

Visualization and analysis of complex networks

Scander Mustapha
École polytechnique

22 janvier 2018

Introduction

Ce rapport a pour objet de présenter les résultats des différents algorithmes implémentés : les deux algorithmes de visualisation et les deux algorithmes de calcul de communautés.

Pour chaque graphe, nous avons indiqué : le temps d'exécution des quatre algorithmes, et les modularités des communautés calculées par l'algorithme de Louvain et l'algorithme glouton. De plus, nous avons indiqué le nombre d'itérations de l'algorithme FRFastlayout nécessaires pour obtenir la visualisation du graphe correspondant, dont nous avons calculé une répartition en communauté par l'algorithme de Louvain.

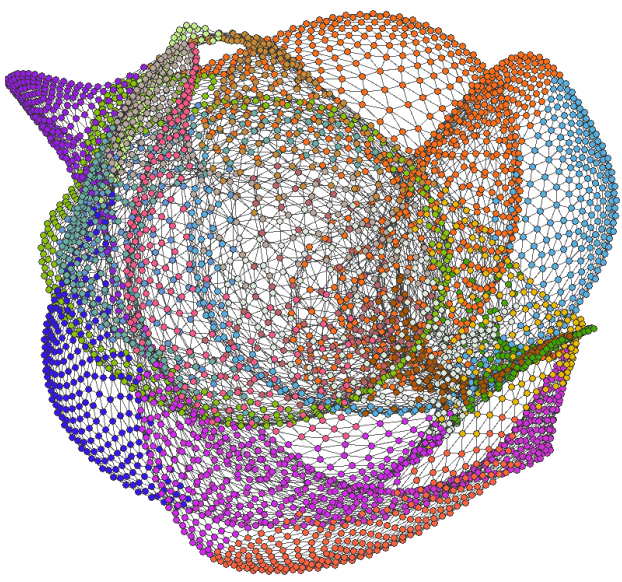
Enfin, nous avons fourni et démontré dans la seconde partie les formules que nous avons utilisées pour l'implémentation de l'algorithme de Louvain et l'algorithme glouton.

1 Résultats

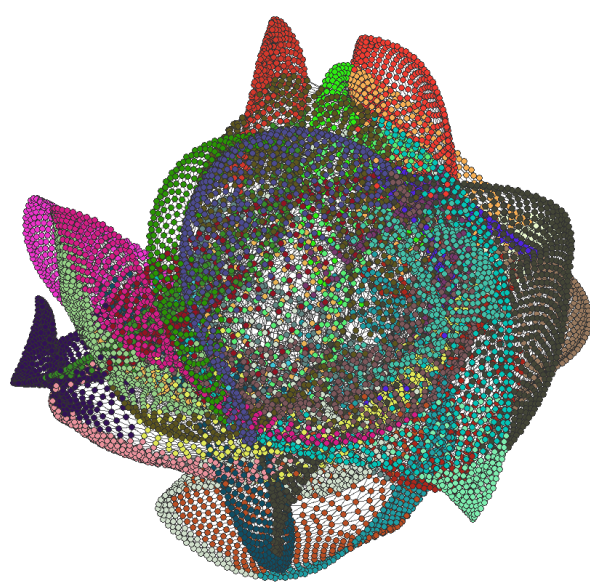
Les graphes sur lesquels nous avons appliqués les algorithmes sont les suivants. Cependant, nous n'avons pas pu obtenir de résultats satisfaisants pour le plus grand d'entre eux bcsstk31. Il n'est donc pas présent dans nos résultats.

Graphe	Nombre de sommets	Nombre d'arrêtes
3elt	4720	13722
ash85	85	304
barth5	15606	61484
dwt_592	592	2848
facebook	1128	1547
fe_4elt2	11143	32818
jagmesh7	1138	4294
karate	34	78
powergrid_4k	4931	6594
qh882	882	3354
siggraph_2005	33	40
west0487	497	1727

