Validation et optimisation d'une décomposition hiérarchique de graphes

François Queyroi

Validation et optimisation d'une décomposition hiérarchique de graphes

François Queyroi*

*Université de Bordeaux I, CNRS, LaBRI, INRIA Bordeaux – Sud-Ouest, France
francois.queyroi@labri.fr,
http://www.labri.fr

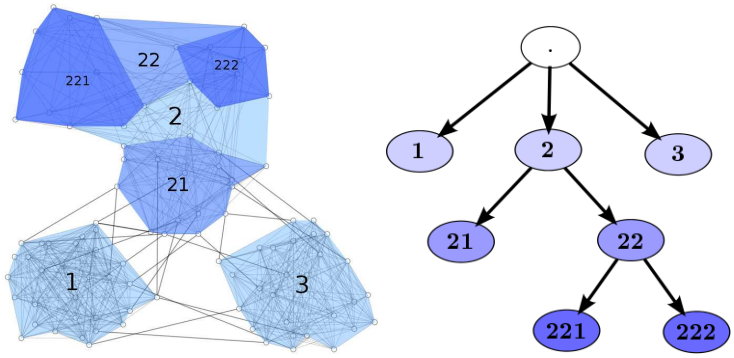
Résumé. De nombreux algorithmes de fragmentation de graphes fonctionnent par agrégations ou divisions successives de sous-graphes menant à une décomposition hiérarchique du réseau étudié. Une question importante dans ce domaine est de savoir si cette hiérarchie reflète la structure du réseau ou si elle n'est qu'un artifice lié au déroulement de la procédure. Nous proposons un moyen de valider et, au besoin, d'optimiser la décomposition multi-échelle produite par ce type de méthode. On applique notre approche sur l'algorithme proposé par Blondel et al. (2008) basé sur la maximisation de la modularité. Dans ce cadre, une généralisation de cette mesure de qualité au cas multi-niveaux est introduite. Nous testons notre méthode sur des graphes aléatoires ainsi que sur des exemples réels issus de divers domaines.

1 Introduction

La compréhension de l'organisation des communautés dans les réseaux est une problématique importante dans le domaine de l'analyse de ces structures. Différents travaux (voir, entre autres, Simon (1962) et Pumain (2006)) suggèrent que ces systèmes que l'on peut qualifier de complexes se composent de groupes d'individus rassemblés à leur tour en groupes d'individus plus grands menant à une hiérarchie de communautés.

Beaucoup d'algorithmes de fragmentation de graphes tentent de reproduire ce phénomène de décompositions de groupes ou d'agrégations de groupes. Les algorithmes agglomératifs basés sur des mesures de similarité (voir Fortunato (2010)) en sont un bon exemple puisqu'ils cherchent à regrouper à chaque étape les deux ensembles d'individus les plus proches. Toutefois, la hiérarchie obtenue est rarement pertinente car chaque niveau ne correspond qu'à la division d'un seul groupe en deux sous-ensembles. Des travaux tels que ceux de Pons et Latapy (2010) permettent dans ce cadre la détection des divisions pertinentes sans toutefois évaluer globalement la hiérarchie produite.

D'autres approches produisent une hiérarchie de communautés directement "exploitable" du point de vue de l'analyse. On peut notamment citer les algorithmes de Blondel et al. (2008), Lancichinetti et al. (2011) et Rosvall et Bergstrom (2011). Dans le premier, la hiérarchie n'est pas réellement l'objectif final de la méthode mais une construction nécessaire à l'obtention du résultat (en l'occurrence un découpage simple des individus). On peut néanmoins se demander si cette hiérarchie a du sens et reflète bien la structure du réseau étudié.



(a) Un graphe doté d'une décomposition hiérarchique.

(b) Un exemple d'arbre de fragmentation annoté

FIG. 1 – *Illustration d'une décomposition hiérarchique d'un graphe (à gauche) représentée sous la forme d'un arbre de fragmentation (à droite).*

Dans cet article, nous nous intéressons à la validation de ce type de hiérarchie. Nous présentons une procédure d'optimisation permettant de filtrer des regroupements inopportuns à différents niveaux. Après avoir introduit une généralisation de la modularité au cadre multi-niveaux, nous appliquons cette méthode à la décomposition hiérarchique produite par l'algorithme de Blondel et al. (2008).

