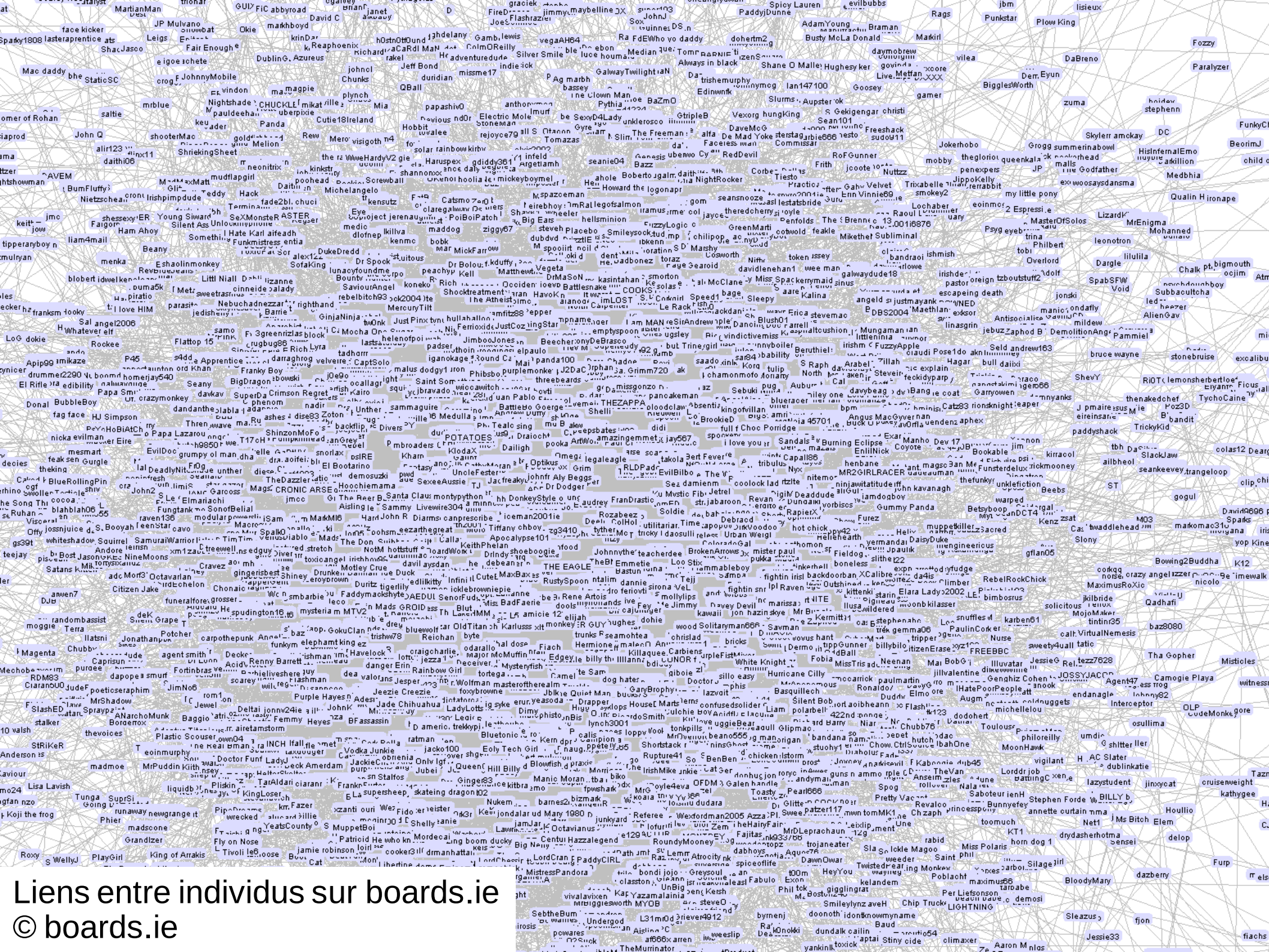
Détection et stabilité : vers des communautés dynamiques

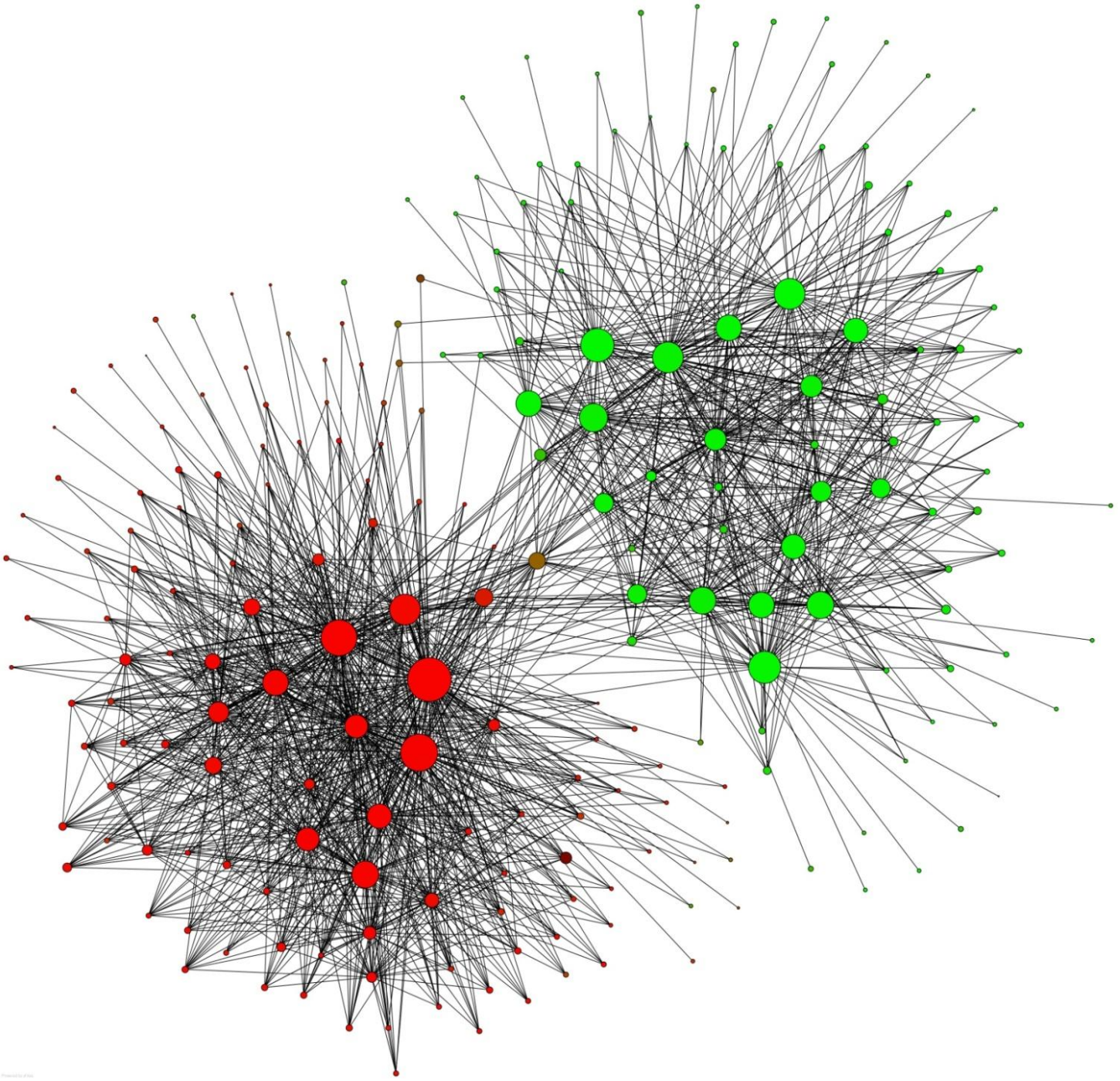
Détection : V. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre

Stabilité : T. Aynaud, J.-L. Guillaume

Jean-loup.guillaume@lip6.fr



Liens entre individus sur boards.ie
© boards.ie



Contexte

Grands graphes/réseaux apparaissant en pratique :

- Sociologie : réseaux de contacts, appels téléphoniques.
- Informatique : Web, Internet, réseaux d'échange pair-à-pair.
- …

On cherche à comprendre :

- La structure de ces graphes.
- Leur évolution.
- Les phénomènes agissant sur ces réseaux.

Quelques applications

Informatique :

- Réseaux : routage, protocoles, sécurité, détection d'anomalies.
- P2P : conception de systèmes, recherche de déviances.
- Web : indexation, moteurs de recherche.

Sociologie :

- Diffusion d'innovations, rumeurs.
- Identification (automatique) de groupes d'individus et de leaders.

Epidémiologie :

- Diffusion de virus, vaccination.

Qu'est-ce qu'une communauté ?

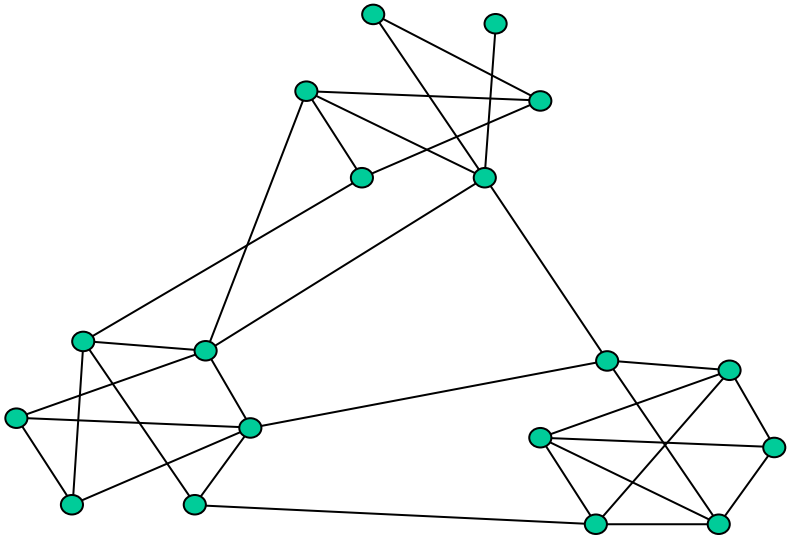
Dans un réseau : communauté = ensemble de sommets qui partagent quelque chose :

- Personnes ayant des intérêts similaires (famille, amis).
- Pages web avec un contenu similaire.
- Blogs sur un même sujet, etc.

Qu'est-ce qu'une communauté ?

On cherche une définition liée à la topologie du réseau :

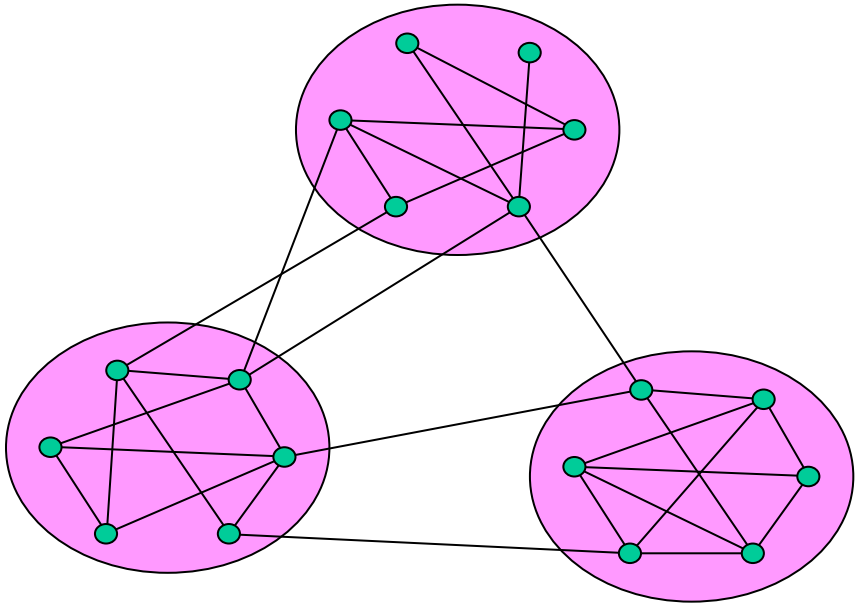
- Groupes de sommets densément connectés.
- Peu de liens entre les groupes.



Qu'est-ce qu'une communauté ?

On cherche une définition liée à la topologie du réseau :

- Groupes de sommets densément connectés.
- Peu de liens entre les groupes.



Applications

Détection (automatique) de communautés :

- Comprendre la structure des réseaux.
- Détecter des communautés d'intérêt :
  - Pages web, fichiers similaires sur un réseau pair-à-pair, …
- Visualisation.
- Amélioration des moteurs de recherche, des techniques de routage, des réseaux pair-à-pair, …

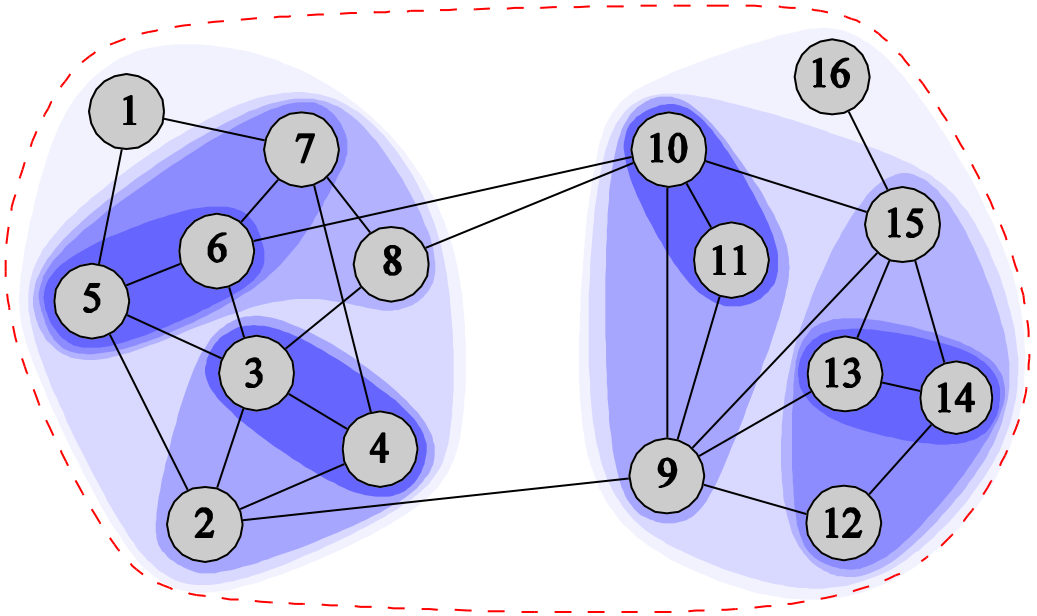
Challenges

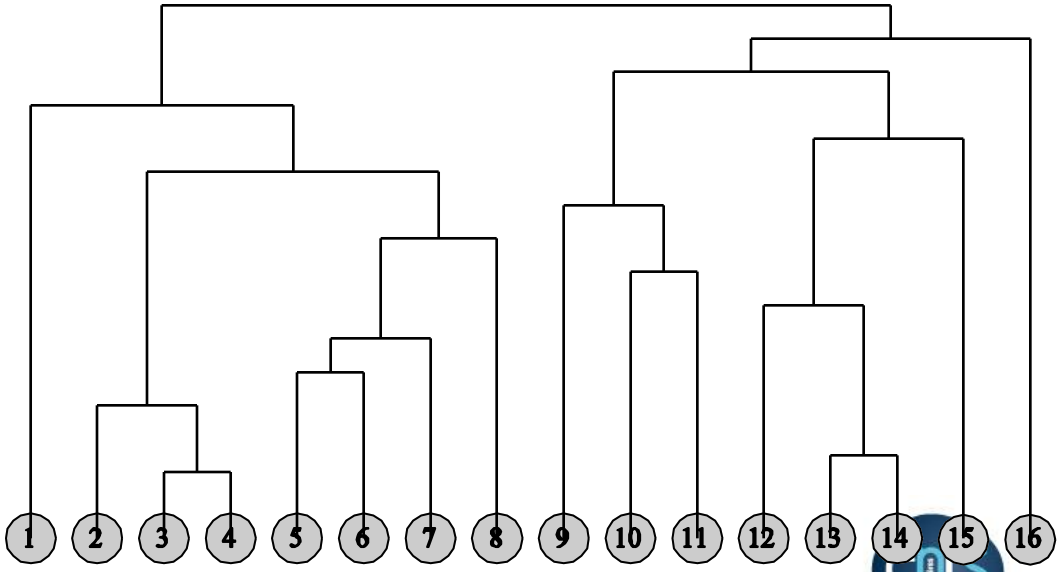
Détection (automatique) de communautés :

- Nombre inconnu de communautés.
- Taille des communautés inconnue.

- Passage à l'échelle :
  - Web ~ milliards de sommets ;
- Hiérarchie de communautés.
- Communautés recouvrantes :
  - Un sommet peut-il appartenir à plusieurs communautés (ou pas) ?
- Communautés dynamiques :
  - Comment évoluent les communautés quand le graphe évolue ?