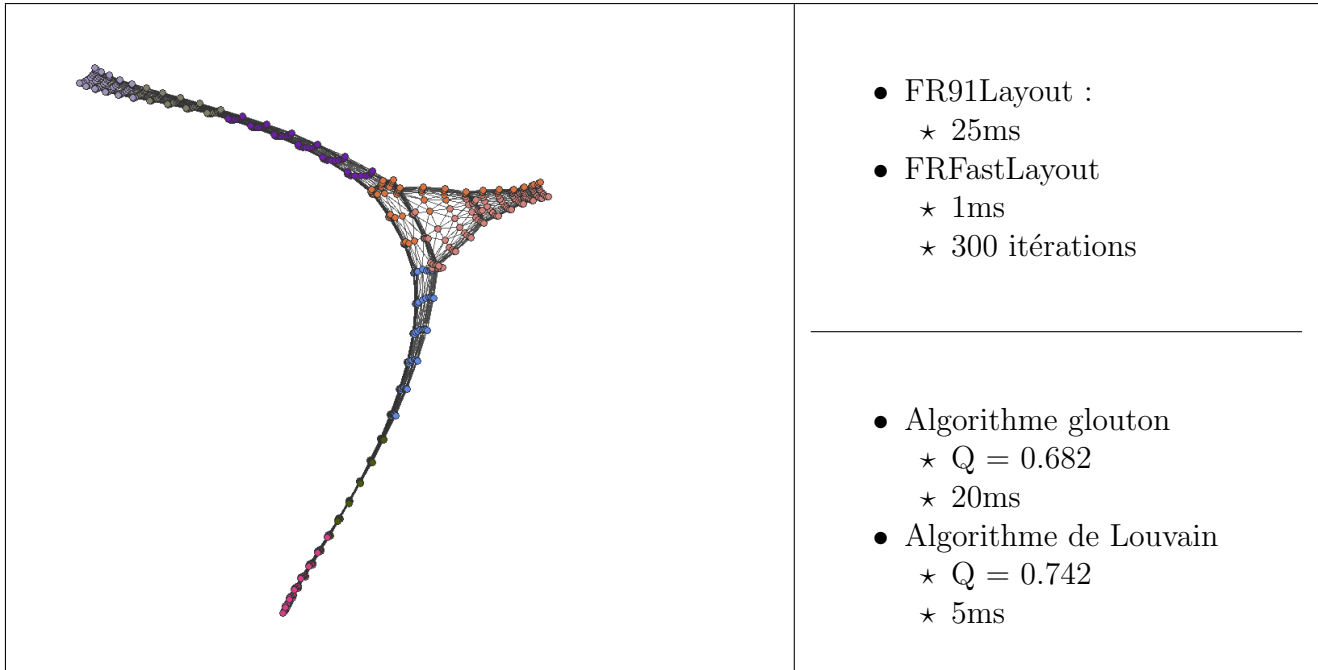
1.1 3elt

	<ul style="list-style-type: none">• FR91Layout :<ul style="list-style-type: none">★ 1ms• FRFastLayout<ul style="list-style-type: none">★ 0.5ms★ 200 itérations
	<ul style="list-style-type: none">• Algorithme glouton<ul style="list-style-type: none">★ $Q = 0.607$★ 4ms• Algorithme de Louvain<ul style="list-style-type: none">★ $Q = 0.618$★ 3ms

