

# GraMoFoNe: a Cytoscape plugin for querying motifs without topology in Protein-Protein Interactions networks

Guillaume Blin, Florian Sikora, Stéphane Vialette

Université Paris-Est, LIGM - UMR CNRS 8049, France  
 {gblin, sikora, vialette}@univ-mlv.fr

## Abstract

During the last decade, data on Protein-Protein Interactions (PPI) has increased in a huge manner. Searching for motifs in PPI Network has thus become a crucial problem to interpret this data. A large part of the literature is devoted to the query of motifs with a given topology. However, the biological data are, by now, so noisy (missing and erroneous information) that the topology of a motif can be irrelevant. Consequently, Lacroix *et al.* [19] defined a new problem, called GRAPH MOTIF, which consists in searching a multiset of colors in a vertex-colored graph. In this article, we present GraMoFoNe, a plugin to Cytoscape based on a Linear Pseudo-Boolean optimization solver which handles GRAPH MOTIF and some of its extensions.

## 1. Introduction

Recent techniques increase data and knowledge about proteins ([14, 15, 29]). Among others proteins properties, the set of all their interactions for a given organism, called Protein-Protein Interactions (PPI) network, have gained huge interest in the last few years. A major stake of comparative analysis of PPI tries to determine to what extend proteins are conserved among species. Indeed, recent research suggests that proteins are functioning together into pathways and tend to evolve in correlated fashion – being preserved or eliminated in new species [21]. Therefore, it has become of foremost importance to identify PPI subnetworks that are *similar* to a given motif (*i.e.*, pathway of proteins), where similarity is measured both in terms of protein-sequence and subnetwork topology conservation.

In this context, most tools consider topology-based motifs (either a path [17, 27], a tree [22, 12], or a graph [12, 8]). However, interactions data were so noisy and incomplete that there is no need for topology information in the motif [9]. According to this remark, Lacroix *et al.* [19] have introduced the following problem named GRAPH MOTIF.

**Definition 1.1 (GRAPH MOTIF [19]).** *Given a vertex-colored graph  $G=(V,E)$  and a multiset of colors  $M$  (the motif), find a connected subset of vertices  $R \subseteq V$  whose multiset of colors equals  $M$  (*i.e.*, there is a bijection  $\sigma : R \rightarrow M$  such that  $\sigma(v) \in col(v)$  for all  $v \in R$ , where  $col(v)$  is the color of  $v$ ).*

In our context, the graph  $G$  represents the PPI network where vertices are the proteins and edges the interactions. The motif is completely defined by adding a color in  $M$  for each different requested proteins. Once the motif is defined, a node  $v \in G$  is colored by a color  $c \in M$  if the protein represented by  $v$  is homologous to the protein represented by  $c$  (*e.g.*, according to a BLASTp [5] analysis). If the protein represented by a node  $v$  is not homologous to any protein of the motif then  $v$  is not colored.

Despite the NP-completeness of the problem [19], some theoretical results exists [7, 13, 11]. Nevertheless, to the best of our knowledge, there is only one implemented tool, called Torque [9]. Torque uses either integer linear programming or dynamic programming conjugated with color coding technique [4]. Limitations of Torque are that it is a web service (therefore it is hard to connect with others services, the performances only depends on the server and it is not possible to perform batch tests), it only give one solution (not all possibles solutions) and, last but not least, it only deals with colorful motif (*i.e.*, at most one occurrence of each color).

When dealing with multiset motif – which may be of interest – two approaches can be highlighted. (i) A functional approach: using a Gene Ontology like classification [10], two proteins have the same color if they belong to the same class. (ii) An evolutive approach: two proteins have the same color in the motif if they are homologous. In our plugin, the second approach is used.

By searching for exact matches of a motif, we provide a new tool to solve GRAPH MOTIF [19]. It is worth noticing that our plugin also deals with some extensions of this problem. Indeed, due to the huge rate of noise in PPI Networks [14, 23], exact match are often too restrictive, and hence one may allow deletions (*i.e.*, proteins which are in the motif but not in the solution). The resulting problem is MAX MOTIF, defined by Dondi *et al.* [11]. Similarly, the resulting subnetwork may contain protein insertions (*i.e.*, proteins which are in the solution but not in the motif) that help to get the connectivity of the result. These proteins can be colored or not, as claimed in [9]. Moreover, our plugin allows to restrict motifs to colorful ones. Finally, since a protein can be homologous to more than one protein, a node  $v \in G$  can have more than one color. Hence, a set of colors (instead of only one

color) can be assigned to any node in order to deal with the LIST-COLORED GRAPH MOTIF problem settled by Betzler *et al.* [7]. In this latter problem, the bijection  $\sigma$  is still valid since  $col(v)$  then returns the list of colors assigned to  $v$ .

## 2. Methods and implementation

Our tool, named GraMoFoNe (which stands for Graph Motif For Networks), has been implemented as a Cytoscape plugin. Cytoscape [26] is a popular open-source software platform for network visualization and analysis, which supports the development of external plugin tools extending its functionalities. Our plugin seeks for occurrences of a user defined motif into a network previously loaded into the Cytoscape workspace (many file format are supported). It uses an exact algorithm to perform this task.