Le reste de ce papier est organisé de la façon suivante. Dans la section 2 nous présentons l'approche utilisée pour évaluer la qualité des décompositions hiérarchiques ainsi qu'une présentation générale de notre méthode. Nous montrons dans la section 3 en quoi cette dernière est pertinente en l'appliquant à l'algorithme de Blondel et al. (2008). Dans la section 4, nous analysons différents résultats expérimentaux qui valident notre approche, d'abord sur un *benchmark* proposé par Lancichinetti et Radicchi (2008) puis sur des exemples réels en se comparant aux algorithmes de Lancichinetti et al. (2011) et de Rosvall et Bergstrom (2011).

2 Optimisation d'une décomposition hiérarchique

2.1 Définitions

Soit un graphe $G = (V, E)$ ayant pour ensemble de sommets V et pour ensemble d'arêtes E. Une décomposition simple de ce graphe revient à séparer les sommets de V en différentes classes (ou communautés) formant des sous-graphes de G. Dans l'exemple disponible en figure 1, les sommets recouverts par les enveloppes annotées $\{1, 2, 3\}$ forment des sous-graphes correspondant à une décomposition simple. Une décomposition hiérarchique apparaît lorsque certaines de ces classes sont à leur tour découpées en sous-classes. C'est le cas dans cet exemple où le sous-graphe annoté 2 est divisé en deux sous-graphes 21 et 22. Cette imbrication fait de l'arbre une représentation naturelle pour les fragmentations hiérarchiques.

On note T un arbre de fragmentation de l'ensemble V. C'est un arbre enraciné tel que chaque nœud $t \in T$ est de degré supérieur ou égal à deux (on parle de *nœud interne*) ou égal à

un (on parle de *feuille*). L'ensemble des feuilles de T est noté $\mathcal{F}(T)$. Dans l'exemple illustré en figure 1, on a $\mathcal{F}(T) = \{1, 21, 221, 222, 3\}$. Chaque nœud $t \in T$ correspond à une communauté $V_t \subset V$. On note $p(t)$ l'ascendant direct de t et $\sigma(t)$ l'ensemble des descendants directs de t dans T. Dans l'exemple, on a $p(22) = 2$ et $\sigma(22) = \{221, 222\}$. Cette relation correspond au fait que $V_t \subset V_{p(t)}$ et que pour chaque nœud $t' \in \sigma(t)$ on a $V_{t'} \subset V_t$.

On note T_t le sous-arbre de T ayant pour nœud racine t. Le graphe induit par l'ensemble de sommets V_t est noté G_t. Dans l'exemple, G_1 est un graphe contenant les sommets sous l'enveloppe notée 1 ainsi que les arêtes ayant leurs deux extrémités dans ce même ensemble. La *hauteur* d'un sommet t dans T désigne le nombre d'arêtes séparant la racine de T et t. On désigne par $N_i(T)$ le i-ème niveau de T c'est-à-dire l'ensemble des feuilles de T auquel on a retiré les nœuds de hauteur supérieure à i. Dans l'exemple en figure 1, on a $N_1(T) = \{1, 2, 3\}$, $N_2(T) = \{1, 21, 22, 3\}$ et $N_3(T) = \mathcal{F}(T)$. Chaque i-ème niveau de T correspond à une décomposition simple de V.

2.2 Évaluation de la qualité multi-échelle

Dans le domaine de la détection de communautés, les mesures de qualité sont souvent utilisées pour comparer plusieurs décompositions simples d'un graphe. Une mesure de qualité Φ peut être vue comme une fonction partant de l'espace des fragmentations possibles d'un ensemble V et allant dans un intervalle réel.

Évaluer la qualité d'une décomposition hiérarchique pose différents problèmes qui sont la prise en compte de l'imbrication des sous-graphes et du niveau auquel une classe intervient. Dans ce cadre, Blanc et al. (2010) ont introduit une mesure exprimée en tant que récursion sur l'arbre de fragmentation et ont ainsi adapté la mesure de Mancoridis et al. (1998) aux décompositions hiérarchiques. On généralise ici l'idée aux mesures de qualité respectant la contrainte d'additivité (Pons et Latapy (2010)).

