

Aspects algorithmiques de la comparaison d'éléments biologiques

Florian Sikora

Université Paris-Est

30 septembre 2011

Plan

Introduction

Complexité paramétrée et algorithmes d'approximation

Recherche exacte de motifs

Recherche approchée de motifs

Logiciels

Perspectives

Plan

Introduction

Complexité paramétrée et algorithmes d'approximation

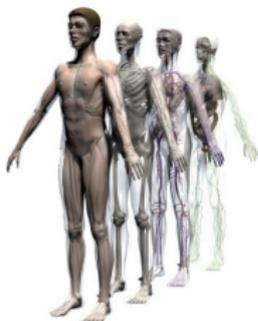
Recherche exacte de motifs

Recherche approchée de motifs

Logiciels

Perspectives

Motivations



25000



30000



45000

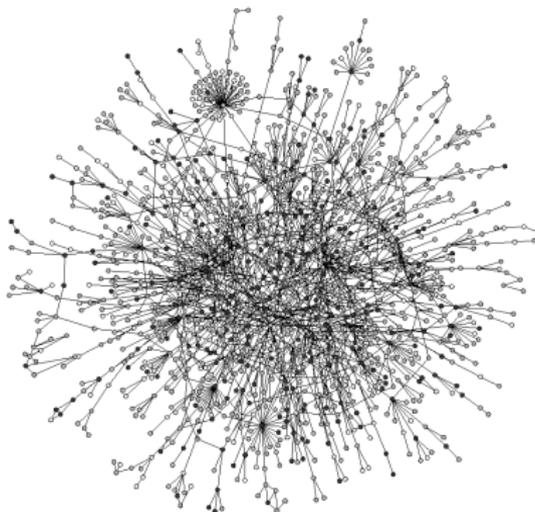
- Complexité de l'homme \Leftrightarrow nombre de gènes ?

Réseaux biologiques

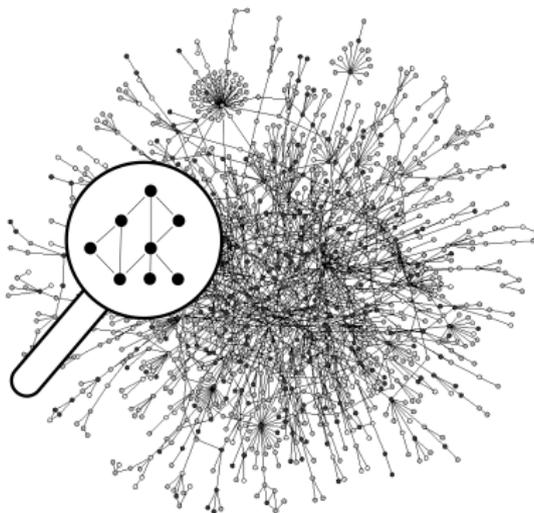
- ▶ Énumérer les composants d'un avion ne suffit pas pour comprendre son fonctionnement.
- ▶ Étudier les relations entre les éléments.

Réseaux biologiques

- ▶ Les éléments biologiques (protéines, métabolites...) peuvent interagir entre-eux.



Réseaux biologiques



- ▶ Modélisation par un **graphe**.
 - ▶ Les **éléments** sont représentés par les **nœuds**.
 - ▶ Les **interactions** sont représentées par les **arêtes**.

Enjeux

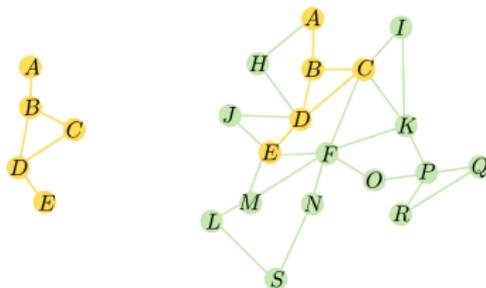
- ▶ Nouvelles techniques : le nombre de données augmente très rapidement [SHARAN ET IDEKER 2006].
 - ▶ 2001 : quelques centaines d'interactions.
 - ▶ 2006 : plusieurs milliers.
- ▶ Beaucoup de bases de données :
 - ▶ BIND,
 - ▶ DIP,
 - ▶ KEGG,
 - ▶ MINT,
 - ▶ ...

Enjeux

- ▶ Nouvelles techniques : le nombre de données augmente très rapidement [SHARAN ET IDEKER 2006].
 - ▶ 2001 : quelques centaines d'interactions.
 - ▶ 2006 : plusieurs milliers.
- ▶ Beaucoup de bases de données :
 - ▶ BIND,
 - ▶ DIP,
 - ▶ KEGG,
 - ▶ MINT,
 - ▶ ...
- ▶ Besoin de l'outil informatique.

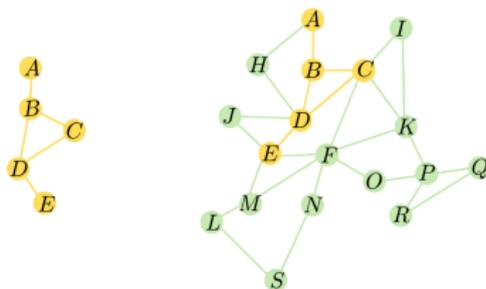
Rechercher des motifs

- ▶ Retrouver un sous-réseau respectant les labels et la topologie du motif.
- ▶ Deux labels correspondent si les éléments ont une fonction similaire (par ex. homologies pour les protéines).



Rechercher des motifs

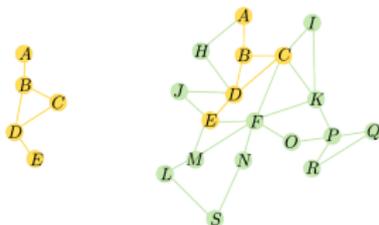
- ▶ Retrouver un sous-réseau respectant les labels et la topologie du motif.
- ▶ Deux labels correspondent si les éléments ont une fonction similaire (par ex. homologues pour les protéines).



- ▶ Chercher des motifs pour retrouver des fonctions connues.
- ▶ **Déduire** les informations d'espèces **peu connues** depuis des espèces **bien connues**.

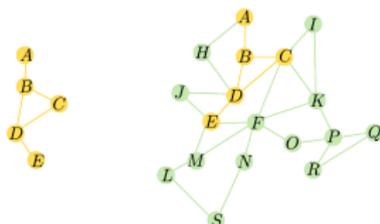
Rechercher des motifs

- ▶ Problèmes rapidement **NP-complets**.
 - ▶ N'existe probablement **pas d'algorithme exact** de complexité pire cas **polynomiale** (par ex. $\mathcal{O}(n^c)$).
 - ▶ Complexité **exponentielle** (par ex. $\mathcal{O}(c^n)$) pour au moins une instance.



Rechercher des motifs

- ▶ Mais tout n'est pas perdu !
- ▶ Idée : exploiter le fait que les **motifs sont plus petits** (~ 5 à 15 nœuds) que le réseau (par ex. ~ 5.000 pour la levure).
- ▶ Confiner la partie exponentielle à k (taille motif) au lieu de n (taille du réseau) : sujet de la **complexité paramétrée**.



Plan

Introduction

Complexité paramétrée et algorithmes d'approximation

Recherche exacte de motifs

Recherche approchée de motifs

Logiciels

Perspectives

Algorithmes FPT

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

Algorithmes FPT

- ▶ Beaucoup de problèmes (paramétrés) sont de la forme :
 - ▶ Entrée : Un objet X , $|X| = n$, un entier k .
 - ▶ Question : Est-ce que X a une propriété selon k ?
- ▶ Plusieurs paramètres possibles (souvent la taille de la solution).

Exemple : Vertex Cover



Exemple : Vertex Cover



Complexité paramétrée



- ▶ Problème dans la classe FPT s'il existe un algorithme **exact exponentiel** seulement en son **paramètre k** (et pas en la taille de l'entrée n) [DOWNEY ET FELLOWS 1999].
- ▶ Complexité $f(k) \cdot n^c$, avec c une constante, et f n'importe quelle fonction (par ex. $\mathcal{O}(2^k \cdot n^2)$).

Complexité paramétrée



- ▶ Problème dans la classe FPT s'il existe un algorithme **exact exponentiel** seulement en son **paramètre k** (et pas en la taille de l'entrée n) [DOWNEY ET FELLOWS 1999].
- ▶ Complexité $f(k) \cdot n^c$, avec c une constante, et f n'importe quelle fonction (par ex. $\mathcal{O}(2^k \cdot n^2)$).
- ▶ Parfois impossible : W[1]-difficulté.

Approximation

- ▶ Un autre possibilité pour contourner la difficulté d'un problème : l'approximation.

Approximation

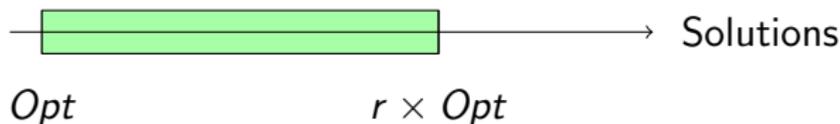
- ▶ Un autre possibilité pour contourner la difficulté d'un problème : l'approximation.
- ▶ Pour un problème d'optimisation NP-difficile, pas d'algorithme **polynomial** possible (sauf si $P = NP$)...
- ▶ ... si la solution doit être optimale !

Approximation

- ▶ Un autre possibilité pour contourner la difficulté d'un problème : l'approximation.
- ▶ Pour un problème d'optimisation NP-difficile, pas d'algorithme **polynomial** possible (sauf si $P = NP$)...
- ▶ ... si la solution doit être optimale !
- ▶ Sacrifier de l'exactitude pour obtenir un algorithme polynomial.

Approximation (problème de minimisation)

- ▶ On borne l'erreur.
- ▶ Un algorithme est de ***r*-approximation** s'il est polynomial et qu'il retourne des solutions dont le coût est au pire $r \times Opt$.



Approximation – Difficulté

- ▶ Possible de prouver qu'il n'existe pas d'algorithme d'approximation inférieur à un certain ratio.

Approximation – Difficulté

- ▶ Possible de prouver qu'il n'existe pas d'algorithme d'approximation inférieur à un certain ratio.
- ▶ But : rapprocher les deux frontières.



Exemple pour VERTEX COVER

Plan

Introduction

Complexité paramétrée et algorithmes d'approximation

Recherche exacte de motifs

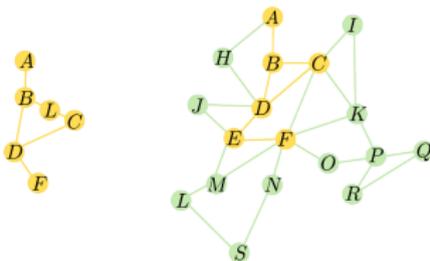
Recherche approchée de motifs

Logiciels

Perspectives

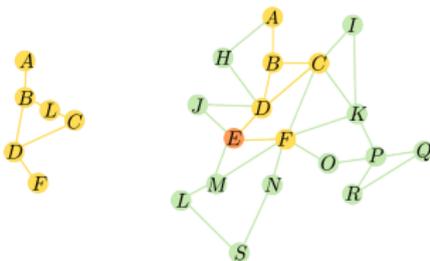
Recherche de motifs

- ▶ Vision “classique” de la recherche de motifs :
 - ▶ Le motif a une certaine topologie (chemin, arbre, graphes spécifiques...).
- ▶ La solution correspond aux labels **et** à la topologie du motif.



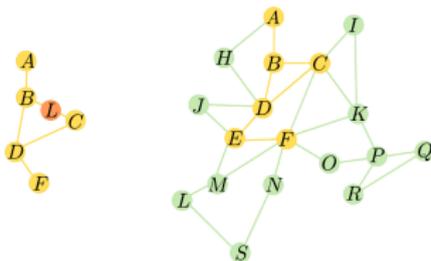
Recherche de motifs

- ▶ Vision “classique” de la recherche de motifs :
 - ▶ Le motif a une certaine topologie (chemin, arbre, graphes spécifiques...).
- ▶ La solution correspond aux labels **et** à la topologie du motif.
- ▶ Un nombre borné d'**insertions** et délétions peut-être autorisé



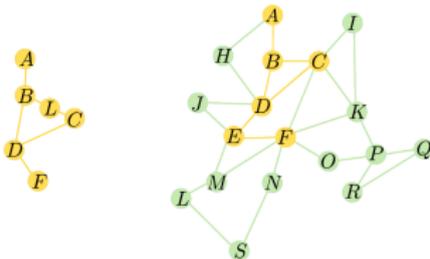
Recherche de motifs

- ▶ Vision “classique” de la recherche de motifs :
 - ▶ Le motif a une certaine topologie (chemin, arbre, graphes spécifiques...).
- ▶ La solution correspond aux labels **et** à la topologie du motif.
- ▶ Un nombre borné d’insertions et **délétions** peut-être autorisé



Recherche de motifs

- ▶ Vision “classique” de la recherche de motifs :
 - ▶ Le motif a une certaine topologie (chemin, arbre, graphes spécifiques...).
- ▶ La solution correspond aux labels **et** à la topologie du motif.
- ▶ Un nombre borné d’insertions et délétions peut-être autorisé
- ▶ Beaucoup d’algorithmes FPT, selon différents types de topologie [SHLOMI ET AL. 2006, DOST ET AL. 2007].
 - ▶ PADA1 [BLIN, S., VIALETTE 2009,2010].