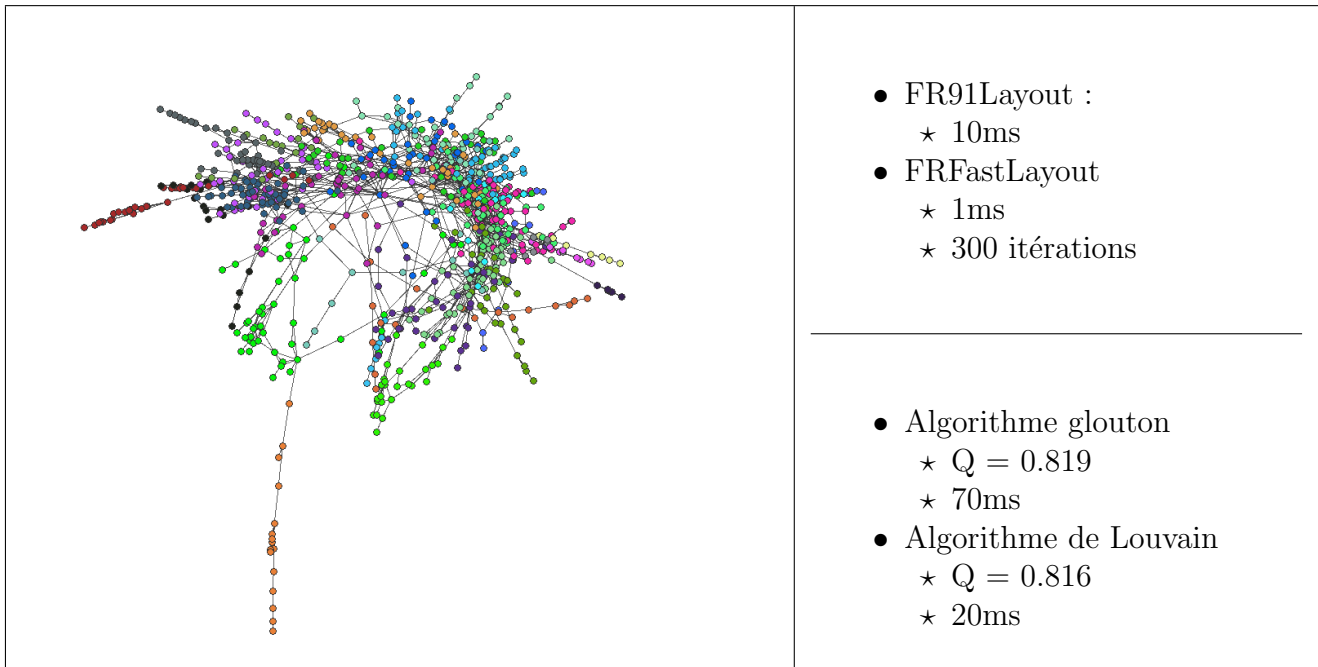
1.2 barth85

	<ul style="list-style-type: none">• FR91Layout :<ul style="list-style-type: none">★ 20s• FRFastLayout<ul style="list-style-type: none">★ 20ms★ 1000 itérations
	<ul style="list-style-type: none">• Algorithme glouton<ul style="list-style-type: none">★ $Q = \text{OutOfMemoryError : Java heap space (matrixB)}$★ 42s• Algorithme de Louvain<ul style="list-style-type: none">★ $Q = 0.92$★ 30s

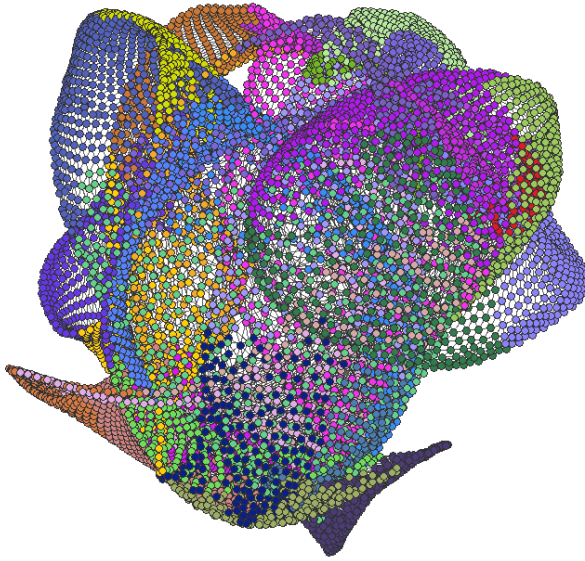
1.3 dwt_592



1.4 facebook



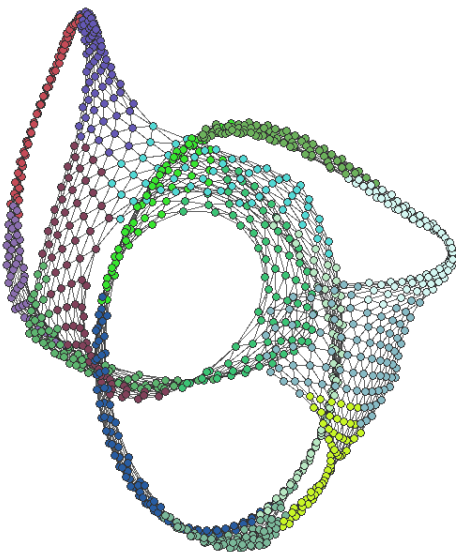
1.5 fe_4elt2



- FR91Layout :
 - ★ 10s
- FRFastLayout
 - ★ 30ms
 - ★ 1000 itérations

-
- Algorithme glouton
 - ★ $Q = \text{OutOfMemoryError}$
 - ★ 20s
 - Algorithme de Louvain
 - ★ $Q = 0.910$
 - ★ 12s