Définition 1 *Une **mesure de qualité** $\Phi(G, C)$ d'une fragmentation plate $C = (C_1, \ldots, C_k)$ appliquée aux sommets V d'un graphe G est dite **additive** si elle peut être formulée sous la forme*

$$\Phi(G, C) = \sum_{i=1}^{k} \phi(G, C_i) \tag{1}$$

*où $\phi(G, C_i) \in [0, \frac{1}{k}]$ correspond à l'**apport** de la communauté i.*

Comme le soulignent Pons et Latapy (2010), beaucoup de mesures de qualité utilisées aujourd'hui sont additives. L'idée de Blanc et al. (2010) est de parcourir l'arbre de fragmentation en appliquant récursivement la mesure de qualité aux sous-arbres rencontrés. Cette approche est donc possible si on dispose d'une mesure additive.

Définition 2 *Soit $\Phi(G, C)$ une mesure de qualité additive. La généralisation de $\Phi(G, C)$ en **mesure de qualité multi-niveaux** notée $\Phi(G, T; q)$ pour un arbre de fragmentation T de racine r appliqué au graphe G s'exprime*

$$\Phi(G, T; q) = \begin{cases} \sum_{t \in \sigma(r)} \phi(G, V(t)) \left(1 + q \times \Phi(G_t, T_t; q)\right) & \textit{si } \sigma(r) > 0 \\ 0 & \textit{sinon} \end{cases} \tag{2}$$

avec $q \in [0, 1]$.

La mesure $\Phi(G, T; q)$ correspond donc à un polynôme en $q \in [0, 1]$. Une valeur de q proche de 1 donne plus de poids aux nœuds étant hauts dans la hiérarchie. D'un autre côté, prendre $q = 0$ revient à évaluer la qualité de $N_1(T)$.

Remarquons également que la qualité d'une classe (un nœud de T) est pondérée par le produit de la qualité de ses ascendants. Cela correspond à l'idée qu'une classe mal définie, par exemple avec un nombre de connections internes relativement plus faible que son nombre de connections externes, ne pourra engendrer que des classes de mauvaise qualité.

La question de la valeur de q à utiliser reste entière. Dans le cas où il n'y a pas de raisons *a priori* de privilégier une décomposition plus ou moins haute on utilise l'indice noté $\Phi(G, T)$ défini ci-dessous.

Définition 3 *On appelle **indice de qualité multi-niveaux** $\Phi(G, T)$ d'un arbre de fragmentation T l'intégrale du polynôme $\Phi(G, T; q)$ sur toutes les valeurs possibles de q*

$$\Phi(G, T) = \int_0^1 \Phi(G, T; q) dq \tag{3}$$

2.3 Optimisation de la hiérarchie

Les formules 2 et 3 permettent donc de comparer différentes décompositions hiérarchiques et de déterminer laquelle est la mieux adaptée pour le réseau étudié.

Comparer différents arbres de fragmentations implique que l'évaluation du gain obtenu après une modification opérée sur un arbre donné est possible. Ainsi on peut par exemple déterminer si il est opportun de retirer un nœud interne de l'arbre. Cette opération de suppression consiste à remplacer un nœud t de nœud parent p par l'ensemble de ses descendants directs $\sigma(t)$. On notera le gain résultat de cette opération $\Delta_t(T) = \Phi(G, T \setminus \{t\}) - \Phi(G, T)$.

Partant d'un arbre de fragmentation T initial, notre procédure d'optimisation de T consiste à supprimer le nœud interne t (si il existe) ayant le gain $\Delta_t(T)$ positif maximum (l'arbre T doit bien sur être mis à jour à chaque étape).

La suppression d'un nœud t de l'arbre T conduit à plusieurs changements dans le calcul de la mesure multi-niveaux. Premièrement, le poids des descendants directs de t devient plus important puisqu'ils interviennent désormais à un niveau hiérarchique plus bas. Deuxièmement, ces descendants ne sont plus une décomposition simple du graphe induit G_t mais une partie de la décomposition simple de $G_{p(t)}$. Cette dernière observation implique que le calcul de $\Delta_t(T)$ passe notamment par le calcul de $\phi(G_{p(t)}, V_{t'})$ pour tout $t' \in \sigma_t \cup \sigma_{p(t)}$. Prenons l'exemple de la figure 1, la suppression du nœud noté 22 fait que les nœuds 221 et 222 deviennent descendants directs du nœud 2. Dans ce cadre, le nombre d'arêtes sortant des sous-graphes 221 et 222 augmente puisqu'il faut y inclure les arêtes allant vers le sous-graphe 21.

3 Application à la maximisation multi-échelle de la modularité

Nous présentons dans cette section l'algorithme de Blondel et al. (2008) qui produit une décomposition hiérarchique de graphe. Nous verrons en particulier que cette dernière peut comporter des biais de construction que notre méthode permet de corriger.