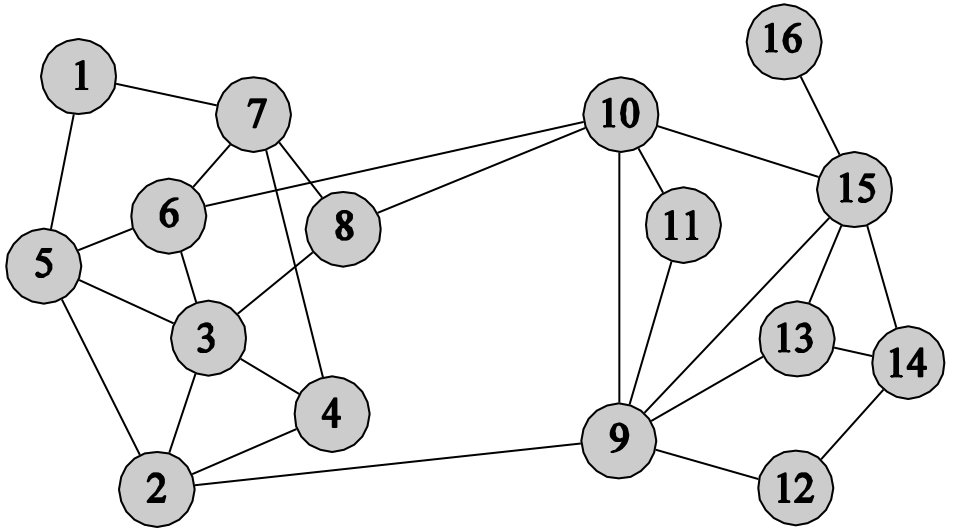
Clustering hiérarchique

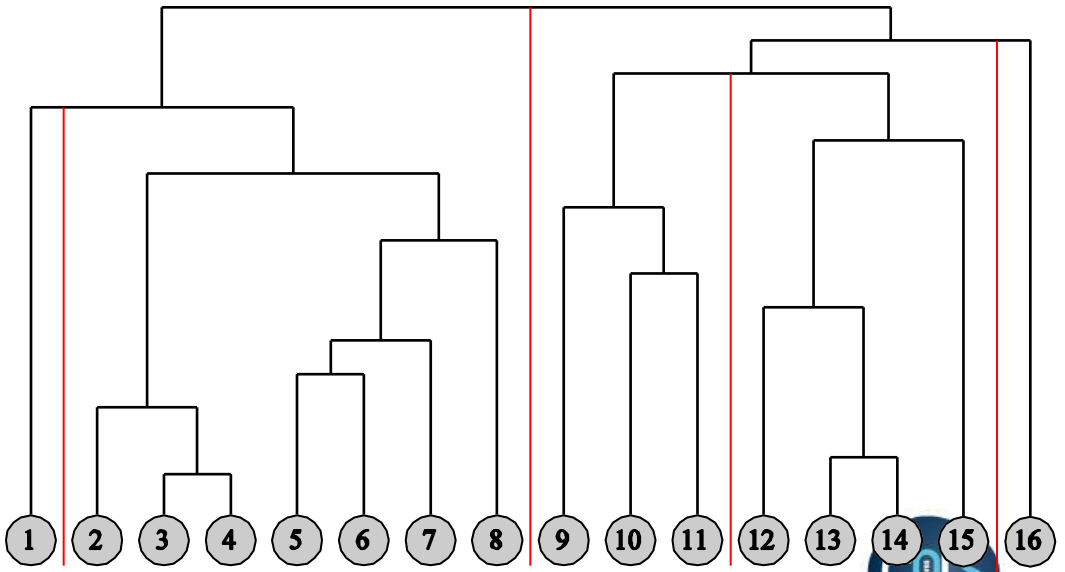




Adapted from slide courtesy of Pascal Pons

Approches divisives





Adapted from slide courtesy of Pascal Pons

Plan de l'exposé

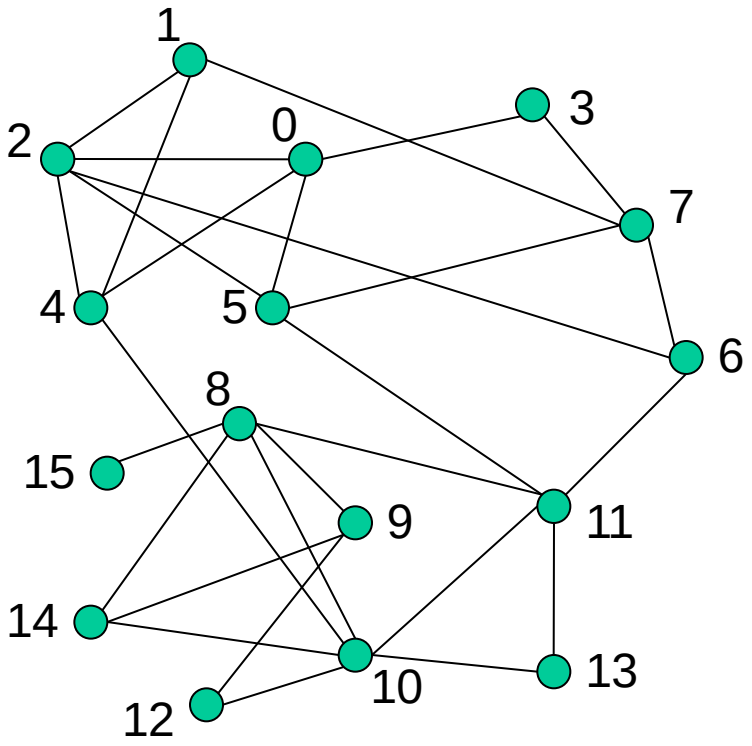
Détection de communautés statiques :

- Algorithme de Louvain.
- Résultats expérimentaux.

Communautés dynamiques :

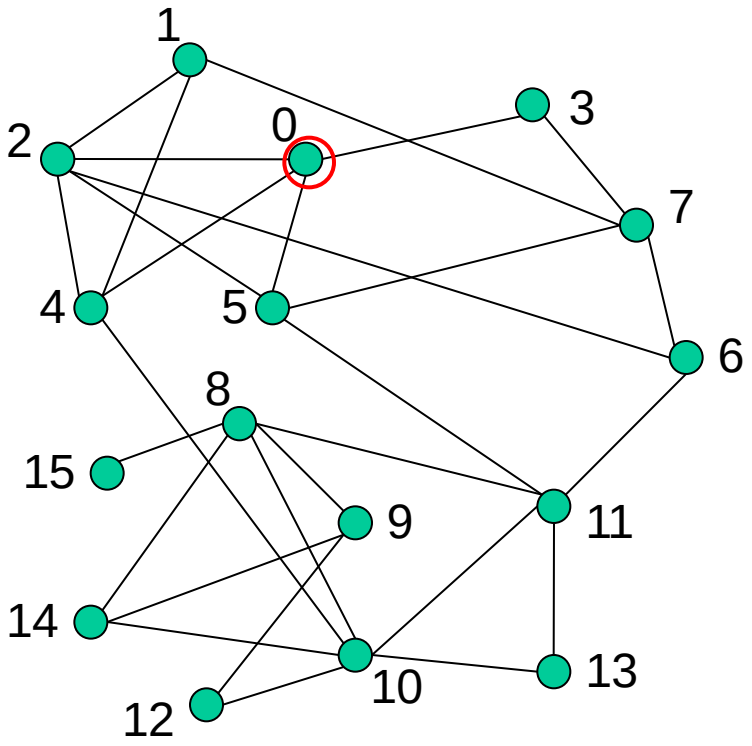
- Quelques études passées.
- Problèmes de stabilité, vers l'étude de la dynamique.

Un exemple

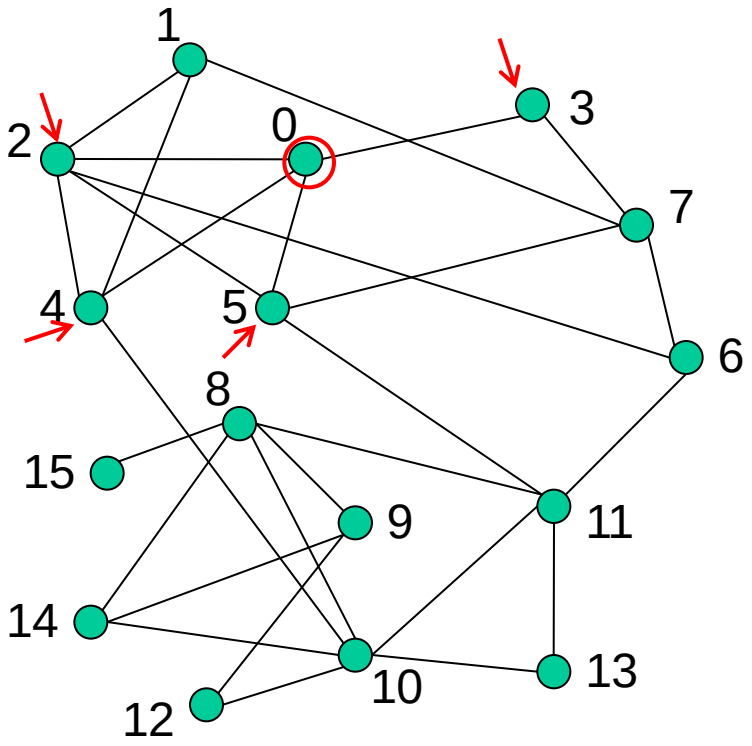


Initialement : chaque sommet est isolé

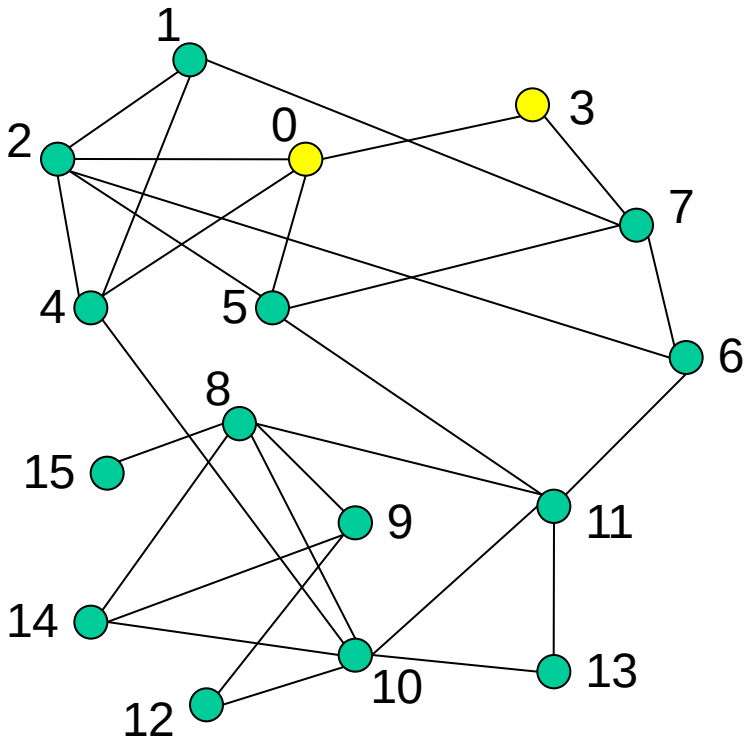
Un exemple



Un exemple

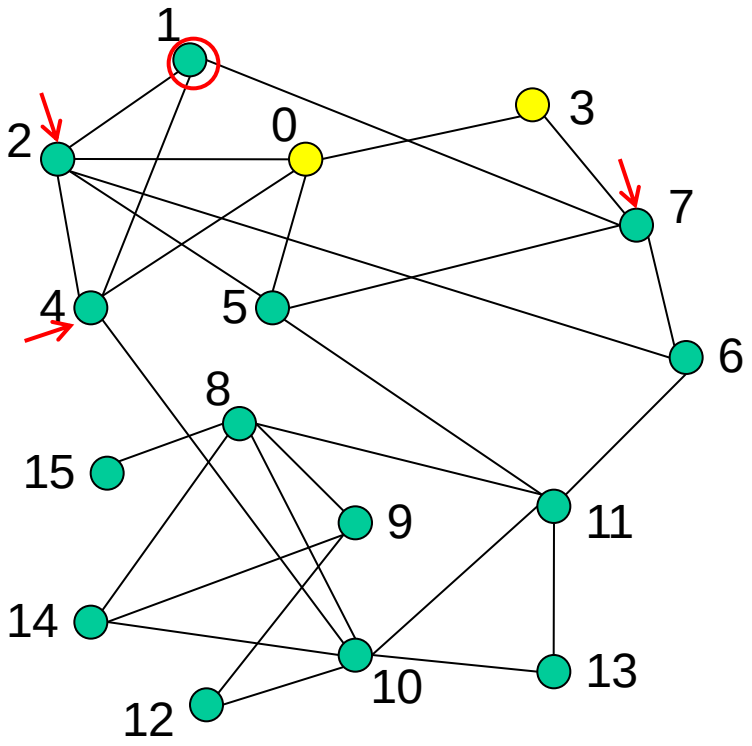


Un exemple



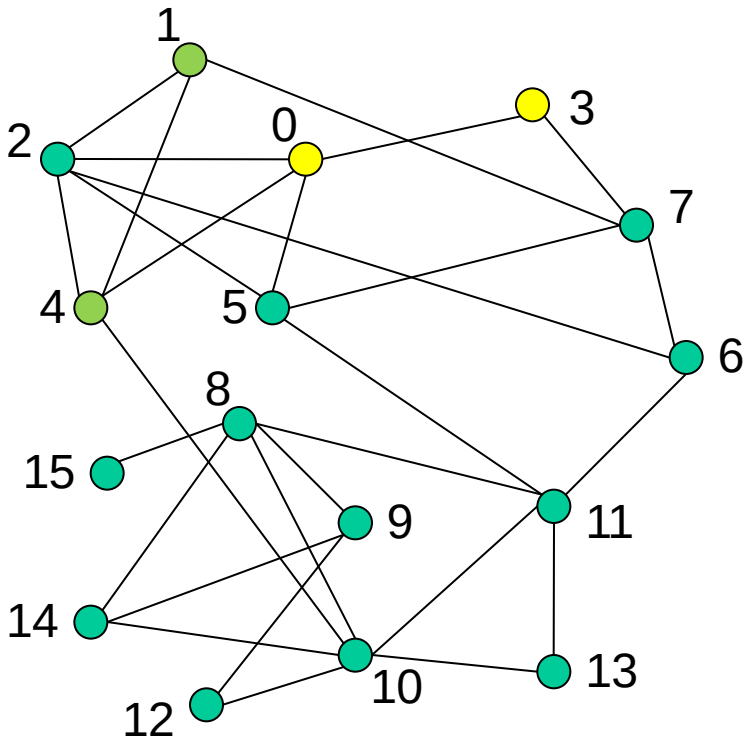
Passe 1 – Itération 1
insérer 0 dans c[3]

Un exemple



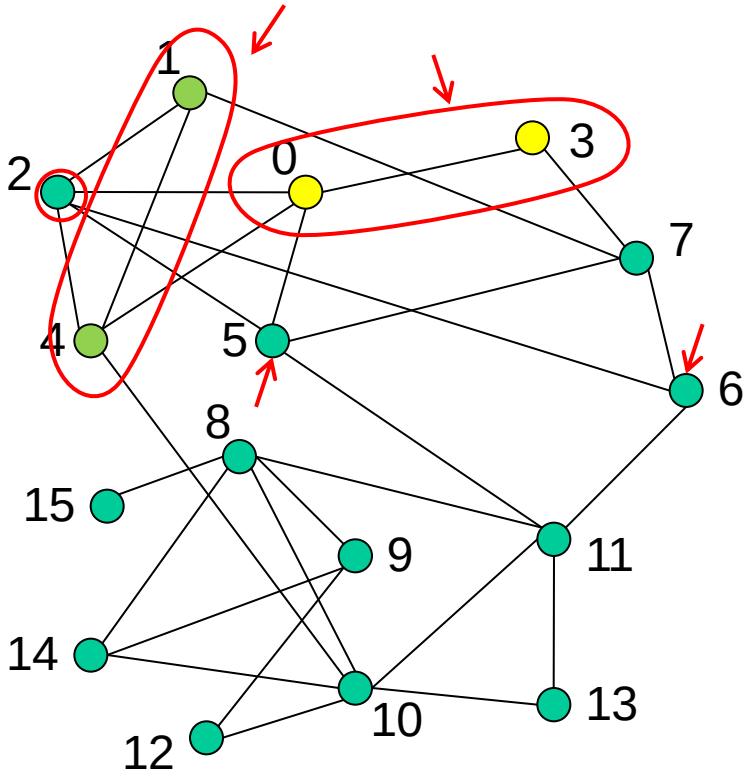
Passe 1 – Itération 1
insérer 0 dans c[3]

Un exemple



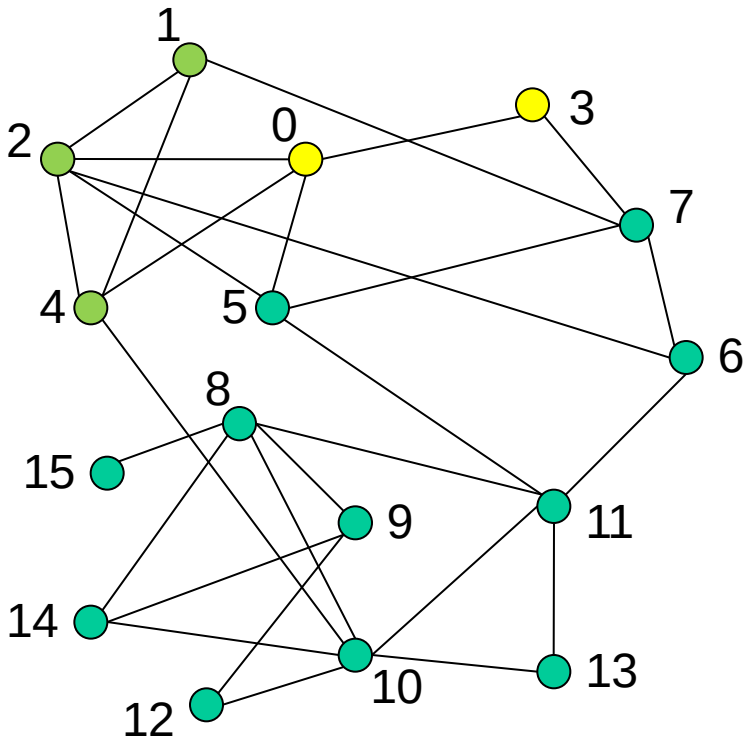
Passe 1 – Itération 1
insérer 0 dans c[3]
insérer 1 dans c[4]