Graph Motif

- ▶ Constat : les données biologiques sont expérimentales donc **très bruitées** [EDWARDS ET AL. 2002]
 - ▶ Manque des informations (faux négatifs). Estimé à 50%.
 - ▶ Informations erronées (faux positifs). Estimé à 50%.
- ▶ La topologie du motif peut ne pas être connue *a priori*.
- ▶ Différents motifs peuvent avoir une même topologie.

Graph Motif

- ▶ Constat : les données biologiques sont expérimentales donc **très bruitées** [EDWARDS ET AL. 2002]
 - ▶ Manque des informations (faux négatifs). Estimé à 50%.
 - ▶ Informations erronées (faux positifs). Estimé à 50%.
- ▶ La topologie du motif peut ne pas être connue *a priori*.
- ▶ Différents motifs peuvent avoir une même topologie.
- ▶ La topologie du motif peut ne pas être pertinente.

Graph Motif – Définition

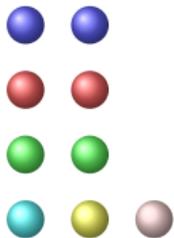
- ▶ Chaque nœud du réseau est coloré selon sa famille.
- ▶ Le motif est juste un (multi-)ensemble de couleurs à retrouver connecté dans le réseau coloré [LACROIX ET AL. 2006].

Graph Motif – Définition

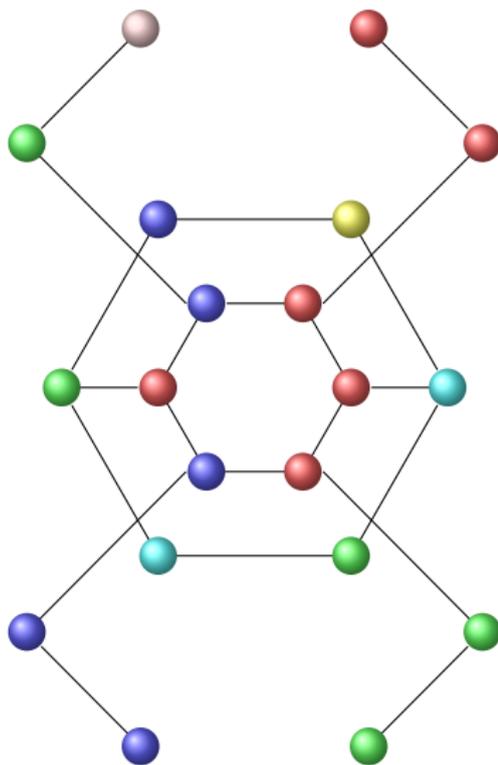
- ▶ Chaque nœud du réseau est coloré selon sa famille.
- ▶ Le motif est juste un (multi-)ensemble de couleurs à retrouver connecté dans le réseau coloré [LACROIX ET AL. 2006].
- ▶ Topologie n'est que la connexité de la solution.

Graph Motif – Un exemple jouet

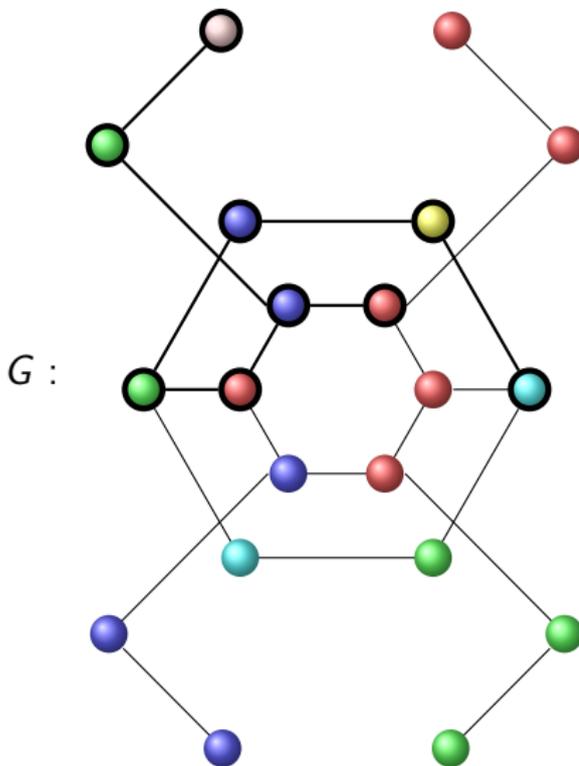
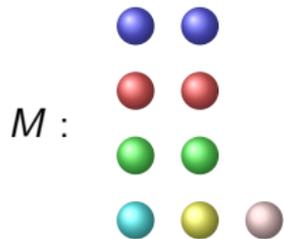
M :



G :

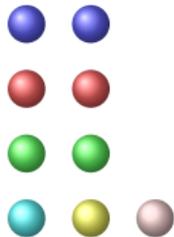


Graph Motif – Un exemple jouet

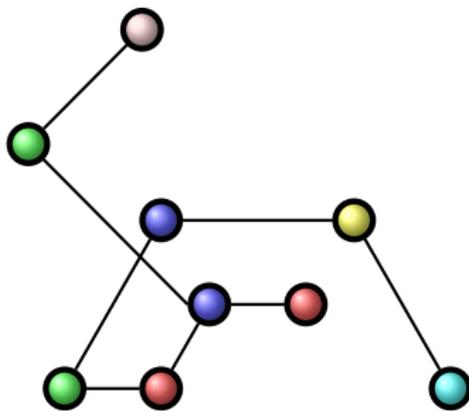


Graph Motif – Un exemple jouet

M :



G :



Graph Motif

- ▶ S'applique à différents types de réseaux biologiques.
 - ▶ Initialement sur les réseaux métaboliques [LACROIX ET AL. 2006].
 - ▶ Utilisable sur les réseaux PPI [BRUCKNER ET AL. 2009].
- ▶ Selon [BETZLER ET AL. 2008], peut être utilisé pour les réseaux sociaux.

Graph Motif – Difficulté

- ▶ Le problème est NP-complet, même si :
 - ▶ Le réseau est un arbre [LACROIX ET AL. 2006],
 - ▶ Cet **arbre est de degré maximum 3** et le motif est un **colorful** [FELLOWS ET AL. 2007],
 - ▶ L'**arbre est de profondeur 2** et le motif est un **colorful** [AMBALATH ET AL. 2010],
 - ▶ Le motif n'est constitué que de **2 couleurs** et le réseau est un graphe biparti de degré maximum 4 [FELLOWS ET AL. 2007].

Graph Motif – Complexité paramétré

- ▶ Le problème est FPT par la taille k du motif :
 - ▶ $\mathcal{O}^*(87^k)$ [FELLOWS ET AL. 2007].
 - ▶ $\mathcal{O}^*(4.3^k)$ (motif multi-ensemble) [BETZLER ET AL. 2008].
 - ▶ $\mathcal{O}^*(3^k)$ (motif colorful) [BETZLER ET AL. 2008].

Graph Motif – Complexité paramétré

- ▶ Le problème est FPT par la taille k du motif :
 - ▶ $\mathcal{O}^*(87^k)$ [FELLOWS ET AL. 2007].
 - ▶ $\mathcal{O}^*(4.3^k)$ (motif multi-ensemble) [BETZLER ET AL. 2008].
 - ▶ $\mathcal{O}^*(3^k)$ (motif colorful) [BETZLER ET AL. 2008].
- ▶ Programmation dynamique et color-coding : **complexité en espace exponentielle** en k .

Graph Motif – Complexité paramétré

- ▶ Le problème est FPT par la taille k du motif :
 - ▶ $\mathcal{O}^*(87^k)$ [FELLOWS ET AL. 2007].
 - ▶ $\mathcal{O}^*(4.3^k)$ (motif multi-ensemble) [BETZLER ET AL. 2008].
 - ▶ $\mathcal{O}^*(3^k)$ (motif colorful) [BETZLER ET AL. 2008].
- ▶ Programmation dynamique et color-coding : **complexité en espace exponentielle** en k .
- ▶ Mais $W[1]$ -difficile si paramétré par le **nombre de couleurs** différentes [FELLOWS ET AL. 2007].

Graph Motif et polynômes

- ▶ Amélioration de la complexité *via* un outil dû à Koutis et Williams [2008,2009].

Graph Motif et polynômes

- ▶ Amélioration de la complexité *via* un outil dû à Koutis et Williams [2008,2009].
- ▶ 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 **monôme multilinéaire de degré k** .

Graph Motif et polynômes

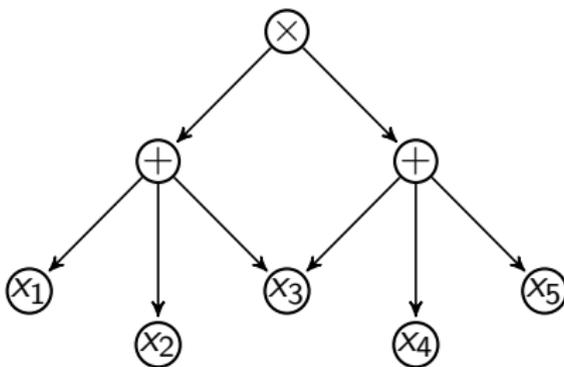
- ▶ Un monôme est **multilinéaire** si chaque variable du monôme n'apparaît qu'une seule fois.
- ▶ Par définition, le degré d'un monôme multilinéaire est son nombre de variables.
- ▶ Exemple : $P(X) = (x_1^2 x_3 x_5 + x_1 x_2 x_4 x_6)$:
 - ▶ $x_1 x_2 x_4 x_6$ est un monôme multilinéaire de degré 4.
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

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 $P(X) = (x_1 + x_2 + x_3)(x_3 + x_4 + x_5)$.



Graph Motif et polynômes

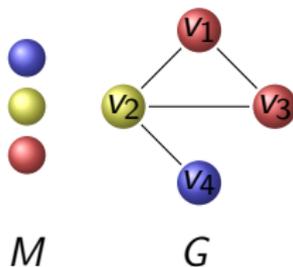
- ▶ Pour résoudre GRAPH MOTIF avec **motif colorful** :
 - ▶ On introduit des **variables correspondant aux couleurs**.
 - ▶ On construit un circuit correspondant aux couleurs de **tous les sous-graphes (connexes) de taille k** de G .
 - ▶ Un monôme multilinéaire de degré k correspond alors à un sous-graphe avec k couleurs différentes (motif colorful), donc k sommets différents.

Exemple

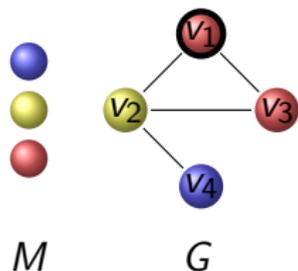
Construction récursive

Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .



Exemple



Construction récursive

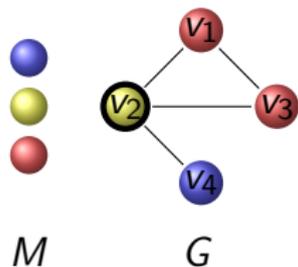
Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .

Pour $k' = 1$:

\times_R

Exemple



Construction récursive

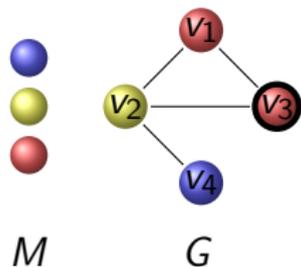
Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .

Pour $k' = 1$:

$$x_R \cdot (P_{2,v_2})$$

Exemple



Construction récursive

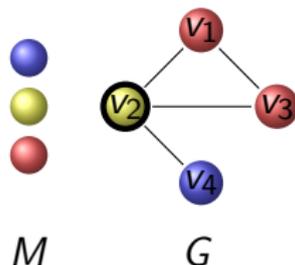
Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .

Pour $k' = 1$:

$$x_R \cdot (P_{2,v_2} + P_{2,v_3})$$

Exemple



Construction récursive

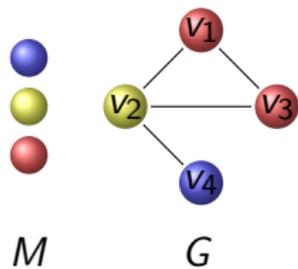
Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .

Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,v_2} + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (P_{1,v_1} + P_{1,v_3} + P_{1,v_4}) + P_{2,v_3})
 \end{aligned}$$

Exemple



Construction récursive

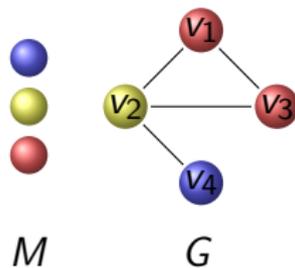
Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .

Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,v_2} + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (P_{1,v_1} + P_{1,v_3} + P_{1,v_4}) + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (x_R + x_R + x_B) + P_{2,v_3})
 \end{aligned}$$

Exemple



Construction récursive

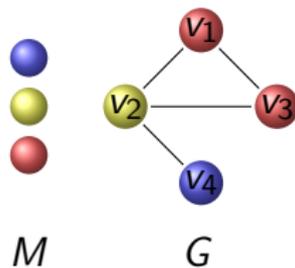
Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .

Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,v_2} + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (P_{1,v_1} + P_{1,v_3} + P_{1,v_4}) + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (x_R + x_R + x_B) + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot x_R + x_J \cdot x_R + x_J \cdot x_B + P_{2,v_3})
 \end{aligned}$$

Exemple



Construction récursive

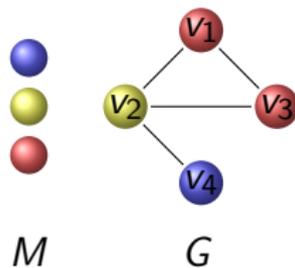
Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .

Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,v_2} + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (P_{1,v_1} + P_{1,v_3} + P_{1,v_4}) + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (x_R + x_R + x_B) + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot x_R + x_J \cdot x_R + x_J \cdot x_B + P_{2,v_3}) \\
 &= x_R x_J x_R + x_R x_J x_R + x_R x_J x_B + \dots
 \end{aligned}$$

Exemple



Construction récursive

Les monômes de degré $k = 3$ contenant le nœud v_1 sont la somme des produits des monômes :

1. de degrés $k' < k$ contenant le nœud v_1 ,
2. de degrés $k - k'$ contenant un voisin de v_1 .

Pour $k' = 1$:

$$\begin{aligned}
 & x_R \cdot (P_{2,v_2} + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (P_{1,v_1} + P_{1,v_3} + P_{1,v_4}) + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot (x_R + x_R + x_B) + P_{2,v_3}) \\
 &= x_R \cdot (x_J \cdot x_R + x_J \cdot x_R + x_J \cdot x_B + P_{2,v_3}) \\
 &= x_R x_J x_R + x_R x_J x_R + x_R x_J x_B + \dots
 \end{aligned}$$

Il y a un monôme multilinéaire (donc une solution).

Graph Motif et polynômes

- ▶ La construction est polynomiale.
- ▶ Il existe **un monôme multilinéaire de degré k** si et seulement s'il existe **un sous-graphe connexe colorful de taille k** .

Graph Motif et polynômes

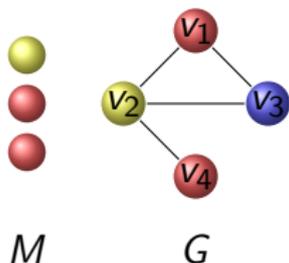
- ▶ La construction est polynomiale.
- ▶ Il existe **un monôme multilinéaire de degré k** si et seulement s'il existe **un sous-graphe connexe colorful de taille k** .
- ▶ On peut résoudre GRAPH MOTIF lorsque le motif est colorful en $\mathcal{O}^*(2^k)$ (algorithme randomisé) et en espace polynomial [GUILLEMOT ET S. 2010].

Graph Motif et polynômes – Multi-ensemble

- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.

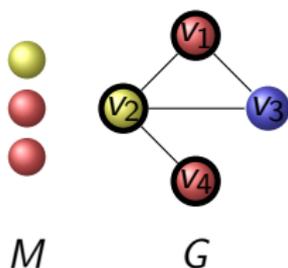
Graph Motif et polynômes – Multi-ensemble

- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.



Graph Motif et polynômes – Multi-ensemble

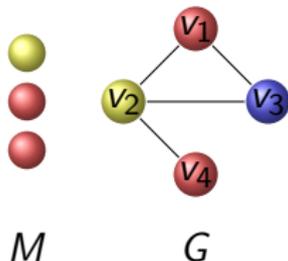
- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.



$$\dots = x_R x_J x_R + x_R x_J x_B + x_R x_J x_R + \dots$$

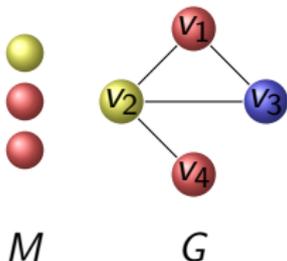
Graph Motif et polynômes – Multi-ensemble

- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $x_{c,1}, x_{c,2}, \dots, x_{c,m}$.



Graph Motif et polynômes – Multi-ensemble

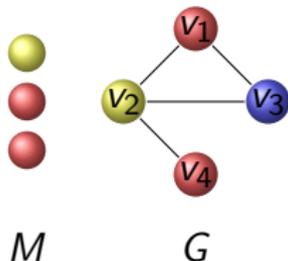
- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $x_{c,1}, x_{c,2}, \dots, x_{c,m}$.



$$\dots = (x_{R,1} + x_{R,2})x_{J,1}(x_{R,1} + x_{R,2}) + \dots$$

Graph Motif et polynômes – Multi-ensemble

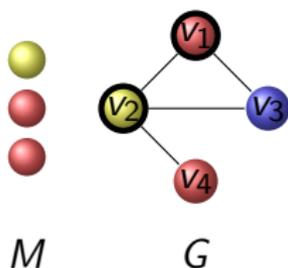
- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $x_{c,1}, x_{c,2}, \dots, x_{c,m}$ (reste insuffisant).



$$\begin{aligned}
 \dots &= (x_{R,1} + x_{R,2})x_{J,1}(x_{R,1} + x_{R,2}) + \dots \\
 &= x_{R,1}x_{J,1}x_{R,1} + x_{R,1}x_{J,1}x_{R,2} + x_{R,2}x_{J,1}x_{R,1} \\
 &\quad + x_{R,2}x_{J,1}x_{R,2} + \dots
 \end{aligned}$$

Graph Motif et polynômes – Multi-ensemble

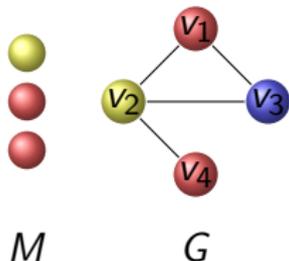
- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $x_{c,1}, x_{c,2}, \dots, x_{c,m}$ (reste insuffisant).



$$\begin{aligned}
 \dots &= (x_{R,1} + x_{R,2})x_{J,1}(x_{R,1} + x_{R,2}) + \dots \\
 &= x_{R,1}x_{J,1}x_{R,1} + x_{R,1}x_{J,1}x_{R,2} + x_{R,2}x_{J,1}x_{R,1} \\
 &\quad + x_{R,2}x_{J,1}x_{R,2} + \dots
 \end{aligned}$$

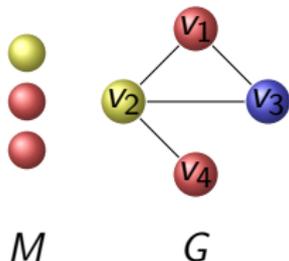
Graph Motif et polynômes – Multi-ensemble

- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $x_{c,1}, x_{c,2}, \dots, x_{c,m}$ (reste insuffisant).
- ▶ Ajout de variables y_v pour les nœuds v du graphe.



Graph Motif et polynômes – Multi-ensemble

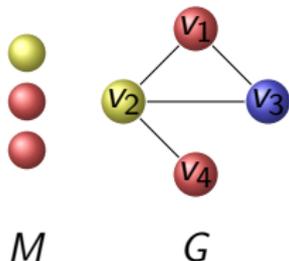
- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $x_{c,1}, x_{c,2}, \dots, x_{c,m}$ (reste insuffisant).
- ▶ Ajout de variables y_v pour les nœuds v du graphe.



$$\dots = y_1(x_{R,1} + x_{R,2}) \cdot y_2 x_{J,1} \cdot y_1(x_{R,1} + x_{R,2}) + \dots$$

Graph Motif et polynômes – Multi-ensemble

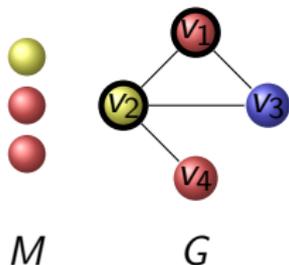
- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $x_{c,1}, x_{c,2}, \dots, x_{c,m}$ (reste insuffisant).
- ▶ Ajout de variables y_v pour les nœuds v du graphe.



$$\begin{aligned} \dots &= y_1(x_{R,1} + x_{R,2}) \cdot y_2 x_{J,1} \cdot y_1(x_{R,1} + x_{R,2}) + \dots \\ &= y_1 x_{R,1} y_2 x_{J,1} y_1 x_{R,1} + y_1 x_{R,1} y_2 x_{J,1} y_1 x_{R,2} + \dots \end{aligned}$$

Graph Motif et polynômes – Multi-ensemble

- ▶ Si le **motif est un multi-ensemble**, un monôme non multilinéaire sur les couleurs est une solution.
- ▶ La construction précédente n'est plus suffisante.
- ▶ Pour chaque couleur c qui apparaît m fois dans le motif, on introduit des variables $x_{c,1}, x_{c,2}, \dots, x_{c,m}$ (reste insuffisant).
- ▶ Ajout de variables y_v pour les nœuds v du graphe.



$$\begin{aligned} \dots &= y_1(x_{R,1} + x_{R,2}) \cdot y_2 x_{J,1} \cdot y_1(x_{R,1} + x_{R,2}) + \dots \\ &= y_1 x_{R,1} y_2 x_{J,1} y_1 x_{R,1} + y_1 x_{R,1} y_2 x_{J,1} y_1 x_{R,2} + \dots \end{aligned}$$

Graph Motif et polynômes – Multi-ensemble

- ▶ La construction est toujours polynomiale.
- ▶ Il existe un **monôme multilinéaire de degré $2k$** si et seulement si il existe une **solution de taille k** .

Graph Motif et polynômes – Multi-ensemble

- ▶ La construction est toujours polynomiale.
- ▶ Il existe un **monôme multilinéaire de degré $2k$** si et seulement si il existe une **solution de taille k** .
- ▶ On peut résoudre GRAPH MOTIF lorsque le motif est un multi-ensemble en $\mathcal{O}^*(4^k)$ (algorithme randomisé) et en espace polynomial [GUILLEMOT ET S. 2010].

Graph Motif et polynômes – Comptage

- ▶ **Compter** le nombre d'occurrences d'un motif **colorful** : $\mathcal{O}^*(2^k)$ (construction similaire).

Graph Motif et polynômes – Comptage

- ▶ **Compter** le nombre d'occurrences d'un motif **colorful** : $\mathcal{O}^*(2^k)$ (construction similaire).
- ▶ Mais, si le motif est un **multi-ensemble**, le problème est $\#W[1]$ -difficile [GUILLEMOT ET S. 2010].

Plan

Introduction

Complexité paramétrée et algorithmes d'approximation

Recherche exacte de motifs

Recherche approchée de motifs

Logiciels

Perspectives

Variantes

- ▶ Données expérimentales, **imprécisions**.
- ▶ Rechercher une occurrence exacte d'un motif peut échouer.

Variantes

- ▶ Données expérimentales, **imprécisions**.
- ▶ Rechercher une occurrence exacte d'un motif peut échouer.
- ▶ Autoriser des sommets supplémentaires dans la solution pour atteindre la connexité (**insertions**).

Variantes

- ▶ Données expérimentales, **imprécisions**.
- ▶ Rechercher une occurrence exacte d'un motif peut échouer.
- ▶ Autoriser des sommets supplémentaires dans la solution pour atteindre la connexité (**insertions**).
- ▶ Autoriser l'absence de couleurs du motif dans la solution (**délétions**).

Variantes

- ▶ Données expérimentales, **imprécisions**.
- ▶ Rechercher une occurrence exacte d'un motif peut échouer.
- ▶ Autoriser des sommets supplémentaires dans la solution pour atteindre la connexité (**insertions**).
- ▶ Autoriser l'absence de couleurs du motif dans la solution (**délétions**).
- ▶ Autoriser **plusieurs composantes connexes** dans la solution.