3.1 Fonctionnement de l'algorithme

L'algorithme de Blondel et al. (2008) est une heuristique d'optimisation de la modularité (Newman (2006)). Pour une décomposition simple C, cette mesure de qualité est définie de la façon suivante :

$$Q(G,C) = \sum_{t=1}^{k} \frac{e_t}{|E(G)|} - \left(\frac{d_t}{2|E(G)|}\right)^2 \tag{4}$$

où e_t désigne le nombre d'arêtes internes à la communauté t, d_t la somme des degrés des sommets et $E(G)$ est l'ensemble des arêtes de G. Il est aisé de vérifier que $Q(G,C)$ est une mesure additive. Ici $\phi(G,V_t)$ correspond à la différence entre la proportion observée d'arêtes internes à la communauté V_t et cette même quantité dans le cas d'un graphe aléatoire avec la même distribution de degrés.

L'algorithme comprend deux étapes. Après avoir placé les sommets dans des communautés distinctes, on parcourt, pour chaque sommet, les communautés présentes dans son voisinage direct en calculant le gain de modularité résultant de l'affectation du sommet dans ces dernières. Le sommet est affecté à la communauté pour laquelle le gain de modularité est le plus grand (il est donc retiré de sa précédente communauté). Cette opération est répétée tant qu'un accroissement de la modularité est possible. La deuxième phase consiste à remplacer chaque communauté par des sommets. Deux "méta-sommets" i et j sont reliés entre eux par des "méta-arêtes" auxquelles on affecte comme poids la somme des poids des arêtes reliant les sommets de i et j. Le graphe obtenu, appelé communément *graphe quotient*, devient le nouveau graphe initial. Les deux phases sont alors répétées jusqu'à atteindre un maximum de la modularité.

3.2 Discussion sur la hiérarchie produite

L'algorithme de Blondel et al. (2008) produit donc une hiérarchie T en appliquant une méthode de décomposition simple à un graphe quotient construit à partir de la décomposition précédente. Chaque niveau de la hiérarchie correspond à un maximum local de la modularité. Les auteurs estiment que le niveau le plus significatif est le dernier identifié $N_1(T)$ puisqu'il correspond au maximum de la modularité. Ils soulignent cependant l'intérêt que peuvent avoir les autres niveaux ainsi que la hiérarchie complète.

La question est de savoir si cette hiérarchie est pertinente pour étudier la structure du réseau étudié. Notons tout d'abord que la première phase de l'algorithme est non-déterministe. En effet, la décomposition simple résultant de cette phase dépend de l'ordre dans lequel on parcourt les sommets.

Prenons l'exemple illustré en figure 1, imaginons qu'une première passe de l'algorithme mène à une identification des sous-graphes $\{1, 21, 221, 222, 3\}$ qui constituent notre premier niveau de fragmentation. Il est possible que dans une seconde passe les sommets 221 et 222 soient regroupés laissant les autres isolés alors que le regroupement des trois sommets 221, 222 et 21 donne une modularité plus grande. Si on examine *a posteriori* l'arbre de fragmentation obtenu, on peut alors dire que le nœud 22 ne correspond qu'à une étape de construction et ne reflète pas la véritable structure du réseau. Même dans le cas contraire, le regroupement en deux temps des sous-graphes 221, 222 et 21 peut se révéler inopportun si on évalue globalement la qualité de la décomposition hiérarchique obtenue.

Cet exemple illustre la nécessité d'appliquer notre méthode à la hiérarchie produite par l'algorithme décrit dans cette section. On cherche donc à filtrer cette hiérarchie en utilisant la procédure décrite dans la section 2.3 et en utilisant comme apport de la classe t :

$$\phi(G, V_t) = \frac{e_t}{|E(G)|} - \left(\frac{d_t}{2|E(G)|}\right)^2 \tag{5}$$

4 Résultats

4.1 Évaluation sur des graphes aléatoires

Pour valider notre méthode, nous utilisons le *benchmark* LFR proposé par Lancichinetti et Radicchi (2008) et étendu au cas multi-niveaux par la suite (Lancichinetti et al. (2011)). Plusieurs travaux sur la fragmentation hiérarchique de réseaux utilisent cette méthode pour évaluer la qualité des algorithmes (voir par exemple Rosvall et Bergstrom (2011)).