Un exemple



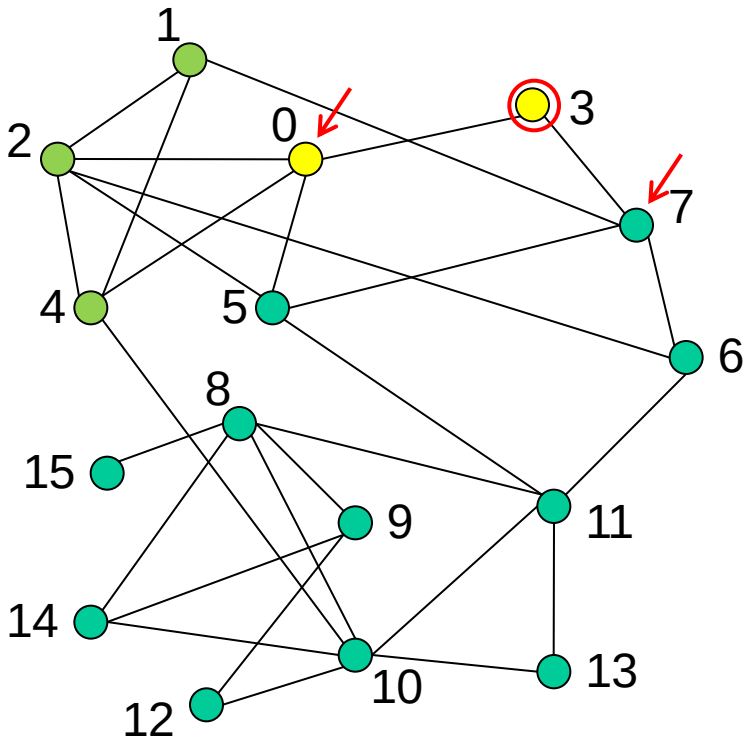
Passe 1 – Itération 1
insérer 0 dans c[3]
insérer 1 dans c[4]

Un exemple



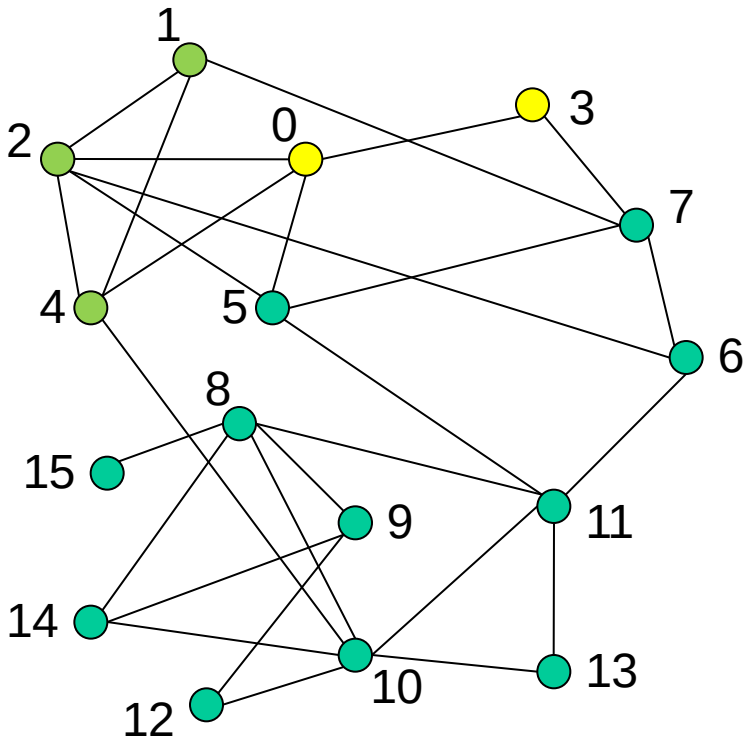
Passe 1 – Itération 1
insérer 0 dans c[3]
insérer 1 dans c[4]
insérer 2 dans c[1,4]

Un exemple



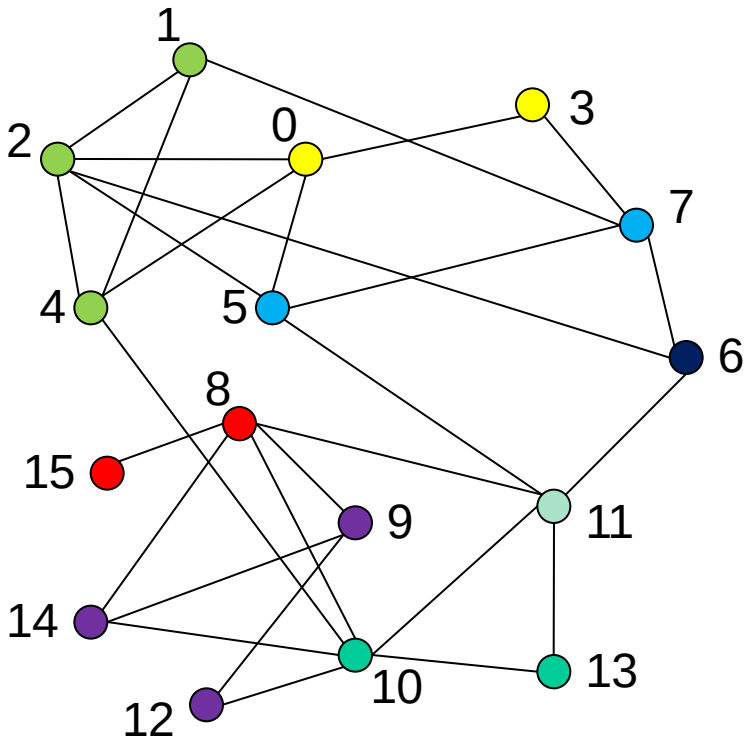
Passe 1 – Itération 1
insérer 0 dans c[3]
insérer 1 dans c[4]
insérer 2 dans c[1,4]

Un exemple



Passe 1 – Itération 1
insérer 0 dans c[3]
insérer 1 dans c[4]
insérer 2 dans c[1,4]
insérer 3 dans c[0]

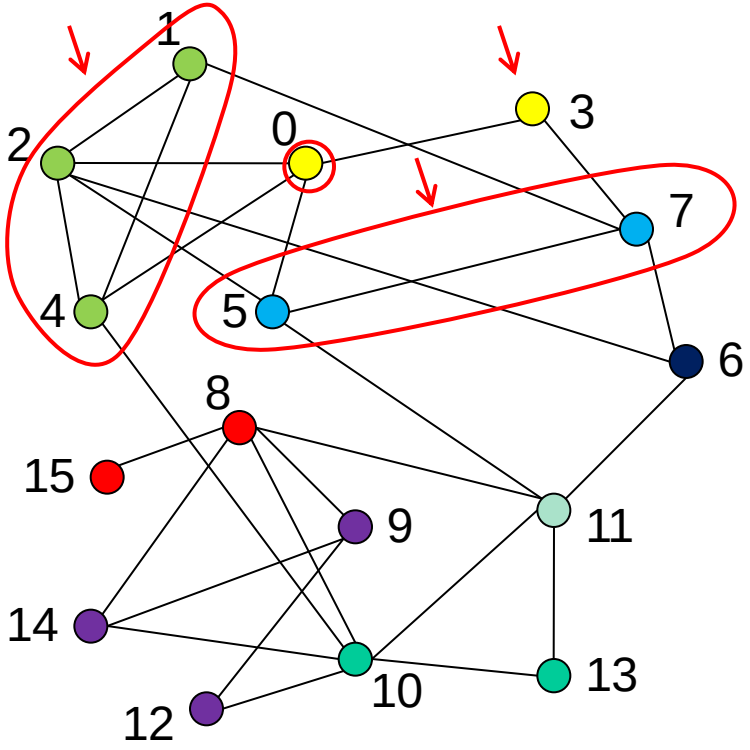
Un exemple



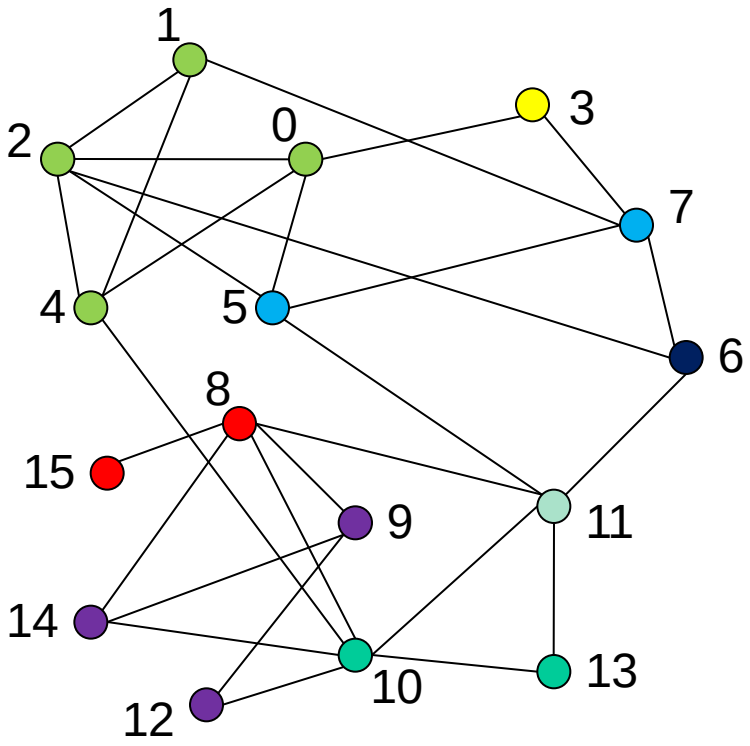
Passe 1 – Itération 1
insérer 0 dans c[3]
insérer 1 dans c[4]
insérer 2 dans c[1,4]
insérer 3 dans c[0]
insérer 4 dans c[1]
insérer 5 dans c[7]
insérer 6 dans c[11]
insérer 7 dans c[5]
insérer 8 dans c[15]
insérer 9 dans c[12]
insérer 10 dans c[13]
insérer 11 dans c[10,13]
insérer 12 dans c[9]
insérer 13 dans c[10,11]
insérer 14 dans c[9,12]
insérer 15 dans c[8]

Un exemple

Passe 1 – Itération 2

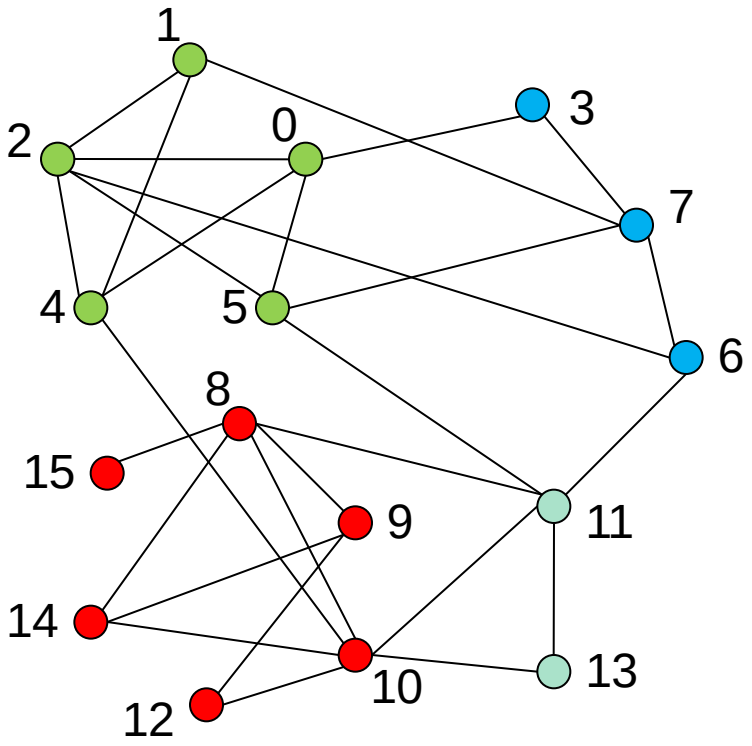


Un exemple



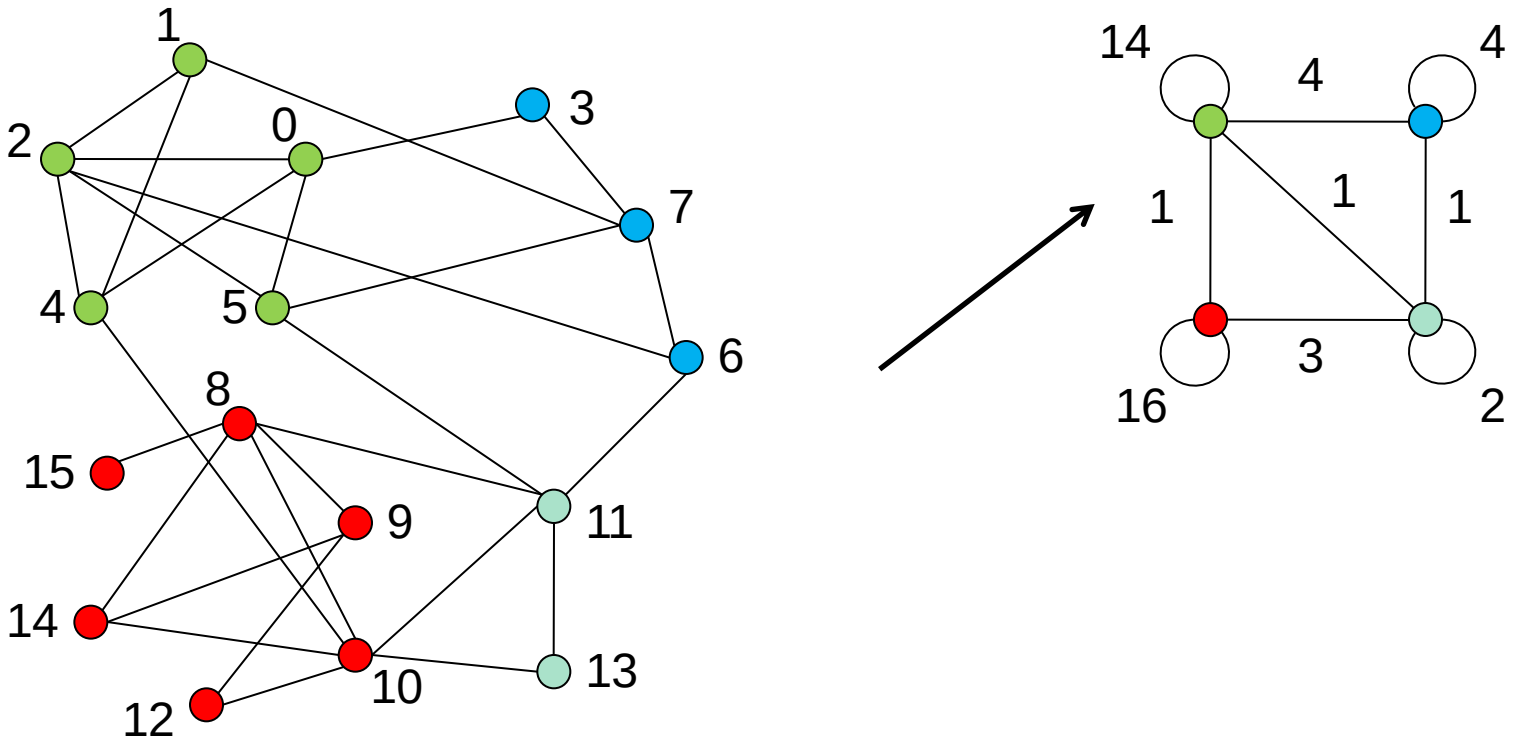
Passe 1 – Itération 2
insérer 0 dans c[4]
…

