



Recherche de motifs dans des graphes colorés

Florian Sikora
(encadré par Guillaume Blin et Stéphane Vialette)

Université Paris-Est, LIGM - UMR CNRS 8049

Séminaire Combi LINA – 06/2010

Plan

Introduction

Complexité paramétrée et motifs avec topologie

Motifs sans topologie

Le problème GRAPH MOTIF

Des logiciels pour GRAPH MOTIF

Conclusion

Plan

Introduction

Complexité paramétrée et motifs avec topologie

Motifs sans topologie

Le problème GRAPH MOTIF

Des logiciels pour GRAPH MOTIF

Conclusion

Motivations



25000



30000



45000

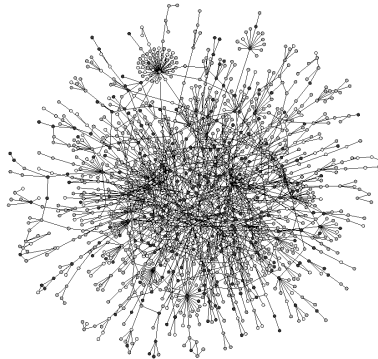
- ▶ Complexité de l'homme $\Leftrightarrow$ # de gènes ?
- ▶ Complexité de l'homme $\Leftrightarrow$ protéines ?

Les protéines..

- ▶ Nouveaux intérêts concernant les protéines...
- ▶ ... et sur leurs interactions: Protein-Protein Interactions (PPI)
- ▶ Obtenues biologiquement... avec beaucoup de bruit !

Réseau de protéines

- Les protéines peuvent interagir avec d'autres protéines



Réseau de protéines



- Modélisation par un **graphe** (éventuellement pondéré)
 - Les **protéines** sont représentées par les **nœuds**
 - Les **interactions** sont représentées par les **arêtes**
 - Les arêtes peuvent être pondérées par la probabilité de l'interaction

Motivations

- ▶ Nouvelles techniques : l'information augmente très rapidement [SHARAN & IDEKER 2006]
 - ▶ 2001: quelques centaines d'interactions
 - ▶ 2006: plusieurs milliers
- ▶ Beaucoup de BDD

Motivations

- ▶ Nouvelles techniques : l'information augmente très rapidement [SHARAN & IDEKER 2006]
 - ▶ 2001: quelques centaines d'interactions
 - ▶ 2006: plusieurs milliers
- ▶ Beaucoup de BDD
- ▶ Chercher des motifs pour retrouver des fonctions connues
- ▶ **Déduire** les informations d'espèces **peu connues** depuis des espèces **bien connues**

Plan

Introduction

Complexité paramétrée et motifs avec topologie

Motifs sans topologie

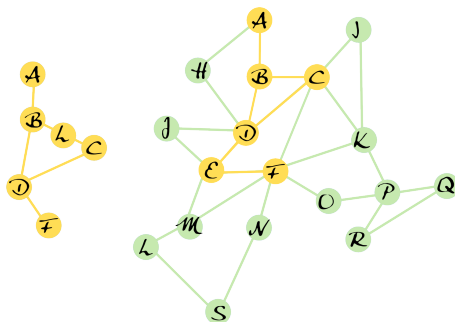
Le problème GRAPH MOTIF

Des logiciels pour GRAPH MOTIF

Conclusion

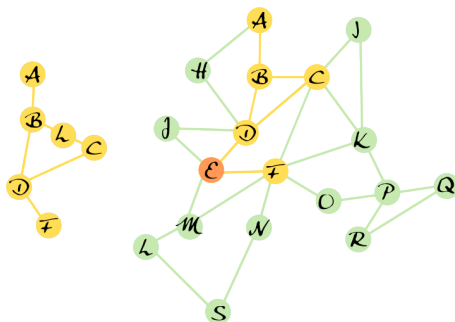
Rechercher des motifs

- ▶ Rechercher des motifs (ici, ensemble de protéines avec une topologie) dans un réseau PPI
- ▶ Une protéine est dite **homologue** à une autre protéine selon une analyse de séquences (avec BLASTp)



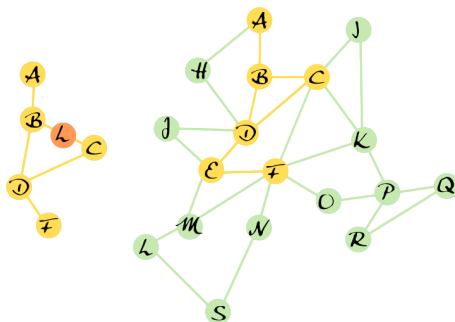
Rechercher des motifs

- ▶ Rechercher des motifs (ici, ensemble de protéines avec une topologie) dans un réseau PPI
- ▶ Une protéine est dite **homologue** à une autre protéine selon une analyse de séquences (avec BLASTp)
- ▶ Un nombre borné d'**insertions** et deletions peut-être autorisé



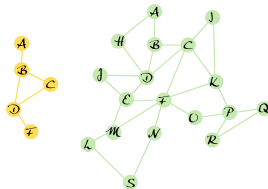
Rechercher des motifs

- ▶ Rechercher des motifs (ici, ensemble de protéines avec une topologie) dans un réseau PPI
- ▶ Une protéine est dite **homologue** à une autre protéine selon une analyse de séquences (avec BLASTp)
- ▶ Un nombre borné d'insertions et **deletions** peut-être autorisé



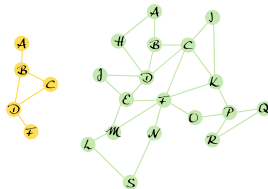
Recherche avec topologie dans un réseau PPI

- Le motif peut être un **chemin**, un **arbre**, un **graphe**
- Problèmes **NP-complets** donc une solution exacte entraîne une **complexité exponentielle**



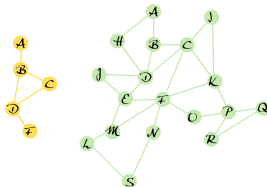
Recherche avec topologie dans un réseau PPI

- ▶ Le motif peut être un **chemin**, un **arbre**, un **graphe**
- ▶ Problèmes **NP-complets** donc une solution exacte entraîne une **complexité exponentielle**
- ▶ Mais tout n'est pas perdu !



Recherche avec topologie dans un réseau PPI

- ▶ Idée: exploiter le fait que les **motifs sont plus petits** ($\sim 5 - 15$) que le réseau (e.g. ~ 5.000 pour la levure)
- ▶ Restreindre la partie exponentielle à k (taille motif) au lieu de n (taille du réseau): **complexité paramétrée**



Algorithmes FPT

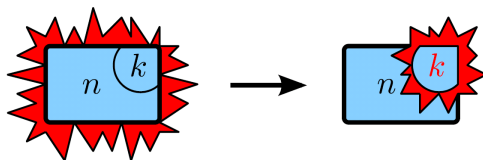
- ▶ Beaucoup de problèmes (*i.e.* problèmes paramétrés) sont de la forme :
 - ▶ Input : Un objet X , $|X| = n$, un entier k
 - ▶ Question : Est-ce que X a une propriété **dépendant de k** ?

Algorithmes FPT

- ▶ Beaucoup de problèmes (*i.e.* problèmes paramétrés) sont de la forme :
 - ▶ Input : Un objet X , $|X| = n$, un entier k
 - ▶ Question : Est-ce que X a une propriété **dépendant de k** ?
- ▶ Exemples :
 - ▶ (VERTEX COVER) Le graphe $G = (V, E)$ contient-il un **sous-ensemble** V' de sommets **de taille k** t.q. pour chaque arête (u, v) de G , soit u soit v est dans V' ?
 - ▶ (LONGEST COMMON SUBSEQUENCE) Existe t-il **une string de taille au moins k** qui soit une sous-séquence de n strings ?
 - ▶ (SET COVER) Etant donné un ensemble S de n ensembles, existe-il un **sous-ensemble $S' \subseteq S$ de k ensembles** t.q. chaque élément dans les ensembles de S est présent dans un ensemble de S' ?

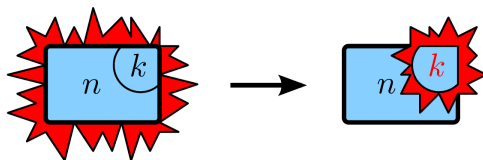
Algorithmes FPT

- ▶ Un algorithme FPT [DOWNEY & FELLOWS 1999]:
algorithme exact **exponentiel** seulement en son **paramètre k** (et pas en la taille de l'entrée n)
- ▶ $f(k) \cdot n^c$, avec c une constante, et f n'importe quelle fonction (e.g. $\mathcal{O}(2^k \cdot n^2)$)
- ▶ L'algorithme devient souvent “praticable”, car même si f est exponentiel, k est petit



Algorithmes FPT

- ▶ Un algorithme FPT [DOWNEY & FELLOWS 1999]:
algorithme exact **exponentiel** seulement en son **paramètre k** (et pas en la taille de l'entrée n)
- ▶ $f(k).n^c$, avec c une constante, et f n'importe quelle fonction (e.g. $\mathcal{O}(2^k.n^2)$)
- ▶ L'algorithme devient souvent “praticable”, car même si f est exponentiel, k est petit
- ▶ Attention, $2^{2^{2^{2^{2^k}}}}.n$ est FPT mais rédibitoire même pour $k = 1$



Algorithmes FPT

- ▶ Si un problème est $W[1]$ -difficile, il n'y a probablement pas d'algorithme FPT possible pour ce problème (plus ou moins l'analogue de la classe NP)
- ▶ Se prouve avec des réductions paramétrées (préservation de la taille de l'instance et du paramètre)