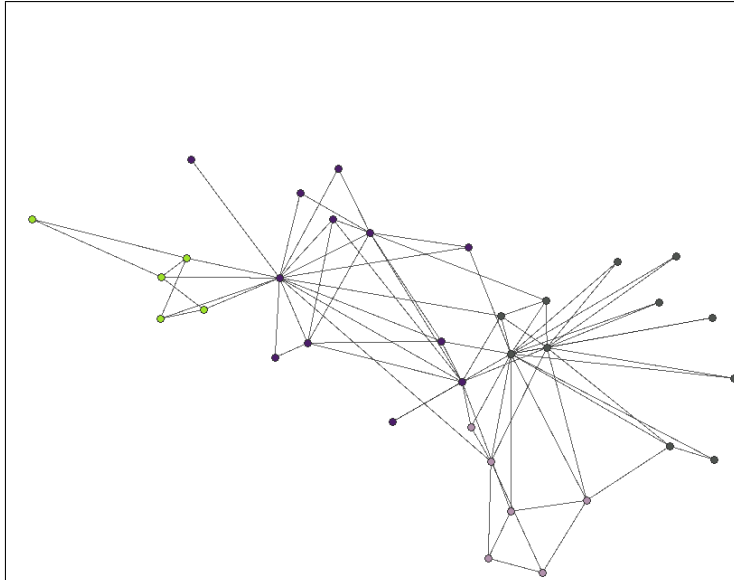
1.6 jagmesh7



- FR91Layout :
 - ★ 0.1s
- FRFastLayout
 - ★ 1ms
 - ★ 400 itérations

-
- Algorithme glouton
 - ★ $Q = 0.759$
 - ★ 70ms
 - Algorithme de Louvain
 - ★ $Q = 0.843$
 - ★ 20ms

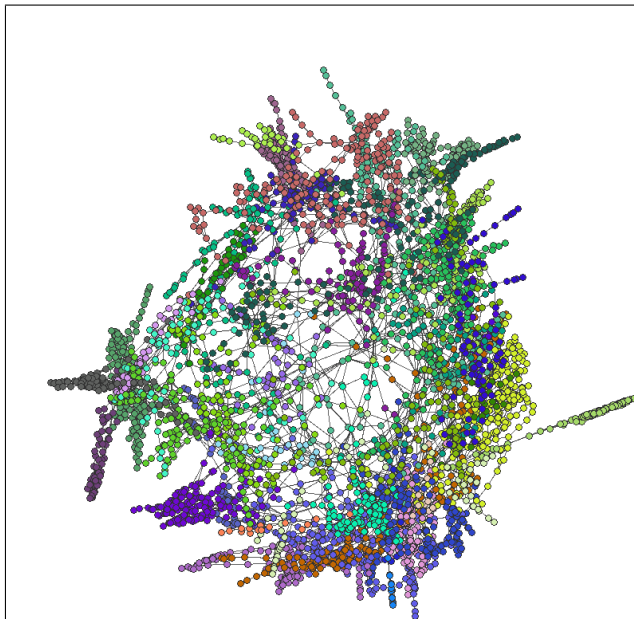
1.7 karate



- FR91Layout :
 - ★ 0ms
- FRFastLayout
 - ★ 0ms
 - ★ 100 itérations

-
- Algorithme glouton
 - ★ $Q = 0.381$
 - ★ 1ms
 - Algorithme de Louvain
 - ★ $Q = 0.419$
 - ★ 0ms