Un exemple

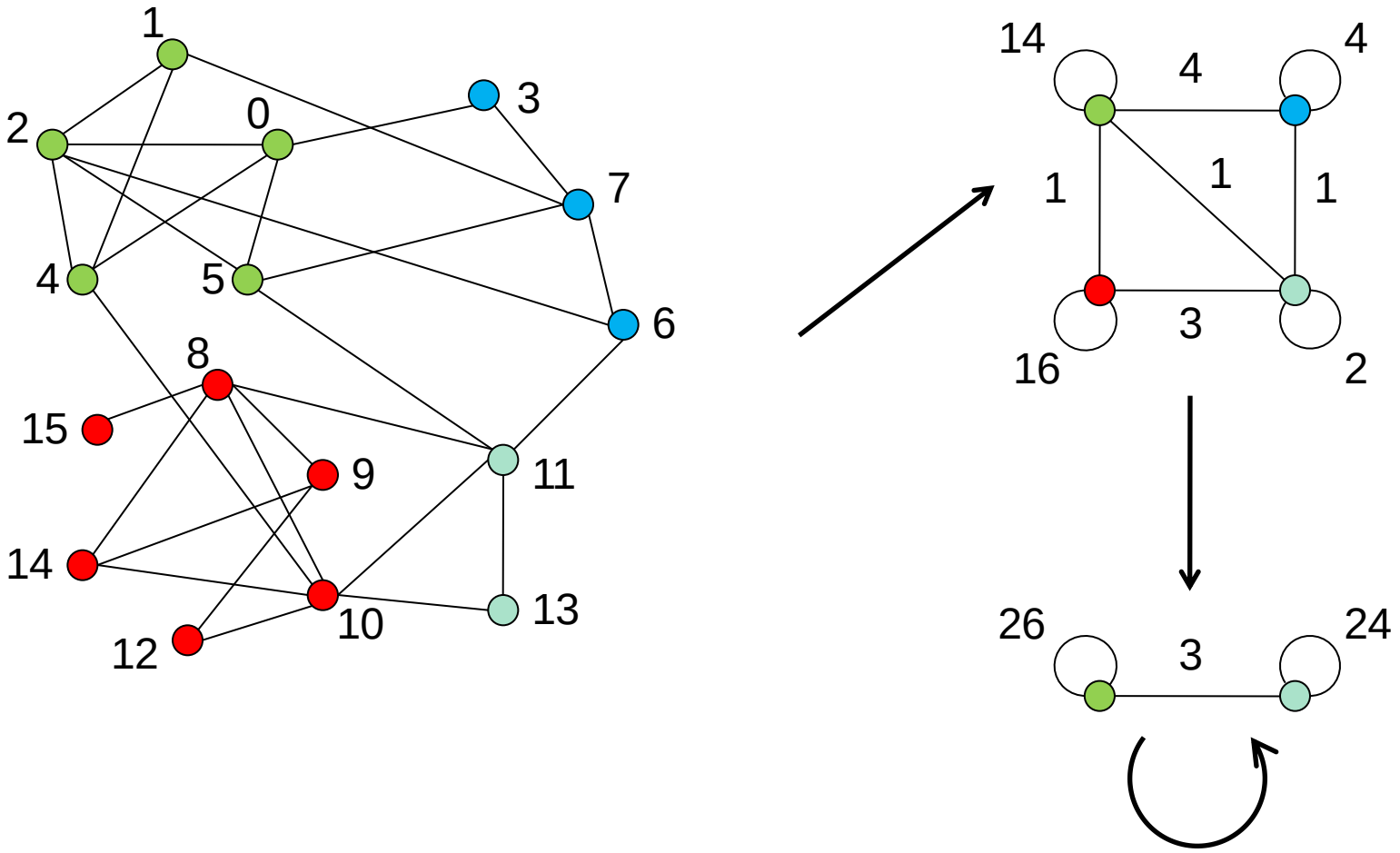


Après 4 itérations, plus aucun changement

Un exemple



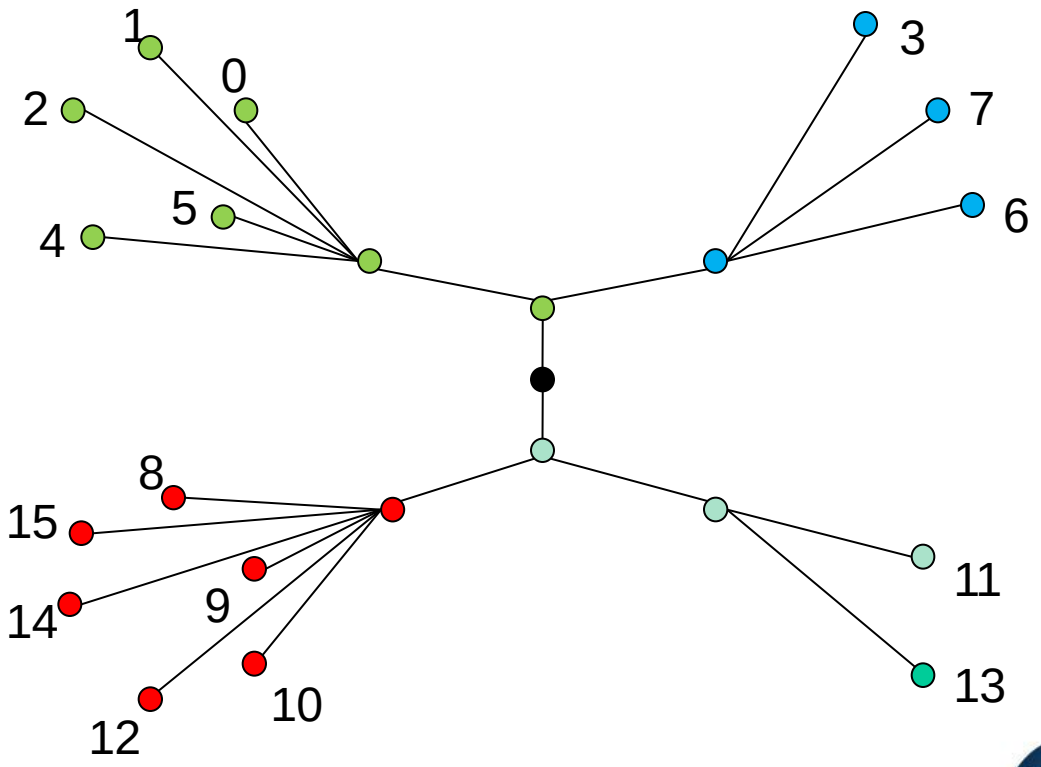
Un exemple



Un exemple

Donne une arborescence (non binaire) :

Chaque niveau est potentiellement intéressant.



Modularité

La mesure de qualité la plus communément utilisée :

$$Q = \frac{1}{2m} \sum_C \left[e_C - \frac{a_C^2}{2m} \right]$$

Liens dans C

Liens avec une extrémité dans C = espérance du nombre de liens dans la communauté (sur un graphe aléatoire)

Contribution d'un sommet isolé :

$$Q(i) = -\left(\frac{k_i}{2m} \right)^2$$

Degré de i

Déplacement d'un sommet

Un sommet isolé 'i' peut être déplacé dans 'C' avec un gain :

$$\Delta Q(C,i) = \left[\frac{e_C + k_{i,C}}{2m} - \left(\frac{a_C + k_i}{2m} \right)^2 \right] - \left[\frac{e_C}{2m} - \left(\frac{a_C}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

Liens de i vers C

Ne dépend que de i et C

Complexité linéaire en le degré de i

Résultats expérimentaux (temps)

	Karate	Arxiv	Internet	Web nd.edu	Téléphone	Web UK-2005	Web Webbase01
	n=34/m=77	9k/24k	70k/351k	325k/1M	2.5M/6.3M	39M / 783M	118M/1B
Newman Girvan Clauset Moore	0s	3.6s	799s	5034s			
Pons Latapy	0s	3.3s	575s	6666s			
Wakita Tsurumi (expected)	0s	0s	8s	52s	1279s	(3days)	
Notre approche	**0s**	**0s**	**<1s**	**<1s**	**47s**	**252s**	**469s**
	3 passes	5 passes	5 passes	5 passes	5 passes	4 passes	5 passes

Résultats expérimentaux(Q)

	Karate	Arxiv	Internet	Web nd.edu	Téléphone	Web UK-2005	Web Webbase01
	n=34/m=77	9k/24k	70k/351k	325k/1M	2.5M/6.3M	39M / 783M	118M/1B
Newman Girvan Clauset Moore	0s 0.38	3.6s 0.772	799s 0.692	5034s 0.927			
Pons Latapy	0s 0.42	3.3s 0.757	575s 0.729	6666s 0.895			
Wakita Tsurumi (expected)	0s	0s	8s	52s	1279s	(3days)	
Notre approche	0s **0.42**	0s **0.813**	<1s **0.781**	<1s **0.935**	47s **0.769**	252s **0.979**	469s **0.984**
	3 passes	5 passes	5 passes	5 passes	5 passes	4 passes	5 passes

Plan de l'exposé

Détection de communautés statiques :

- Algorithme de Louvain.
- Résultats expérimentaux.

Communautés dynamiques :

- Quelques études passées.
- Problèmes de stabilité, vers l'étude de la dynamique.

Communautés dynamiques

Louvain (et autres) fonctionne bien pour les réseaux statiques mais la plupart des réseaux évoluent :

- De nouvelles pages/sites sont créés sur le Web.
- On rencontre de nouvelles personnes.
- Des posts sont créés sur des blogs, …

Communautés dynamiques

Principe de base :

Graphe dynamique = succession de réseaux statiques.

On dispose de techniques efficaces pour les réseaux statiques.

-> calculer les communautés à chaque instant.

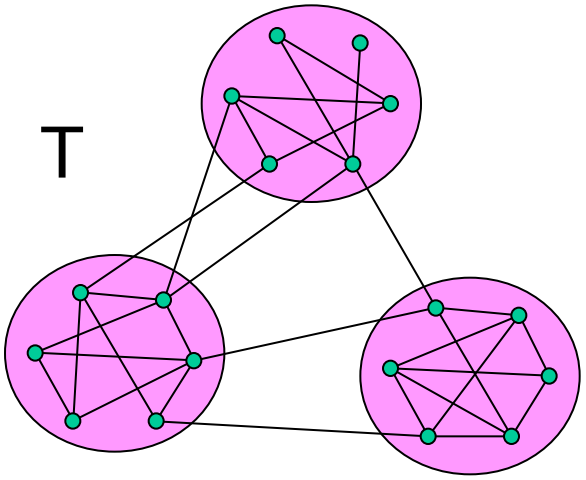
Communautés dynamiques

Principe de base :

Graphe dynamique = succession de réseaux statiques.

On dispose de techniques efficaces pour les réseaux statiques.

-> calculer les communautés à chaque instant.



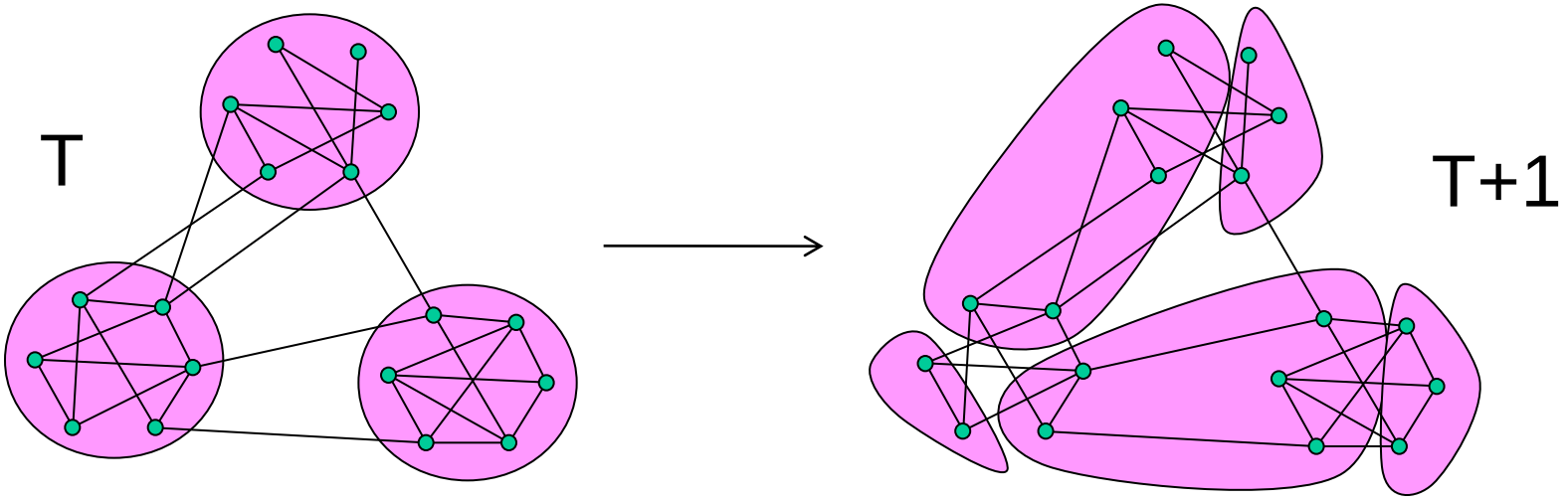
Communautés dynamiques

Principe de base :

Graphe dynamique = succession de réseaux statiques.

On dispose de techniques efficaces pour les réseaux statiques.

-> calculer les communautés à chaque instant.



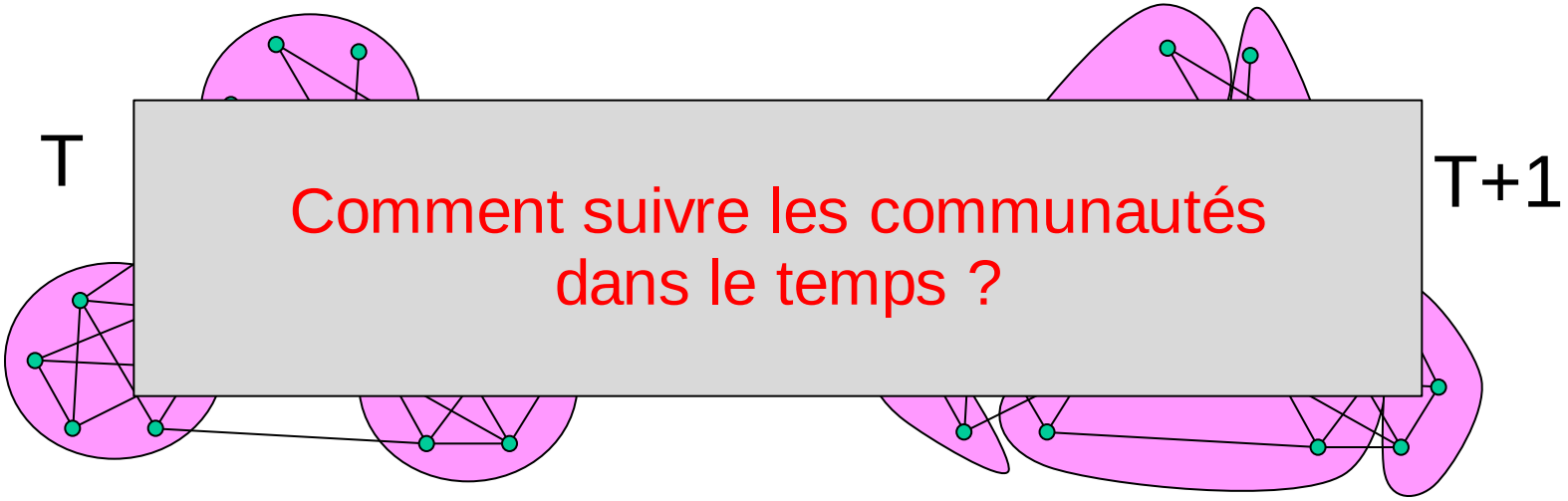
Communautés dynamiques

Principe de base :

Graphe dynamique = succession de réseaux statiques.

On dispose de techniques efficaces pour les réseaux statiques.

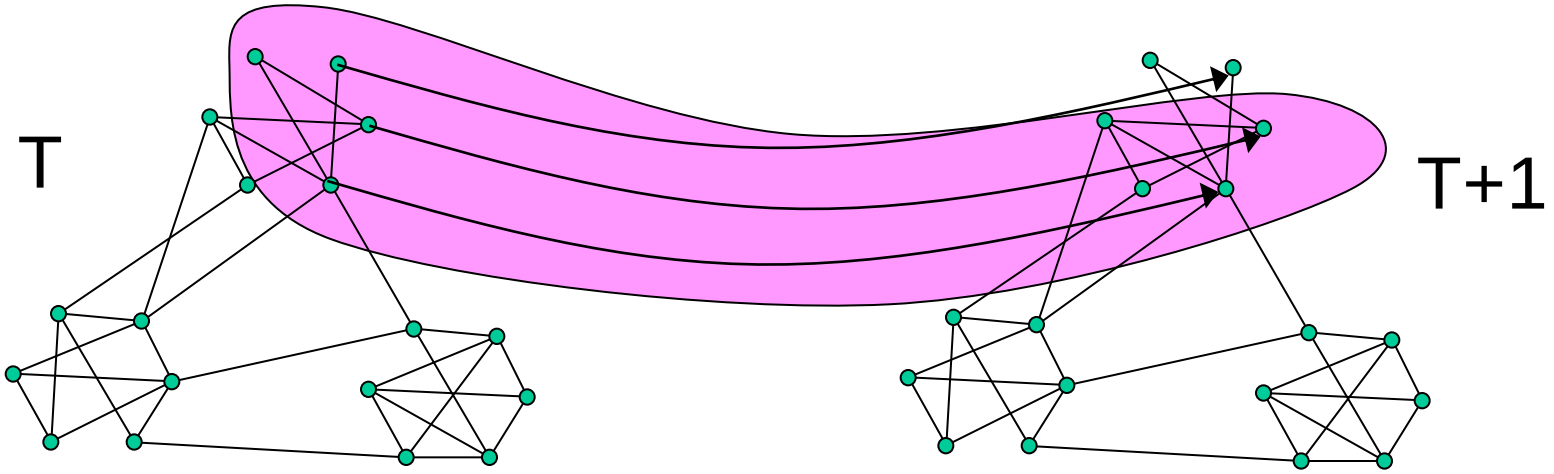
-> calculer les communautés à chaque instant.



Quelques approches

Construire un réseau temporel :

- Ajouter des liens temporel entre les graphes à chaque instant.
- Utiliser n'importe quel algorithme pour les graphes statiques.
- Donne des communautés temporelles.

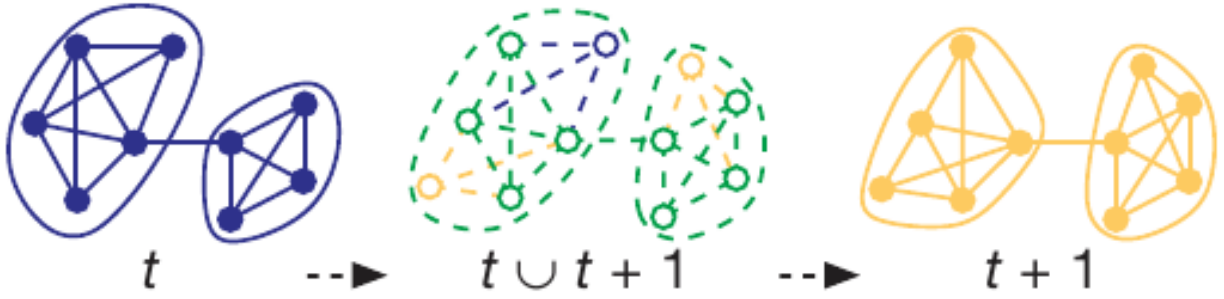


Quelques approches

Construire un réseau temporel

Modifier la définition de communauté :

- Par exemple des cliques recouvrantes.
- Permettent de suivre l'évolution de manière plus simple.