Color-coding [ALON ET AL. 1995]

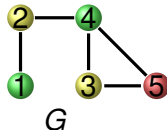
- ▶ Probleme : Trouver un chemin de taille k sans cycle dans un graphe avec n noeuds
- ▶ NP-Complet (Chemin hamiltonien où $k = n$)
- ▶ Algorithme naïf : n^k (tous les chemins de taille k)

Color-coding [ALON ET AL. 1995]

- ▶ (Randomized) Color-coding :
 - ▶ Choisir k couleurs différentes
 - ▶ Choisir aléatoirement une couleur pour chaque noeud de G



Couleurs introduites

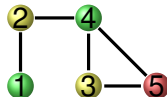


Color-coding [ALON ET AL. 1995]

- ▶ (Randomized) Color-coding :
 - ▶ Choisir k couleurs différentes
 - ▶ Choisir aléatoirement une couleur pour chaque noeud de G
 - ▶ Chercher un chemin “colorful” sur les k couleurs $\Rightarrow$ chemin sans cycle de taille k
 - ▶ S’effectue en seulement $\mathcal{O}(2^k)$ (1 bit pour chaque couleur)



Couleurs introduites



G

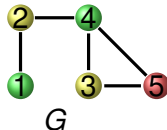
$$D(v, S) = \bigvee_u D(u, S \setminus \{col(u)\}) \text{ t.q. } u \text{ voisin de } v \text{ et } col(u) \in S$$

Color-coding [ALON ET AL. 1995]

- ▶ (Randomized) Color-coding :
 - ▶ Choisir k couleurs différentes
 - ▶ Choisir aléatoirement une couleur pour chaque noeud de G
 - ▶ Chercher un chemin “colorful” sur les k couleurs $\Rightarrow$ chemin sans cycle de taille k
 - ▶ S’effectue en seulement $\mathcal{O}(2^k)$ (1 bit pour chaque couleur)
 - ▶ Un “bon chemin” est colorful avec la probabilité $\frac{k!}{k^k} \geq e^{-k}$



Couleurs introduites

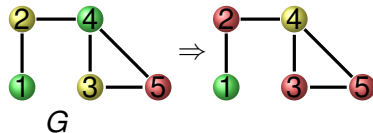


Color-coding [ALON ET AL. 1995]

- ▶ (Randomized) Color-coding :
 - ▶ Choisir k couleurs différentes
 - ▶ Choisir aléatoirement une couleur pour chaque noeud de G
 - ▶ Chercher un chemin “colorful” sur les k couleurs $\Rightarrow$ chemin sans cycle de taille k
 - ▶ S’effectue en seulement $\mathcal{O}(2^k)$ (1 bit pour chaque couleur)
 - ▶ Un “bon chemin” est colorful avec la probabilité $\frac{k!}{k^k} \geq e^{-k}$
 - ▶ Donc, relancer la procédure e^k fois pour avoir une probabilité d’erreur faible



Couleurs introduites



Color-coding [ALON ET AL. 1995]

- ▶ (Randomized) Color-coding :
 - ▶ Choisir k couleurs différentes
 - ▶ Choisir aléatoirement une couleur pour chaque noeud de G
 - ▶ Chercher un chemin “colorful” sur les k couleurs $\Rightarrow$ chemin sans cycle de taille k
 - ▶ S’effectue en seulement $\mathcal{O}(2^k)$ (1 bit pour chaque couleur)
 - ▶ Un “bon chemin” est colorful avec la probabilité $\frac{k!}{k^k} \geq e^{-k}$
 - ▶ Donc, relancer la procédure e^k fois pour avoir une probabilité d’erreur faible
 - ▶ Exponentiel en k seulement !

Recherche de motifs

- ▶ Motif est un chemin : QPath, $\mathcal{O}(2^k |E|)$ [SHLOMI ET AL. 2006]
 - ▶ Application directe du color-coding
- ▶ Motif est un arbre : QNet, $\mathcal{O}^*(2^{\mathcal{O}(k)})$ [DOST ET AL. 2007]
 - ▶ Extension du color-coding (programmation dynamique sur l'arbre)
- ▶ Motif est un graphe : W[1]-dur [DOWNEY & FELLOWS 1999] (pas d'algo FPT possible)
 - ▶ Mais FPT si le motif est un graphe à treewidth bornée, ou à Vertex Feedback Set borné [DOST ET AL. 2007, BLIN ET AL. 2009]

Plan

Introduction

Complexité paramétrée et motifs avec topologie

Motifs sans topologie

Le problème GRAPH MOTIF

Des logiciels pour GRAPH MOTIF

Conclusion

GRAPH MOTIF

- ▶ Constat : les données biologiques sont très bruitées
 - ▶ Manque des informations, faux négatifs. Estimé à 50% [GAVIN ET AL. 2002]
 - ▶ Informations erronées, faux positifs. Estimé à 65% [REGULY ET AL. 2006]
- ▶ La topologie du motif peut ne pas être connue *a priori*
- ▶ La topologie du motif peut ne pas être pertinente

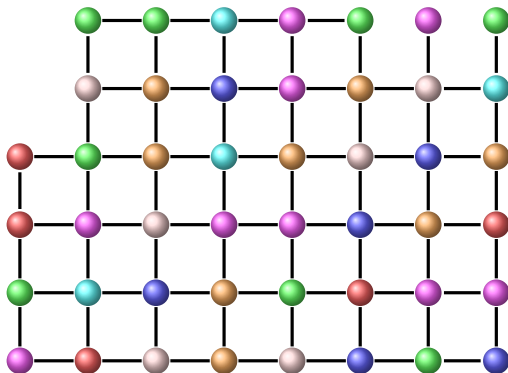
GRAPH MOTIF

- ▶ Chaque nœud du réseau est coloré par sa “fonction”
- ▶ Le motif sera juste un ensemble (ou multi ensemble) de couleurs à rechercher dans un réseau coloré [LACROIX ET AL. 2006]

GRAPH MOTIF

- ▶ S'applique à différent type de réseaux biologiques
 - ▶ Dans les réseaux PPI, chaque protéine du motif recoit une couleur
 - ▶ Le réseau est coloré selon les homologues avec les protéines du motif
- ▶ Selon [BETZLER ET AL. 2008], peut être utilisé pour réseaux sociaux ou autres réseaux complexes...

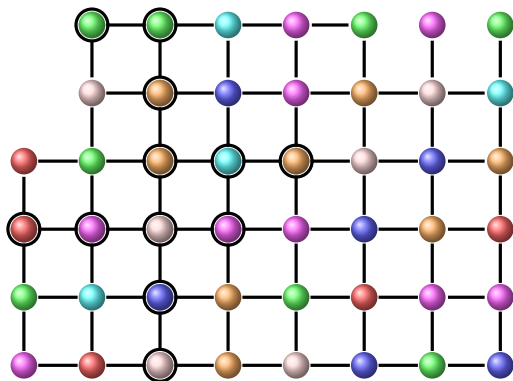
GRAPH MOTIF – Un exemple

 M  $G = (V, E)$

GRAPH MOTIF – Un exemple

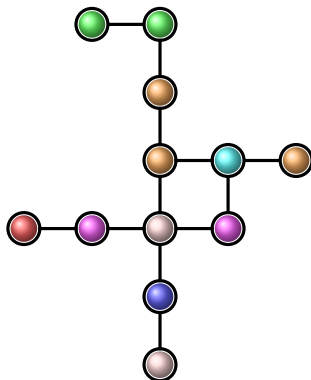


M



$G = (V, E)$
a possible $V' \subseteq V$

GRAPH MOTIF – Un exemple

 M  $G[V']$

GRAPH MOTIF – NP-C

- ▶ Le problème est NP-Complet, même si
 - ▶ Le réseau est un arbre [LACROIX ET AL. 2006]
 - ▶ Cet arbre est de degré max 3 et le motif est un ensemble simple [FELLOWS ET AL. 2008]
 - ▶ Le motif n'est constitué que de 2 couleurs et le réseau est un graphe biparti de degré max 4 [FELLOWS ET AL. 2008]

GRAPH MOTIF – Complexité paramétré

- ▶ Le problème est FPT par la taille du motif [LACROIX ET AL. 2006]

GRAPH MOTIF – Complexité paramétré

- ▶ Le problème est FPT par la taille du motif [LACROIX ET AL. 2006]
- ▶ Mais est $W[1]$ -difficile si paramétré par le nombre de couleurs [FELLOWS ET AL. 2008] (pas d'algo FPT possible)

GRAPH MOTIF – FPT – Motif colorful

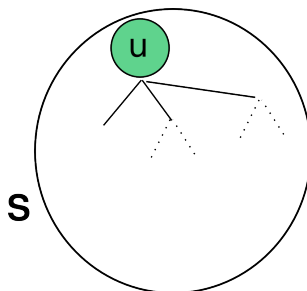
- Complexité en $\mathcal{O}^*(3^k)$ si le motif est un ensemble simple (colorful), avec k taille du motif [BETZLER ET AL. 2008]

GRAPH MOTIF – FPT – Motif colorful

- ▶ Complexité en $\mathcal{O}^*(3^k)$ si le motif est un ensemble simple (colorful), avec k taille du motif [BETZLER ET AL. 2008]
- ▶ Idée :
 1. Solution connecté : on cherche un arbre
 2. Motif colorful : on cherche donc un arbre colorful
 3. k couleurs différentes $\Rightarrow k$ nœuds différents