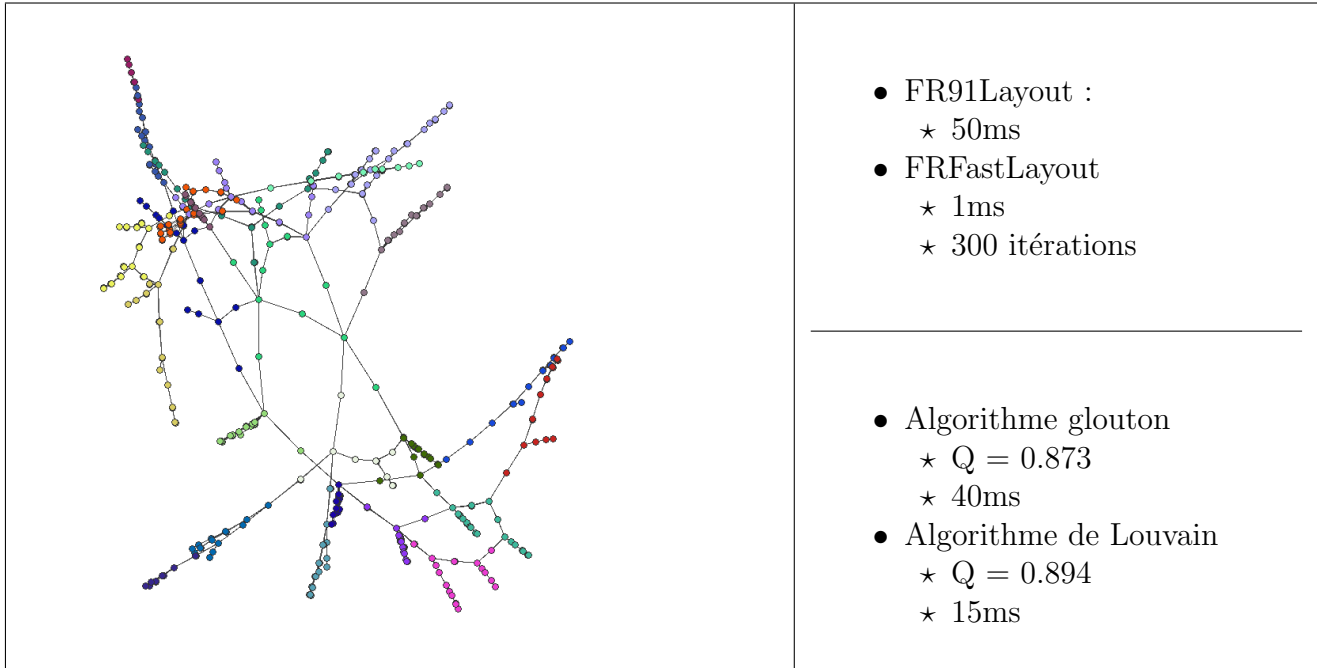
1.8 powergrid_4k



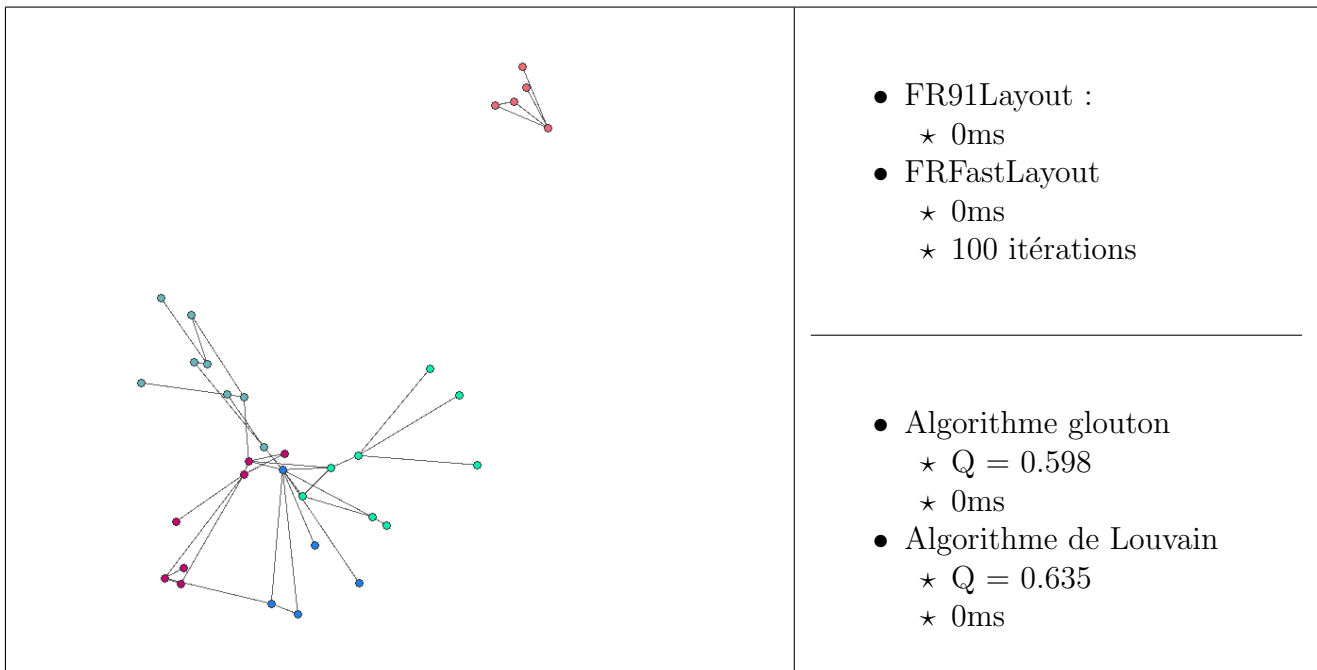
- FR91Layout :
 - ★ 2s
- FRFastLayout
 - ★ 10ms
 - ★ 300 itérations