To this end, we choose to express our problem as a linear pseudo-boolean optimization problem (LPB), *i.e.*, as a linear program [25] whose variables are boolean. In a LPB problem, the objective is to find an assignment of boolean variables such that all constraints are satisfied and such the value of the linear objective function is optimized. Our LPB formulation is composed of 23 constraints defined upon 9 domains of variables (details are provided in the sequel). A large number of LPB solvers – which are generalization of SAT solvers – exists. We decided to use java SAT4JPseudo library [20] for (i) efficient java integration, (ii) its good result in the PB Evaluation 07 [2], and (iii) its free availability (efficient pure linear programming solver are indeed often expensive).

Our LPB program seeks for a connected occurrence of a multiset of colors, called motif,  $M$  (with  $k = |M|$ ) into the vertex-colored edge-weighted undirected graph  $G = (V, E, w)$ , where  $w$  is a function assigning a weight to any edge of  $E$ . Let  $R \subseteq V$  be a solution. Let  $N(v)$  represents the set of neighbors of  $v$  (*i.e.*,  $N(v) = \{u : \{u, v\} \in E\}$ ) and  $G[R]$  represents the subgraph of  $G$  induced by the set  $R$ . In the motif  $M$ , let  $occ_M(c)$ ,  $c \in \mathcal{C}$ , denotes the number of occurrences of color  $c$  in  $M$ . Let  $col : V \rightarrow 2^{\mathcal{C}}$  be a function which returns the list of colors of  $\mathcal{C}$  associated to any node of  $V$ .

As said previously, looking for exact matching can be too restrictive. We will allow insertions and deletions of proteins, and then,  $|R|$  would be different of  $k$ . Indeed, when  $|R| < k$  (resp.  $|R| > k$ ), we say that there are at least  $k - |R|$  deletions (resp.  $|R| - k$  insertions). The maximal number of deletions (resp. insertions) is denoted by  $N_{del}$  (resp.  $N_{ins}$ ). However, comparing  $k$  and  $|R|$  is not a sufficient condition for determining the number of indels (*i.e.*, insertions-deletions) in the solution. Indeed, if there is one deletion for a color and one insertion for another color, we certainly have  $|R| = k$  whereas  $R$  may be not in bijection with  $M$ . To deal with this fact, we have to consider for each color  $c$ , the difference between the

number of occurrences of  $c$  in the motif and the number of occurrences of nodes colored by  $c$  in  $R$ . Moreover, if a node  $v$  in the solution  $R$  is colored with more than one color,  $v$  must match only one color of  $M$  since  $\sigma$  is a bijection – other colors of  $v$  can not match other colors of  $M$ . Our LPB program deals with these two constraints.

Hereafter, we present the variables, the objective function and the constraints of our LPB program.

**Variables.** For any node  $v \in V$ , we have a variable  $x_v \in \{0, 1\}$  to denote the presence of  $v$  in the solution  $R$ :  $x_v = 1$  iff  $v \in R$ . For any edge  $\{u, v\} \in E$ , we have a variable  $e_{uv} \in \{0, 1\}$  to denote the presence of  $\{u, v\}$  in  $G[R]$ :  $e_{uv} = 1$  iff  $\{u, v\} \in G[R]$ .

As we will explain soon, there is  $k + N_{ins}$  different integers labels associated to nodes in  $R$  to ensure the connectivity of  $G[R]$ . Note that a node is labeled only if it is part of the solution. Thus, for any node  $v \in V$ , we have  $k + N_{ins}$  variables  $Label[v][i] \in \{0, 1\}$ , with  $1 \leq i \leq k + N_{ins}$ , to represent the “label” of  $v$ :  $Label[v][i] = 1$  iff  $v$  has the label  $i$ .

For any node  $v \in V$  and color  $c \in col(v)$ , we have variables  $ColV[v][c]$ , to represent the color of  $v$  used in the coloring function:  $ColV[v][c] = 1$  iff  $v$  is considered to have the color  $c$  in  $R$ . These variables are used to choose which color among the  $|col(v)|$  colors of  $v$  is chosen in the bijection with  $M$ . In fact, by allowing a list of colors for  $v$ , if  $v$  is in the solution,  $v$  may match up to  $|col(v)|$  colors of the motif. Since we want a bijection between colors of  $R$  and  $M$ , we have to choose which unique color will be considered for a given node.

For any color  $c \in \mathcal{C}$ , we have  $N_{ins} + 1$  variables  $nins_c[i]$ , with  $0 \leq i \leq N_{ins}$ , to represent the number of insertions for the color  $c$ :  $nins_c[i] = 1$  iff there are  $i$  insertions of nodes with the color  $c$ . Similarly, for any color  $c \in \mathcal{C}$ , we have  $N_{del} + 1$  variables  $ndel_c[i]$ , with  $0 \leq i \leq N_{del}$ , to represent the number of deletions for the color  $c$ :  $ndel_c[i] = 1$  iff there are  $i$  deletions of nodes with the color  $c$ .

For any color  $c \in \mathcal{C}$ , we have three variables  $IsExact_c$ ,  $IsIns_c$ ,  $IsDel_c$ , to indicate if there are some nodes colored with  $c$  in  $R$  which are inserted or deleted:  $IsExact_c = 1$  (resp.  $IsIns_c = 1$ ,  $IsDel_c = 1$ ) iff the number of nodes in  $R$  with the color  $c$  is equal to (resp. greater than, lower than)  $occ_M(c)$ . These variables are used for ease of exposition (*i.e.* there is an equivalence between these variables, and  $nins_c[0]$  and  $ndel_c[0]$ ).