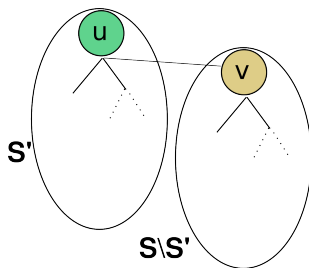
GRAPH MOTIF – Motif colorful – Programmation dynamique

- But : trouver un arbre, enraciné en u , colorful sur S .
 $D(u, S) = \text{True} ?$



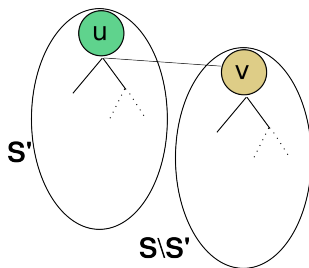
GRAPH MOTIF – Motif colorful – Programmation dynamique

- But : trouver un arbre, enraciné en u , colorful sur S .
 $D(u, S) = \text{True} ?$
- Oui si u à un voisin v t.q.
 1. u enracine un arbre, colorful sur S' couleurs
 2. v enracine un arbre, colorful sur $S \setminus S'$ couleurs



GRAPH MOTIF – Motif colorful – Programmation dynamique

- ▶ But : trouver un arbre, enraciné en u , colorful sur S .
 $D(u, S) = \text{True} ?$
- ▶ Oui si u à un voisin v t.q.
 1. u enracine un arbre, colorful sur S' couleurs
 2. v enracine un arbre, colorful sur $S \setminus S'$ couleurs
- ▶ Regarder pour chaque voisin v et chaque $S' \subset S$
- ▶ $D(u, S) = D(u, S') \wedge D(v, S \setminus S')$



GRAPH MOTIF – Motif colorful – Programmation dynamique

- ▶ But : trouver un arbre, enraciné en u , colorful sur S .
 $D(u, S) = \text{True} ?$
- ▶ Oui si u à un voisin v t.q.
 1. u enracine un arbre, colorful sur S' couleurs
 2. v enracine un arbre, colorful sur $S \setminus S'$ couleurs
- ▶ Regarder pour chaque voisin v et chaque $S' \subset S$
- ▶ $D(u, S) = D(u, S') \wedge D(v, S \setminus S')$
- ▶ Récursivement...
- ▶ ...jusqu'à $S = \text{col}(u)$

GRAPH MOTIF – FPT – Multiset

- ▶ Programmation dynamique valable que si le motif est un simple ensemble, car S et S' doivent être distincts (sinon plus assuré d'avoir k nœuds différents)
- ▶ On peut ramener le cas multi-ensemble au cas colorful avec le color-coding, en $\mathcal{O}^*(4.32^k)$ [BETZLER ET AL. 2008]

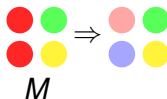
GRAPH MOTIF – FPT – Multiset

- Pour chaque couleur c du motif dont $occ_M(c) \geq 2$
- Créer $occ_M(c)$ nouvelles couleurs et remplacer c dans le motif par ces nouvelles couleurs



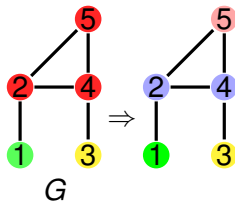
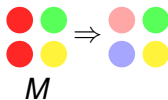
GRAPH MOTIF – FPT – Multiset

- Pour chaque couleur c du motif dont $occ_M(c) \geq 2$
- Créer $occ_M(c)$ nouvelles couleurs et remplacer c dans le motif par ces nouvelles couleurs



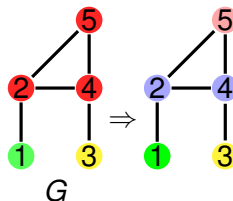
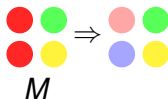
GRAPH MOTIF – FPT – Multiset

- ▶ Pour chaque couleur c du motif dont $occ_M(c) \geq 2$
- ▶ Créer $occ_M(c)$ nouvelles couleurs et remplacer c dans le motif par ces nouvelles couleurs
- ▶ Recolorer aléatoirement les nœuds du réseau portant la couleur c par une des nouvelles couleurs



GRAPH MOTIF – FPT – Multiset

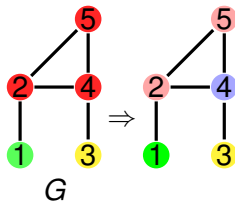
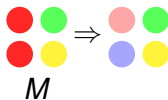
- Pour chaque couleur c du motif dont $occ_M(c) \geq 2$
- Créer $occ_M(c)$ nouvelles couleurs et remplacer c dans le motif par ces nouvelles couleurs
- Recolorer aléatoirement les nœuds du réseau portant la couleur c par une des nouvelles couleurs



- Finalement, le motif est colorful : en chercher une occurrence dans le réseau modifié avec l'algo précédent

GRAPH MOTIF – FPT – Multiset

- ▶ Pour chaque couleur c du motif dont $occ_M(c) \geq 2$
- ▶ Créer $occ_M(c)$ nouvelles couleurs et remplacer c dans le motif par ces nouvelles couleurs
- ▶ Recolorer aléatoirement les nœuds du réseau portant la couleur c par une des nouvelles couleurs



- ▶ Finalement, le motif est colorful : en chercher une occurrence dans le réseau modifié avec l'algo précédent
- ▶ Il faut répéter $|\ln(\epsilon)|e^k$ fois cette coloration aléatoire pour avoir une probabilité $1 - \epsilon$ de succès

GRAPH MOTIF et polynômes

- ▶ On peut diminuer la complexité de GRAPH MOTIF en utilisant un framework dû à Koutis et Williams [ICALP 08,09]

GRAPH MOTIF et polynômes

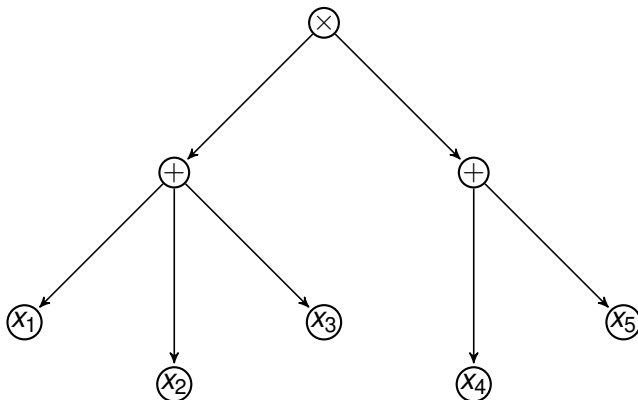
- ▶ On peut diminuer la complexité de GRAPH MOTIF en utilisant un framework dû à Koutis et Williams [ICALP 08,09]
- ▶ Résultat clef : On peut déterminer par un algorithme randomisé, en **temps $\mathcal{O}^*(2^k)$ et en espace polynomial**, si un polynôme représenté par un circuit arithmétique contient un **monome multilinéaire de degré k**

GRAPH MOTIF et polynômes

- ▶ Un monôme est **multilinéaire** si chaque variable du monôme n'apparaît qu'une seule fois (pas de carré)
- ▶ Par définition, le degré d'un monôme multilinéaire est son nombre de variables
- ▶ Exemple : $P(X) = (x_1^2 x_3 x_5 + x_1 x_2 x_4 x_6)$
 - ▶ $x_1 x_2 x_4 x_6$ est un monôme multilinéaire de degré 4
 - ▶ $x_1^2 x_3 x_5$ n'est pas un monôme multilinéaire

GRAPH MOTIF et polynômes

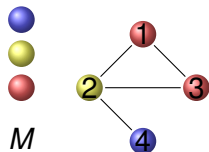
- ▶ Un **circuit arithmétique** sur un ensemble de variables X est un DAG t.q. :
 - ▶ les noeuds internes sont les opérations $\times$ ou $+$
 - ▶ les feuilles sont des éléments de X
- ▶ Exemple pour $P(X) = (x_1 + x_2 + x_3)(x_4 + x_5)$



GRAPH MOTIF et polynômes

- Pour résoudre GRAPH MOTIF avec motif colorful :
 - On introduit des variables correspondant aux couleurs
 - On construit un circuit correspondant aux couleurs de tous les sous arbres de taille k dans G
 - Un monôme multilinéaire de degré k correspond alors à un sous arbre avec k couleurs différentes (motif colorful)

Exemple

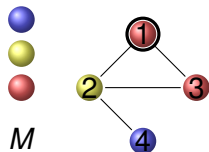


Construction récursive.

Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :

1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.

Exemple



Construction récursive.

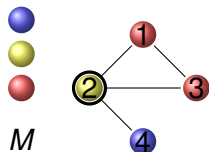
Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :

1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.

Pour $k' = 1$:

x_R

Exemple



Construction récursive.

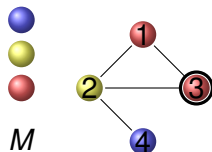
Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :

1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.

Pour $k' = 1$:

$$x_R \cdot (P_{2,2})$$

Exemple



Construction récursive.

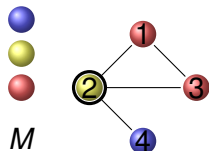
Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :

1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.

Pour $k' = 1$:

$$x_R \cdot (P_{2,2} + P_{2,3})$$

Exemple



Construction récursive.

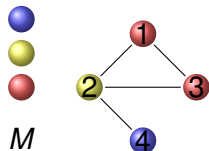
Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :

1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.

Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,2} + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (P_{1,1} + P_{1,3} + P_{1,4}) + P_{2,3})
 \end{aligned}$$

Exemple



Construction récursive.

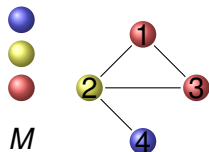
Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :

1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.

Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,2} + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (P_{1,1} + P_{1,3} + P_{1,4}) + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (x_R + x_R + x_B) + P_{2,3})
 \end{aligned}$$

Exemple



Construction récursive.

Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :

1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.

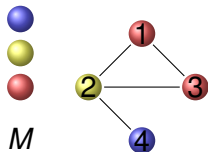
Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,2} + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (P_{1,1} + P_{1,3} + P_{1,4}) + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (x_R + x_R + x_B) + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot x_R + x_Y \cdot x_R + x_Y \cdot x_B + P_{2,3})
 \end{aligned}$$

Exemple

Construction récursive.

Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :



1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.

Pour $k' = 1$:

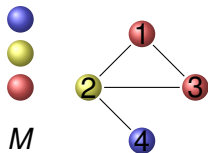
$$\begin{aligned}
 & x_R \cdot (P_{2,2} + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (P_{1,1} + P_{1,3} + P_{1,4}) + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (x_R + x_R + x_B) + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot x_R + x_Y \cdot x_R + x_Y \cdot x_B + P_{2,3}) \\
 &= x_R x_Y x_R + x_R x_Y x_R + x_R x_Y x_B + \dots
 \end{aligned}$$

Exemple

Construction récursive.

Les monômes de degré $k = 3$ contenant le noeud 1 sont la somme des produits des monômes :

1. de degré $k' < k$ contenant le noeud 1,
2. de degré $k - k'$ contenant un voisin de 1.



Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,2} + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (P_{1,1} + P_{1,3} + P_{1,4}) + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot (x_R + x_R + x_B) + P_{2,3}) \\
 &= x_R \cdot (x_Y \cdot x_R + x_Y \cdot x_R + x_Y \cdot x_B + P_{2,3}) \\
 &= x_R x_Y x_R + x_R x_Y x_R + x_R x_Y x_B + \dots
 \end{aligned}$$

Il y a un monôme multilinéaire (donc une solution)

GRAPH MOTIF et polynômes

- ▶ Si le **motif est un multiensemble**, un monôme non multilinéaire sur les couleurs est une solution
- ▶ La construction précédente n'est plus suffisante
- ▶ On introduit des variables pour les noeuds du graphe

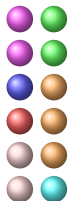
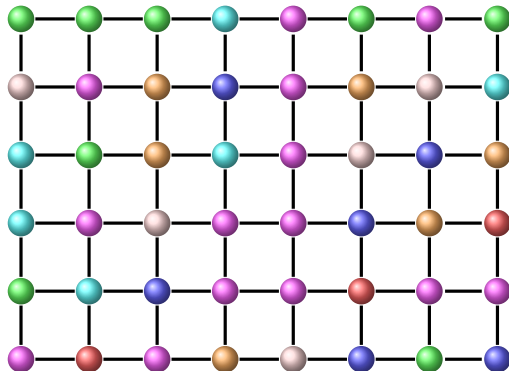
Une variante de GRAPH MOTIF : MAX MOTIF

- ▶ Comme données bruitées, chercher une occurrence exacte peut-être impossible
- ▶ Autoriser insertions et deletions

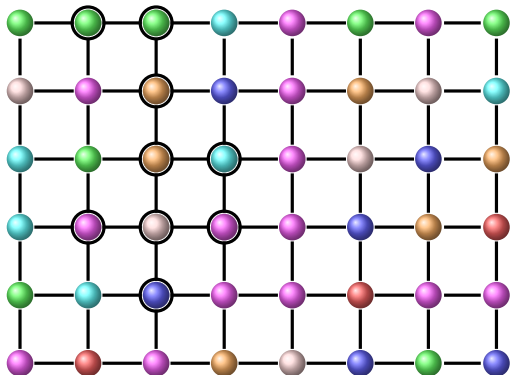
Une variante de GRAPH MOTIF : MAX MOTIF

- ▶ Comme données bruitées, chercher une occurrence exacte peut-être impossible
- ▶ Autoriser insertions et deletions
- ▶ Un exemple : MAX MOTIF
- ▶ Trouver une occurrence qui matche "le plus possible" de couleurs du motif

Exemple MAX MOTIF

 M  G

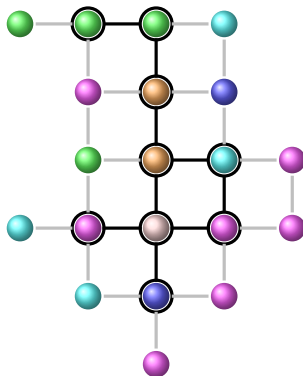
Exemple MAX MOTIF

 M  G

Exemple MAX MOTIF

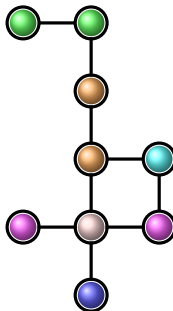


M



G

Exemple MAX MOTIF

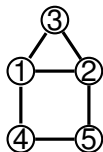
 M  G

MAX MOTIF – Innapproximabilité

- ▶ (Malheureusement,)MAX MOTIF est aussi dur à approximer que INDEPENDANT SET , même si
 - ▶ Le réseau est un arbre
 - ▶ Le motif est colorful
 - ▶ Chaque couleur apparait au plus 2 fois dans le réseau
- ▶ INDEPENDANT SET : Dans un graphe $G = (V, E)$, trouver le plus grand $V' \subseteq V$ t.q. aucun nœud dans V' est relié à un autre nœud de V'

MAX MOTIF – Innapproximabilité

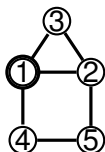
- A partir d'un graphe $G = (V_G, E_G)$, on construit un arbre



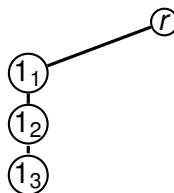
$$G = (V_G, E_G)$$

MAX MOTIF – Innapproximabilité

- Chemins de taille $d(u)$, $\forall u \in V_G$



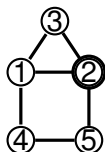
$$G = (V_G, E_G)$$



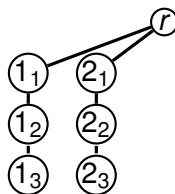
T

MAX MOTIF – Innapproximabilité

- Chemins de taille $d(u)$, $\forall u \in V_G$



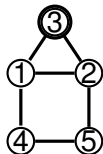
$$G = (V_G, E_G)$$



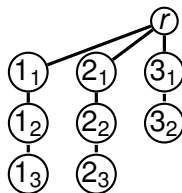
T

MAX MOTIF – Innapproximabilité

- Chemins de taille $d(u)$, $\forall u \in V_G$



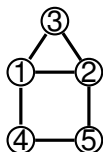
$$G = (V_G, E_G)$$



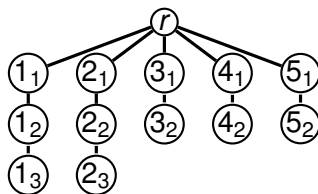
T

MAX MOTIF – Innapproximabilité

- Chemins de taille $d(u)$, $\forall u \in V_G$



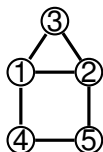
$$G = (V_G, E_G)$$



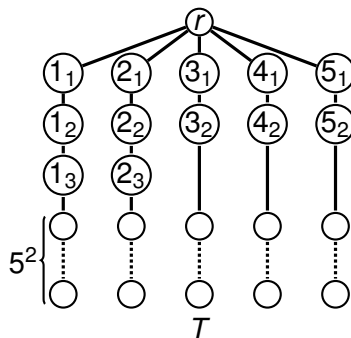
T

MAX MOTIF – Innapproximabilité

- Ajout de $|V_G|^2$ nœuds à chaque chemin (“boîtes noires”)

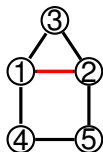


$$G = (V_G, E_G)$$



MAX MOTIF – Innapproximabilité

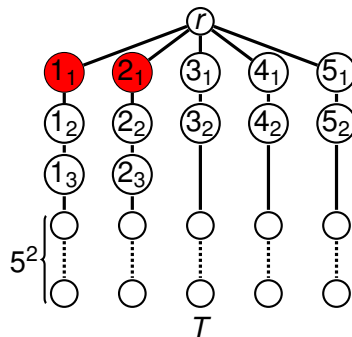
- Coloration (2 occurrences...) + motif (colorful)



$$G = (V_G, E_G)$$

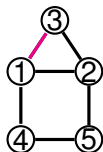


M



MAX MOTIF – Innapproximabilité

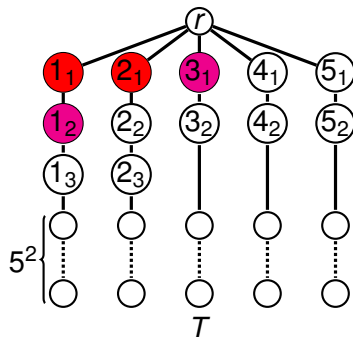
- Coloration (2 occurrences...) + motif (colorful)



$$G = (V_G, E_G)$$

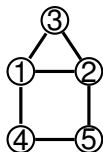


M



MAX MOTIF – Innapproximabilité

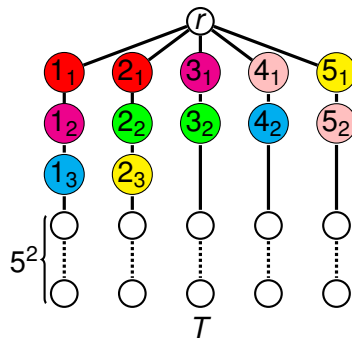
- Coloration (2 occurrences...) + motif (colorful)