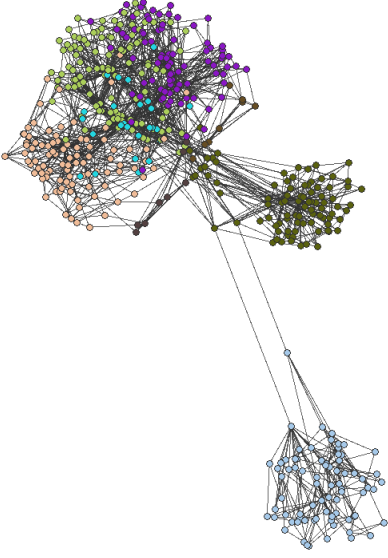
-
- Algorithme glouton
 - ★ $Q = 0.873$
 - ★ 2ms
 - Algorithme de Louvain
 - ★ $Q = 0.894$
 - ★ 0.950ms

1.9 qh882



1.10 siggraph2005



	<ul style="list-style-type: none"> • FR91Layout : <ul style="list-style-type: none"> ★ 30ms • FRFastLayout <ul style="list-style-type: none"> ★ 1ms ★ 300 itérations <hr/> <ul style="list-style-type: none"> • Algorithme glouton <ul style="list-style-type: none"> ★ $Q = 0.598$ ★ 20ms • Algorithme de Louvain <ul style="list-style-type: none"> ★ $Q = 0.635$ ★ 5ms
---	--

1.11 west0497

2 Détection de communautés

Soit $G = (V, E)$ un graphe non orienté, avec V l'ensemble des sommets de cardinal $n \in \mathbb{N}$ et E l'ensemble des arrêtes.

On suppose que $V = \{1, \dots, n\}$, que les arrêtes de G possèdent des poids et que G possède éventuellement des arrêtes allant d'un sommet vers lui-même.

Notons pour une arrête $\{i, j\} \in E$ son poids $A_{ij} := A_{\{i,j\}} = A_{\{j,i\}}$, et définissons $k_i = \sum_{\substack{j=1 \\ \{i,j\} \in E}}^n A_{i,j}$

pour $i \in V$. On note aussi pour un sommet $r \in V$, $C(r) = \{s \in V | c_s = c_r\}$ l'ensemble des sommets de la même communauté.

On note ainsi $2m$ la somme des poids des arrêtes :

$$2m = \sum_{\{i,j\} \in E} A_{ij}$$

Considérons une répartition des sommets en communautés : on les note C_1, \dots, C_p et c_i la communauté du sommet $i \in V$. On définit la modularité du graphe muni de ces communautés par :

$$Q(C) = \frac{1}{2m} \sum_{i,j} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j)$$

2.1 Algorithme glouton

Soit $C = (C(r_1), \dots, C(r_p))$ une répartition en communautés

On définit la communauté $\bar{C} = (C(r_1) \cup C(r_2), C(r_3), \dots, C(r_p))$ en fusionnant $C(r_1)$ et $C(r_2)$.