Ce *benchmark* permet la génération de graphes possédant une distribution des degrés en loi de puissance et une structure de communautés à deux niveaux. Ces deux niveaux correspondent à des macro-communautés découpées en micro-communautés. Le nombre d'arêtes et de communautés ainsi fixés, le *benchmark* propose de faire varier la cohésion de ces communautés à chacun des deux niveaux de façon à rendre la hiérarchie plus ou moins identifiable. On utilise ainsi deux paramètres μ_1 et μ_2 qui correspondent à la proportion d'arêtes entre macro-communautés (respectivement micro-communautés issues de la même macro-communauté).

La performance des algorithmes à bien détecter les deux niveaux de fragmentation peut être évaluée en utilisant l'information mutuelle normalisée telle que définie par Danon et al. (2005). Cette mesure évalue la similarité entre deux partitions d'un même ensemble aboutissant à un score compris entre 0 (les deux partitions sont complètements différentes) et 1 (les partitions sont identiques).

Nous fixons le nombre de sommets à 10000 avec une distribution de degré d'exposant 2 avec une moyenne de 20 et un maximum de 100. La taille des macro-communautés est comprise entre 400 et 4000 sommets et celle des micro-communautés entre 10 et 100.

La figure 2 détaille les résultats obtenus. Pour chaque graphique, l'axe des abscisses correspond à la valeur de $\mu_1 + \mu_2$ c'est-à-dire la proportion d'arêtes externes aux communautés. Pour chaque valeur de μ_1, on fait varier μ_2 dans $[\mu_1, 1]$. L'axe des ordonnées représente l'information mutuelle entre les fragmentations comparées. On compare les micro-communautés réelles aux fragmentations $N_2(T)$ et $\mathcal{F}(T)$ (courbes oranges et rouges respectivement) et les macro-communautés à la fragmentation $N_1(T)$ (courbes bleues). Les résultats correspondent à une moyenne sur cent simulations.

Nous analysons tout d'abord la décomposition hiérarchique obtenue avec l'algorithme de Blondel et al. (2008) sans optimisation (colonne de gauche dans la figure 2). Les micro-communautés sont assez bien identifiées à partir du moment où la proportion d'arêtes entre ces sous-graphes est faible $\mu_2 < 0.5$. Il en est de même pour les macro-communautés à condition que la mixité entre micro-communautés soit bien supérieure à la mixité entre macro-communautés. On constate en revanche que les arbres de fragmentation produits par l'algorithme contiennent souvent des niveaux supplémentaires. Dans le cas idéal, $N_2(T)$ devrait correspondre à $\mathcal{F}(T)$ et aux micro-communautés or cela n'est pas le cas ici.

Cette dernière observation confirme les risques évoqués en section 3.2. Un décalage peut se créer entre les différentes exécutions de l'algorithme, ce qui entraîne l'ajout dans la hiérarchie d'étapes intermédiaires non pertinentes.

L'analyse des résultats obtenus en considérant les décompositions optimisées par notre méthode (colonne de droite dans la figure 2) montre que ces groupements intermédiaires sont supprimés et que $N_2(T)$ correspond bien aux micro-communautés. De plus, la similarité entre $N_1(T)$ et les macro-communautés ne change pas, de même pour la similarité entre $\mathcal{F}(T)$ et les micro-communautés. Cela signifie que l'on ne supprime pas à tort certains regroupements.

4.2 Exemples réels

Nous analysons maintenant les résultats obtenus sur des réseaux réels. Nous confrontons ces derniers aux décompositions hiérarchiques produites par les algorithmes *Oslom* proposé par Lancichinetti et al. (2011) et *Infomap* proposé par Rosvall et Bergstrom (2011). Notons que ces deux algorithmes sont également non-déterministes. En particulier l'algorithme *Oslom* produit des fragmentations hiérarchiques et chevauchantes *i.e.* un sommet peut être inclus dans plusieurs communautés. Ce dernier cas de figure n'apparaît toutefois que rarement dans les deux exemples suivants.

4.2.1 Réseau de collaboration

Nous nous intéressons à un graphe de co-publications dans le domaine de l'analyse de réseaux (voir Fortunato (2010)). Il comprend 515 auteurs (sommets) et 1318 relations.

Les algorithmes *Oslom* et *Infomap* identifient au premier niveau des sous-graphes de grande taille (une centaine d'individus). Bien que facilement séparables du reste du réseau, ces sous-graphes ne sont pas très denses, ils comportent notamment beaucoup de composantes biconnexes.