Quelques approches

Construire un réseau temporel

Modifier la définition de communauté

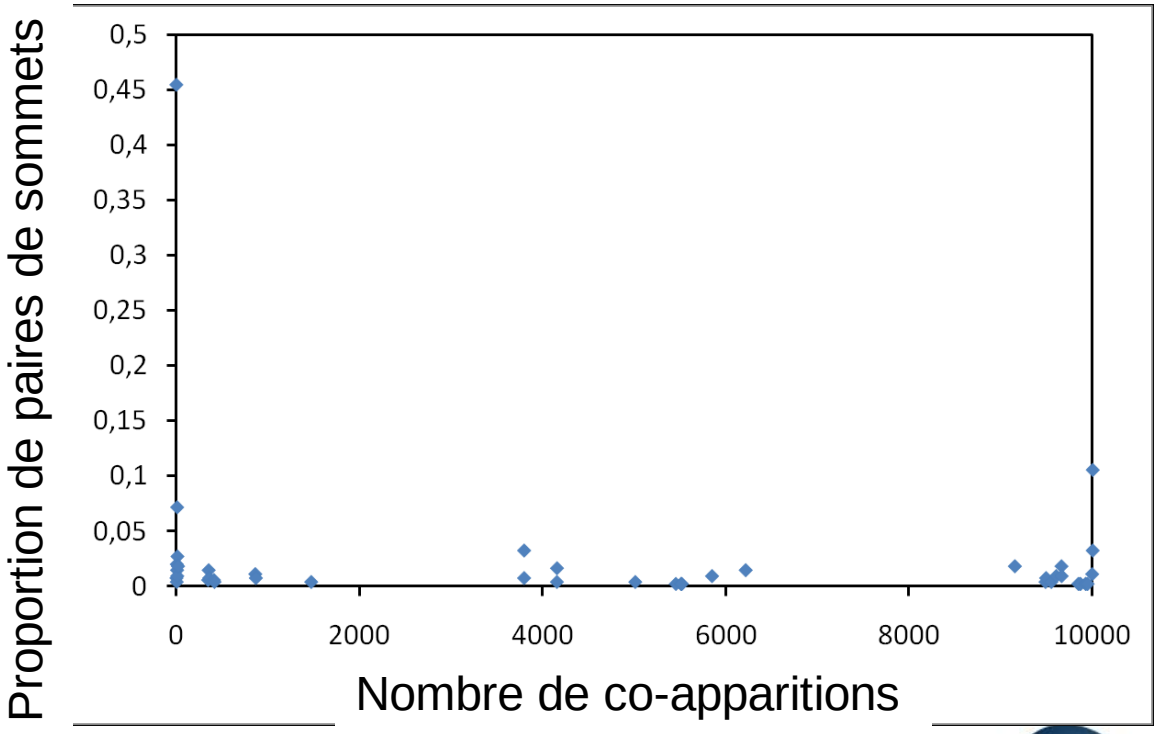
Ne s'intéresser qu'à certaines communautés :

- Recherche de communautés très stables (peu ou pas modifiées si le graphe est légèrement modifié).
- Ces communautés devraient aussi être peu modifiées lors de changements plus importants.

Notre approche

Les algorithmes classiques sont très peu stables

- 10000 calculs sur le même graphe.
- On observe chaque paire de sommets.



Notre approche

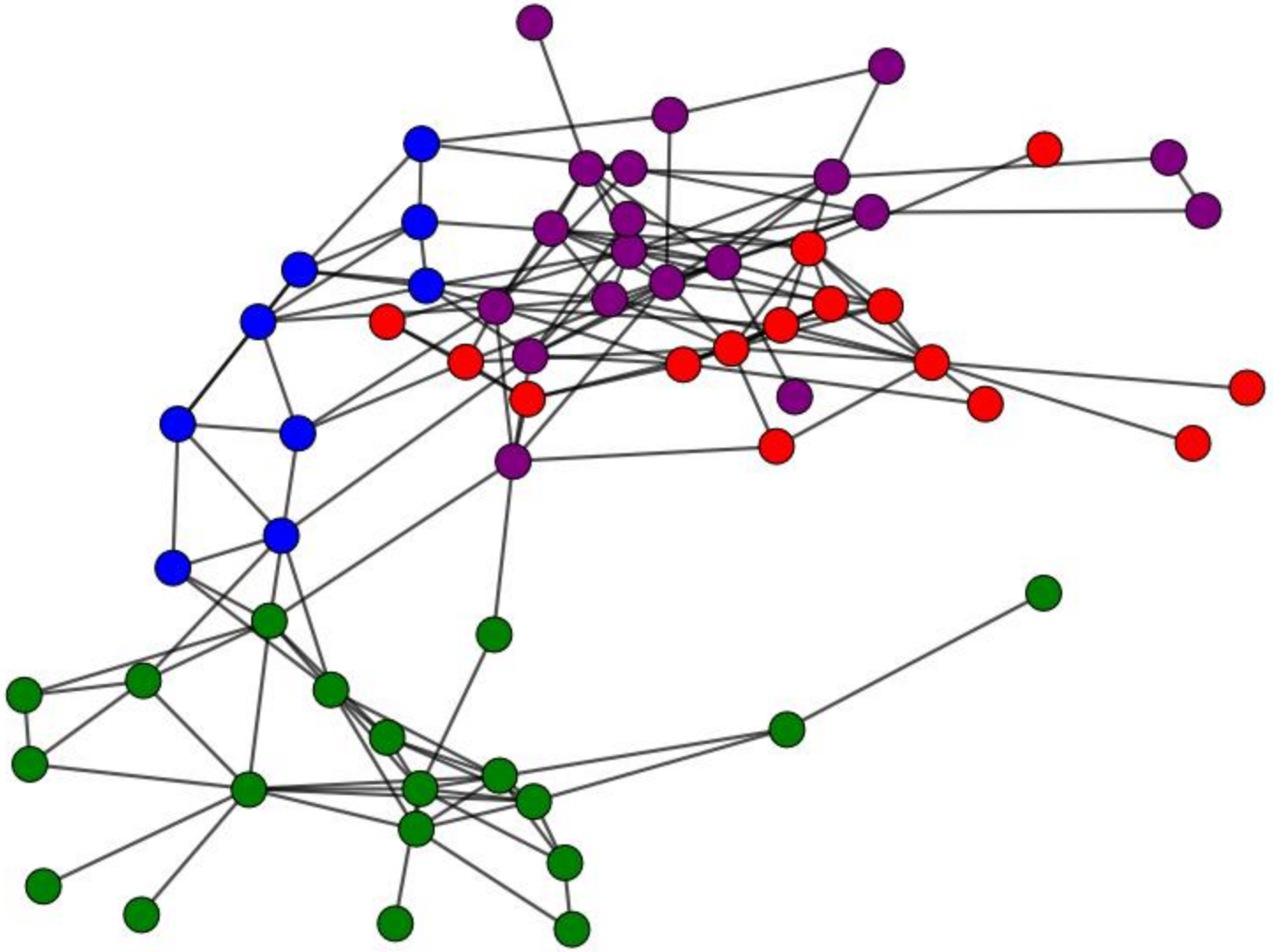
Les algorithmes classiques sont très peu stables.

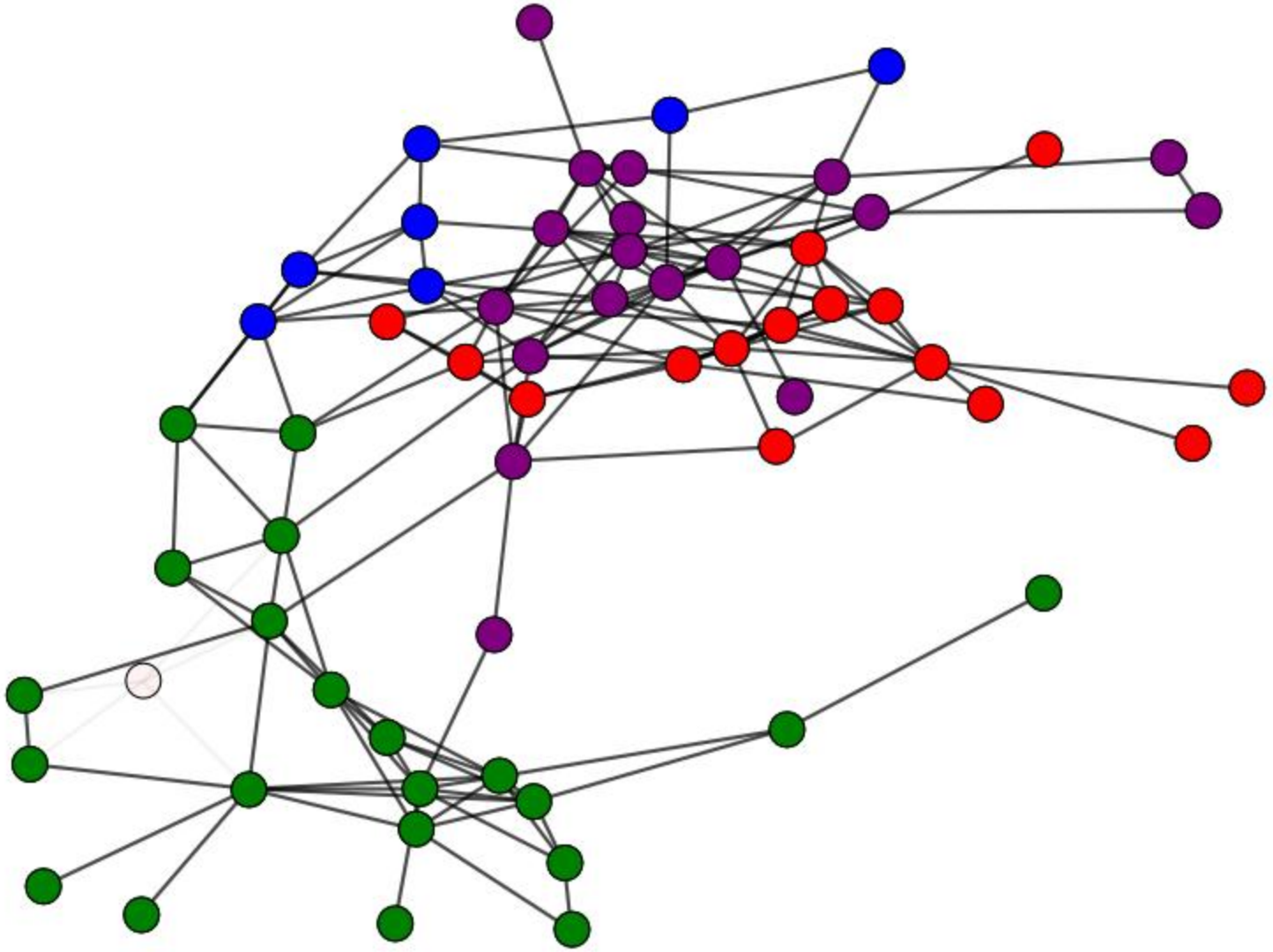
Utilisation d'algorithmes classiques de détection :

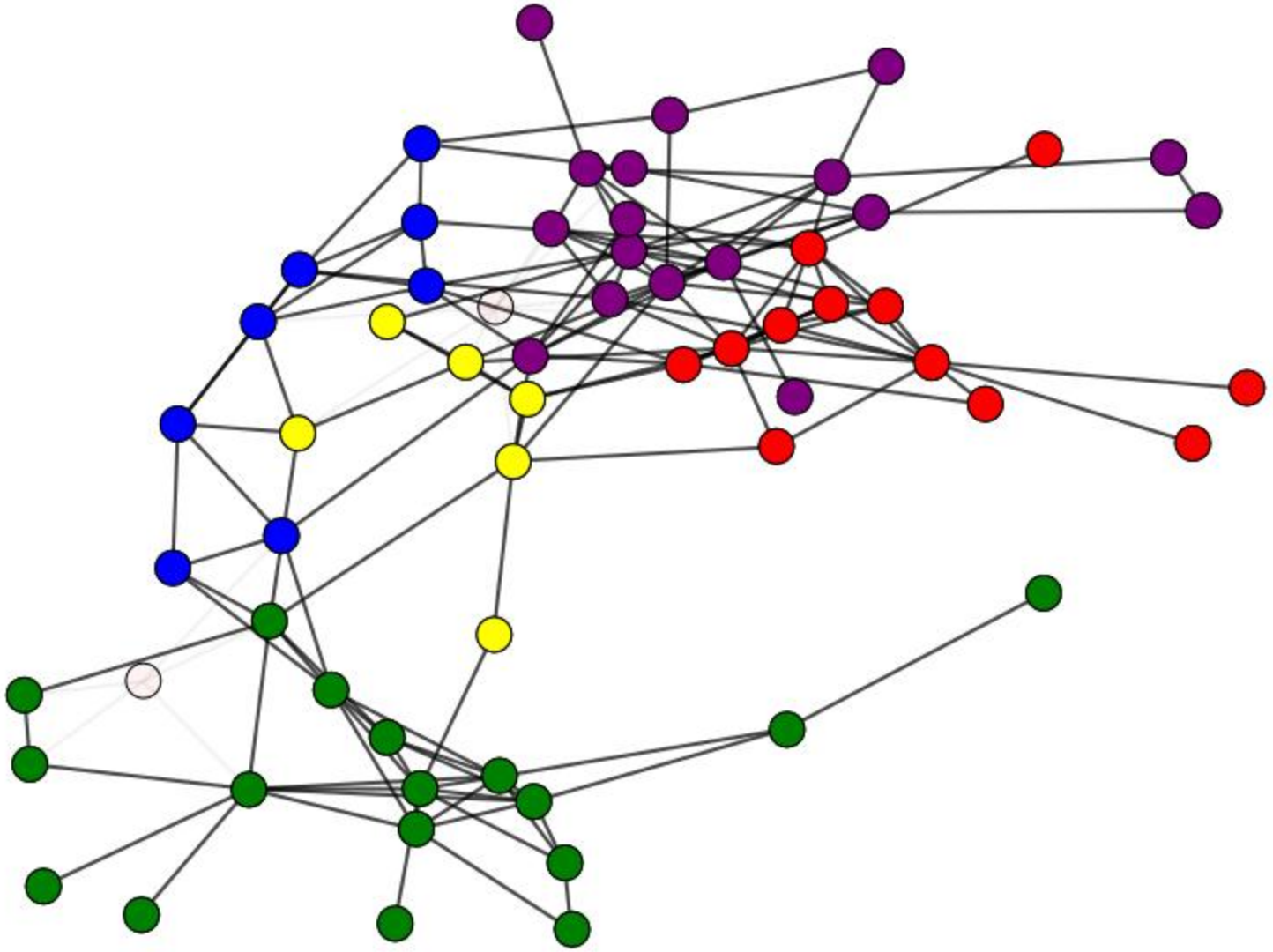
- Algorithme de Newman : optimisation gloutonne de la modularité.
- Walktrap : optimisation gloutonne utilisant des marches aléatoires.
- Algorithme de Louvain.

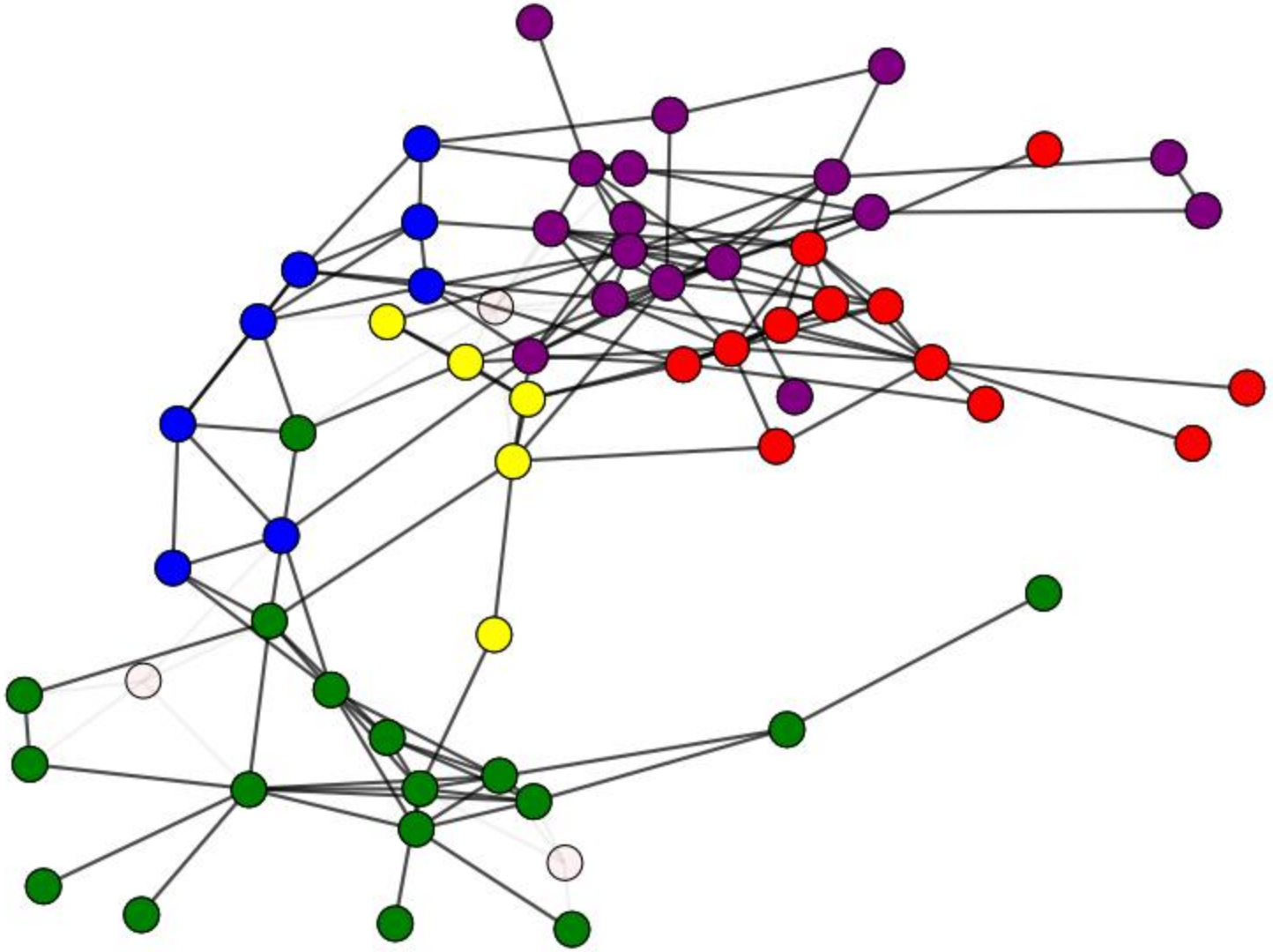
Simulation d'une évolution simple :