Variantes

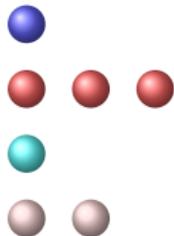
- ▶ Données expérimentales, **imprécisions**.
- ▶ Rechercher une occurrence exacte d'un motif peut échouer.
- ▶ Autoriser des sommets supplémentaires dans la solution pour atteindre la connexité (**insertions**).
- ▶ Autoriser l'absence de couleurs du motif dans la solution (**délétions**).
- ▶ Autoriser **plusieurs composantes connexes** dans la solution.
- ▶ Autoriser une **liste de couleurs** pour chaque sommet du réseau.

Une variante de Graph Motif : Minimum Substitutions

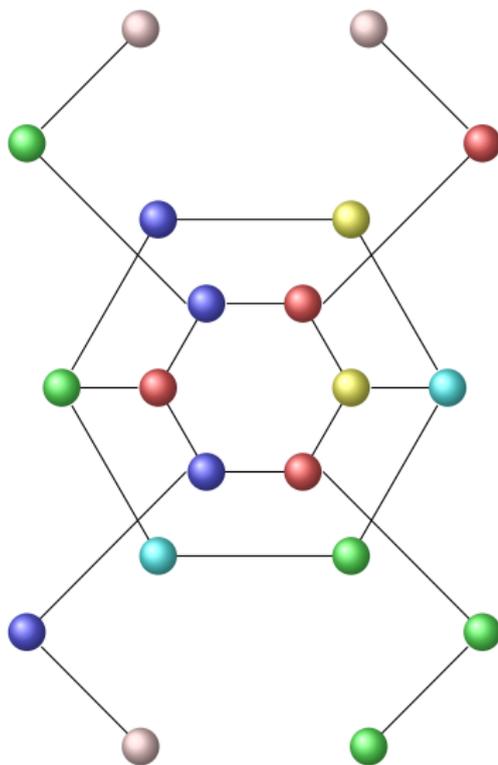
- ▶ Une autre variante : MINIMUM SUBSTITUTIONS [DONDI ET AL. 2011].
- ▶ Trouver une occurrence qui reprend "le plus de couleurs possible" du motif, mais de même taille que le motif.
- ▶ On *substitue* des couleurs du motif par de nouvelles couleurs.

Minimum Substitutions : Exemple

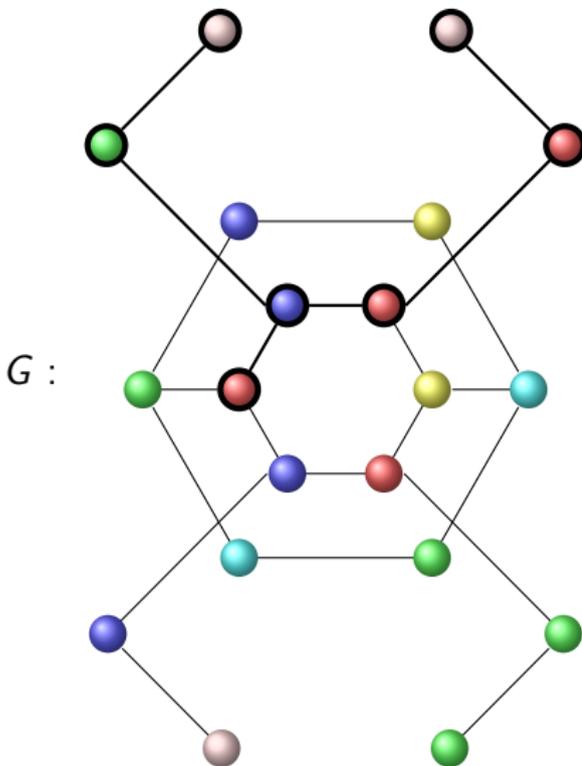
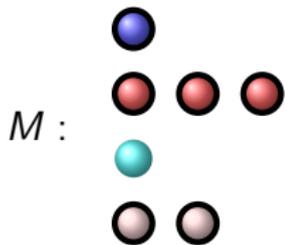
M :



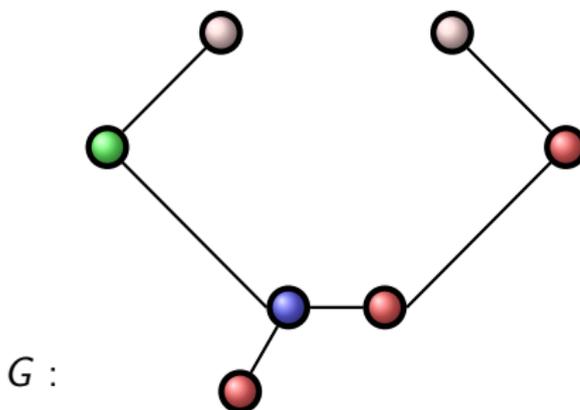
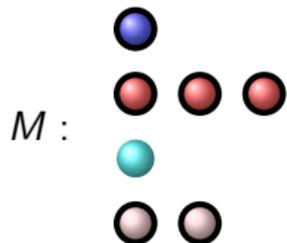
G :



Minimum Substitutions : Exemple



Minimum Substitutions : Exemple



Une variante de Graph Motif : Minimum Substitutions

- ▶ Il n'existe pas de ratio d'approximation inférieur à $c \log |V|$, sauf si $P = NP$, même lorsque le motif est colorful et G est un arbre de profondeur 2.