La décomposition hiérarchique obtenue par l'algorithme de Blondel et al. (2008) sans optimisation donne des résultats semblables. Notre méthode d'optimisation supprime ce premier niveau pour aboutir à des sous-graphes de diamètre plus faible pouvant correspondre à des collaborations proches entre équipes. Une visualisation est disponible en figure 3. Les niveaux $N_1(T)$ et $N_2(T)$ y sont dessinés à l'aide d'enveloppes concaves bleues et grises respectivement.

Ce genre d'observations rejoint la remarque faite par Blondel et al. (2008) : la hiérarchie fournit une alternative au problème de *résolution limite* observé notamment par Fortunato et Barthélemy (2007) et Good et al. (2010). En effet, l'optimisation directe de la modularité peut mener à des agrégations excessives de plusieurs communautés. Dans notre cas, il est possible de déterminer si cette opération est justifiée ou non.

Comme on peut le voir dans la figure 3, le niveau $N_2(T)$ (enveloppes grises) rassemble des sous-graphes globalement de petite taille qui semblent correspondre aux collaborations entre collègues. On retrouve également ce type de regroupement dans le dernier niveau fourni par *Infomap*.

4.2.2 Réseau de migrations

Le réseau auquel on s'intéresse maintenant représente des flux migratoires au sein des États-Unis (voir Cui et al. (2008)). Les 1650 sommets correspondent à des comtés reliés entre eux avec des arêtes pondérées par le nombre de personnes étant parties du comté A pour s'installer dans le comté B entre 1995 et 2000. On compte au total environ 6500 arêtes dans ce réseau.

Les résultats sont illustrés dans la figure 4. Les deux premiers niveaux de la fragmentation y sont présentés en utilisant un *mapping* de couleur car la position des sommets correspond ici aux coordonnées géographiques des comtés. Nous retrouvons la composante géographique également présente dans ceux obtenus en utilisant *Oslom* et *Infomap*, à savoir que les comtés proches géographiquement sont susceptibles d'appartenir à la même communauté.

Infomap ne détecte pas de structure hiérarchique dans ce réseau. Les plus grosses communautés identifiées correspondent à la Californie, au Texas et à l'est du pays. A l'inverse, *Oslom* propose une fragmentation hiérarchique assez haute commençant par deux grandes communautés se situant dans les zones de l'ouest/*Mid-West* et de l'est des États-Unis. Ces deux très grandes communautés sont ensuite découpées sur deux niveaux pour arriver à des classes semblables à celles détectées par *Infomap*.

Les résultats obtenus par notre méthode sur cet exemple offrent un bon compromis. En effet, le premier niveau (voir figure 4(a)) correspond à des zones relativement larges tandis que le second niveau (voir figure 4(b)) rejoint bien la décomposition proposée par *Infomap*.

5 Conclusion

Nous avons introduit une procédure de post-traitement d'un arbre de fragmentation permettant d'améliorer la qualité de la décomposition en supprimant certains nœuds internes. La méthode a été appliquée à l'algorithme de Blondel et al. (2008) en utilisant une généralisation de la modularité au cadre multi-niveaux. Des tests réalisés sur des graphes aléatoires montrent que la hiérarchie optimisée est bien plus proche de la vraie configuration du réseau. Bien que nous n'ayons pas pu tous les présenter ici, les résultats obtenus sur plusieurs exemples réels sont prometteurs.

On peut cependant noter que notre méthode de filtrage hiérarchique ne permet pas de déterminer si il est opportun de conserver ou non les feuilles d'un arbre de fragmentation. Il faudrait pour cela déterminer si une décomposition est préférable à l'absence de décomposition, en utilisant par exemple un seuil minimum de modularité. Cependant, la gestion de ces sous-graphes est moins problématique. En effet, du point de vue de l'analyse, on peut supposer que les séparations correspondant aux premiers niveaux contiennent plus d'informations. Par exemple, on s'attend à ce qu'une bonne décomposition fasse apparaître les composantes connexes du réseau au premier niveau de la hiérarchie.

Références

Blanc, C., M. Delest, J.-M. Fédou, G. Mélançon, et F. Queyroi (2010). Évaluer la qualité d'une fragmentation de graphe multi-niveaux. In *Journées MARAMI 2010*, Toulouse, France.