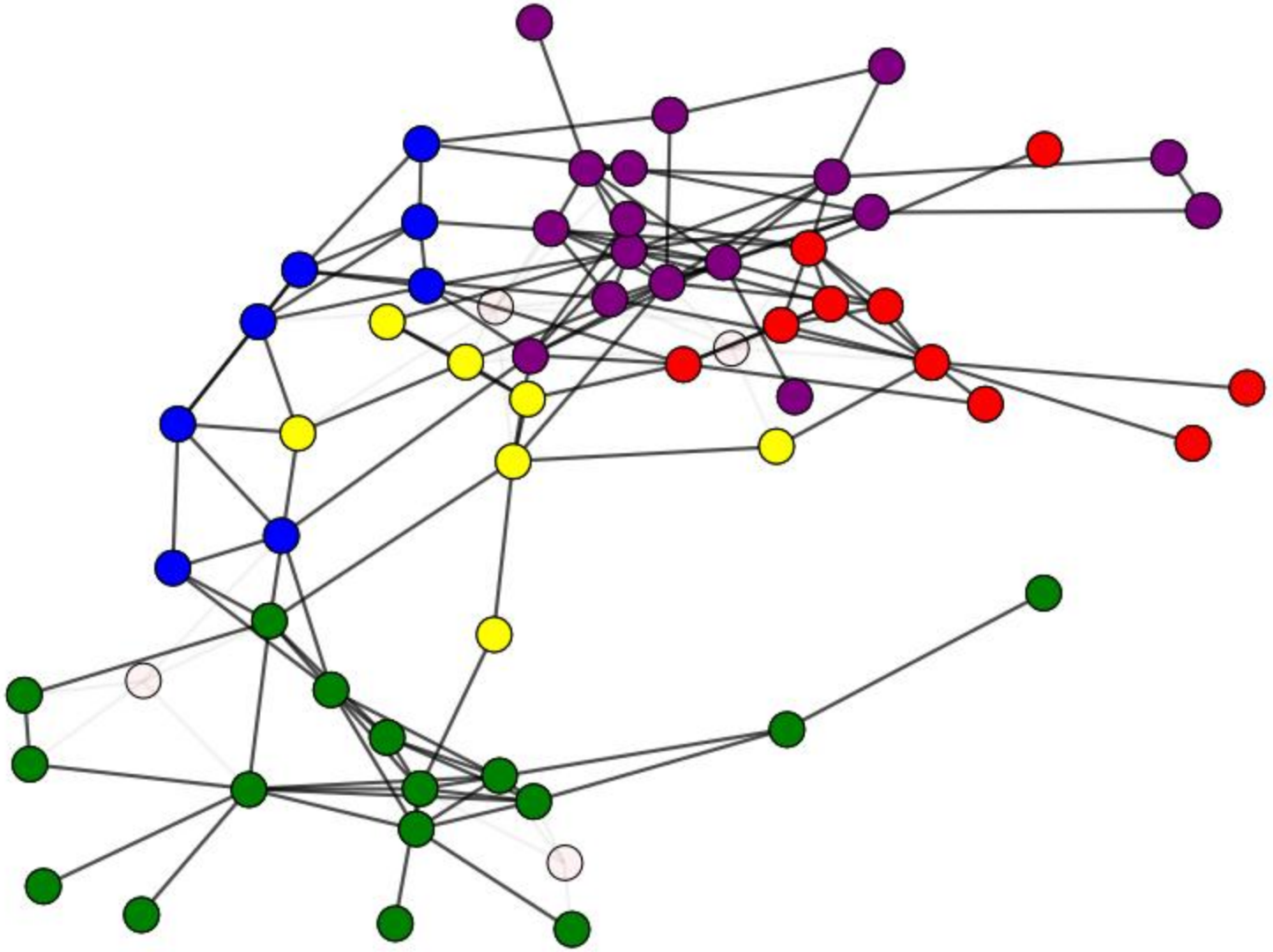
- Suppression de sommets un par un de manière aléatoire.
- Calcul des communautés après chaque suppression.

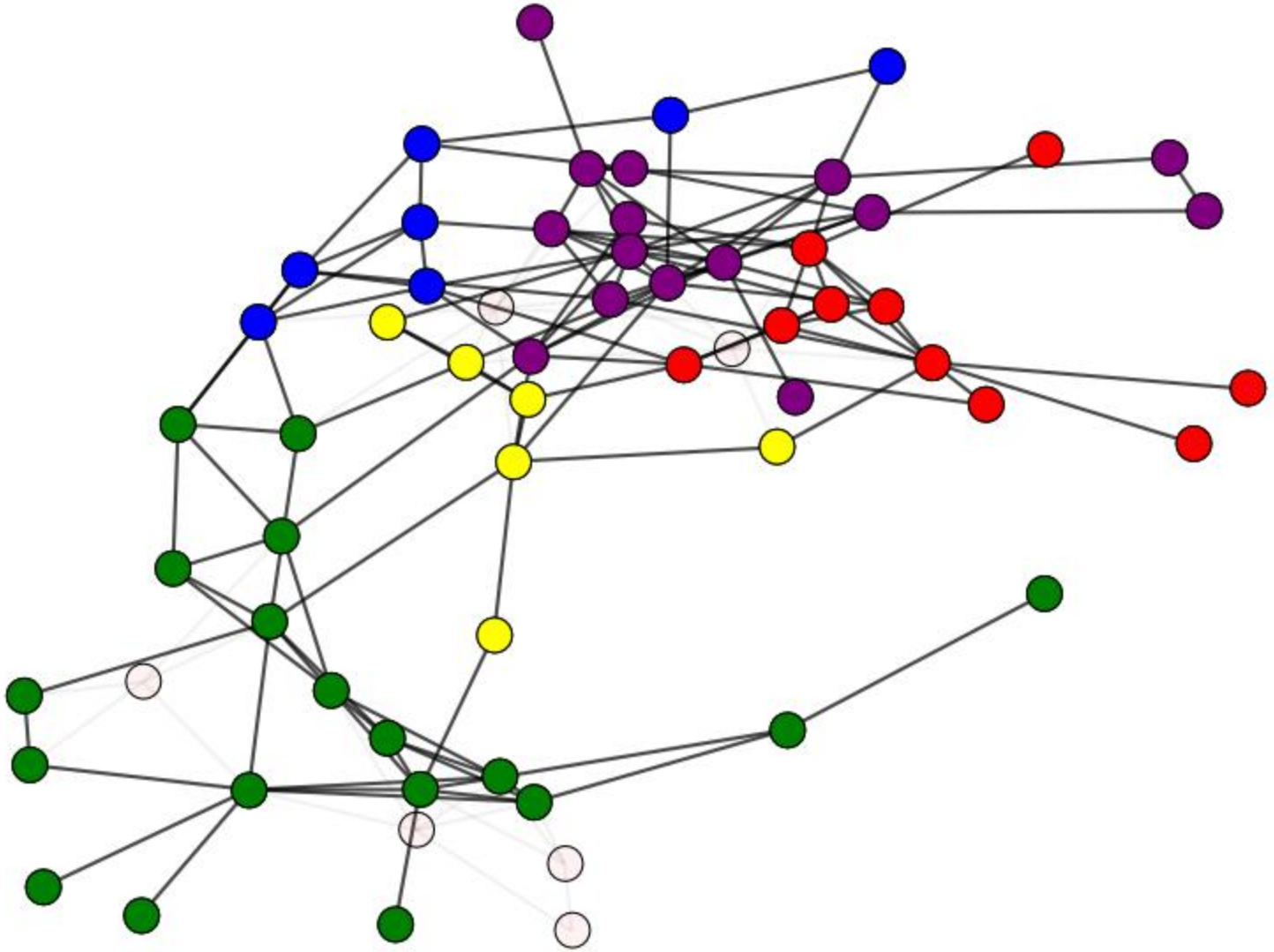


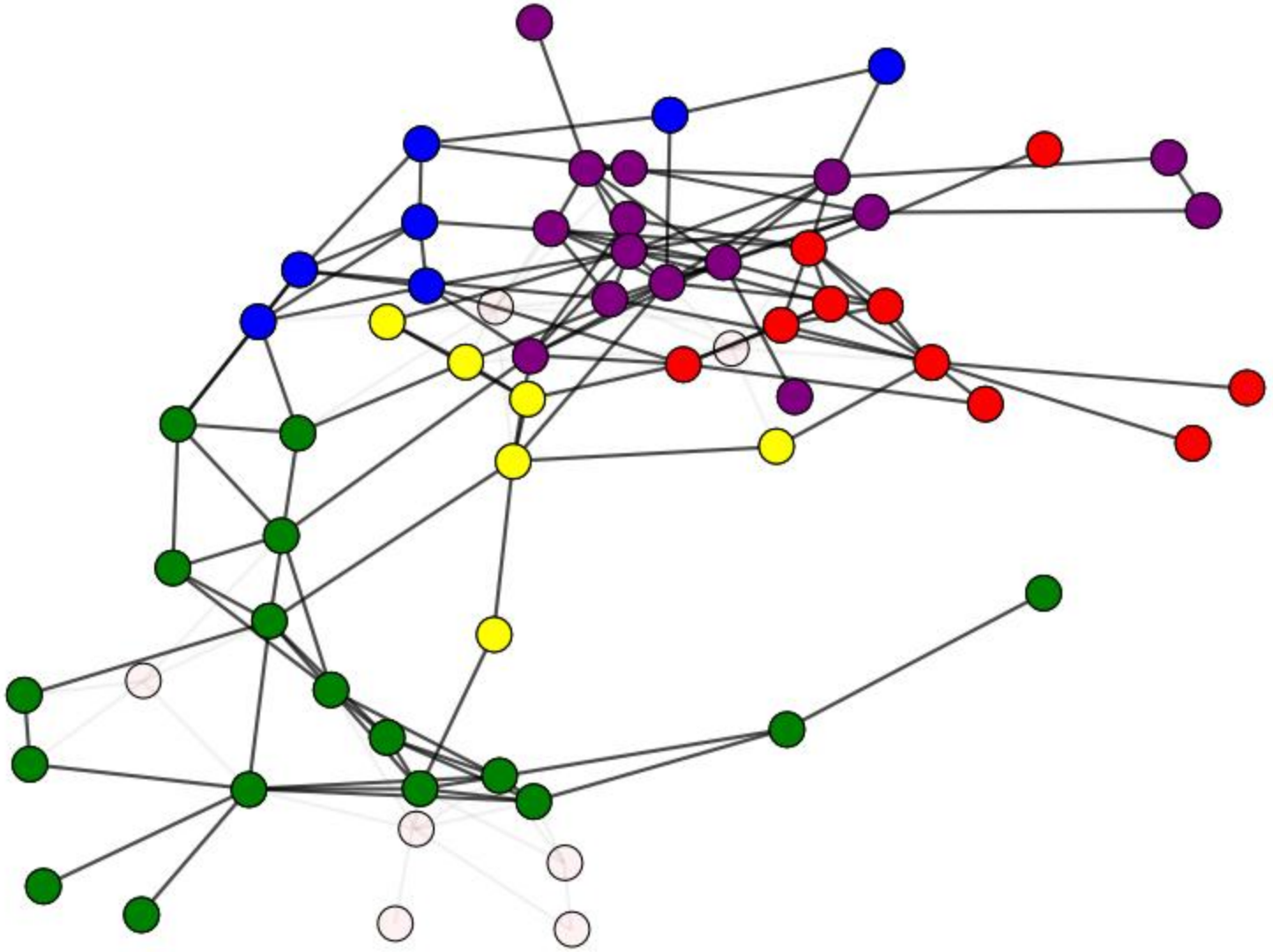










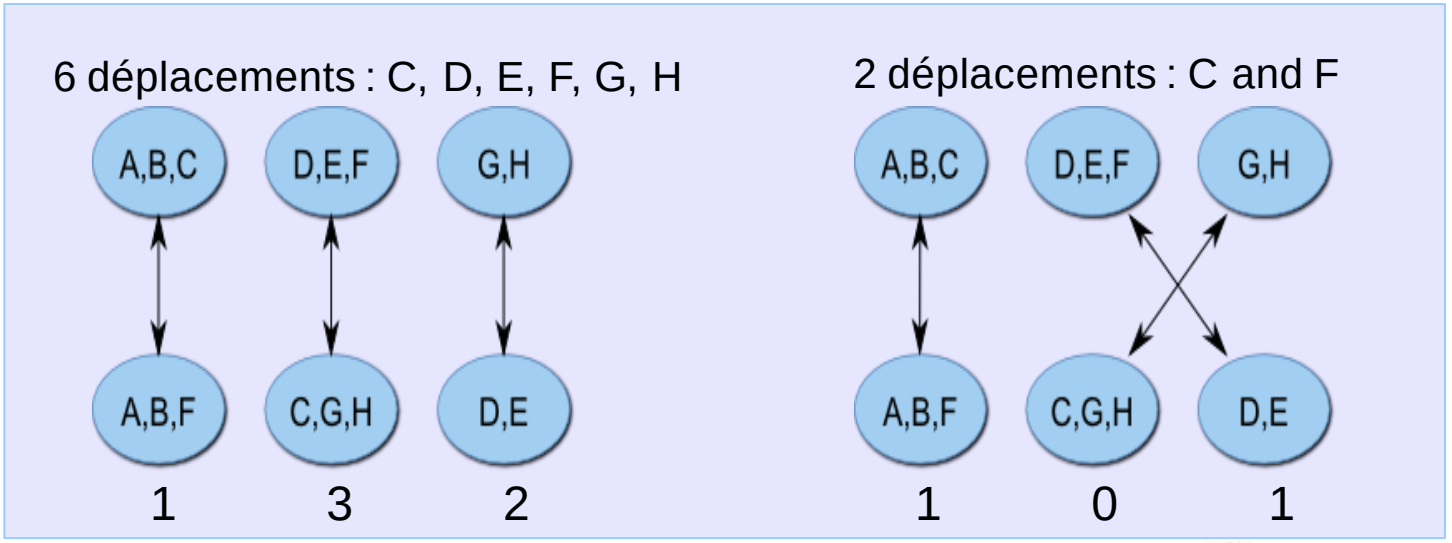


Evaluation – deux paramètres

Qualité de l'algorithme à chaque étape (modularité)

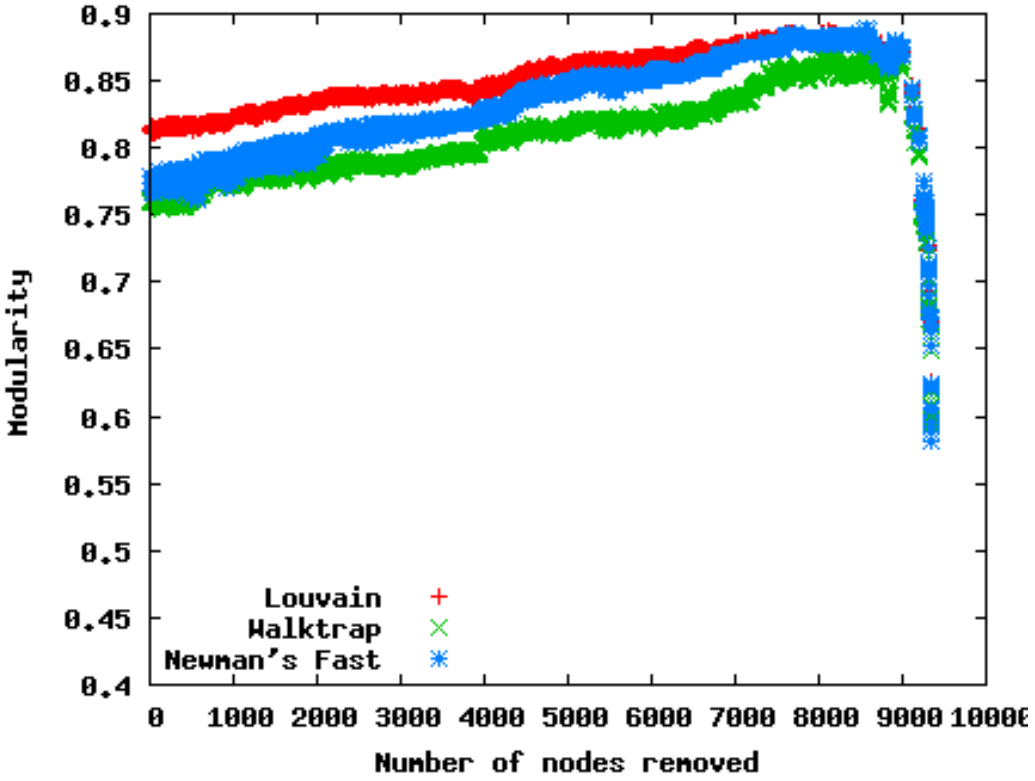
Stabilité de l'algorithme :

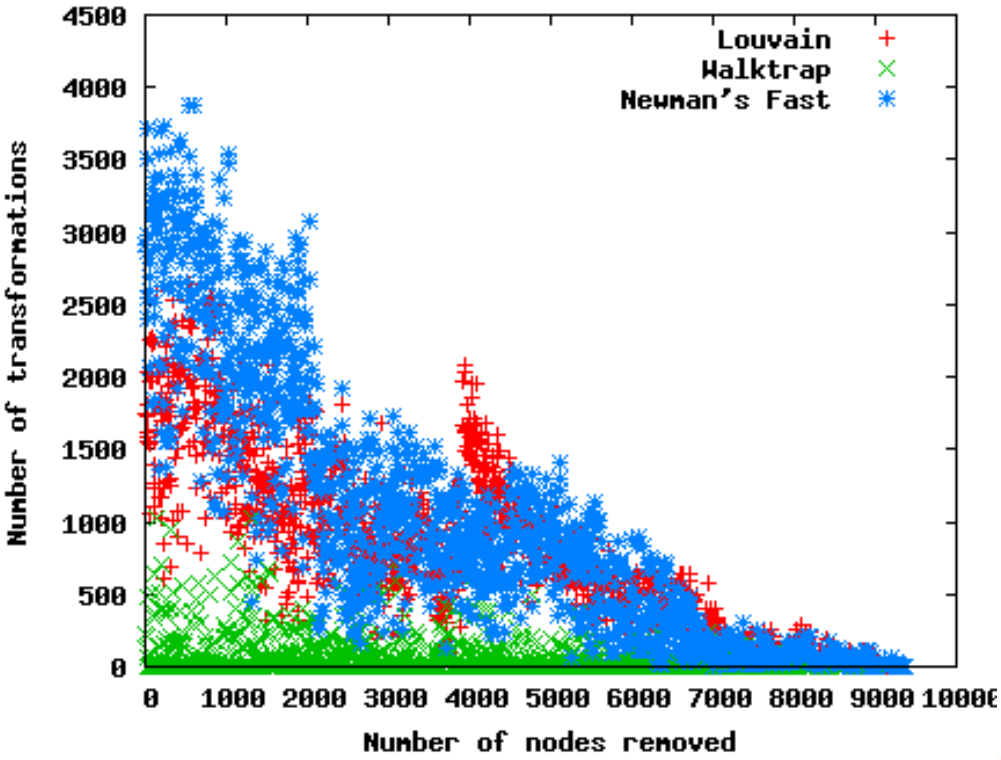
Nombre minimal de transformations pour passer d'une partition à l'autre.