Construction

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$

Construction



$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$

Construction


$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$

Construction

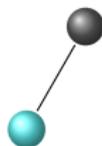

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$

 M G

Construction

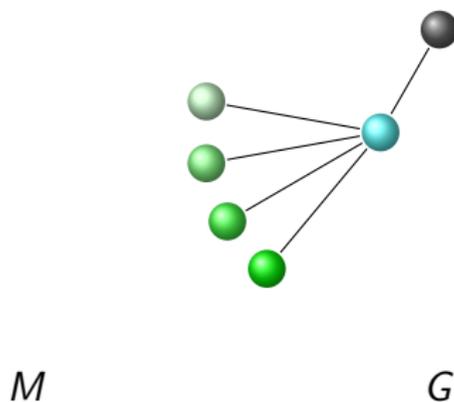
○○○○○ ●●●●● ○○○○ ● ● ●

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$

 M G

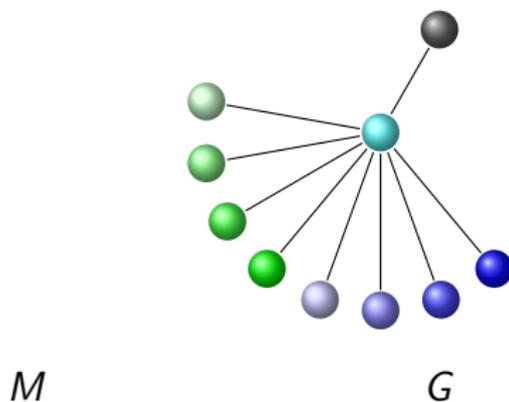
Construction

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



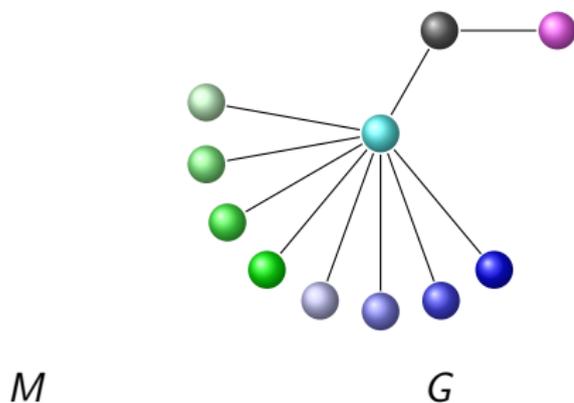
Construction

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



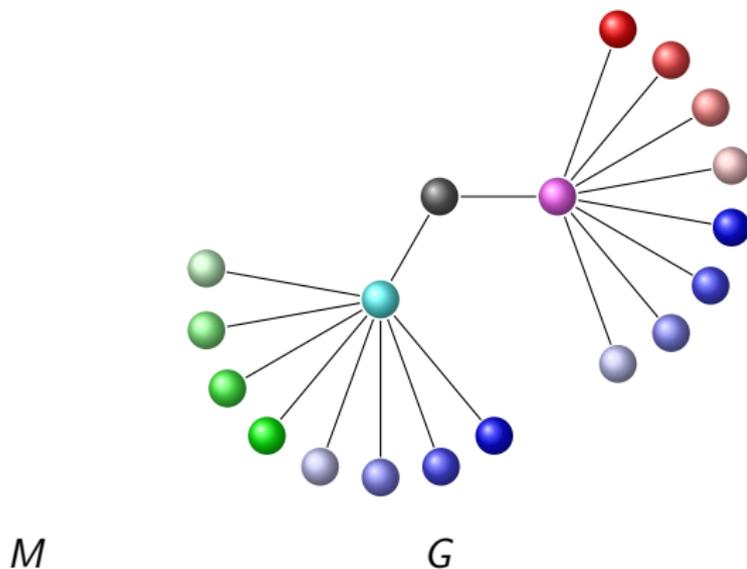
Construction

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



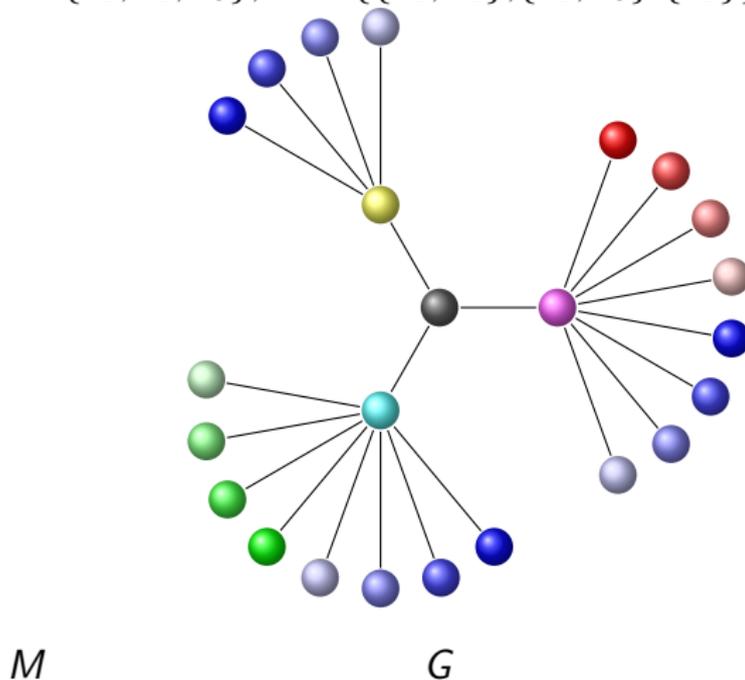
Construction

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



Construction

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



Construction

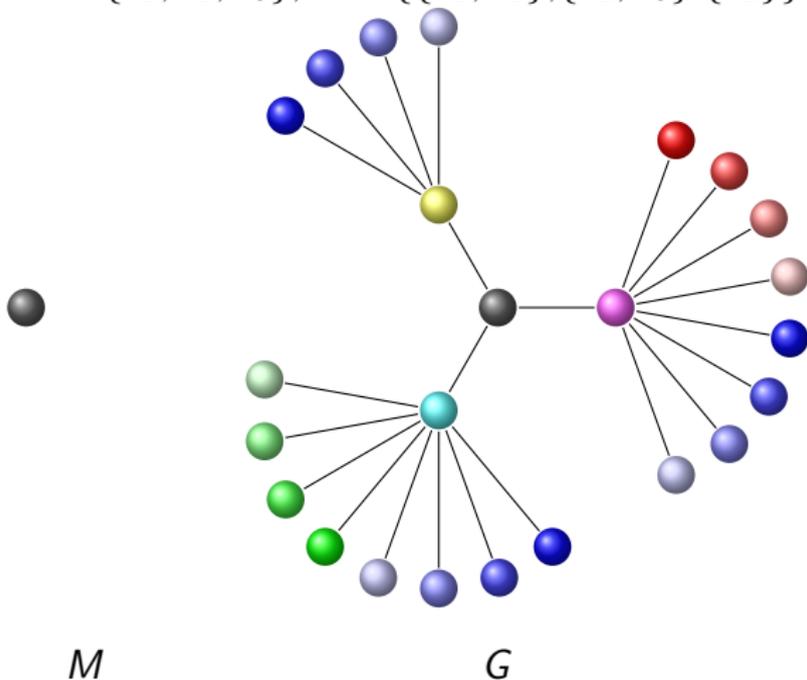
●●●●● ●●●●● ●●●●●

●

●

●

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



Construction

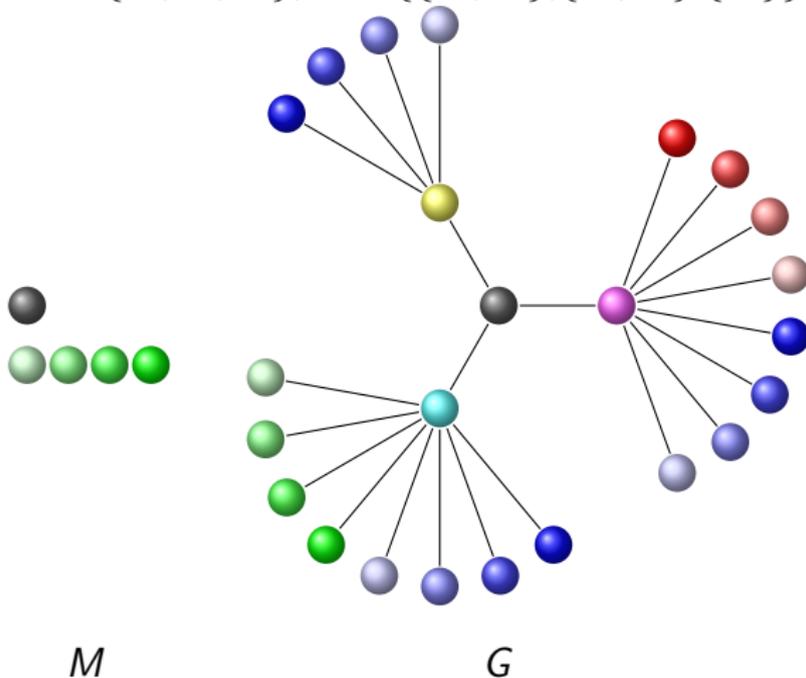
○○○○○ ○○○○○ ○○○○○

●

●

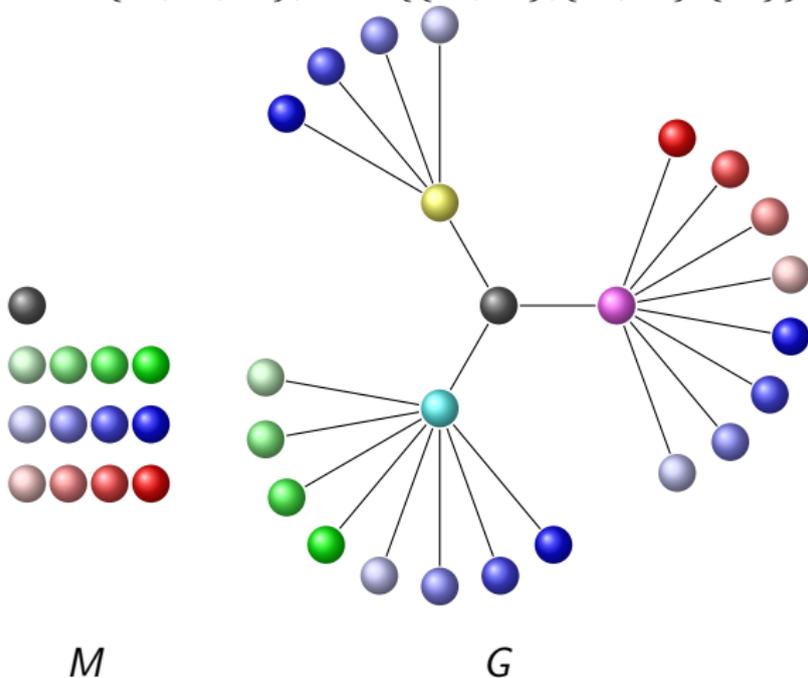
●

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



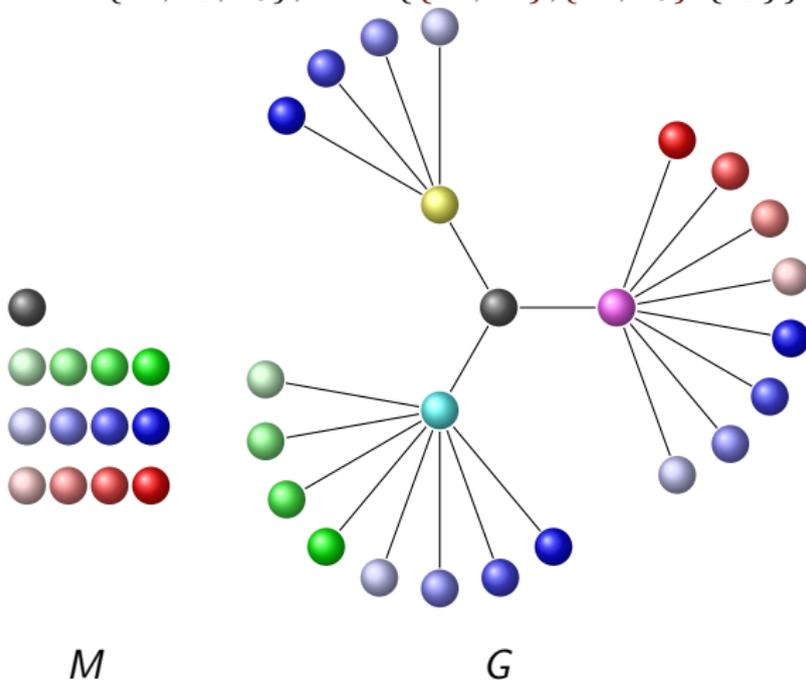
Construction

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



Construction

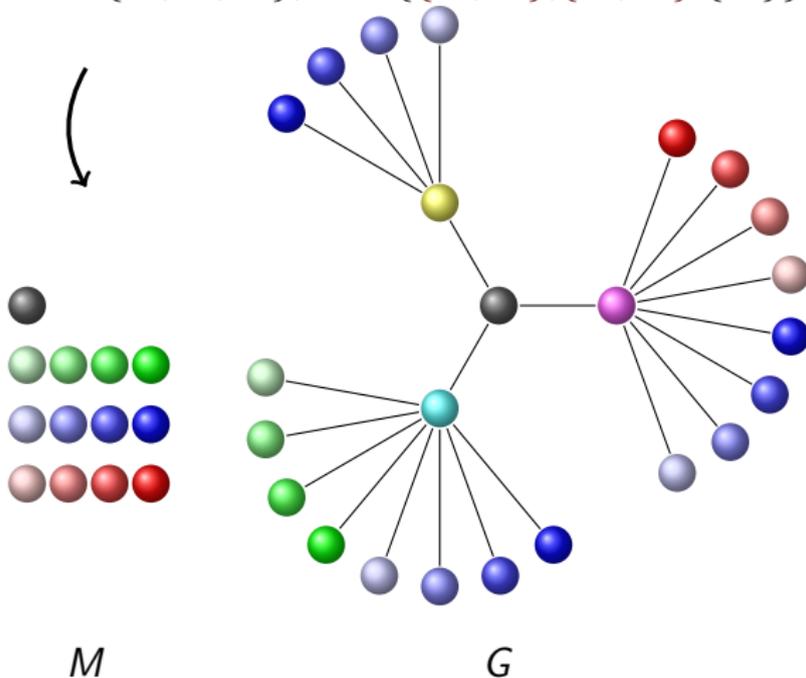
$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



$|\mathcal{S}'| = \text{nombre de substitutions}$

Construction

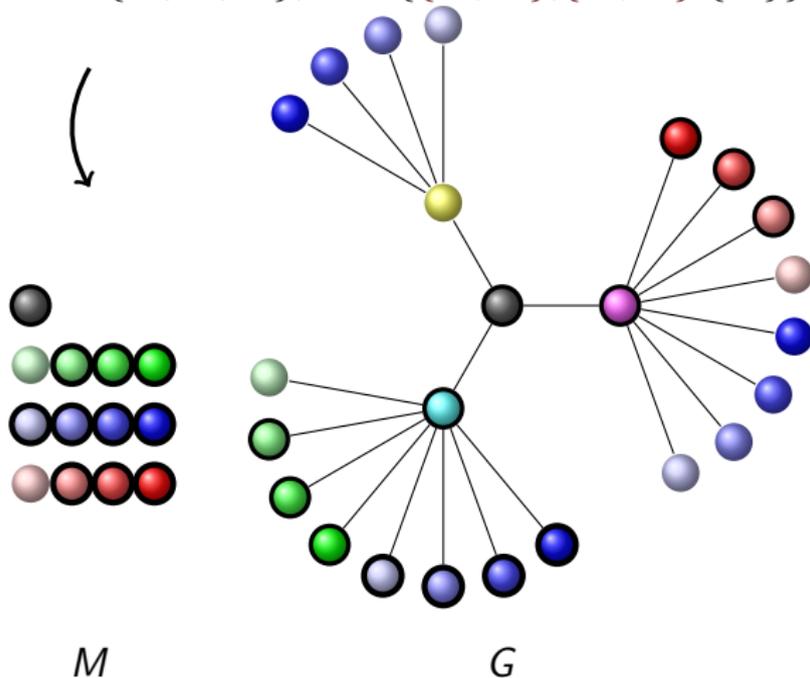
$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



$|\mathcal{S}'| = \text{nombre de substitutions}$

Construction

$$X = \{x_1, x_2, x_3\}, \mathcal{S} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_2\}\}$$



$|\mathcal{S}'| = \text{nombre de substitutions}$

Résultat

- ▶ Il n'existe pas de ratio d'approximation meilleur que $c \log |X|$ pour MINIMUM SET COVER (sauf si $P = NP$) [RAZ ET SAFRA 1997].

Résultat

- ▶ Il n'existe pas de ratio d'approximation meilleur que $c \log |X|$ pour MINIMUM SET COVER (sauf si $P = NP$) [RAZ ET SAFRA 1997].
- ▶ Il n'existe pas de ratio d'approximation meilleur que $c \log |V|$ pour MINIMUM SUBSTITUTIONS (sauf si $P = NP$) [RIZZI ET S., EN PRÉPARATION].

Plan

Introduction

Complexité paramétrée et algorithmes d'approximation

Recherche exacte de motifs

Recherche approchée de motifs

Logiciels

Perspectives

Logiciels pour Graph Motif

- ▶ **Motus** [LACROIX ET AL. 2006] : testé sur les réseaux métaboliques.
 - ▶ Mode recherche : trouver toutes les occurrences d'un motif donné.
 - ▶ Mode inférence : trouver tous les motifs (répétés) d'une certaine taille (testé avec 3 et 4).
- ▶ **Torque** [BRUCKNER ET AL. 2009] : un service web pour les réseaux PPI, qui ne gère que les motifs colorfults.
- ▶ **GraMoFoNe** [BLIN, S., VIALETTE 2010] : un greffon Cytoscape (environ 350 téléchargements).

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'intégration d'annotations à ces réseaux.
- ▶ Largement utilisé ("des centaines" d'articles l'utilisent pour de l'analyse).
- ▶ Maintenu...



Cytoscape

- ▶ Principal avantage : greffons.
- ▶ Ajout de fonctions :
 - ▶ Analyses,
 - ▶ Nouveaux layouts,
 - ▶ Formats de fichiers,
 - ▶ Connexion avec des bases de données,
 - ▶ ...

Programmation Pseudo-Booléenne

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

Programmation Pseudo-Booléenne

- ▶ 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 – Interface

File Edit View Select Layout Plugins Help

Search:

Control Panel

Network \VizMapper™ \ Editor \ Filter <

Pattern

Mouse_Complexes_CORUM.txt

DNA synthesome complex

ENSMUSG00000026134
ENSMUSG00000070544
ENSMUSG00000025395
ENSMUSG00000024833

Set pattern as multiset

FASTA

Choose Fasta File as Pattern

naFiles/Mouse_Biomart.fasta Choose F. Pat
 fastaFiles/SC_Torque.sif.fasta Choose F. Net

Blast parameters

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

Threshold: 16.0 LogEZero: 100.0

Options

NIns: 2 NDel: 3
insPenalty: 50 delPenalty: 60
Timeout: 500

Get only the first founded solution

Launch Close GraMoFoNe

SC_Torque.sif

Solution

YKL045W YOR510W YDL102W
YNL102W YBL035C YNL088W
YLR008C YIR006W YDL164C
YBR088C YOL006C YDL229W

Results Panel

GraMoFoNe results

Result	Details
	Score = 17.0 Rank = 1 Nb Nodes = 12 Nb Del = 3 Nb Ins = 2 (0 C + 2 NC)
	Score = 17.0 Rank = 2 Nb Nodes = 12 Nb Del = 3 Nb Ins = 2 (0 C + 2 NC)
	Score = 16.0 Rank = 3 Nb Nodes = 12 Nb Del = 3 Nb Ins = 2 (0 C + 2 NC)
	Score = 16.0 Rank = 4 Nb Nodes = 12 Nb Del = 3 Nb Ins = 2 (0 C + 2 NC)
	Score = 15.0 Rank = 5 Nb Nodes = 12 Nb Del = 3 Nb Ins = 2 (0 C + 2 NC)