$$G = (V_G, E_G)$$

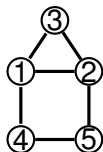


M

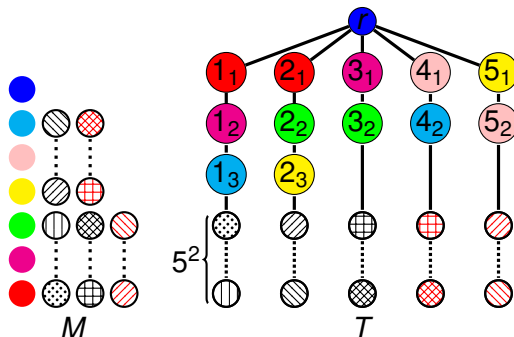


MAX MOTIF – Innapproximabilité

- Des couleurs différentes pour la racine et les boîtes noires

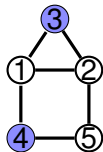


$$G = (V_G, E_G)$$

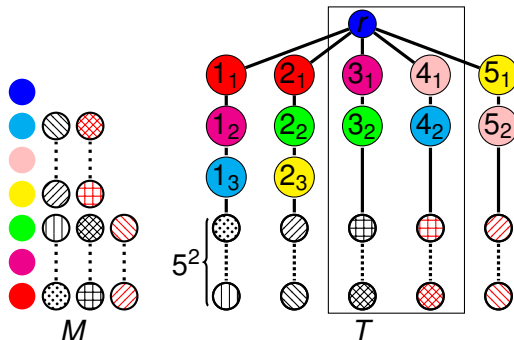


MAX MOTIF – Innapproximabilité

- INDEPENDANT SET $\Rightarrow$ MAX MOTIF colorful

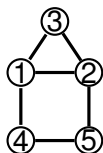


$$G = (V_G, E_G)$$

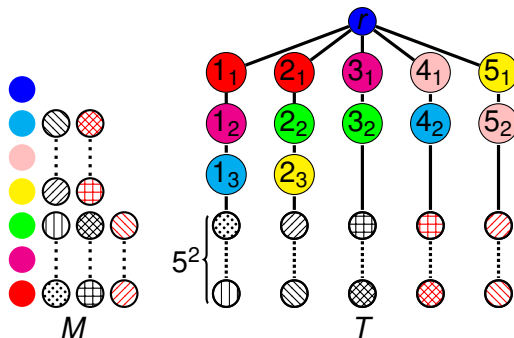


MAX MOTIF – Innapproximabilité

- On prouve que s'il existait un algo d'approx. pour MAX MOTIF, il y aurait un algo d'approx. pour INDEPENDANT SET



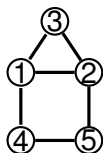
$$G = (V_G, E_G)$$



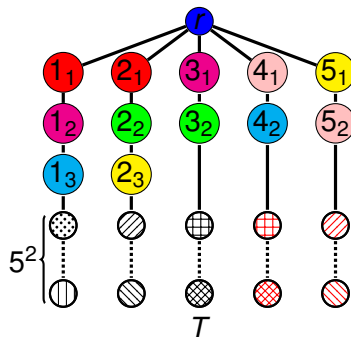
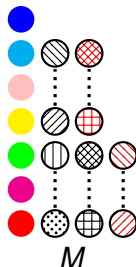
MAX MOTIF – Innapproximabilité

► Idées :

- La racine doit être prise pour rendre la solution connexe
- Les boîtes noires sont données “gratuitement” à la solution si le chemin depuis la racine est dans la solution.
- Une boîte noire est de plus grande taille que l'ensemble des autres noeuds



$$G = (V_G, E_G)$$



Logiciels pour GRAPH MOTIF

- ▶ Torque [BRUCKNER ET AL. 2009] : un web service, qui ne gère que les motifs colorfals
- ▶ GraMoFoNe [BLIN ET AL. 2010] : un plugin cytoscape

GraMoFoNe – Cytoscape (2002)

- ▶ Plateforme java open-source, gratuite, permettant
 - ▶ l'importation / l'exportation nombreux formats, BDD,...
 - ▶ la visualisation et l'analyse de réseaux d'interactions
 - ▶ l'integration d'annotations à ces réseaux
- ▶ Largement utilisé ("des centaines" d'articles l'utilisent pour de l'analyse)
- ▶ Maintenu...



GraMoFoNe – Cytoscape

- ▶ Principal avantage : plugins
- ▶ Ajout de fonctions :
 - ▶ Analyses
 - ▶ Nouveaux layouts
 - ▶ Support de fichiers
 - ▶ Connection avec des bases de données
 - ▶ ...

GraMoFoNe – PB

- ▶ On exprime GRAPH MOTIF avec de la programmation linéaire Pseudo-Booléenne
- ▶ *i.e.* de la programmation linéaire avec des variables booléennes

GraMoFoNe – PB

- ▶ Trouver un assignement de variables satisfaisant les contraintes et maximisant l'objectif
- ▶ Un exemple simple :
 - ▶ **Variables** : $x_i \in \{0, 1\}, \forall i = 1, 2, 3$
 - ▶ **Objectif** : $\max x_1 + 2x_2 - x_3$
 - ▶ **Contraintes** :
 1. $x_1 - 2x_2 + 3x_3 \geq 1$
 2. $x_1 + x_2 + x_3 = 1$
 3. $2x_1 + x_2 + x_3 < 3$
- ▶ Solution : $x_1 = 1, x_2 = 0, x_3 = 0$

GraMoFoNe – PB

- ▶ Donne une solution exacte
- ▶ Beaucoup de solvers efficaces existent
- ▶ Dont beaucoup de gratuits ($\neq$ CPLEX)

GraMoFoNe – PB

- ▶ Donne une solution exacte
- ▶ Beaucoup de solvers efficaces existent
- ▶ Dont beaucoup de gratuits ($\neq$ CPLEX)
- ▶ On utilise SAT4JPseudo [LE BERRE, PARRAIN 2007]
 - ▶ Gratuit
 - ▶ Maintenu
 - ▶ Bonnes performances lors des compétitions
 - ▶ En java

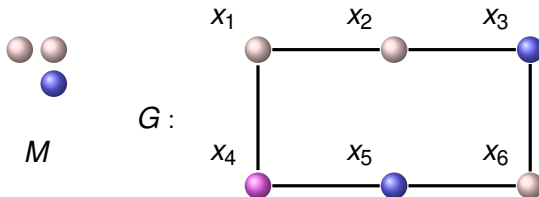


GraMoFoNe – PB

- ▶ On utilise 23 contraintes et 9 domaines de variables
- ▶ On cherche une occurrence d'un motif M dans un graphe G
- ▶ A respecter :
 1. La taille de la solution
 2. La coloration de la solution par rapport au motif
 3. La connexité de la solution (partie difficile)

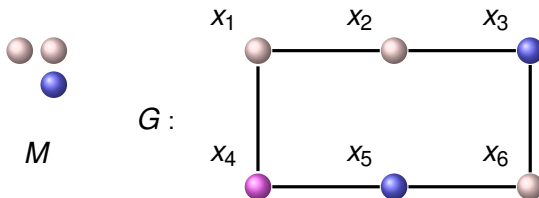
GraMoFoNe – Variables

- Une variable x pour chaque nœud



GraMoFoNe – Variables

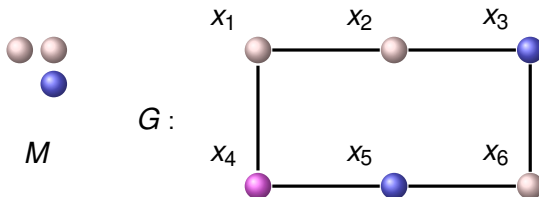
- Une variable x pour chaque nœud



- Contrainte "taille solution" : $\sum_{v \in V} x_v = |M|$:
 - $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$

GraMoFoNe – Variables

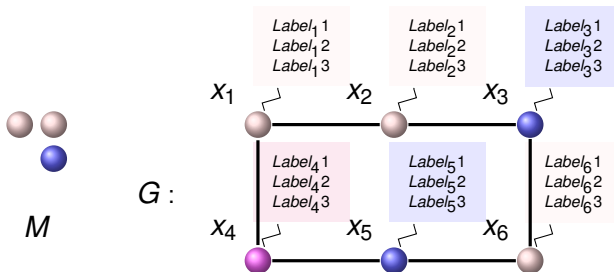
- Une variable x pour chaque nœud



- Contrainte "taille solution" : $\sum_{v \in V} x_v = |M|$:
 - $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$
- Contraintes "coloration" $\sum_{\substack{v \in V \\ c \in col(v)}} x_v = occ_M(c)$:
 - $x_1 + x_2 + x_6 = 2$ (gris)
 - $x_3 + x_5 = 1$ (bleu)
 - $x_4 = 0$ (rose)

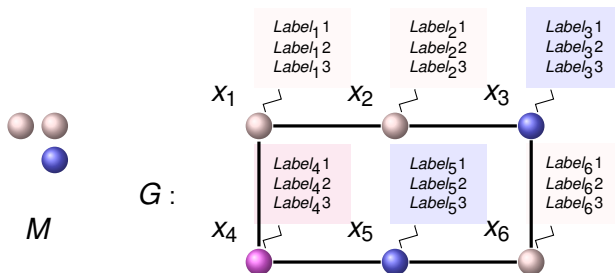
GraMoFoNe – Variables

- Connexité : $|M|$ variables $Label_v$ pour chaque nœud v



GraMoFoNe – Variables

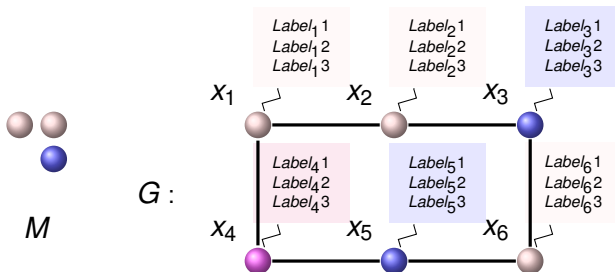
- Connexité : $|M|$ variables $Label_v$ pour chaque nœud v