**Objective.** The objective of the LPB program is to maximize the score of the solution. Our program maximizes the sum of all variables  $e_{uv}$  times their corresponding edge weight. In other words, it corresponds to maximizing the sum of edge weights of the solution. Formally, the objective is :  $\max \sum_{\{u,v\} \in E} e_{uv} w(\{u, v\})$

**Constraints.** The two following constraints ensure that the solution  $G[R]$  is a graph of correct size (according to  $k$ ,  $N_{ins}$  and  $N_{del}$ ).

$$\forall u, v \in V, \quad e_{uv} \Leftrightarrow x_u \wedge x_v \quad (1)$$

$$k - N_{del} \leq \sum_{v \in V} x_v \leq k + N_{ins} \quad (2)$$

Constraint (1) ensures that  $\{u, v\} \in G[R]$  iff both  $u$  and  $v \in R$ . Constraint (2) controls the number of nodes in the solution. When no indels are allowed, the size of the solution must be equal to  $k$ , the number of elements in the motif. When allowing insertions (resp. deletions), the size of the solution can be larger (resp. smaller) than  $k$ .

The four following constraints ensure the connectivity of  $G[R]$ .

$$\forall v \in V, \quad x_v \Rightarrow \left( \sum_{i=1}^{k+N_{ins}} Label[v][i] = 1 \right) \quad (3)$$

$$\forall v \in V, \quad \neg x_v \Rightarrow \left( \sum_{i=1}^{k+N_{ins}} Label[v][i] = 0 \right) \quad (4)$$

$$\forall 1 \leq i \leq k + N_{ins}, \quad \sum_{v \in V} Label[v][i] \leq 1 \quad (5)$$

$$\forall v \in V, \forall 1 \leq i < k + N_{ins},$$

$$Label[v][i] \Rightarrow \left( \sum_{u \in N(v)} \sum_{j>i} Label[u][j] \geq 1 \right) \quad (6)$$

Constraint (3) ensures that if  $v \in R$ , then  $v$  has exactly one label, an integer between 1 and  $k + N_{ins}$ . Constraint (4) ensures that if  $v \notin R$ , then  $v$  is unlabeled. Constraint (5) ensures that any label is attributed to at most one node. Due to deletions, some labels may be not attributed. Constraint (6) ensures the connectivity of  $G[R]$ : any node of  $R$ , except the one with the maximal label, must have a neighbor in  $G[R]$  with a label greater than its own.

The two following constraints ensure that  $G[R]$  has enough colored vertex according to  $occ_M(c)$  for any  $c \in \mathcal{C}$ ,  $N_{ins}$  and  $N_{del}$ .

$$\forall c \in \mathcal{C}, occ_M(c) - N_{del} \leq \sum_{\substack{v \in V \\ c \in col(v)}} x_v \leq occ_M(c) + N_{ins} \quad (7)$$

$$\forall v \in V, \quad \sum_{c \in col(v)} ColV[v][c] = x_v \quad (8)$$

Constraint (7) ensures that for any color  $c$  in  $M$ , there is enough vertices colored with  $c$  in  $G[R]$ . Where no indels are allowed, a solution must contain  $occ_M(c)$  occurrences of  $c$ , for each color  $c$ . Since insertions of colored

nodes (resp. deletions) are allowed, the number of occurrences of a color can be larger (resp. smaller). Constraint (8) ensures that a unique color for any node  $v$  in  $R$  is selected among its  $|col(v)|$  associated colors.

The three following constraints ensure that either all occurrences of a color  $c \in \mathcal{C}$  in  $M$  are matched, or at least one of them is inserted or deleted.

$$\forall c \in \mathcal{C}, \quad IsExact_c + IsIns_c + IsDel_c = 1 \quad (9)$$

$$\forall c \in \mathcal{C}, \quad \sum_{v \in V} ColV[v][c] - occ_M(c) \leq IsIns_c \cdot N_{ins} - IsDel_c \quad (10)$$

$$\forall c \in \mathcal{C}, \quad \sum_{v \in V} ColV[v][c] - occ_M(c) \geq$$

$$- IsExact_c - IsDel_c - IsDel_c \cdot N_{del} \quad (11)$$

Constraint (9) ensures the above assertion whereas constraints (10) and (11) ensure the consistency between  $ColV$ ,  $IsExact$ ,  $IsIns$ ,  $IsDel$ :  $\forall c \in \mathcal{C}$ ,  $IsExact_c$  (resp.  $IsIns_c$ ,  $IsDel_c$ ) = 1 iff  $\sum_{v \in V} ColV[v][c] - occ_M(c) = 0$  (resp.  $> 0$ ,  $< 0$ ).