Blondel, V. D., J.-L. Guillaume, R. Lambiotte, et E. Lefebvre (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment 2008*, P10008.

Cui, W., H. Zhou, H. Qu, P. Wong, et X. Li (2008). Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics 14*(6), 1277–1284.

Danon, L., A. Diaz-Guilera, J. Duch, et A. Arenas (2005). Comparing community structure identification. *Journal of Statistical Mechanics : Theory and Experiment 2005*, P09008.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports 486*(3-5), 75–174.

Fortunato, S. et M. Barthélemy (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences 104*(1), 36.

Good, B., Y. De Montjoye, et A. Clauset (2010). Performance of modularity maximization in practical contexts. *Physical Review E 81*(4), 46106.

Lancichinetti, A. et F. Radicchi (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E 78*(4), 046110.

Lancichinetti, A., F. Radicchi, et J. Ramasco (2011). Finding statistically significant communities in networks. *PloS one 6*(4), e18961.

Mancoridis, S., B. Mitchell, C. Rorres, Y. Chen, et E. Gansner (1998). Using automatic clustering to produce high-level system organizations of source code. In *Proceedings of the 6th International Workshop on Program Comprehension*, pp. 45–52. IEEE.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences, USA 103*, 8577–8582.

Pons, P. et M. Latapy (2010). Post-processing hierarchical community structures : Quality improvements and multi-scale view. *Theoretical Computer Science 412*, 892–900.

Pumain, D. (Ed.) (2006). *Hierarchy in Natural and Social Sciences*, Volume 3 of *Methodos Series*. Springer.

Rosvall, M. et C. Bergstrom (2011). Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PloS one 6*(4), e18209.

Simon, H. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society 106*(6), 467–482.

Summary

Many graph clustering algorithms perform successive divisions or aggregations of subgraphs leading to a hierarchical decomposition of the network. An important question in this domain is to know if this hierarchy reflects the structure of the network or if it is only an artifice due to the conduct of the procedure. We propose a method to validate and, if necessary, to optimize the multi-scale decomposition produced by such methods. We apply our approach on the algorithm proposed by Blondel et al. (2008) based on the maximisation of the modularity. In this context, a generalization of this measure of quality in the multi-level case is introduced. We test our method on random graphs and on real examples from various fields.