Notons $\Delta Q = Q(\tilde{C}) - Q(C)$, $B_{C(p)C(q)} = \sum_{\substack{i \in C(p) \\ j \in C(q)}} A_{ij}$ et $K_{C(l)} = \sum_{i \in C(l)} k_i$

On a alors la formule :

Formule 1.

$$\Delta Q = \frac{2}{2m} (B_{C(r_1)C(r_2)} - \frac{K_{C(r_1)}K_{C(r_2)}}{2m})$$

Démonstration. Sans nuire à la généralité $r_1 = 1$ et $r_2 = 2$.

Notons $C_1 = C(1)$, $C_2 = C(2)$ et $\alpha_{ij} = A_{ij} - \frac{k_i k_j}{2m}$ pour $(i, j) \in \{1, \dots, n\}^2$.

Alors :

$$\begin{aligned} 2mQ(\tilde{C}) &= \sum_{i,j} \alpha_{ij} \delta(\tilde{c}_i, \tilde{c}_j) \\ &= \sum_{i \in C_1, j} \alpha_{ij} \delta(c_2, \tilde{c}_j) + \sum_{i \notin C_1, j} \alpha_{ij} \delta(c_i, \tilde{c}_j) \\ &= \sum_{i \in C_1, j \in C_1} \alpha_{ij} + \sum_{i \in C_1, j \in C_2} \alpha_{ij} + \sum_{i \notin C_1, j \in C_1} \alpha_{ij} \delta(c_i, c_2) + \sum_{i \notin C_1, j \notin C_1} \alpha_{ij} \delta(c_i, c_j) \\ &= \sum_{i \in C_1, j \in C_1} \alpha_{ij} + \sum_{i \in C_1, j \in C_2} \alpha_{ij} + \sum_{i \in C_2, j \in C_1} \alpha_{ij} + \sum_{i \notin C_1, j \notin C_1} \alpha_{ij} \delta(c_i, c_j) \end{aligned}$$

On a aussi :

$$2mQ(C) = \sum_{i \in C_1, j \in C_1} \alpha_{ij} + \sum_{i \notin C_1, j \notin C_1} \alpha_{ij}$$

Il vient :

$$2m\Delta Q = 2 \sum_{i \in C_1, j \in C_2} \alpha_{ij} = 2 \left(\sum_{\substack{i \in C_1 \\ j \in C_2}} A_{ij} - \frac{(\sum_{i \in C_1} k_i)(\sum_{i \in C_2} k_i)}{2m} \right)$$

D'où le résultat :

$$\Delta Q = \frac{2}{2m} (B_{C_1 C_2} - \frac{K_{C_1} K_{C_2}}{2m})$$

□

De plus, on calcule facilement \tilde{B} et \tilde{K} pour \tilde{C} :

- $\tilde{B}_{C(r_1) \cup C(r_2) C(r_j)} = B_{C(r_1) C(r_2)} + B_{C(r_2) C(j)}$ pour $j \geq 3$
- $\tilde{B}_{C(r_1) C_1 \cup C(r_2)} = B_{C(r_1) C(r_j)} + B_{C(r_2) C(j)}$ pour $j \geq 3$
- $\tilde{B}_{C(r_1) \cup C(r_2) C(r_1) \cup C(r_2)} = B_{C(r_1) C(r_1)} + B_{C(r_2) C(r_2)} + 2B_{C(r_1) C(r_2)}$
- $\tilde{K}_{C(r_1) \cup C(r_2)} = K_{C(r_1)} + K_{C(r_2)}$

Les coefficients restent les mêmes sinon.

2.2 Algorithme de Louvain

Soit $C = (C(r_1), \dots, C(r_p))$ une répartition en p communautés, $r \in \{r_1, \dots, r_p\}$ un sommet et $s \in \{r_1, \dots, r_p\} \setminus \{r\}$ un autre sommet qui n'est pas dans $C(r)$. On souhaite calculer le changement de modularité en faisant passer r de la communauté $C(r)$ à la communauté $C(s)$.

On définit alors \tilde{C} par $\tilde{c}_u = \begin{cases} c_s & \text{si } u = r \\ c_u & \text{sinon} \end{cases}$.