Qualité / Stabilité

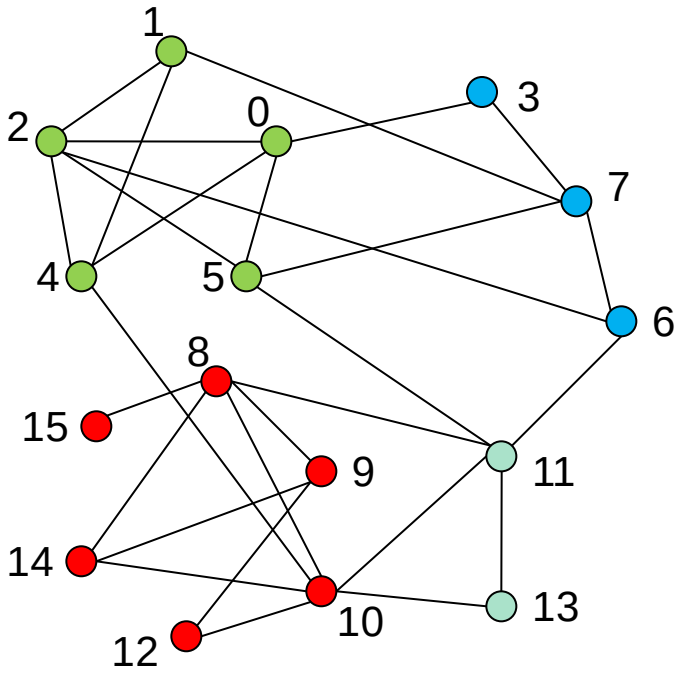
Louvain et Newman's sont très instables (9000 sommets).
Walktrap est plus stable mais avec une qualité plus faible.





Stabilisation de Louvain

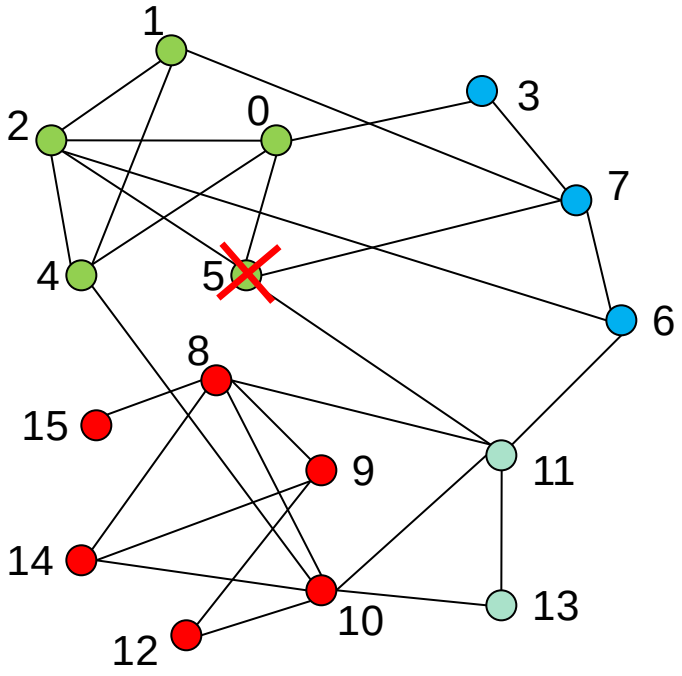
Avant modification : calcul classique.

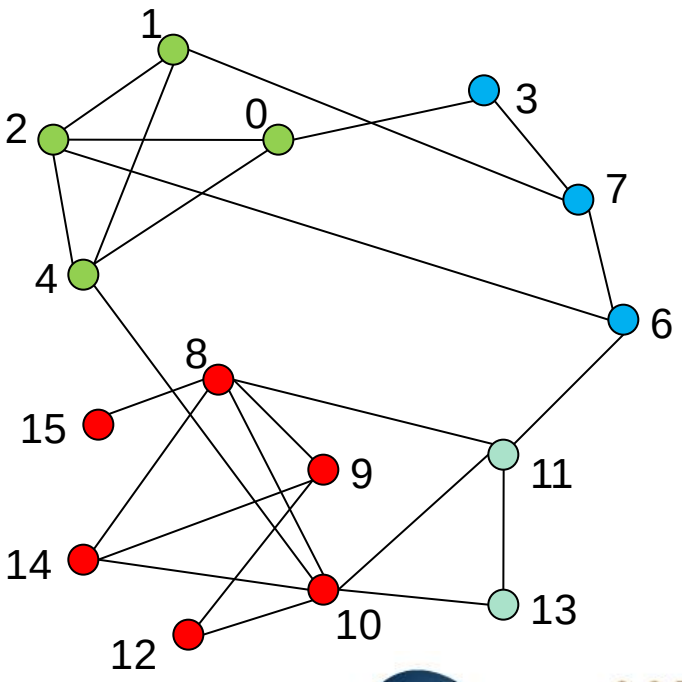


Stabilisation de Louvain

Avant modification : calcul classique.

Après modification : on initialise avec les communautés précédentes et on recommence (passe 1, itération 2).

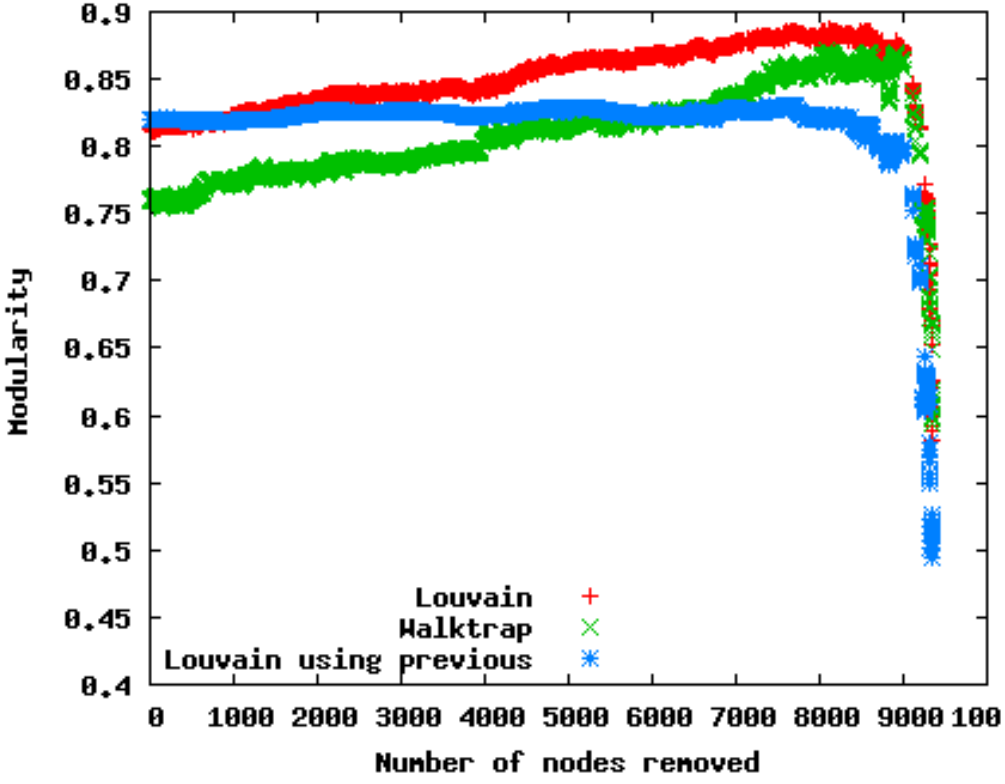


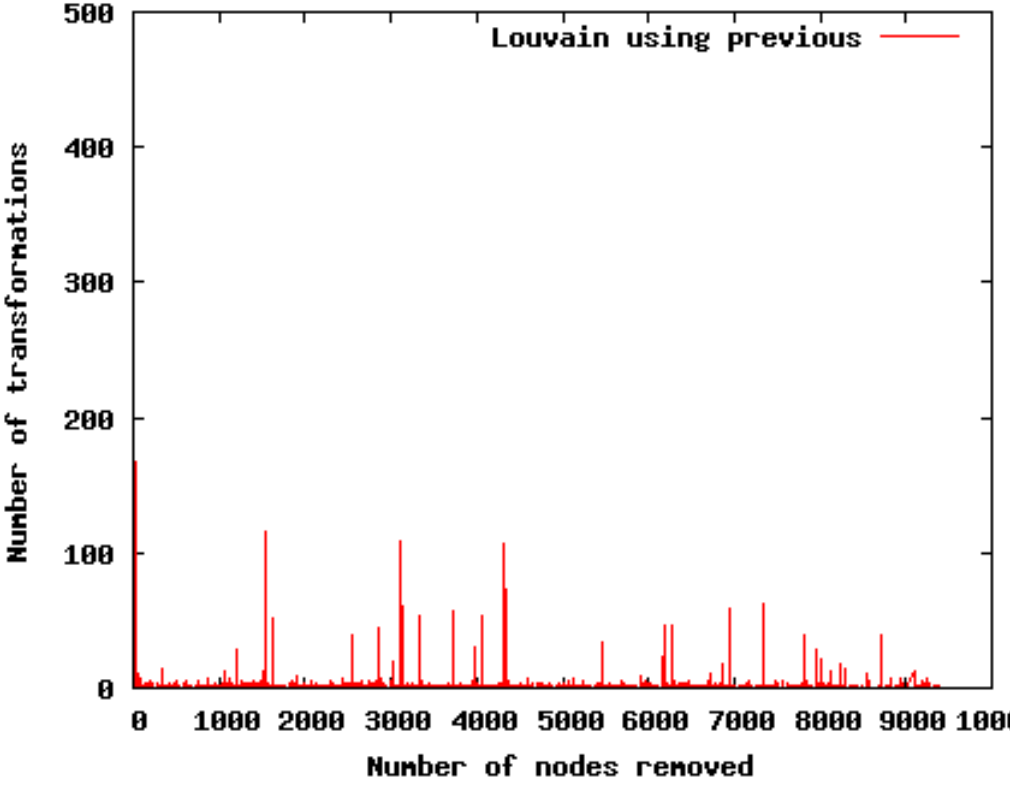


Qualité / Stabilité

Compromis :

- Perte de qualité mais toujours efficace.
- Plus stable avec quelques pics.



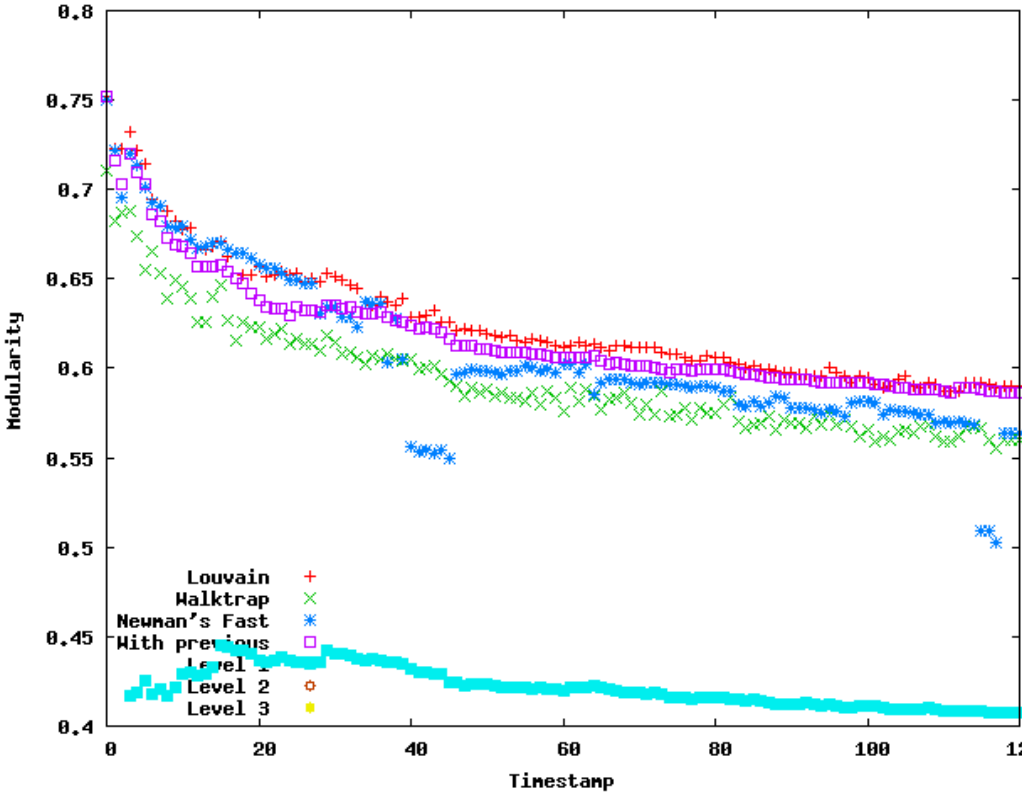


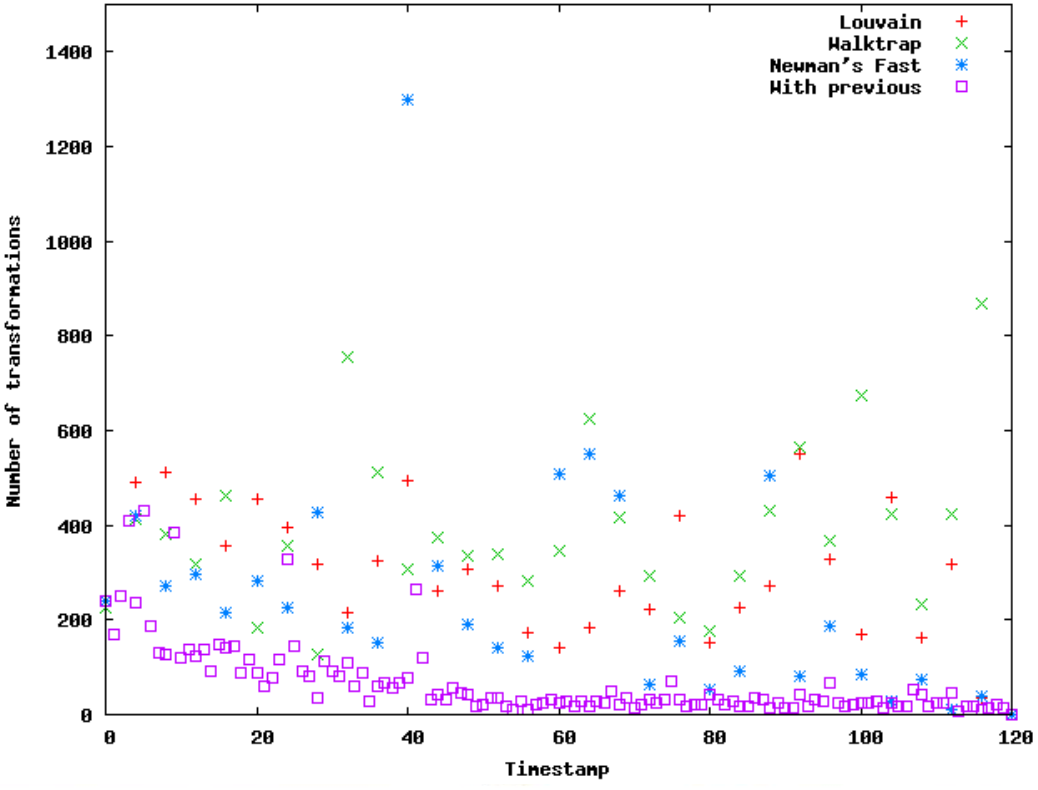
Sur un réseau réel

Réseau de blogs (hyperliens entre blogs).

Les résultats sont similaires :

- Toujours des événements mais assez stable.
- Louvain stabilisé fournit de bons résultats.



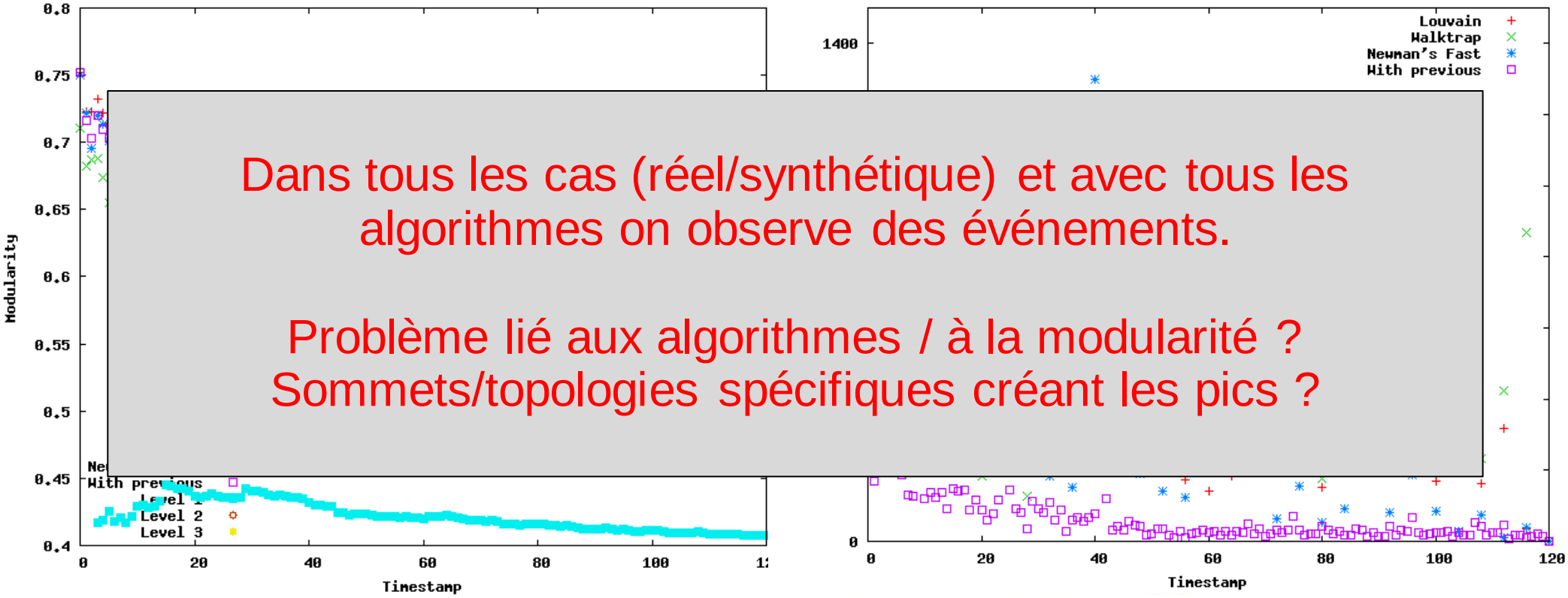


Sur un réseau réel

Réseau de blogs (hyperliens entre blogs).