- $|V|$ contraintes "un seul label par nœud dans la solution" :
 - Pour chaque v , $x_v \Rightarrow (\sum_{i=1}^{|M|} Label_v i = 1)$

GraMoFoNe – Variables

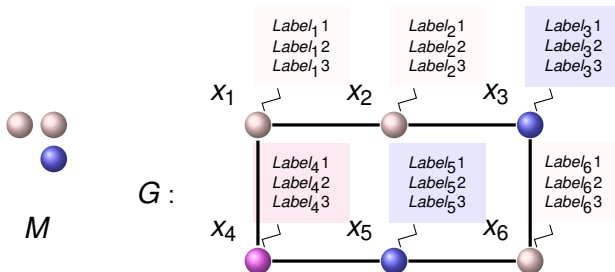
- Connexité : $|M|$ variables $Label_v$ pour chaque nœud v



- $|V|$ contraintes "un seul label par nœud dans la solution" :
 - Pour chaque v , $x_v \Rightarrow (\sum_{i=1}^{|M|} Label_{v,i} = 1)$
- $|M|$ contraintes "un seul nœud par label donné"
 - Pour un label i donné, $\sum_{v \in V} Label_{v,i} = 1$

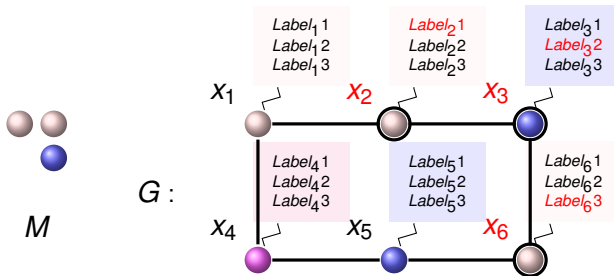
GraMoFoNe – Variables

- ▶ Connexité : $|M|$ variables $Label_v$ pour chaque nœud v

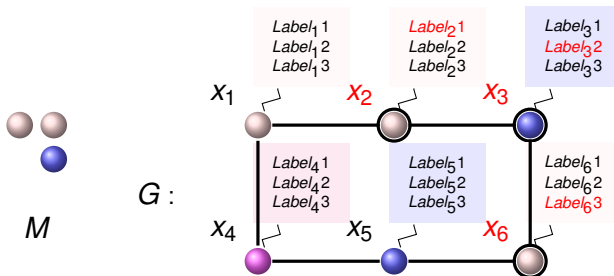


- ▶ $|V|$ contraintes "un seul label par nœud dans la solution" :
 - ▶ Pour chaque v , $x_v \Rightarrow (\sum_{i=1}^{|M|} Label_v i = 1)$
- ▶ $|M|$ contraintes "un seul nœud par label donné"
 - ▶ Pour un label i donné, $\sum_{v \in V} Label_v i = 1$
- ▶ $|V| \cdot |M|$ contraintes "un nœud avec un label a un voisin avec un label supérieur" (sauf le dernier)
 - ▶ $Label_v i \Rightarrow (\sum_{u \in N(v)} \sum_{j > i} Label_u j \geq 1)$

GraMoFoNe – Une solution

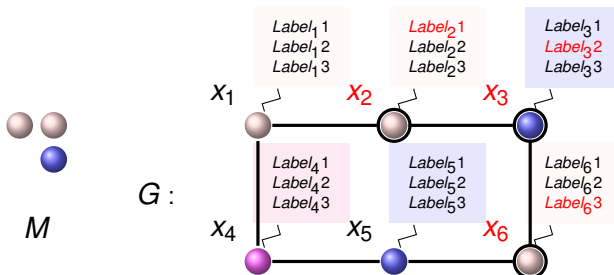


GraMoFoNe – Une solution



- ▶ “taille” : $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$
- ▶ “coloration”:
 - ▶ $x_1 + x_2 + x_6 = 2$ (gris)
 - ▶ $x_3 + x_5 = 1$ (bleu)
 - ▶ $x_4 = 0$ (rose)

GraMoFoNe – Une solution



- ▶ “un label par nœud” : $\forall v, x_v \Rightarrow (\sum_{i=1}^{|M|} Label_v i = 1)$
- ▶ “un nœud par label” : $\sum_{v \in V} Label_v i = 1$
- ▶ “voisin avec label supérieur” :
 $Label_v i \Rightarrow (\sum_{u \in N(v)} \sum_{j > i} Label_u j \geq 1)$

GraMoFoNe – PB

- ▶ Gère GRAPH MOTIF "classique" (colorful et multi-ensemble)
- ▶ PB permet d'avoir toutes les solutions possibles...

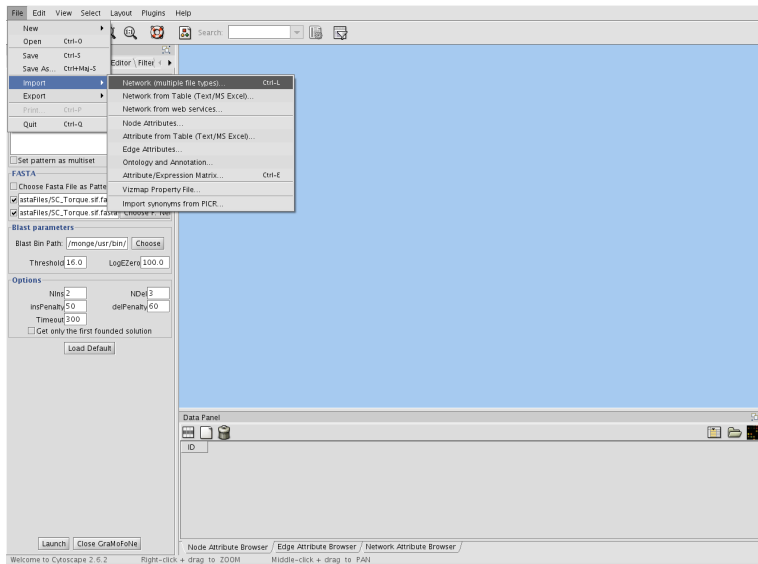
GraMoFoNe – PB

- ▶ Gère GRAPH MOTIF "classique" (colorful et multi-ensemble)
- ▶ PB permet d'avoir toutes les solutions possibles...
- ▶ Avec variables et contraintes supplémentaires, on peut gérer
 - ▶ Les insertions
 - ▶ Les deletions
 - ▶ Une liste de couleurs associée à chaque nœud du graphe

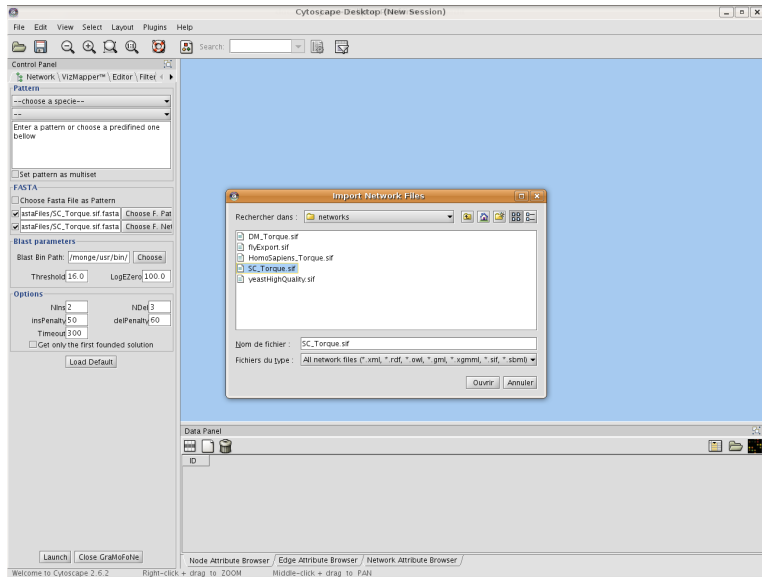
GraMoFoNe – PB

- ▶ Gère GRAPH MOTIF "classique" (colorful et multi-ensemble)
- ▶ PB permet d'avoir toutes les solutions possibles...
- ▶ Avec variables et contraintes supplémentaires, on peut gérer
 - ▶ Les insertions
 - ▶ Les deletions
 - ▶ Une liste de couleurs associée à chaque nœud du graphe
- ▶ Devient complexe (compensations "une deletion compense une insertion", garder la bijection entre motif et graphe si plusieurs couleurs dans le graphe,...)