Notons pour des sommets $(u, v) \in V^2$:

$$\begin{aligned} - L(C(r)) &= \sum_{i \in C(r)} k_i \\ - \Sigma(i, C(r)) &= \sum_{j \in C(r)} A_{ij} \end{aligned}$$

On a alors la formule :

Formule 2.

$$\Delta Q = \frac{2}{2m}(\Sigma(r, C(s)) - \Sigma(r, C(r)) + \frac{k_r}{2m}(L(C(r)) - L(C(s))) + (A_{rr} - \frac{k_r^2}{2m}))$$

Démonstration. Sans nuire à la généralité $r = 1$ et $s = 2$. Notons $\alpha_{ij} = A_{ij} - \frac{k_i k_j}{2m}$. On a alors $\tilde{c}_1 = c_2$ et $\tilde{c}_l = c_l$ pour $2 \leq l \leq n$. Donc :

$$\begin{aligned} 2m\tilde{Q} &= \sum_{i,j} \alpha_{ij} \delta(\tilde{c}_i, \tilde{c}_j) \\ &= \sum_{i=2}^n \sum_{j=1}^n \alpha_{ij} \delta(c_i, \tilde{c}_j) + \sum_{j=1}^n \alpha_{1j} \delta(c_2, \tilde{c}_j) \\ &= \sum_{i=2}^n \sum_{j=2}^n \alpha_{ij} \delta(c_i, c_j) + \sum_{i=2}^n \alpha_{i1} \delta(c_i, c_2) + \sum_{j=1}^n \alpha_{1j} \delta(c_2, \tilde{c}_j) \\ &= \sum_{i=2}^n \sum_{j=2}^n \alpha_{ij} \delta(c_i, c_j) + 2 \sum_{j \in C(2)} (A_{1j} - \frac{k_1 k_j}{2m}) + A_{11} - \frac{k_1^2}{2m} \\ &= \sum_{i=2}^n \sum_{j=2}^n \alpha_{ij} \delta(c_i, c_j) + 2\Sigma(1, 2) - 2\frac{k_1 L(2)}{2m} + A_{11} - \frac{k_1^2}{2m} \end{aligned}$$

De plus :

$$\begin{aligned}
2mQ &= \sum_{i,j} \alpha_{ij} \delta(c_i, c_j) \\
&= \sum_{i=2}^n \sum_{j=1}^n \alpha_{ij} \delta(c_i, c_j) + \sum_{j=1}^n \alpha_{1j} \delta(c_1, c_j) \\
&= \sum_{i=2}^n \sum_{j=2}^n \alpha_{ij} \delta(c_i, c_j) + \sum_{i=2}^n \alpha_{i1} \delta(c_i, c_1) + \sum_{j=1}^n \alpha_{1j} \delta(c_1, c_j) \\
&= \sum_{i=2}^n \sum_{j=2}^n \alpha_{ij} \delta(c_i, c_j) + 2 \sum_{j=1}^n \alpha_{1j} \delta(c_1, c_j) - \alpha_{11} \\
&= \sum_{i=2}^n \sum_{j=2}^n \alpha_{ij} \delta(c_i, c_j) + 2(\Sigma(1, 1) - \frac{k_1 L(1)}{2m}) - (A_{11} - \frac{k_1^2}{2m})
\end{aligned}$$

Il vient :

$$\begin{aligned}
2m\Delta Q &= -2\Sigma(1, C(1)) + (A_{11} - \frac{k_1^2}{2m}) + 2\Sigma(1, C(2)) - \frac{k_1 L(C(2))}{2m} + (A_{11} - \frac{k_1^2}{2m}) \\
&= 2(\Sigma(1, C(2)) - \Sigma(1, C(1))) + \frac{k_1(L(C(1)) - L(C(2)))}{2m} + (A_{11} - \frac{k_1^2}{2m})
\end{aligned}$$

D'où le résultat :

$$\Delta Q = \frac{2}{2m}(\Sigma(1, C(2)) - \Sigma(1, C(1))) + \frac{k_1(L(C(1)) - L(C(2)))}{2m} + (A_{11} - \frac{k_1^2}{2m})$$

□

De plus, on calcule facilement les matrices $\tilde{\Sigma}$ et \tilde{L} pour \tilde{C} :

- $\tilde{\Sigma}(i, C(r)) = \Sigma(i, C(r)) - A_{ir}$
- $\tilde{\Sigma}(i, C(s)) = \Sigma(i, C(r)) + A_{ir}$
- $\tilde{L}(C(r)) = L(C(r)) - k_r$
- $\tilde{L}(C(s)) = L(C(s)) + k_s$