The six following constraints ensure that the number of insertions is less than  $N_{ins}$ .

$$\forall c \in \mathcal{C}, \quad \sum_{i=0}^{N_{ins}} nins_c[i] = 1 \quad (12)$$

$$\forall c \in \mathcal{C}, \quad IsIns_c \Rightarrow nins_c[0] = 0 \quad (13)$$

$$\forall c \in \mathcal{C}, \quad \neg IsDel_c + nins_c[0] \geq 1 \quad (14)$$

$$\forall c \in \mathcal{C}, \forall 0 \leq i \leq N_{ins},$$

$$\sum_{v \in V} ColV[v][c] - occ_M(c) \leq i \cdot nins_c[i] + \neg nins_c[i] \cdot N_{ins} \quad (15)$$

$$\forall c \in \mathcal{C}, \forall 0 \leq i \leq N_{ins},$$

$$\neg nins_c[i] + \sum_{v \in V} ColV[v][c] - occ_M(c) + N_{del} \cdot IsDel_c \geq$$

$$i \cdot nins_c[i] \quad (16)$$

$$\sum_{c \in \mathcal{C}} \sum_{i=1}^{N_{ins}} i \cdot nins_c[i] + \sum_{\substack{v \in V \\ col(v)=\emptyset}} x_v \leq N_{ins} \quad (17)$$

Constraint (12) ensures that, for a given color  $c \in \mathcal{C}$ , there is a unique variable  $nins_c$  that corresponds to the number of insertions of nodes with color  $c$ . Constraint (13) ensures that variables  $nins_c$  and  $IsIns_c$  are consistent. Constraint (14) ensures that for a color  $c \in \mathcal{C}$  there

are either insertions or deletions. Constraint (15) and (16) ensure that  $nins_c[i] = 1$  iff there are  $i$  insertions of nodes with the color  $c \in \mathcal{C}$  (i.e. if the difference between  $\sum_{v \in V} ColV[v][c]$  and  $occ_M(c)$  is equal to  $i$ ). Constraint (17) ensures that the number of insertions is bounded by  $N_{ins}$ . The sum of all the insertions for a given color in addition to insertions of not colored nodes have to be less than  $N_{ins}$ .

We also give six constraints, which are built similarly to constraints (12)-(17).

**Lemma 2.1** *Our LPB program correctly solves GRAPH MOTIF.*

Proof omitted due to space constraints.

Let us now define two preprocessing steps to speed up our LPB program.

First, let us remark that a protein in the motif without any homologous protein in the network will be considered as a deletion in any feasible result. Let  $D$  be the set of all colors corresponding to such proteins in the motif  $M$ . If the size of  $D$  exceeds  $N_{del}$ , then no solution is possible for this motif. Otherwise, we already know that all proteins corresponding to colors in  $D$  will be deleted in any solution. Thus, we launch the LPB program over the motif  $M \setminus D$ , with  $N_{del} - |D|$  allowed deletions.

Then, we prune the network and run the LPB solver on each connected component as shown in [9]. Indeed, a not colored node of  $G$  can be too “far” from any colored node, in terms of shortest path length, to be inserted in the solution in regards to the maximum number of allowed insertions (i.e.,  $N_{ins}$ ). According to this remark, we only keep a colored node  $u$  in  $G$  if there exist two colored nodes  $v_1$  and  $v_2$  such that  $dist(u, v_1) + dist(u, v_2) \leq N_{ins} + 1$ , where  $dist(u, v)$  is the length of the shortest path between  $u$  and  $v$ . Otherwise,  $u$  would never be part of a solution, and hence can be safely deleted from  $G$ .

Once  $G$  is pruned, the LPB program is used on each *valid* connected component of  $G$ . A component is said to be *valid* if it contains at least  $k - N_{del}$  colored nodes. Otherwise, a connected solution would never be found in this component, and hence there is no need to consider it. As stated in [9], there is in practice only 5% of colored nodes in  $G$ .

### 3. GraMoFoNe Functionalities

Screenshots of our plugin can be seen on the GraMoFoNe website<sup>1</sup>. The user can provide input data and parameters on the left sidebar, networks are drawn in the center and results are presented on the right panel. We now describe inputs and outputs of GraMoFoNe.

#### Inputs

*The network and the motif.* The network has to be loaded into the Cytoscape environment. The motif is either (1) a predefined motif, (2) or given manually in a textbox, (3) or loaded as a FASTA file.

*BLASTp.* Since we consider two proteins as homologous according to their sequence similarity by a BLASTp analysis, we need FASTA files of the motif and the network. These last can be provided by the user; otherwise, our plugin tries to retrieve them from the Uniprot database Archive (Uniparc) [6] using EBI Web Services [18]. The user has also to provide the BLASTp threshold value: two proteins are homologous if their  $-\log(eValue)$  value is above this threshold.

*Indels.* The user can provide a maximum number of deletions and insertions allowed in a solution, and the corresponding penalty costs used to compute the score of a result.

#### Outputs

Once GraMoFoNe routine is launched, the plugin provides the potential subnetworks list, ordered by their scores, while Torque only provides the best solution. The user may see each of these subnetworks highlighted in the full network. The plugin also provides the correspondence between proteins in the result and the motif. Finally, the plugin allows an exportation of any such subnetwork as a new network.

## 4. Results and comparison

To validate our plugin on real data, we launched a batch mode of our plugin (not available through Cytoscape) which tries to retrieve motifs (protein complexes) of six different species in three large different PPI networks.

#### Data acquisition and parameters

The PPI networks of *Saccharomyces cerevisiae* (Yeast, about 5.500 proteins and 40.000 interactions), *Drosophila melanogaster* (Fly, about 6.500 proteins and 21.000 interactions) and *Homo sapiens* (about 8.000 proteins and 29.000 interactions) were downloaded from the Torque website. They obtain these data from recent papers and public databases.

The motifs data for Yeast, Fly, Human, Mouse, Bovine and Rat were kindly supplied by Torque authors which obtained them from the databases SGD [3], AmiGo[1] and Corum[24].

Fasta files for Yeast, Fly and Human were downloaded from the Torque website, while data for Mouse, Bovine and Rat were downloaded from Biomart [28]. Missing informations have been manually added from Uniprot [6] and Ensembl [16] databases.