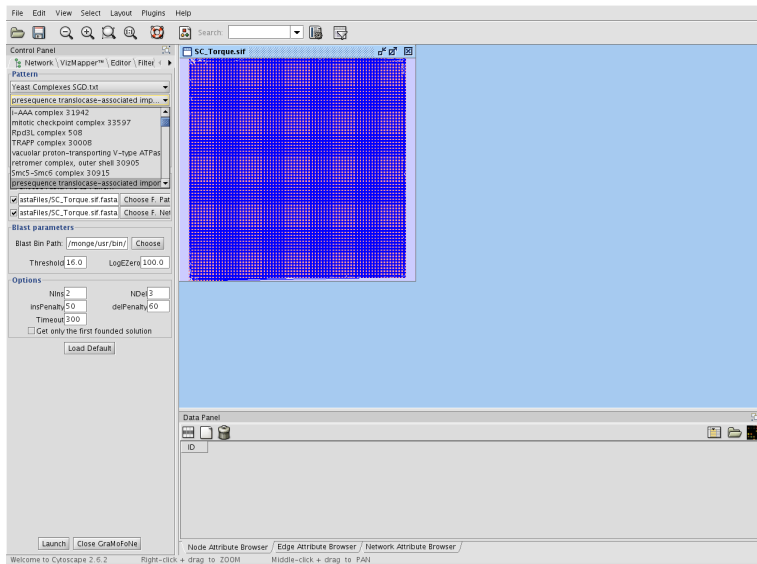
GraMoFoNe – IG



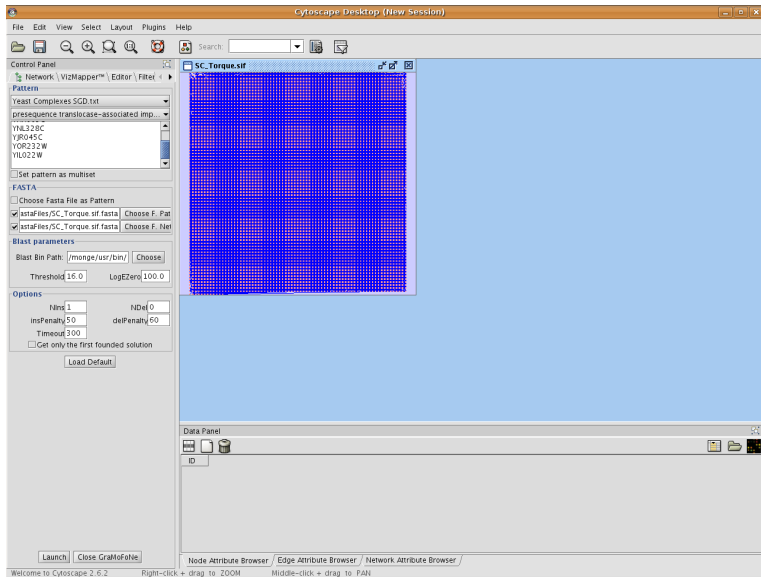
GraMoFoNe – IG



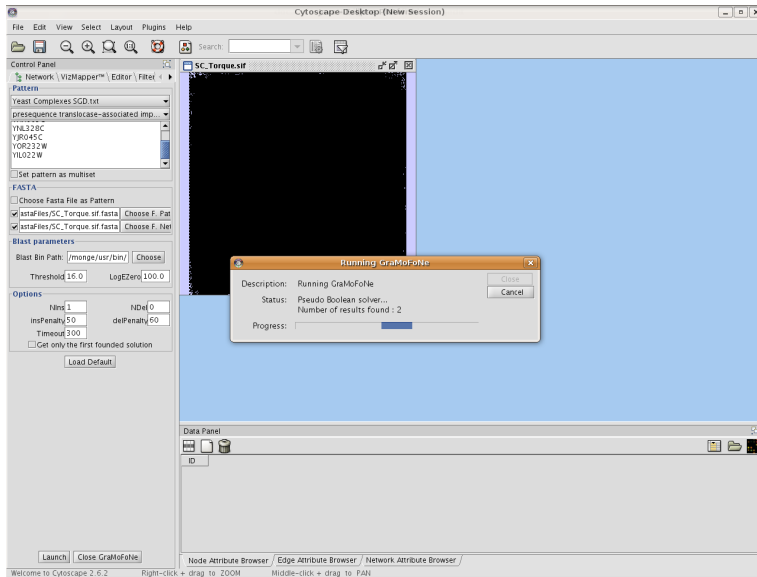
GraMoFoNe – IG



GraMoFoNe – IG



GraMoFoNe – IG



GraMoFoNe – IG

The screenshot shows the Cytoscape Desktop (New Session) interface. The main window displays a network graph titled "SC_Torque.sif". The interface is divided into several panels:

- Control Panel:** Contains settings for the network, pattern, and blast parameters.
 - Pattern:** A list of patterns including "Yeast Complexes SGD.txt", "presequence translocase-associated imp...", "YNL328C", "YJR045C", "YOR232W", and "YIL022W".
 - FASTA:** Options to set pattern as multiset, choose FASTA file as pattern, and select files for FASTA and pattern.
 - Blast parameters:** Blast Bin Path, Threshold (16.0), and LogEZero (100.0).
 - Options:** Nins (1), NDel (0), insPenalty (50), delPenalty (60), Timeout (300), and a checkbox for "Get only the first founded solution".
- Results Panel:** Displays the results of the GraMoFoNe search. It shows a table with columns "Result" and "Details".

Result	Details
[Red dots]	Score = 15.0 Rank = 1 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Red dots]	Score = 14.0 Rank = 2 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Red dots]	Score = 13.0 Rank = 3 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Red dots]	Score = 13.0 Rank = 4 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Red dots]	Score = 13.0 Rank = 5 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
- Data Panel:** Shows the network data, including a table with columns "ID" and "Name".

At the bottom of the interface, there are buttons for "Launch", "Close GraMoFoNe", and "Discard Result". The status bar at the very bottom indicates "Welcome to Cytoscape 2.6.2" and provides instructions for zooming and panning.

GraMoFoNe – IG

Cytoscape Desktop (New Session)

File Edit View Select Layout Plugins Help

Search: []

Control Panel

Network \ VizMapper™ \ Editor \ Filter

Pattern

Yeast Complexes SGD.txt
 presequence translocase-associated imp...
 YNL328C
 YJR045C
 YOR232W
 YIL022W

☐ Set pattern as multiset

FASTA

☐ Choose Fasta File as Pattern

☒ IstaFiles/SC_Torque.sif fasta Choose F. Pat
☒ IstaFiles/SC_Torque.sif fasta Choose F. Net

Blast parameters

Blast Bin Path: [/monge/usr/bin/] Choose
 Threshold 16.0 LogEZero 100.0

Options

Nins 1 NDel 0
 insPenalty 50 delPenalty 60
 Timeout 300
☐ Get only the first founded solution

Load Default

Results Panel

GraMoFoNe results

Result	Details
[Pattern]	Score = 15.0 Rank = 1 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Pattern]	Score = 14.0 Rank = 2 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Pattern]	Score = 13.0 Rank = 3 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Pattern]	Score = 13.0 Rank = 4 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Pattern]	Score = 13.0 Rank = 5 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)

YJL104W matched with : YJL104W
 YLR008C matched with : YLR008C
 YNR017W Inserted as a not colored node
 YKR065C matched with : YKR065C
 YNL328C matched with : YNL328C
 YJR045C matched with : YJR045C
 YOR232W matched with : YOR232W
 YIL022W matched with : YIL022W

Discard Result

Data Panel

ID

YJR045C
 YJL104W
 YOR232W
 YNR017W
 YLR008C
 YKR065C
 YIL022W
 YNL328C

Launch Close GraMoFoNe

Node Attribute Browser / Edge Attribute Browser / Network Attribute Browser

Welcome to Cytoscape 2.6.2 Right-click + drag to ZOOM Middle-click + drag to PAN

GraMoFoNe – IG

Cytoscape Desktop (New Session)

File Edit View Select Layout Plugins Help

Search:

Control Panel

Network \VizMapper™ \ Editor \ Filter

Pattern

Yeast Complexes SGD.txt

presequence translocase-associated imp...

YNL328C

YJR045C

YOR232W

YIL022W

☐ Set pattern as multiset

FASTA

☐ Choose Fasta File as Pattern

☒ istaFiles/SC_Torque.sif fasta Choose F. Pat

☒ istaFiles/SC_Torque.sif fasta Choose F. Net

Blast parameters

Blast Bin Path: /monge/usr/bin/ Choose

Threshold 16.0 LogEZero 100.0

Options

Nins 1 NDel 0

insPenalty 50 delPenalty 60

Timeout 300

☐ Get only the first founded solution

Load Default

Launch Close GraMoFoNe

SC_Torque.sif

Results Panel

GraMoFoNe results

Result	Details
	Score = 15.0 Rank = 1 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
	Score = 14.0 Rank = 2 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
	Score = 13.0 Rank = 3 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
	Score = 13.0 Rank = 4 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
	Score = 13.0 Rank = 5 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)

YJL104W matched with : YIL104W
YLR008C matched with : YLR008C
YNR017W inserted as a not colored node
YKR065C matched with : YKR065C
YNL328C matched with : YNL328C
YJR045C matched with : YJR045C
YOR232W matched with : YOR232W
YIL022W matched with : YIL022W

Discard Result

Data Panel

ID

YJR045C

YJL104W

YOR232W

YNR017W

YLR008C

YKR065C

YIL022W

YNL328C

Node Attribute Browser / Edge Attribute Browser / Network Attribute Browser

Welcome to Cytoscape 2.6.2 Right-click + drag to ZOOM Middle-click + drag to PAN

GraMoFoNe – IG

The screenshot displays the GraMoFoNe software interface, which is used for pattern matching in colored graphs. The interface is divided into several panels:

- Control Panel:** Contains settings for the pattern (e.g., "Yeast Complexes SGD.txt"), FASTA file selection, and blast parameters (Threshold: 16.0, LogEzern: 100.0).
- Network Visualization:** A central window showing a network graph with nodes and edges. The nodes are colored (yellow, blue, red) and labeled with IDs like YJL104W, YLR008C, etc.
- Results Panel:** Displays the search results for the pattern "SC_Torque.sif". It shows a table of results with columns for "Result" and "Details". The table lists matches for various nodes and their associated scores and ranks.
- Data Panel:** A table at the bottom showing the IDs of the nodes found in the network.