Les résultats sont similaires :

- Toujours des événements mais assez stable.
- Louvain stabilisé fournit de bons résultats.



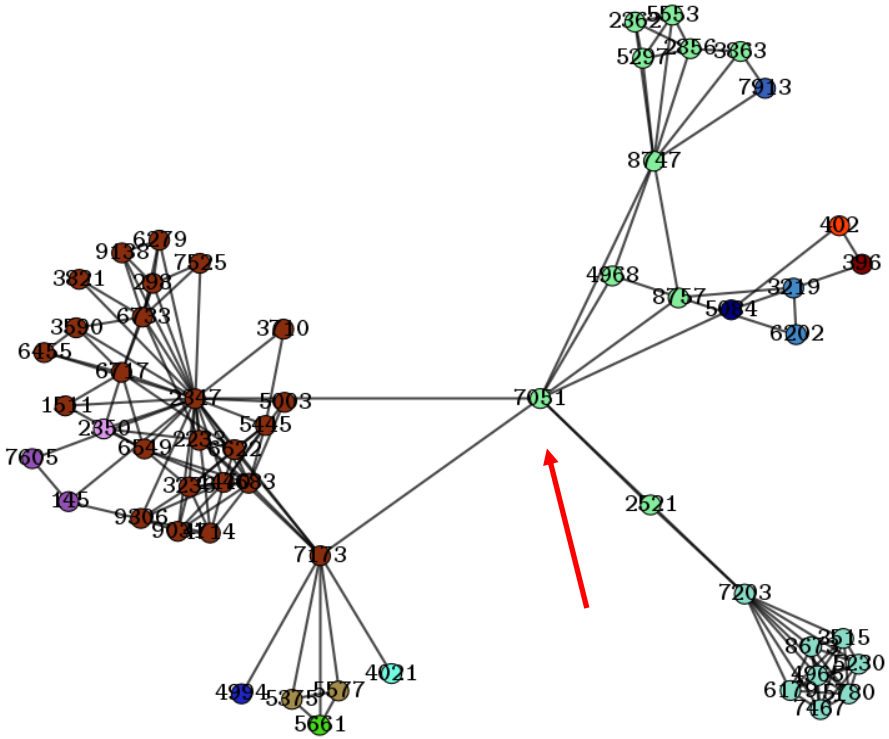
Dans tous les cas (réel/synthétique) et avec tous les algorithmes on observe des événements.

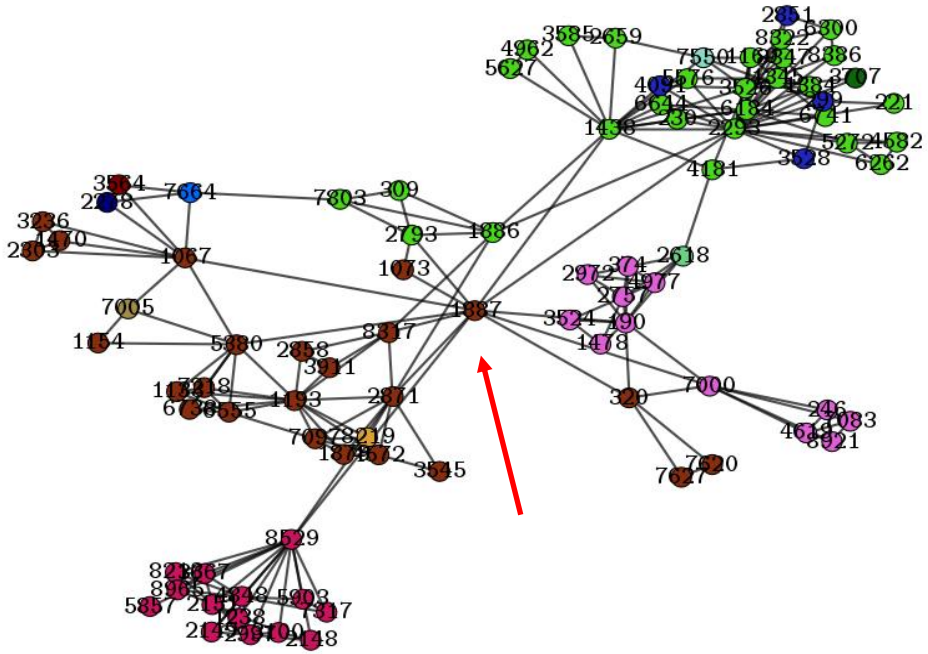
Problème lié aux algorithmes / à la modularité ?
Sommets/topologies spécifiques créant les pics ?

Comprendre les événements

Pour chaque sommet du réseau :

- Calculer les communautés, supprimer le sommet et recalculer.
- Chercher le nombre de modifications.
- Trouver des propriétés topologiques pour expliquer les changements.

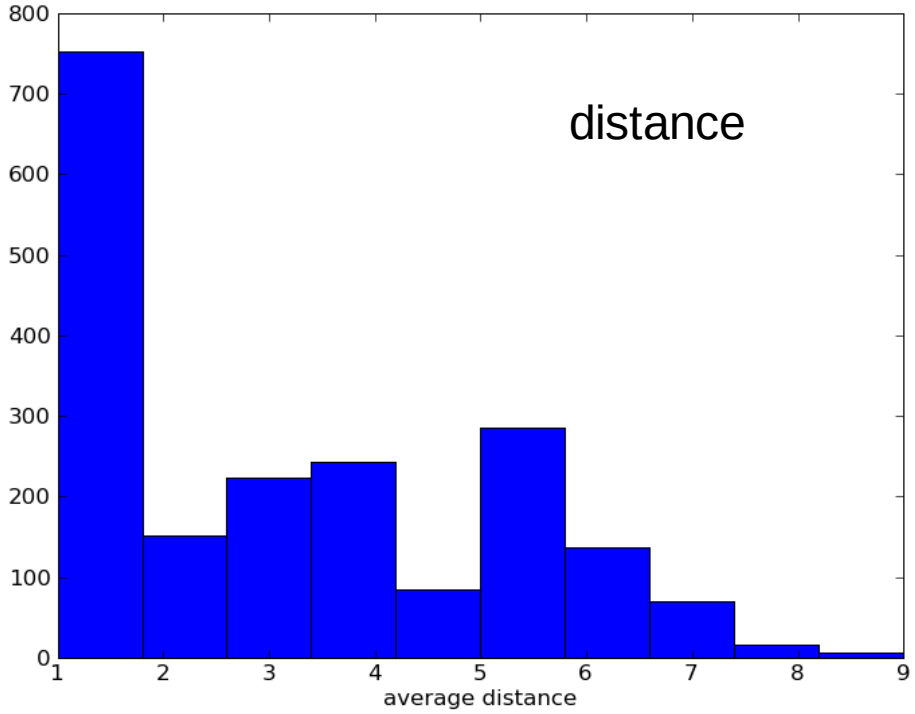


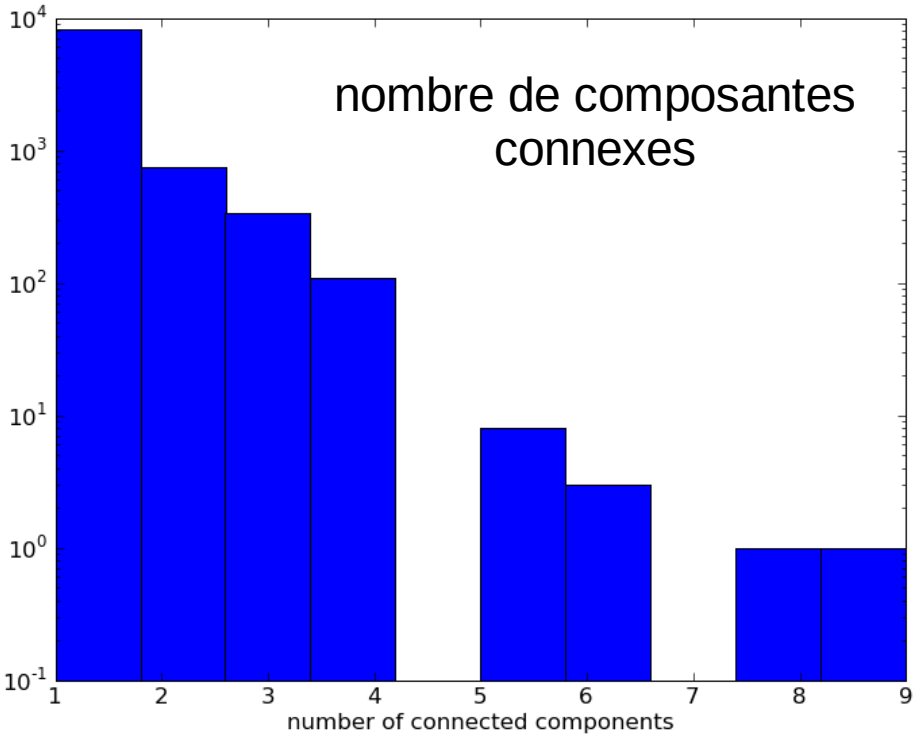


Quels sommets bougent ?

Proches du sommet supprimé en général

Peu de composantes connexes



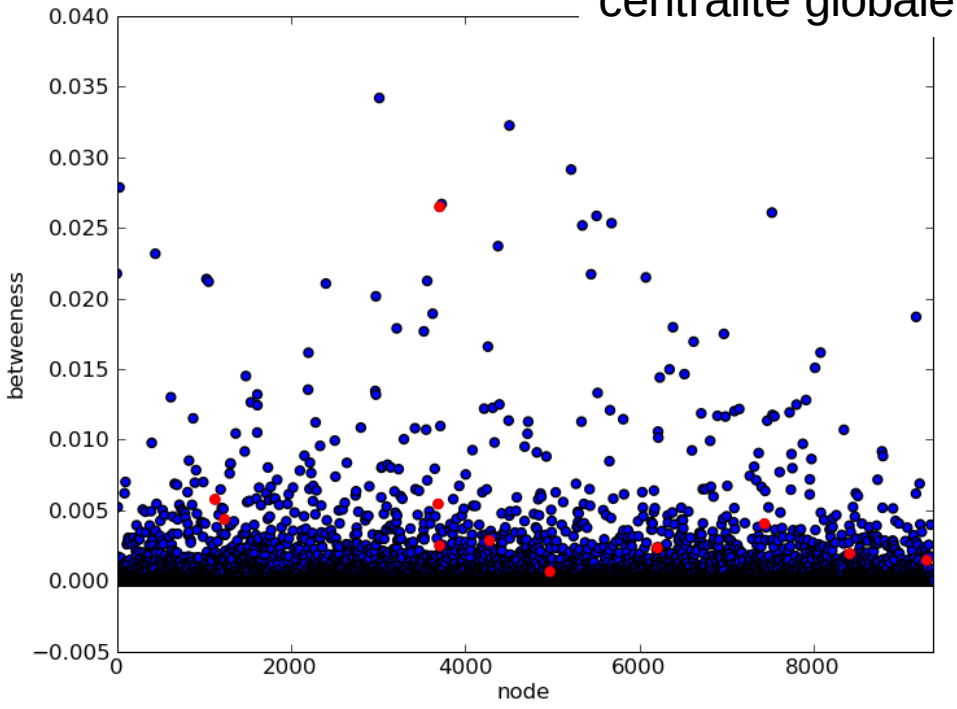


Centralité (betweenness)

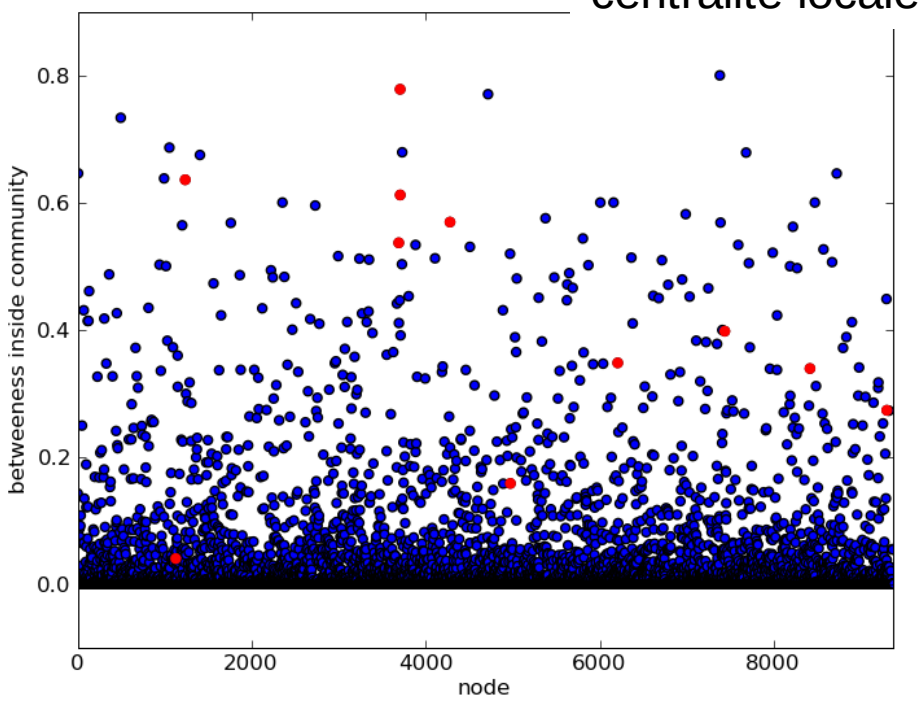
En rouge : les sommets provoquant le plus de modifications.

=> Les paramètres locaux sont plus pertinents.

centralité globale



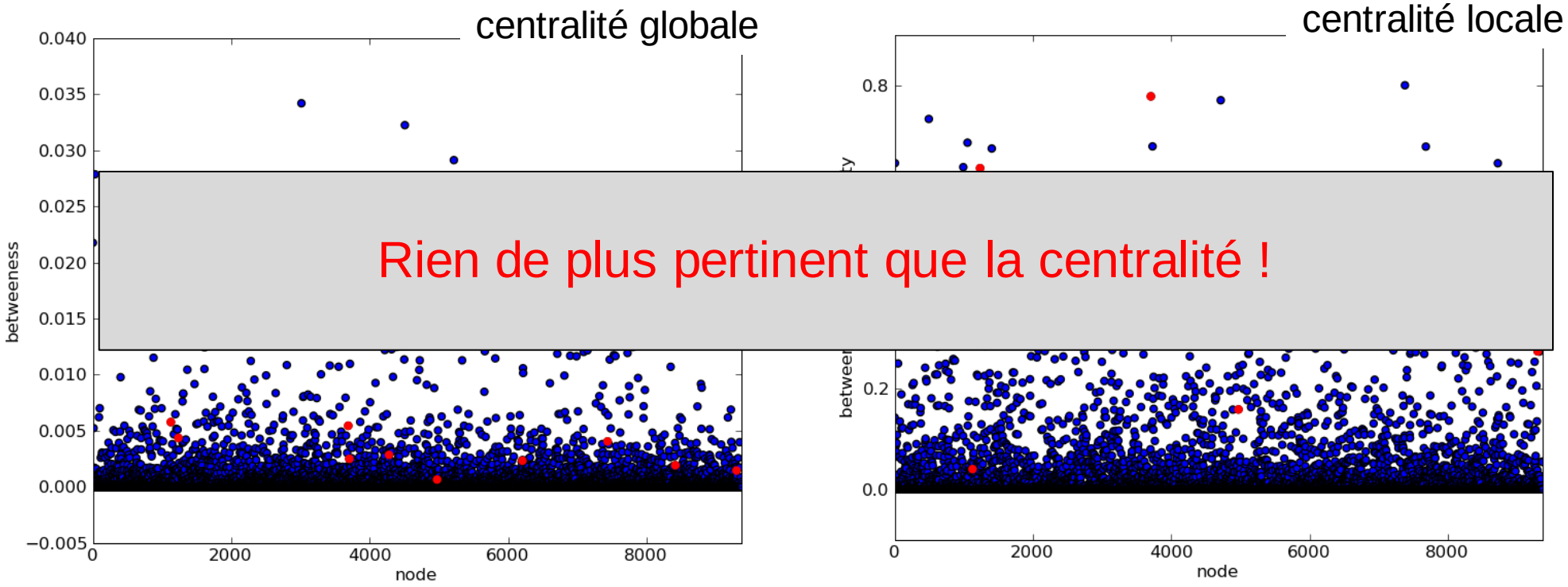
centralité locale



Centralité (betweenness)

En rouge : les sommets provoquant le plus de modifications.

=> Les paramètres locaux sont plus pertinents.



Rien de plus pertinent que la centralité !

Conclusion / perspectives

Les algorithmes classiques ne sont pas utilisables tel quels.

- Louvain stabilisé est plus stable que les autres algorithmes.
- Mais n'est-il pas trop contraignant ?
- Perte de qualité : recommencer à zéro si la qualité diminue trop ?

Importance de la localité :

- Les sommets qui bougent sont toujours proches des sommets supprimés.
- L'impact est généralement plus fort avec des propriétés locales.

Etudier des évolutions plus complexes.

Questions ?

Travaux soutenus par les projets :

- MAPAP SIP-2006-PP-221003 (Europe)
- MAPE (ANR)
- Webfluence (ANR)