The parameters have been set as similar as possible as in Torque. Therefore, the threshold value for BLASTp has been set to  $-\log(10^{-7}) \simeq 16.1$ . Two insertions ( $N_{ins}$ ) and deletions ( $N_{del}$ ) were allowed for small motifs (size  $< 7$ ), three for medium motifs (size 8-14), four for larger ones. The timeout for the LPB program was set to 500 seconds.

#### Experiments

Our tests were done on a 3GHz Personal Computer, with 2Go RAM memory. Torque values were not com-

<sup>1</sup><http://igm.univ-mlv.fr/AlgoB/gramofone/>

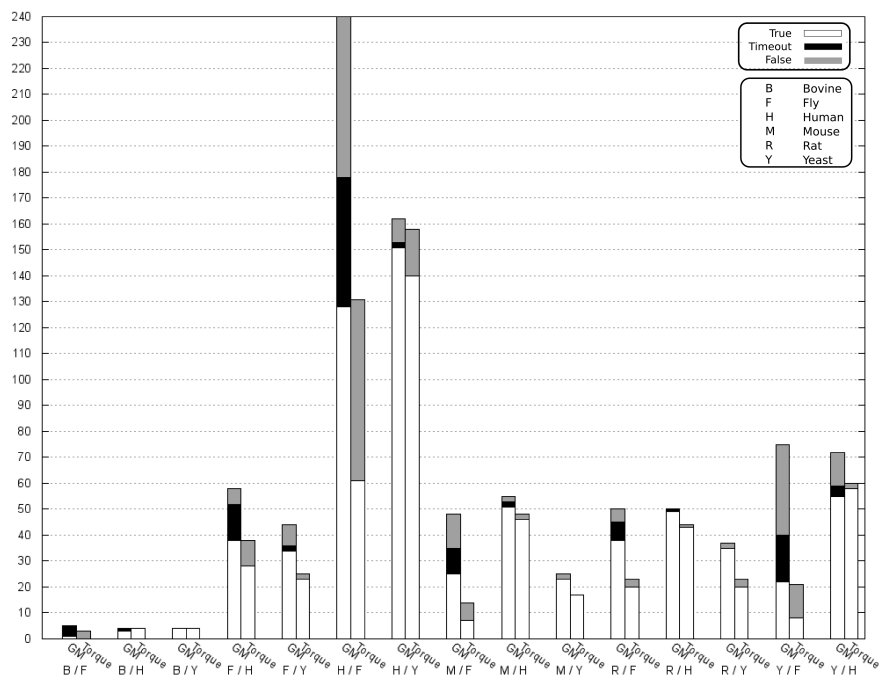


Figure 1: Comparison of the number of matches between our software (GM) and Torque [9]. Each histogram labeled by  $X/Y$  corresponding to retrieving a list of motif of specie  $X$  in the network of specie  $Y$ . White (resp. grey) bars corresponds to feasible motifs founded (resp. not founded) in the network. Black bars correspond to motifs where the timeout limit has been reached before any result. Hence, the whole bar correspond to feasible motifs.

puted by ourself since there is only a web service for Torque. We obtained values (number of matches) from the Torque paper. Values for GraMoFoNe were computed as follows.

From the list of motifs of a given species, we kept only feasible ones. We performed preprocessing on motif and network as described previously. Then, we considered a motif as feasible when, (i) its size was between 4 and 25, (ii) there were less than  $N_{del}$  proteins in the motif without homologous proteins in the network, and (iii) there was at least one connected component in the network with enough colors to match the motif.

Afterwards, for a feasible motif, the LPB program could found a solution (True in Figure 1), or found that this motif can not be matched in this network (False in Figure 1), or not finish under the time limit (Timeout in Figure 1).

### Results

Comparisons between our plugin GraMoFoNe and Torque are given in Figure 1. For most experiments, our plugin finds more feasible motifs (*i.e.*, the sum of “true”, “timeout” and “false” in the figure, or the height of each whole bar) and also more matches (*i.e.*, height of white bars) than Torque. These results can be due to differences in our preprocessing methods and to our manual addition of missing information in Fasta files.

As Torque, we can query motifs where there is no in-

formation about the motif topology (Bovine, Mouse and Rat). Also as in Torque, we had more unmatched feasible motifs when they are requested in the fly network. According to Torque authors, this is because the fly data is more noisy, with a high rate of false negatives. A motif can not be found if a false negative disconnects a potential solution. Conversely, false positives does not disturb the connectivity, but can create “bad” solutions.

With the set of parameters defined previously, there is no significant differences in terms of number of matches when we use a motif as a multiset (*i.e.* when two homologous proteins in the motif has the same color) or not.

Knowing if there is a match can be computed in seconds (5-20 for small motifs, 40-60 for larger ones), but the time to found the best solution can be longer. But, due to the use of a LPB solver as a “black box”, it is very hard to predict times.

## 5. Conclusion

In this paper, we presented GraMoFoNe, a new tool to request motifs (multiset of proteins without topology) into Protein-Protein Interactions network by solving GRAPH MOTIF and some of its extensions, to increase knowledge about biological network. This tool is given as a plugin for Cytoscape, a popular software to manage such networks. GraMoFoNe use the free Linear Pseudo Boolean

solver Sat4JPseudo to give all possible solutions, including the best one.