The Results Panel shows the following matches:

Result	Details
YJL104W	Score = 15.0 Rank = 1 Nb Nodes = 0 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
YLR008C	Score = 13.0 Rank = 3 Nb Nodes = 0 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
YNR017W	Score = 13.0 Rank = 4 Nb Nodes = 0 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
YJL104W	Score = 13.0 Rank = 5 Nb Nodes = 0 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)

The Data Panel shows the following IDs:

ID
YJR045C
YJL104W
YOR232W
YNR017W
YLR008C
YKR065C
YIL022W
YNL328C

The interface also includes a "Launch" button and a "Close GraMoFoNe" button at the bottom left.

GraMoFoNe – IG

Cytoscape Desktop (New Session)

File Edit View Select Layout Plugins Help

Search: []

Control Panel

Network \ VizMapper™ \ Editor \ Filter

Pattern

Yeast Complexes SGD.txt
 presequence translocase-associated imp...
 YNL328C
 YJR045C
 YOR232W
 YIL022W

☐ Set pattern as multiset

FASTA

☐ Choose Fasta File as Pattern

☒ istaFiles/SC_Torque.sif fasta Choose F. Pat
☒ istaFiles/SC_Torque.sif fasta Choose F. Net

Blast parameters

Blast Bin Path: [/monge/usr/bin/] Choose
 Threshold 16.0 LogEZero 100.0

Options

Nins 1 NDel 0
 insPenalty 50 delPenalty 60
 Timeout 300
☐ Get only the first founded solution

Load Default

Results Panel

GraMoFoNe results

Result	Details
[Red dots]	Score = 15.0 Rank = 1 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Red dots]	Score = 14.0 Rank = 2 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Red dots]	Score = 13.0 Rank = 3 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Red dots]	Score = 13.0 Rank = 4 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
[Red dots]	Score = 13.0 Rank = 5 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)

YIL104W matched with: YIL104W
 YLR008C matched with: YLR008C
 YNR017W inserted as a not colored node
 YKR065C matched with: YKR065C
 YNL328C matched with: YNL328C
 YJR045C matched with: YJR045C
 YOR232W matched with: YOR232W
 YIL022W matched with: YIL022W

Discard Result

Data Panel

ID

Node Attribute Browser / Edge Attribute Browser / Network Attribute Browser

Welcome to Cytoscape 2.6.2 Right-click + drag to ZOOM Middle-click + drag to PAN

Launch Close GraMoFoNe

GraMoFoNe – “batch mode”

- ▶ Utilisé pour tests grande échelle
- ▶ Recherche de dizaines de complexes protéiques dans des réseaux d'autres espèces

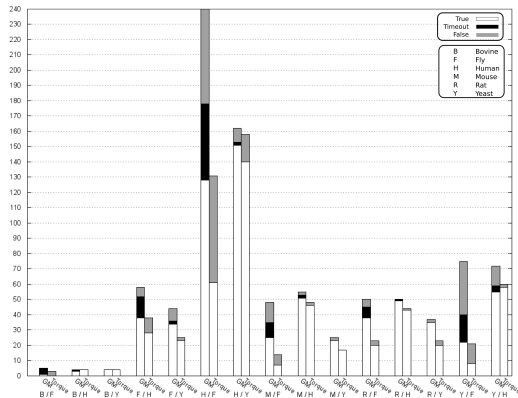
GraMoFoNe – “batch mode”

- ▶ Utilisé pour tests grande échelle
- ▶ Recherche de dizaines de complexes protéiques dans des réseaux d'autres espèces
- ▶ Données :
 - ▶ Motifs de 6 espèces (levure, drosophile, homo sapiens, souris, boeuf, rat)
 - ▶ Réseaux de 3 espèces (levure, drosophile, homo sapiens)
 - ▶ De 2 à 4 indels autorisés selon la taille du motif

GraMoFoNe – “batch mode”

- ▶ Chaque motif est
 1. Trouvé dans le temps imparti
 2. Marqué comme non trouvé dans le temps imparti
 3. Inconnu (temps imparti écoulé)

GraMoFoNe – “batch mode”



- ▶ + de bruit pour la mouche (faux négatifs déconnectent la solution, faux positifs donnent des “mauvaises” solutions)
- ▶ 5-20s (petits M), 40-60s (grands). Mais PB “boite noire”

Plan

Introduction

Complexité paramétrée et motifs avec topologie

Motifs sans topologie

Le problème GRAPH MOTIF

Des logiciels pour GRAPH MOTIF

Conclusion

Conclusion

- ▶ Deux vues pour la recherche de motifs
 1. Motifs avec topologie
 2. Motifs sans topologie
- ▶ Problèmes difficiles donc :
 - ▶ Complexité paramétrée
 - ▶ Approximation

Conclusion

- ▶ Le bruit pousse à la recherche d'extensions plus souples
 - ▶ contrainte couleurs
 - ▶ contrainte connexité
 - ▶ Simple connexité : FPT
 - ▶ 2-connexité : $W[1]$ -difficile
 - ▶ Modules ?
 - ▶ ?
 - ▶ ?

Questions sur la recherche de motifs dans les graphes colorés ?

Florian Sikora
(encadré par Guillaume Blin et Stéphane Vialette)

Université Paris-Est, LIGM - UMR CNRS 8049

Séminaire Combi LINA – 06/2010

GRAPH MOTIF et polynômes

- ▶ Si le **motif est un multiensemble**, un monôme non multilinéaire sur les couleurs est pourtant une solution

GRAPH MOTIF et polynômes

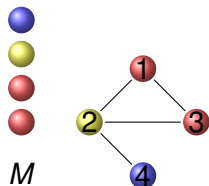
- ▶ Si le **motif est un multiensemble**, un monôme non multilinéaire sur les couleurs est pourtant une solution
- ▶ On introduit des variables x_u pour les noeuds du graphe
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $y_{c,1}, y_{c,2}, \dots, y_{c,m}$
- ▶ On modifie la construction du circuit :
$$P_{u,1} = x_u \cdot (y_{c,1} + y_{c,2} + \dots + y_{c,m})$$
 - ▶ Les variables x assurent qu'un noeud n'est utilisé qu'une fois
 - ▶ Les variables y donnent le bon nombre de couleurs demandés par le motif

GRAPH MOTIF et polynômes

- ▶ Si le **motif est un multiensemble**, un monôme non multilinéaire sur les couleurs est pourtant une solution
- ▶ On introduit des variables x_u pour les noeuds du graphe
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $y_{c,1}, y_{c,2}, \dots, y_{c,m}$
- ▶ On modifie la construction du circuit :

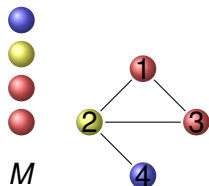
$$P_{u,1} = x_u \cdot (y_{c,1} + y_{c,2} + \dots + y_{c,m})$$
 - ▶ Les variables x assurent qu'un noeud n'est utilisé qu'une fois
 - ▶ Les variables y donnent le bon nombre de couleurs demandés par le motif
- ▶ On cherche maintenant un monôme de degré $2k$ (noeuds + couleurs)

Exemple



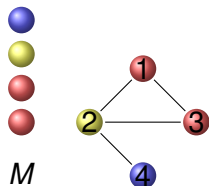
$$x_1(y_{R,1} + y_{R,2}) \cdot x_2 y_{Y,1} \cdot x_3(y_{R,1} + y_{R,2}) \cdot x_4 y_{B,1} + \dots$$

Exemple



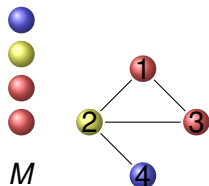
$$\begin{aligned}
 & x_1(y_{R,1} + y_{R,2}) \cdot x_2 y_{Y,1} \cdot x_3(y_{R,1} + y_{R,2}) \cdot x_4 y_{B,1} + \dots \\
 &= x_1 \textcolor{red}{y}_{R,1} \cdot x_2 y_{Y,1} \cdot x_3 \textcolor{red}{y}_{R,1} \cdot x_4 y_{B,1} + \dots
 \end{aligned}$$

Exemple



$$\begin{aligned}
 & x_1(y_{R,1} + y_{R,2}) \cdot x_2 y_{Y,1} \cdot x_3(y_{R,1} + y_{R,2}) \cdot x_4 y_{B,1} + \dots \\
 &= x_1 \textcolor{red}{y}_{R,1} \cdot x_2 y_{Y,1} \cdot x_3 \textcolor{red}{y}_{R,1} \cdot x_4 y_{B,1} + \\
 & \quad x_1 \textcolor{red}{y}_{R,1} \cdot x_2 y_{Y,1} \cdot x_3 \textcolor{red}{y}_{R,2} \cdot x_4 y_{B,1} + \dots
 \end{aligned}$$

Exemple



$$\begin{aligned}
 & x_1(y_{R,1} + y_{R,2}) \cdot x_2 y_{Y,1} \cdot x_3(y_{R,1} + y_{R,2}) \cdot x_4 y_{B,1} + \dots \\
 &= x_1 y_{R,1} \cdot x_2 y_{Y,1} \cdot x_3 y_{R,1} \cdot x_4 y_{B,1} + \\
 & x_1 y_{R,1} \cdot x_2 y_{Y,1} \cdot x_3 y_{R,2} \cdot x_4 y_{B,1} + \dots
 \end{aligned}$$

- Un monôme multilinéaire solution est de degré 8.