YNL102W matched with: ENSMUSG00000038644
YBR088C matched with: ENSMUSG00000027342
YDL229W inserted as a not colored node
YBL035C matched with: ENSMUSG00000024833
YNL088W matched with: ENSMUSG00000017485
ENSMUSG00000030726 ENSMUSG00000024854 ENSMUSG0000002091
YDL164C matched with: ENSMUSG00000056394
YOL006C matched with: ENSMUSG00000070544
YIR006W matched with: ENSMUSG00000020471

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

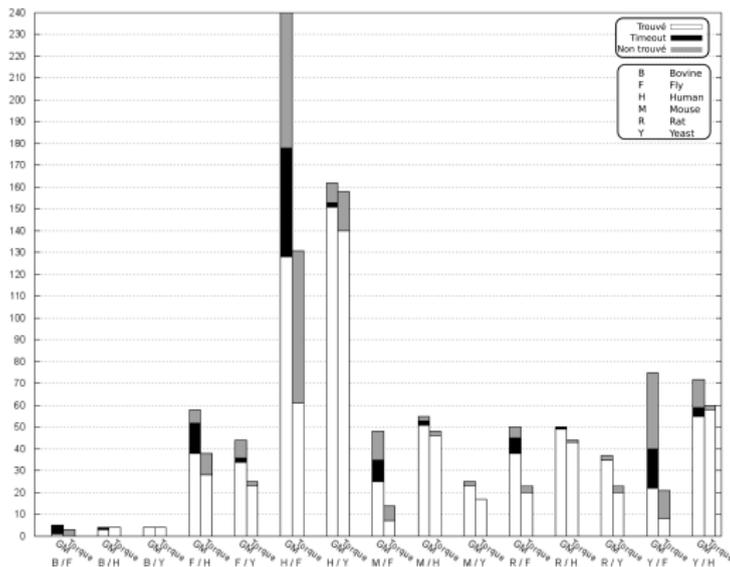
GraMoFoNe – “batch mode”

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

GraMoFoNe – “batch mode”

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

GraMoFoNe – “batch mode”



- Plus de bruit pour la mouche (faux négatifs déconnectent la solution, faux positifs donnent des “mauvaises” solutions).

GraMoFoNe – Temps de calcul

- ▶ 5 à 20 secondes pour de petits motifs, 40 à 60 pour des plus grands.
- ▶ Utiliser un solveur externe empêche de prédire les temps de calcul.

Plan

Introduction

Complexité paramétrée et algorithmes d'approximation

Recherche exacte de motifs

Recherche approchée de motifs

Logiciels

Perspectives

Perspectives... ou ce dont je n'ai pas parlé !

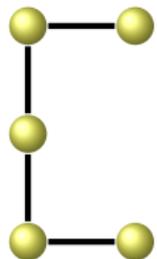
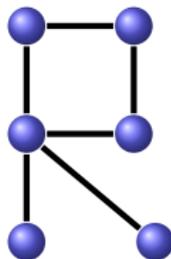
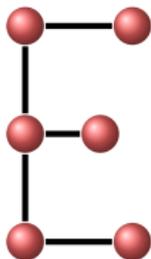
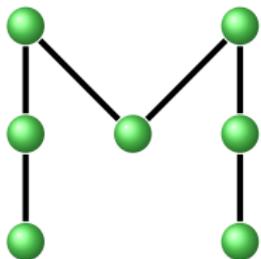
- ▶ Comparaison de réseaux biologiques hétérogènes [BLIN, FERTIN, MOHAMED-BABOU, RUSU, S. ET VIALETTE 2011].
- ▶ Réarrangements génomiques [BLIN, FERTIN, S., VIALETTE 2009], [BLIN, RIZZI, S. ET VIALETTE 2011], [YANG, S., BLIN, HAMEL, RIZZI ET ALURU, EN COURS DE SOUMISSION].
- ▶ Minimisation des duplications de gènes [BLIN, BONIZZONI, DONDI, RIZZI ET S., 2012].
- ▶ Plus longue sous-séquence commune sans répétition [BLIN, BONIZZONI, DONDI ET S., EN COURS DE SOUMISSION].

Perspectives

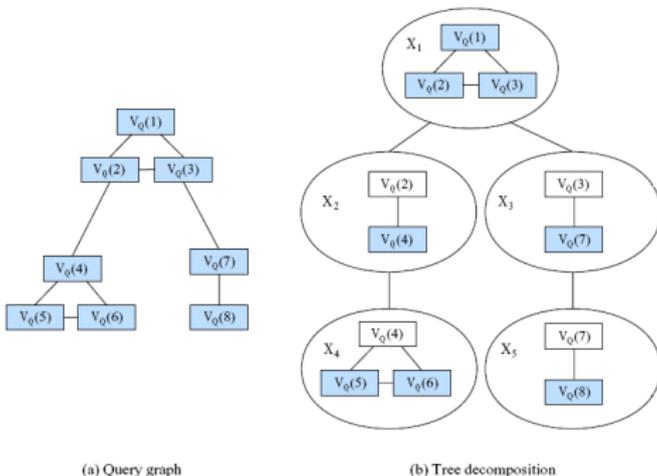
- ▶ Algorithme en $\mathcal{O}^*(2^k)$ pour GRAPH MOTIF quand le motif est un multi-ensemble ?

Perspectives

- ▶ Algorithme en $\mathcal{O}^*(2^k)$ pour GRAPH MOTIF quand le motif est un multi-ensemble ?
- ▶ Le bruit pousse à la recherche d'extensions plus souples :
 - ▶ Contraintes sur les couleurs.
 - ▶ Contraintes sur la connexité :
 - ▶ Simple connexité : FPT,
 - ▶ 2-connexité : W[1]-difficile,
 - ▶ Modules.



Décomposition arborescente

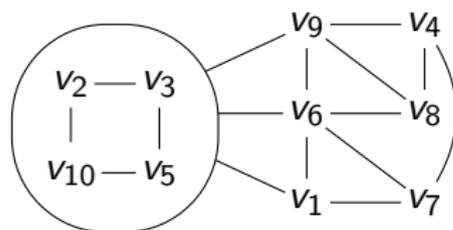
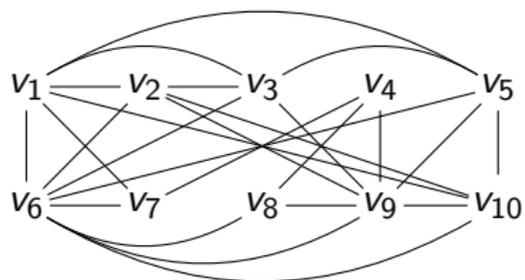


- ▶ l'union de tous les X_i de X est égale à V ,
- ▶ pour toute arête $\{v, w\}$ de E , il existe un nœud X_i de l'arbre T qui contient v et w ,
- ▶ si X_i et X_j contiennent un même sommet v , alors tous les nœuds X_z de T sur le chemin entre X_i et X_j contiennent v .

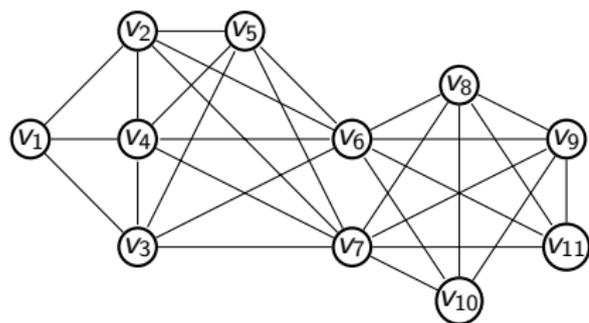
Matroïdes

- ▶ $\mathcal{M} = (S, I)$ défini sur :
 - ▶ Un ensemble fini $S = \{s_1, \dots, s_{|S|}\}$
 - ▶ Ensemble I de sous-ensembles sur S (indépendants)
- ▶ \mathcal{M} est un matroïde si :
 - ▶ si $X \subseteq Y$ et $Y \in I$, alors $X \in I$ (tout sous-ensemble d'un indépendant est un indépendant)
 - ▶ si $X, Y \in I$ et $|Y| > |X|$, alors $\exists s \in Y \setminus X$ t.q. $X \cup \{s\} \in I$ (existe un élément de Y absent de X qui peut être ajouté à X t.q. X reste indépendant)
- ▶ \mathcal{M} est matroïde de partition où il existe une partition de S en k parts disjointes $\{s_1, \dots, s_k\}$, k entiers positifs n_1, \dots, n_k et $I = \{X \subseteq S : |X \cap S_i| \leq n_i, 1 \leq i \leq k\}$

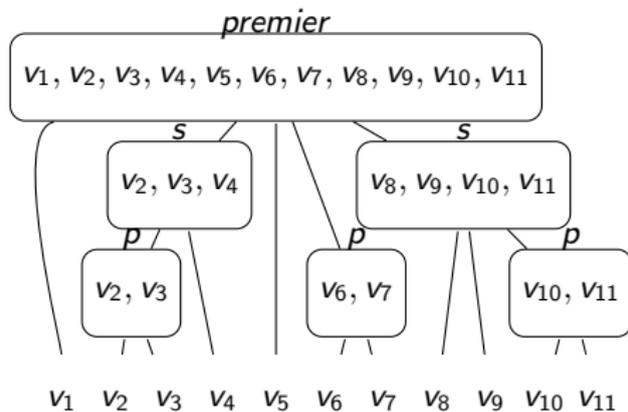
Modules



Modules



a)



b)

GraMoFoNe – Programmation Pseudo-Booléenne

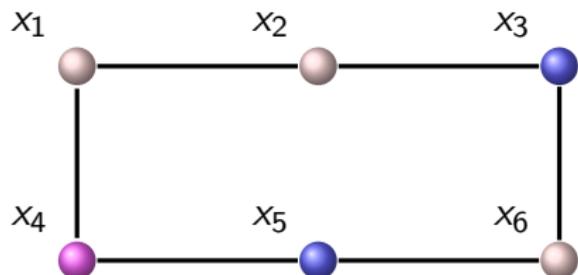
- ▶ 23 contraintes et 9 domaines de variables
- ▶ A respecter :
 1. La taille de la solution
 2. La coloration de la solution par rapport au motif
 3. La connexité de la solution (partie difficile)

GraMoFoNe – Variables

- ▶ Une variable x_v pour chaque nœud v
- ▶ $x_v = 1$ si v est dans la solution ($x_v = 0$ sinon)

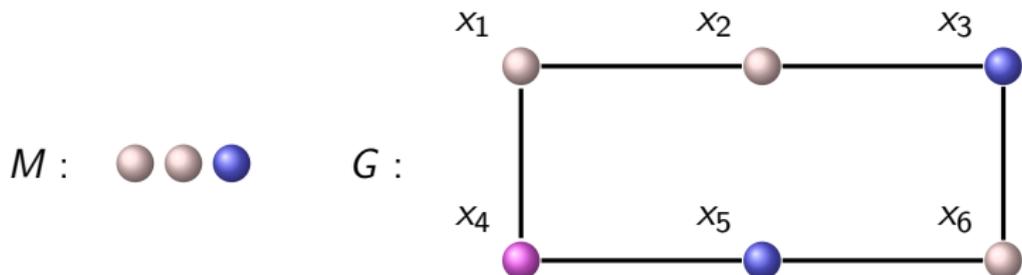


G :



GraMoFoNe – Variables

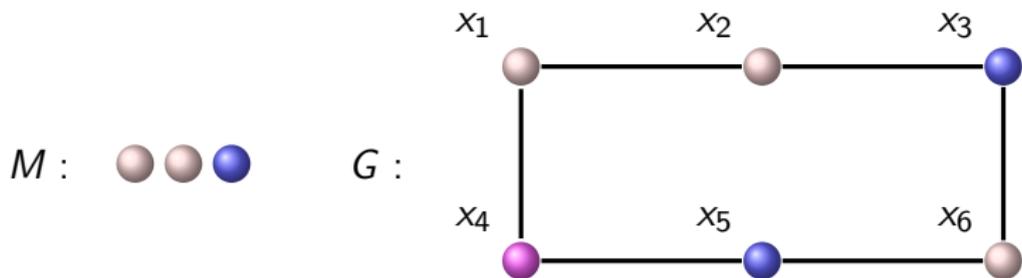
- ▶ Une variable x_v pour chaque nœud v
- ▶ $x_v = 1$ si v est dans la solution ($x_v = 0$ sinon)