Since giving all solution can take time, our tool can also give the first solution founded by the LPB solver in short time. However, in this case, we do not know the quality of this solution compared to the best one (*i.e.* if there is another solution with less indels). A future work could be to find a fast heuristic to find a “good” solution in most case, and to compare this last with GraMoFoNe.

Our coloration method is only given in terms of sequence similarity. Therefore, it would be interesting to extend it to other measures. In the same way, our threshold for homologies is fixed. It would be also interesting to have a variable threshold.

The GraMoFoNe plugin and batch program are under GPL license and available at the website <http://igm.univ-mlv.fr/AlgoB/gramofone/>

## 6. Acknowledgement

The authors would like to thank Anne Parrain for her help and her quick response to our requests for SAT4JPseudo. We also thanks Sharon Bruckner for providing motifs data, Fasta files and Torque technical details. We thanks Vincent Lacroix for his ideas about using multiset motif.

## 7. References

- [1] Go consortium. amigo. <http://amigo.geneontology.org>, sept 2008.
- [2] Pb evaluation 07 – special event of the sat 2007 conference. <http://www.cril.univ-artois.fr/PB07/>.
- [3] Sgd project. “saccharomyces genome database”. <http://www.yeastgenome.org>, sept 2008.
- [4] N. Alon, R. Yuster, and U. Zwick. Color coding. *JACM*, 42(4):844–856, 1995.
- [5] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *JMB*, 215(3):403–410, 1990.
- [6] A. Bairoch, R. Apweiler, et al. The universal protein resource (UniProt). *NAR*, 33:D154, 2005.
- [7] N. Betzler, M. Fellows, C. Komusiewicz, and R. Niedermeier. Parameterized algorithms and hardness results for some graph motif problems. In *CPM*, volume 5029 of *LNCS*, pages 31–43, 2008.
- [8] G. Blin, F. Sikora, and S. Vialette. Querying Protein-Protein Interaction Networks. In *ISBRA*, volume 5542 of *LNBI*, pages 52–62, 2009.
- [9] S. Bruckner, F. Hüffner, R. M. Karp, R. Shamir, and R. Sharan. Topology-free querying of protein interaction networks. In *RECOMB*. Springer, 2009.
- [10] T. G. O. Consortium. Gene Ontology: tool for the unification of biology. *Nature Genet*, 25:25–29, 2000.
- [11] R. Dondi, G. Fertin, and S. Vialette. Maximum Motif Problem in Vertex-Colored Graphs. In *CPM*, 2009.
- [12] B. Dost, T. Shlomi, N. Gupta, E. Ruppin, V. Bafna, and R. Sharan. QNet: A Tool for Querying Protein Interaction Networks. *RECOMB*, pages 1–15, 2007.
- [13] M. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In *ICALP*, volume 4596 of *LNCS*, pages 340–351, 2007.
- [14] A. Gavin, M. Boshe, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 414(6868):141–147, 2002.
- [15] Y. Ho, A. Gruhler, et al. Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, 415(6868):180–183, 2002.
- [16] T. Hubbard, B. Aken, et al. Ensembl 2009. *NAR*, 37:D690, 2009.
- [17] B. Kelley, R. Sharan, R. Karp, T. Sittler, D. E. Root, B. Stockwell, and T. Ideker. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *PNAS*, 100(20):11394–11399, 2003.
- [18] A. Labarga, F. Valentin, M. Anderson, and R. Lopez. Web services at the European bioinformatics institute. *NAR*, 35:W6, 2007.
- [19] V. Lacroix, C. Fernandes, and M.-F. Sagot. Motif search in graphs: application to metabolic networks. *TCCB*, 3(4):360–368, 2006.
- [20] D. Le Berre and A. Parrain. On extending sat solvers for pb problems. In *RCRA*, 2007.
- [21] M. Pellegrini, E. Marcotte, M. Thompson, D. Eisenberg, and T. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *PNAS*, 96(8):4285–4288, 1999.
- [22] R. Pinter, O. Rokhlenko, E. Yeager-Lotem, and M. Ziv-Ukelson. Alignment of metabolic pathways. *Bioinformatics*, 21(16):3401–3408, 2005.
- [23] T. Reguly, A. Breitkreutz, et al. Comprehensive curation and analysis of global interaction networks in *saccharomyces cerevisiae*. *Journal of Biology*, 2006.
- [24] A. Ruepp, B. Brauner, I. Dunger-Kaltenbach, et al. CO-REM: the comprehensive resource of mammalian protein complexes. *NAR*, 2007.
- [25] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, 1998.
- [26] P. Shannon, A. Markiel, O. Ozier, et al. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, 13:2498–2504, 2003.
- [27] T. Shlomi, D. Segal, E. Ruppin, and R. Sharan. QPath: a method for querying pathways in a protein-protein interaction network. *BMC Bioinformatics*, 7:199, 2006.
- [28] D. Smedley, S. Haider, et al. BioMart – biological queries made easy. volume 10, page 22. BioMed Central Ltd, 2009.
- [29] P. Uetz, L. Giot, et al. A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, 403(6770):623–627, 2000.

