[a]	926	949	156	208
Motif hits[b]	817	615	13	14
At interaction site	319 (39%)	224 (36%)	8 (62%)	8 (57%)
Random $\geq$ at interaction site[c]	6%	50%	4%	8%

[a]Both proteins of a pair should contain at least one motif from the result.

[b]Number of motif-protein hits after filtering data such that only motif hits for which a complementary motif hit is present in an interacting protein (with both protein having an associated structure) are kept.

[c]The percentage of randomly generated motif hit datasets that have more hits at interaction sites than the result of the method.

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 proteins, whereas highly frequent motifs in yeast occur in up to 60 proteins as can be seen from the histogram in Fig. 11. Therefore, mining correlated motifs is very different from mining frequent motifs.

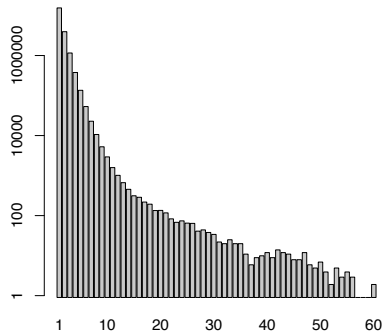


Fig. 11. Number of (8,3)-motifs (y-axis) selecting a given number of proteins in the yeast network (x-axis).

IX. 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 generic metaheuristic SLIDER based on the sliding window neighborhood 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 such as investigating candidate generation for motif pairs. A detailed comparison with interaction-driven approaches [12]–[14] should be done, 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 and our results suggest that this model suffers from false positives caused by indirect interactions. Although more expressive models exist (e.g., Position Weight Matrix or Hidden Markov Model), (ℓ, d)-motifs are very common in the field of bioinformatics. Moreover, Van Dijk et al. [18] already showed how motifs generated by D-STAR can be used to predict protein interactions in 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 [11] 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.

ACKNOWLEDGMENT

Peter Boyen is funded by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). A. D. J. van Dijk is supported by an NWO (Netherlands Organisation for Scientific Research) VENI grant (863.08.027). This research is supported by the BioRange programme (SP 2.3.1) of the Netherlands Bioinformatics Centre (NBIC), which is supported through the Netherlands Genomics Initiative (NGI) and the Research Programme of the Research Foundation Flanders (FWO) (G030607). This work was also sponsored by the BiG Grid project for the use of computing and storage facilities, with financial support from NWO.

REFERENCES

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

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

[3] H. Berman et al., "The Protein Data Bank", *Nucleic Acids Research*, 28 pp. 235–242 (2000)

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

[5] P. Boyen, F. Neven, D. Van Dyck, A. van Dijk, and R. van Ham, "SLIDER: Mining correlated motifs in protein-protein interaction networks," *Proc. The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 (ICDM 2009)*, pp. 716-721, Dec. 2009, doi:10.1109/ICDM.2009.92

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

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

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

[9] S. Hubbard, and J. Thornton, "NACCESS", Computer Program, Department of Biochemistry and Molecular Biology, University College London, 1993.

[10] T. Kawabata et al., "GTOP: a database of protein structures predicted from genome sequences", *Nucleic Acids Research*, 30: 294–298, 2002.

[11] H. Leung, M. Siu, S. Yiu, F. Chin, and K. Sung, "Finding linear motif pairs from protein interaction networks: A probabilistic approach," In *Computational Systems Bioinformatics (CSB)*, pp. 111–120, 2006.

[12] 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.

[13] 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.

[14] 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.

[15] T. Prasad et al., "Human Protein Reference Database - 2009 update", *Nucleic Acids Research* 37, D767–D772

[16] M. Stumpf, T. Thorne, E. de Silva, R. Stewart, H. An, M. Lappe, and C. Wiuf, "Estimating the size of the human interactome," *Proc Natl Acad Sci U S A*, 105(19):6959–64, 2008.

[17] S. Tan, W. Hugo, W. Sung, and S. Ng, "A correlated motif approach for finding short linear motifs from protein interaction networks," *BMC Bioinformatics*, 7:502+, November 2006.

[18] A. van Dijk, C. ter Braak, R. Immink, G. Angenent, and R. van Ham, "Predicting and understanding transcription factor interactions based on sequence level determinants of combinatorial control," *Bioinformatics*, 24(1):26–33, 2008.

[19] C. von Mering, R. Krause, B. Snel, M. Cornell, S. Oliver, S. Fields, and P. Bork, "Comparative assessment of large-scale data sets of protein-protein interactions," *Nature*, 417:399–403, 2002.

[20] 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.

Peter Boyen received the BSc degree in computer science from Hasselt University in 2005 and the MSc degree in computer science from Hasselt University in 2007. He is currently a PhD student with the Department of Computer Science at Hasselt University. His research interests include protein interaction networks and graph alignment.

Dries Van Dyck received his PhD degree in computer science from Ghent University in 2004 and currently holds a position at Hasselt University as postdoctoral teaching assistant. His main research interests are algorithmic graph theory, heuristics for hard (graph) problems, graph mining and mining biological data. For his PhD, he also worked on graph structural properties of cubic graphs.

Frank Neven received his PhD degree in Computer Science from Limburgs Universitair Centrum in 1999. Since 2001, he is a professor at Hasselt University. His main research interests are databases and data mining, formal languages and automata, and logic in computer science.

Roeland C.H.J. van Ham received his PhD degree in Biology from Utrecht University in 1994. At present, he is group leader Bioinformatics at Plant Research International, Wageningen, and associate professor Bioinformatics at Wageningen University. His research interest is broadly in plant bioinformatics and methodology for integrative analysis of omics data.

Aalt D.J. van Dijk received his PhD degree in Chemistry from Utrecht University in 2006. At present, he is a researcher in the Bioinformatics group at Plant Research International, Wageningen University and Research Centre. His main research interests are protein-protein interactions, transcription factor - DNA interactions, computational structural biology, and modelling the dynamics of networks of interacting proteins.