- ▶ 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

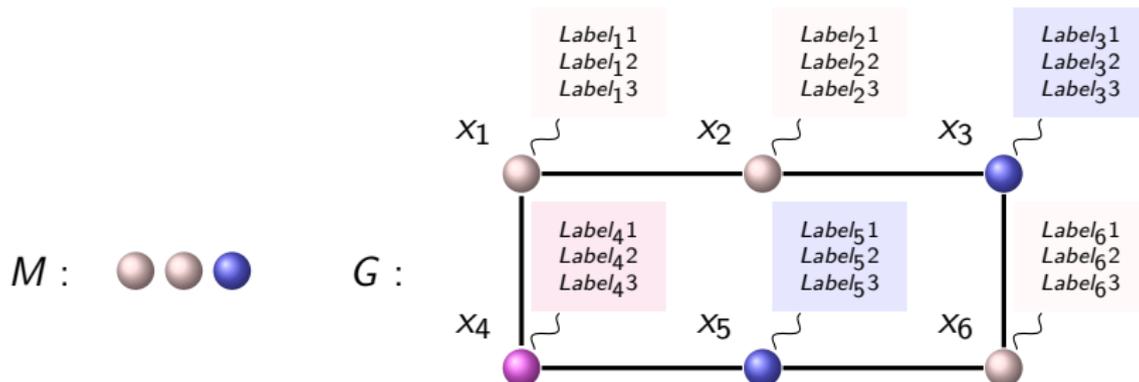
- ▶ Une variable x_v pour chaque nœud v
- ▶ $x_v = 1$ si v est dans la solution ($x_v = 0$ sinon)



- ▶ 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)

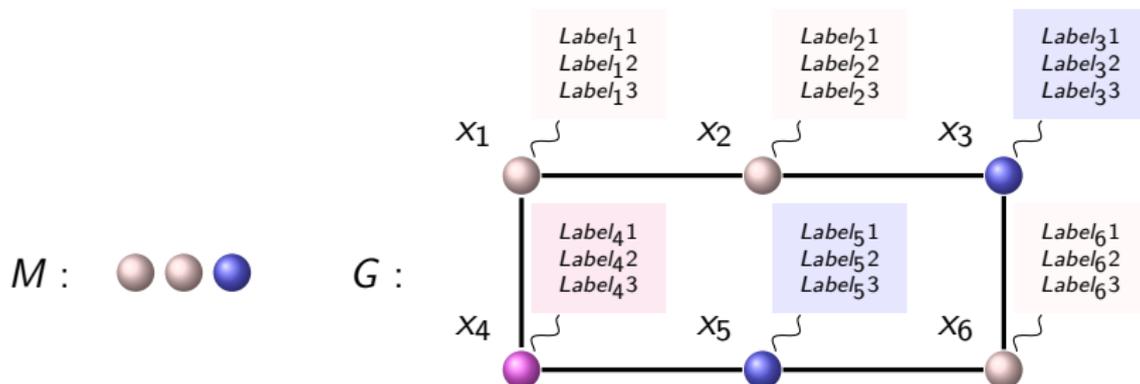
GraMoFoNe – Variables

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



GraMoFoNe – Variables

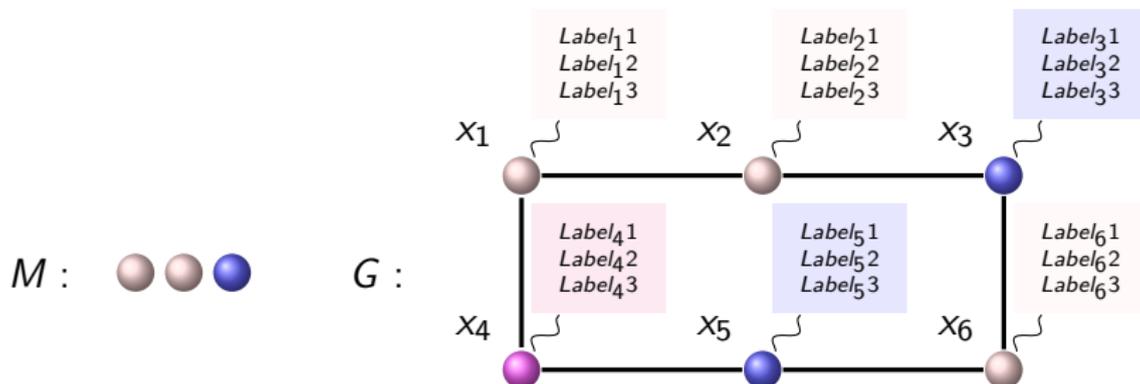
- ▶ Connexité : $|M|$ variables $Label_v$ pour chaque nœud v



- ▶ $|V|$ 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

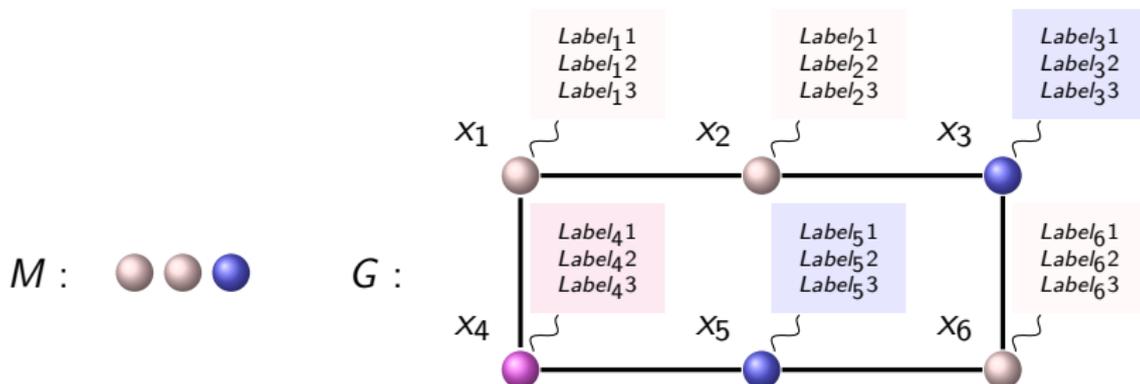
- ▶ Connexité : $|M|$ variables $Label_v$ pour chaque nœud v



- ▶ $|V|$ 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

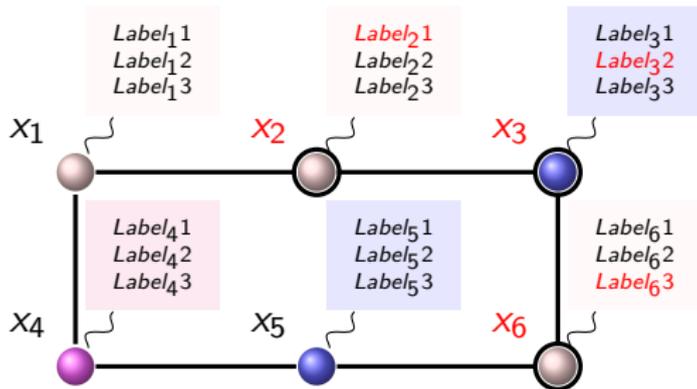


- ▶ $|V|$ contraintes "un seul label par nœud dans la solution" :
 - ▶ Pour chaque v , $x_v \Rightarrow (\sum_{i=1}^{|M|} Label_{v,i} = 1)$
- ▶ $|M|$ contraintes "un seul nœud par label donné"
 - ▶ Pour un label i donné, $\sum_{v \in V} Label_{v,i} = 1$
- ▶ $|V| \cdot |M|$ contraintes "un nœud avec un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

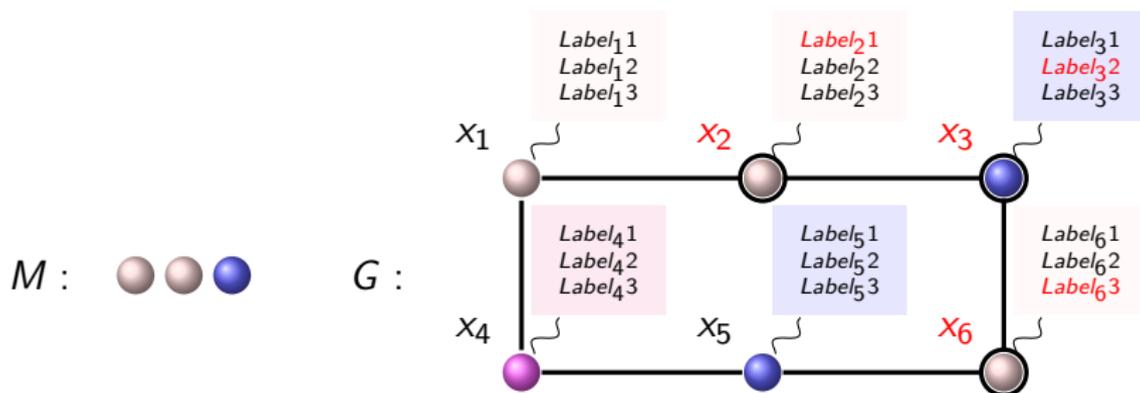
GraMoFoNe – Une solution

M : ● ● ●

G :