## 8. Appendix

### 8.1. Constraints to bound the number of deletions

The six following constraints ensure that the number of deletions is lower than  $N_{del}$ .

$$\forall c \in \mathcal{C}, \quad \sum_{i=0}^{N_{del}} ndel_c[i] = 1 \quad (18)$$

$$\forall c \in \mathcal{C}, \quad \text{IsDel}_c \Rightarrow ndel_c[0] = 0 \quad (19)$$

$$\forall c \in \mathcal{C}, \quad \neg \text{IsIns}_c + ndel_c[0] \geq 1 \quad (20)$$

$$\begin{aligned} \forall c \in \mathcal{C}, \forall 0 \leq i \leq N_{del}, \\ - \sum_{v \in V} \text{ColV}[v][c] + occ_M(c) \leq i \cdot ndel_c[i] + \neg ndel_c[i] \cdot N_{del} \end{aligned} \quad (21)$$

$$\begin{aligned} \forall c \in \mathcal{C}, \forall 0 \leq i \leq N_{del}, \\ - ndel_c[i] - \sum_{v \in V} \text{ColV}[v][c] + occ_M(c) + N_{ins} \cdot \text{IsIns}_c \geq i \cdot ndel_c[i] \end{aligned} \quad (22)$$

$$\sum_{c \in \mathcal{C}} \sum_{i=1}^{N_{del}} i \cdot ndel_c[i] \leq N_{del} \quad (23)$$

Constraint (18) ensures that, for a given color  $c \in \mathcal{C}$ , there is a unique variable  $ndel_c$  that corresponds to the number of deletions for  $c$ . Constraint (19) ensures that variables  $ndel_c$  and  $\text{IsDel}_c$  are consistent. Constraint (20) ensures that for a color  $c \in \mathcal{C}$  there are either deletions or insertions. Constraint (21) and (22) ensure that  $ndel_c[i] = 1$  iff there are  $i$  deletions for the color  $c \in \mathcal{C}$  (*i.e.* if the difference between  $occ_M(c)$  and  $\sum_{v \in V} \text{ColV}[v][c]$  is equal to  $i$ ). Constraint (23) ensures that the number of deletions is bounded by  $N_{del}$ . The sum of all the deletions for a given color have to be less than  $N_{del}$ .

### 8.2. Proof of Lemma 2.1

**Proof** We first prove the Lemma considering that no indels are allowed. The extension to the case allowing indels is straightforward and is given afterwards.

Let us first prove that a solution to GRAPH MOTIF can be found by our LPB program *i.e.* that it has a corresponding variables assignment that respects all the LPB constraints previously defined.

Given a solution  $G[R]$  to GRAPH MOTIF, set  $x_v = 1$  if  $v \in R$ ;  $x_v = 0$  otherwise and  $e_{u,v} = 1$  if  $u$  and  $v \in R$ ,  $e_{u,v} = 0$  otherwise. Find a spanning tree  $T$  of  $G[R]$  (this tree exists since  $G[R]$  is connected) and label the nodes of  $G[R]$  according to a postorder traversal of  $T$ .

Since no indels are allowed,  $|R| = k$ . By definition, exactly  $k$  variables  $x_v$  are equal to 1 and thus constraints (1) and (2) hold. The labeling induced by the postorder traversal of  $T$  ensures that all variables of  $R$  have a unique and distinct label. Therefore, constraints (3) and (5) hold. Since no label are given in nodes not belonging to  $R$ , Constraint (4) holds. Moreover, in  $T$ , according to the postorder traversal, the father of any node  $v$ , except the root, has a label greater than  $v$ . Therefore, in  $G[R]$ , any node has at least one neighbor with a greater label. Thus, Constraint (6) holds. Since  $\sigma : R \rightarrow M$  is a bijection, there is exactly  $occ_M(c)$  occurrences of each color  $c \in \mathcal{C}$  in  $R$ , and hence, Constraint (7) holds. Moreover, there is only one image  $\sigma(v)$  associated to any  $v \in R$ , thus the sum in (8) is equal to 1 and the Constraint holds when  $x_v = 1$ . By the bijection  $\sigma : R \rightarrow M$ , any element in  $M$  is associated to an element in  $R$ . Thus, no node  $v \notin R$  has an image in  $M$ , the sum in (8) is equal to 0 and the Constraint holds also when  $x_v = 0$ . Since no indels are allowed, any color  $c$  is matched. Thus, Constraint (9) holds. Moreover,  $\sum_{v \in V} \text{ColV}[v][c] - occ_M(c)$  is equal to 0. Constraints (10) and (11) hold iff  $\text{IsExact}_c = 1$ . Indeed, if  $\text{IsIns}_c = 1$ , Constraint (11) does not hold ( $0 \geq 1$ ), and if  $\text{IsDel}_c = 1$ , Constraint (10) does not hold ( $0 \leq -1$ ). For each color  $c$ , constraints (12) to (23) hold if  $nins_c[0] = 1$  and  $ndel_c[0] = 1$ , which is the case when no indels are allowed.

Let us now prove that a solution to our LPB program corresponds to a solution to GRAPH MOTIF.

Given a LPB solution, for any  $v \in V$ , add  $v$  to  $R$  if  $x_v = 1$ . Constraint (2) ensures that we have  $|R| = k$ . According to constraints (2) and (7),  $R$  and  $M$  are finite sets with exactly the same number of elements (*i.e.*,  $|R| = |M|$ ). By Constraint (8), if  $\sigma(v)$  is defined, then  $v \in R$  (otherwise,  $x_v = 0$  and all variables  $ColV[v][c]$  are equal to 0 for this  $v$  and for all  $c \in col(v)$ ). By constraints (7) and (8), for any  $c \in M$ , there is only one  $v \in R$  such that  $\sigma(v) = c$  (a node  $v$  can match at most one color and there are exactly the same number of elements in  $R$  and  $M$ ). Thus, on the whole,  $\sigma : R \rightarrow M$  is a bijection. It remains to show that  $G[R]$  is connected.

By Constraint (3), every node in  $R$  has a label. Let  $r$  be the node with the greatest label. By Constraint (5), this label is unique. Let us show that there exists a path in  $R$  connecting any node  $v \in R$  to  $r$ . To do so, let us prove by induction that there is a path from  $v$  to  $r$  with increasing labels. The case  $v = r$  is trivial. Suppose there exists a path  $p$  in  $R$  of length  $l$  starting from  $v$  with increasing labels. Let  $s_p$  be the sink of  $p$  (*i.e.* the last node in  $p$ ). If  $s_p = r$ , then we are done. Otherwise, by Constraint (6),  $s_p$  has at least one neighbor  $u$  with a label greater than its own. Then, there is a path  $p \cup \{u\}$  of length  $l + 1$  with increasing labels.

Let us prove Lemma 2.1 when indels are allowed.

Let first show that constraints (9) to (11) are consistent when indels are allowed. We already have shown that these constraints hold if there is no indels.

- If there are  $i$  insertions for a color  $c$ , then,  $\sum_{v \in V} ColV[v][c] - occ_M(c) = i$ . Constraint (9) ensure that only one variable among  $IsExact_c$ ,  $IsIns_c$  and  $IsDel_c$  is equal to 1.
  - If  $IsExact_c = 1$ , then constraints (10) ( $i \leq 0$ ) and (11) ( $i \geq 0$ ) are both true iff  $i = 0$ .
  - If  $IsIns_c = 1$ , then constraints (10) ( $i \leq N_{ins}$ ) and (11) ( $i \geq 0$ ) are both true iff  $0 \leq i \leq N_{ins}$ , which is the case here.
  - If  $IsDel_c = 1$ , then Constraint (10) ( $i \leq -1$ ) does not hold since the number of insertions is positive ( $i > 0$ ).
- If there are  $d$  deletions for a color  $c$ , then,  $\sum_{v \in V} ColV[v][c] - occ_M(c) = -d$ .
  - If  $IsExact_c = 1$ , then constraints (10) ( $-d \leq 0$ ) and (11) ( $-d \geq 0$ ) are both true iff  $d = 0$ .
  - If  $IsIns_c = 1$ , then Constraint (11) ( $-d \geq 1$ ) does not hold since the number of deletions is positive ( $d > 0$ ).
  - If  $IsDel_c = 1$ , constraints (10) ( $-d \leq -1$ ) and (11) ( $-d \geq 1 - 1 - N_{del}$ ) are both true iff  $-N_{del} \leq -d \leq -1$ , which is the case here.

Let us now show that constraints (12) to (16) and constraints (18) to (22) are consistent with the number of indels for a given color in a solution of GRAPH MOTIF.

- If there are  $i$  insertions for a color  $c$ , then,  $\sum_{v \in V} ColV[v][c] - occ_M(c) = i$  and  $IsIns_c = 1$ ,  $IsDel_c = IsExact_c = 0$ . Constraints (12) and (13) ensure that there is one  $i \neq 0$  s.t.  $nins_c[i] = 1$ . Constraint (14) holds since  $IsDel_c = 0$ . Constraints (15) and (16) both hold iff  $nins_c[i] = 1$  ( $i \leq i$  and  $i \geq i$ ). Otherwise, if  $nins_c[j] = 1, j \neq i$ , constraints (15) and (16) hold iff we have  $j \leq i \leq j$  ( $i \leq j$  and  $i \geq j$ ), which is impossible since  $j \neq i$ .  
Since  $IsIns_c = 1$ , Constraint (20) holds iff  $ndel_c[0] = 1$ . Then, Constraint (18) holds. Variable  $IsDel_c = 0$ , thus Constraint (19) holds. Hereafter, constraints (21) ( $-i \leq 0$ ) and (22) ( $-i + N_{ins} \geq 0$ ) hold when  $ndel_c[0] = 1$ .
- If there are  $d$  deletions for a color  $c$ , then  $\sum_{v \in V} ColV[v][c] - occ_M(c) = -d$  and  $IsDel_c = 1$ ,  $IsIns_c = IsExact_c = 0$ . Thus, Constraint (13) holds. Since  $IsDel_c = 1$ , Constraint (14) holds iff  $nins_c[0] = 1$ . Thus, Constraint (12) holds. Hereafter, constraints (15) ( $-d \leq N_{ins}$ ) and (16) ( $-d + N_{del} \geq 0$ ) hold when  $nins_c[0] = 1$ .  
Constraints (20) holds since  $IsIns_c = 1$ . Constraints (18) and (19) ensure that there is one  $i \neq 0$  s.t.  $ndel_c[i] = 1$ . Constraints (21) and (22) both holds iff  $ndel_c[d] = 1$  ( $d \leq d$  and  $d \geq d$ ). Otherwise, if  $ndel_c[j] = 1, j \neq d$ , constraints (21) and (22) hold iff we have  $j \leq d \leq j$  ( $d \leq j$  and  $d \geq j$ ), which is impossible since  $j \neq d$ .

In both case, constraints (17) and (23) hold iff the overall number of insertions and deletions are respectively less than  $N_{ins}$  and  $N_{del}$ .