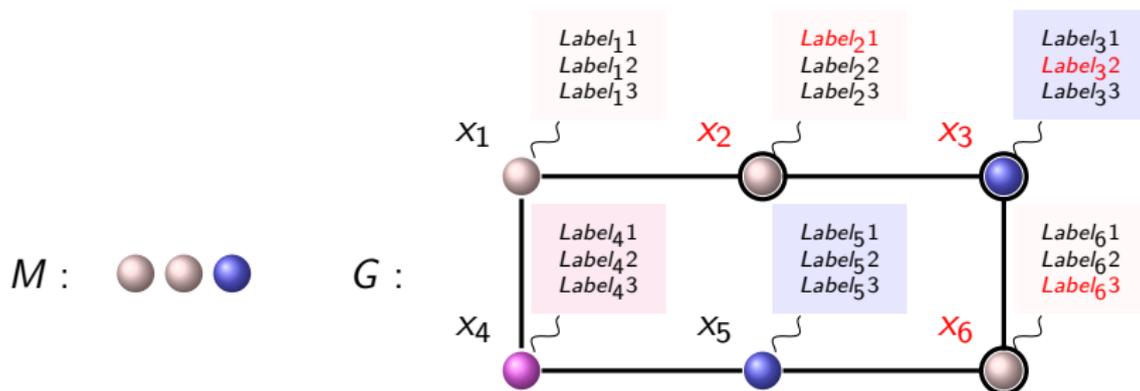


GraMoFoNe – Une solution



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

GraMoFoNe

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

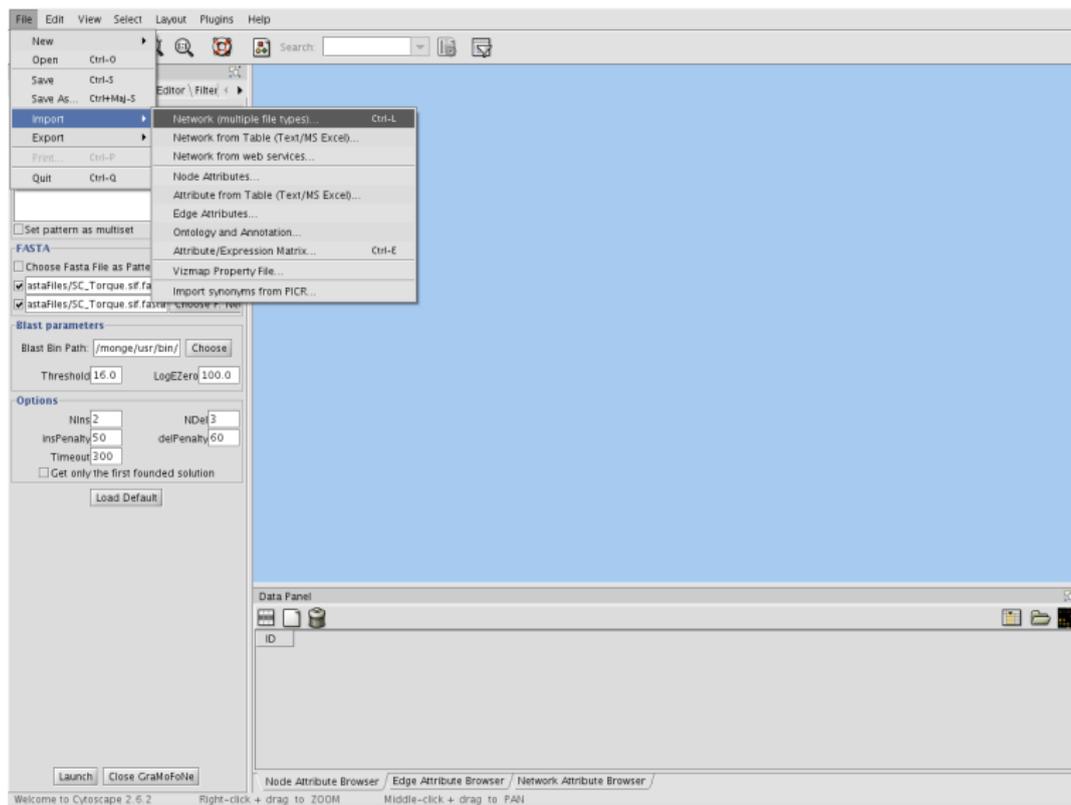
GraMoFoNe

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

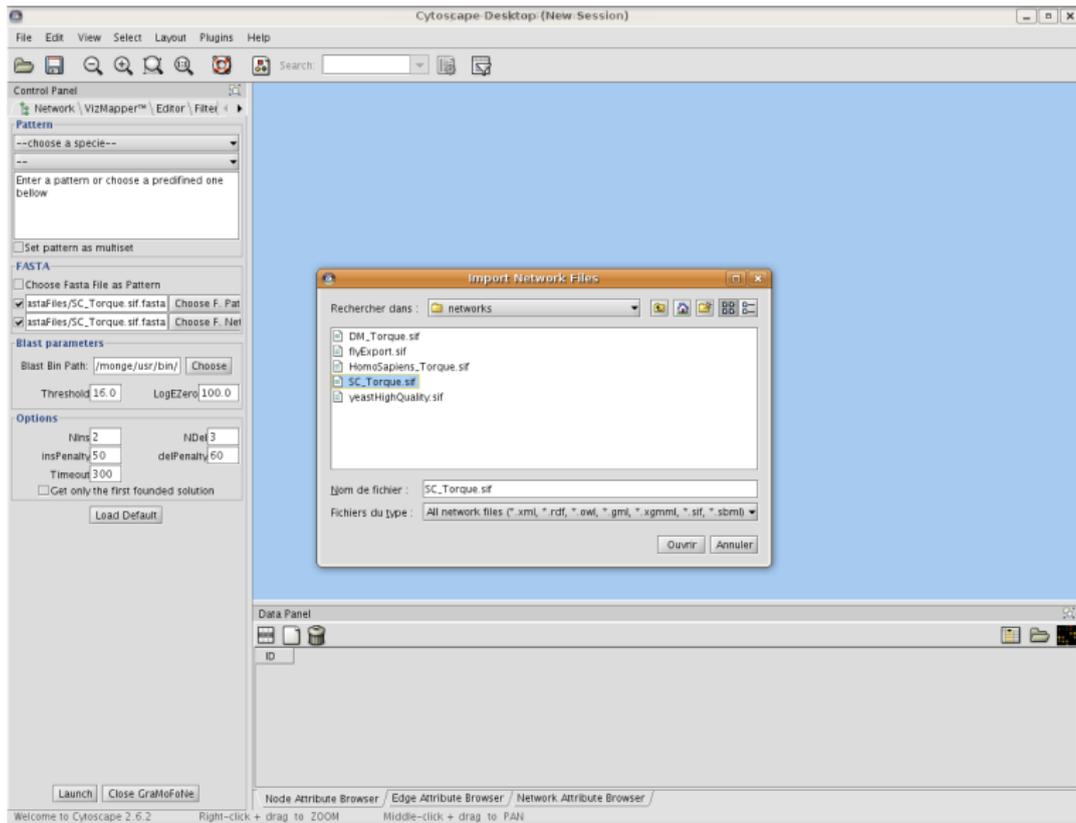
GraMoFoNe

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

GraMoFoNe – Interface



GraMoFoNe – Interface



GraMoFoNe – Interface

File Edit View Select Layout Plugins Help

Search: [] [] []

Control Panel

Network \VizMapper™\Editor \Filter

Patterns

Yeast Complexes SGD.txt

presequence translocase-associated imp...

h-AAA complex 31942

mitotic check-point complex 33597

Rpp3L complex 508

TRAPP complex 30008

vacuolar proton-transporting V-type ATPase

retromer complex, outer shell 30905

Smc5-Smc6 complex 30915

presequence translocase-associated impo...

astaFiles/SC_Torque.sif.fasta Choose F. Pat

astaFiles/SC_Torque.sif.fasta Choose F. Net

Blast parameters

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

Threshold 16.0 LogEZero 100.0

Options

NInq 2 NDel 3

insPenalty 50 delPenalty 60

Timeout 300

Get only the first founded solution

Load Default

Launch Close GraMoFoNe

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

GraMoFoNe – Interface

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...

YHL328C

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: /munge/usr/bin/ Choose

Threshold 16.0 LogEzero 100.0

Options

Nins 1 NDef 0

insPenalty 50 delPenalty 60

Timeout 200

Get only the first founded solution

Load Default

Launch Close GraMoFoNe

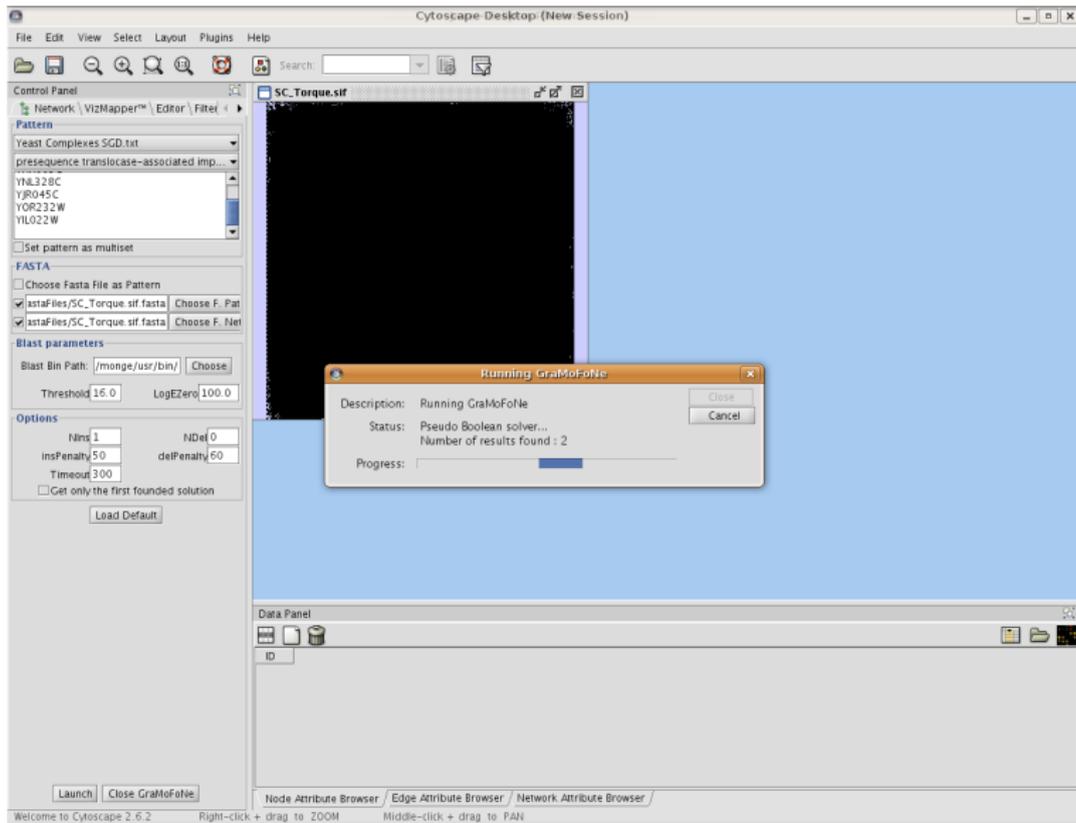
Data Panel

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

GraMoFoNe – Interface



GraMoFoNe – Interface

Control Panel

Network \ VizMapper™ \ Editor \ Filter

Pattern

Yeast Complexes SGD.txt

presequence translocase-associated imp...

YIL328C
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 NDef: 0

insPenalty: 50 delPenalty: 60

Timeout: 200

Get only the first founded solution

Load Default

Launch Close GraMoFoNe

Results Panel

GraMoFoNe results

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

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

GraMoFoNe – Interface

The screenshot displays the GraMoFoNe interface within a window titled "Cytoscape Desktop (New Session)". The interface is divided into several panels:

- Control Panel:** Contains settings for pattern selection (e.g., "Yeast Complexes SGD.txt"), FASTA file handling, blast parameters (Threshold: 16.0, LogEZero: 100.0), and options (Nms: 1, NDe: 0, etc.).
- Central Visualization:** A large black area with several yellow dots, representing a network or motif search results.
- Results Panel:** Displays "GraMoFoNe results" in a table format.

Result	Details
***	Score = 15.0 Rank = 1 Nb Nodes = 0 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
***	Score = 14.0 Rank = 2 Nb Nodes = 0 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
***	Score = 13.0 Rank = 3 Nb Nodes = 0 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
***	Score = 13.0 Rank = 4 Nb Nodes = 0 Nb Del = 0 Nb Ins = 1 (0 C + 1 NC)
***	Score = 13.0 Rank = 5
- Data Panel:** Lists node IDs: YJR045C, YJL104W, YOR232W, YNR017W, YLR008C, YKR065C, YIL022W, YNL328C.

At the bottom, there are buttons for "Launch", "Close GraMoFoNe", and "Discard Result". The footer includes the text: "Welcome to Cytoscape 2.6.2 Right-click + drag to ZOOM Middle-click + drag to PAN".

GraMoFoNe – Interface

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...

YJL328C
YJR045C
YOR232W
YILO22W

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

Nms 1 NDe 0

insPenalty 50 delPenalty 60

Timeout 200

Get only the first founded solution

Load Default

SC_Torque.sif

Results Panel

GraMoFoNe results

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

YJL104W matched with : YJL104W
YLR008C matched with : YLR008C
YNR017W inserted as a not colored node
YKR065C matched with : YKR065C
YJL328C matched with : YJL328C
YJR045C matched with : YJR045C
YOR232W matched with : YOR232W
YILO22W matched with : YILO22W

Discard Result

Data Panel

ID

YJR045C
YJL104W
YOR232W
YNR017W
YLR008C
YKR065C
YILO22W
YJL328C

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 – Interface

The screenshot displays the GraMoFoNe software interface, which is divided into several panels:

- Control Panel:** Contains settings for the pattern file (SC_Torque.sif), FASTA file selection, and Blast parameters (Threshold: 16.0, LogEZero: 100.0).
- Visualization Area:** A central black area showing a network visualization with yellow nodes and edges.
- Results Panel:** Displays the results of the search, including a table of results and a list of matches.
- Data Panel:** Lists the IDs of the nodes found in the network.

Results Panel Table:

Result	Details
Score = 15.0 Rank = 1 Nb Nodes = 8	
Export in a new network Export in a new network with neighbor	
Rank = 2 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 HC)	
Score = 13.0 Rank = 3 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 HC)	
Score = 13.0 Rank = 4 Nb Nodes = 8 Nb Del = 0 Nb Ins = 1 (0 C + 1 HC)	
Score = 13.0 Rank = 5 Nb Nodes = 8	

Matches:

- YJL104W matched with: YJL104W
- YLR008C matched with: YLR008C
- YKR017W inserted as a not colored node
- YKR065C matched with: YKR065C
- YNL328C matched with: YNL328C
- YJR045C matched with: YJR045C
- YOR232W matched with: YOR232W
- YILO22W matched with: YILO22W

Data Panel:

ID
YJR045C
YJL104W
YOR232W
YKR017W
YLR008C
YKR065C
YILO22W
YNL328C

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 – Interface

The screenshot displays the GraMoFoNe software interface, titled "Cytoscape Desktop (New Session)". The interface is divided into several panels:

- Control Panel:** Contains settings for network visualization and analysis. It includes a "Pattern" dropdown set to "Yeast Complexes SGD.txt", a "FASTA" section with checked options for "Choose Fasta File as Pattern", "astaFiles/SC_Torque.sif.fasta", and "astaFiles/SC_Torque.sif.fasta", and "Blast parameters" with a "Threshold" of 16.0 and "LogEZero" of 100.0. There are also "Options" for "Nms" (1), "NDe" (0), "insPenalty" (50), and "delPenalty" (60).
- Main View:** Shows a network graph with nodes and edges. A sub-graph titled "child pure" is highlighted in blue, showing nodes YL022W, YR232W, YL104W, YR008C, YNR017W, YNL32BC, YKRO65C, and YR045C.
- Results Panel:** Displays "GraMoFoNe results" in a table format.

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

 Below the table, it lists matches: YJL104W matched with: YJL104W; YLR008C matched with: YLR008C; YNR017W inserted as a not colored node; YKRO65C matched with: YKRO65C; YNL32BC matched with: YNL32BC; YR045C matched with: YR045C; YOR232W matched with: YOR232W; YL022W matched with: YL022W. A "Discard Result" button is at the bottom.
- Data Panel:** Located at the bottom, it contains a table with columns for "ID" and is currently empty.

At the bottom of the window, there are buttons for "Launch" and "Close GraMoFoNe", and a footer with the text: "Welcome to Cytoscape 2.6.2 Right-click + drag to ZOOM Middle-click + drag to PAN Node Attribute Browser / Edge Attribute Browser / Network Attribute Browser".

Programmation Pseudo-Booléenne

- ▶ Trouver une assignation 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 : $x_1 = 1, x_2 = 0, x_3 = 0$