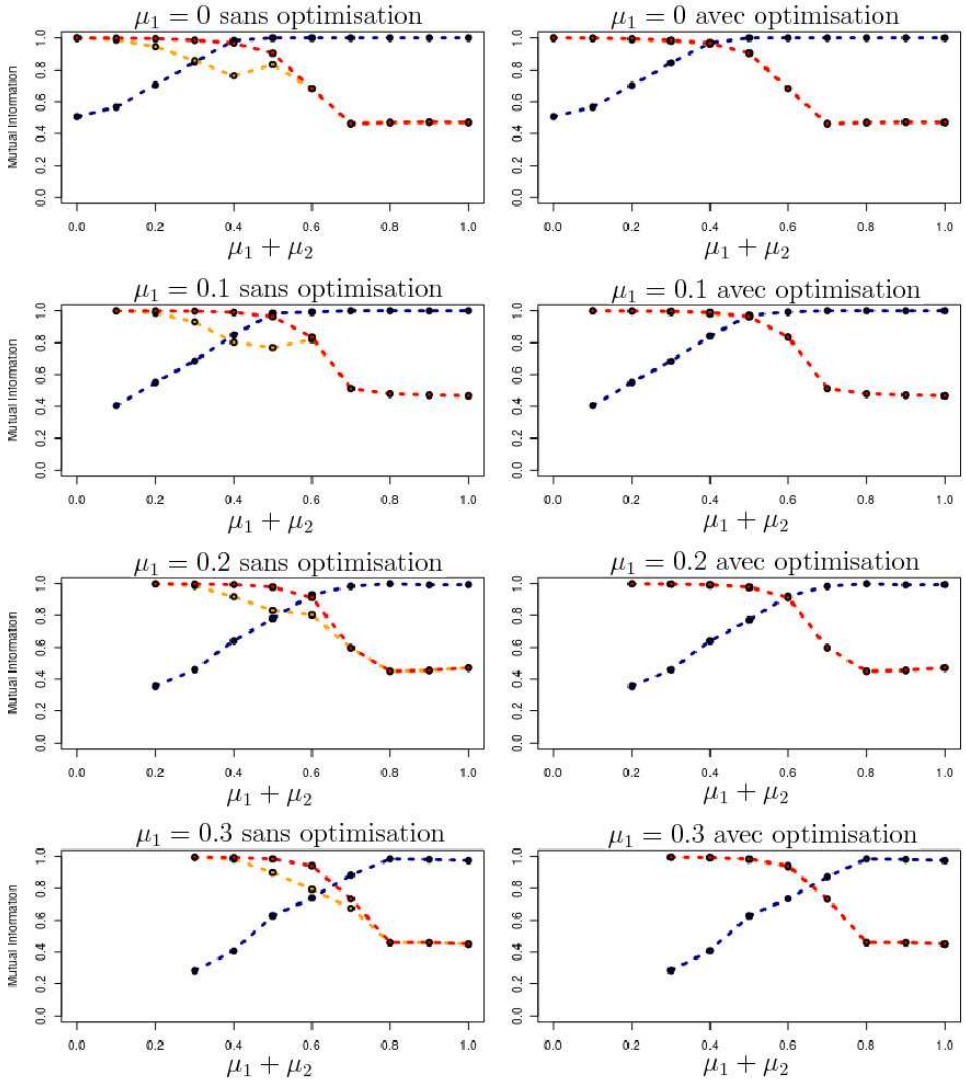


FIG. 2 – ***Tests sur le benchmark LFR hiérarchique*** *pour différentes valeurs de μ_1 et μ_2. La courbe bleue correspond à l'information mutuelle entre $N_1(T)$ et les vraies macro-communautés. La courbe rouge entre $\mathcal{F}(T)$ et les vraies micro-communautés. Enfin, la courbe orange entre $N_2(T)$ et les vraies micro-communautés.*

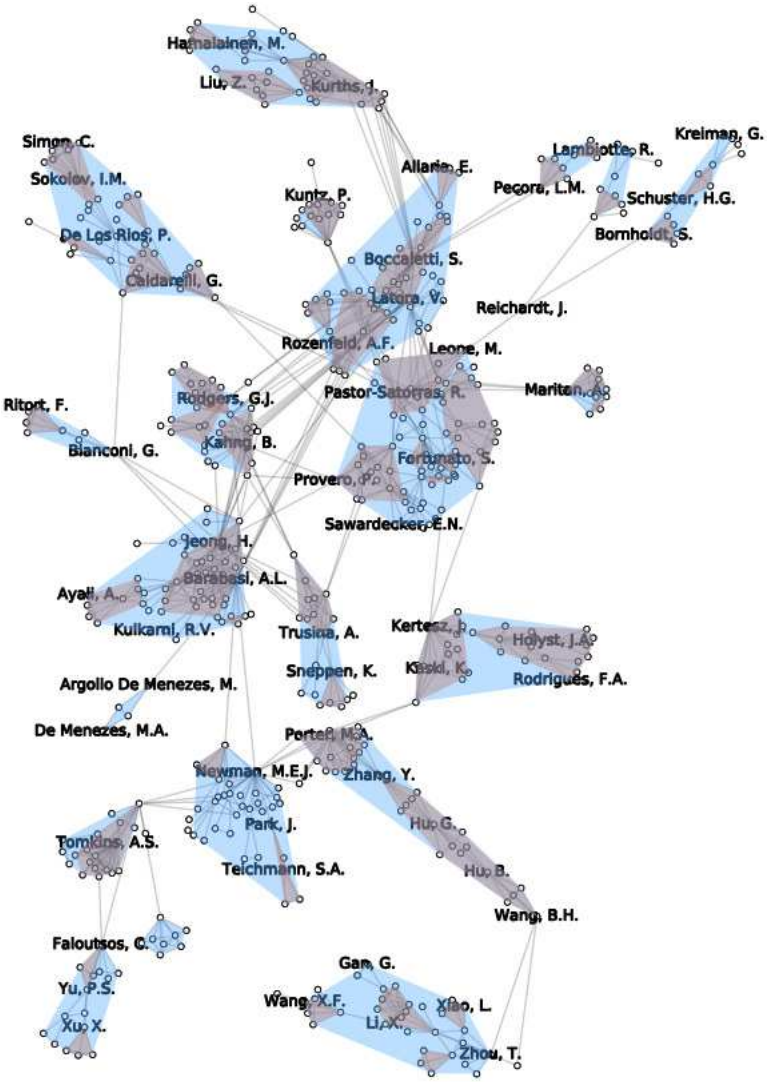
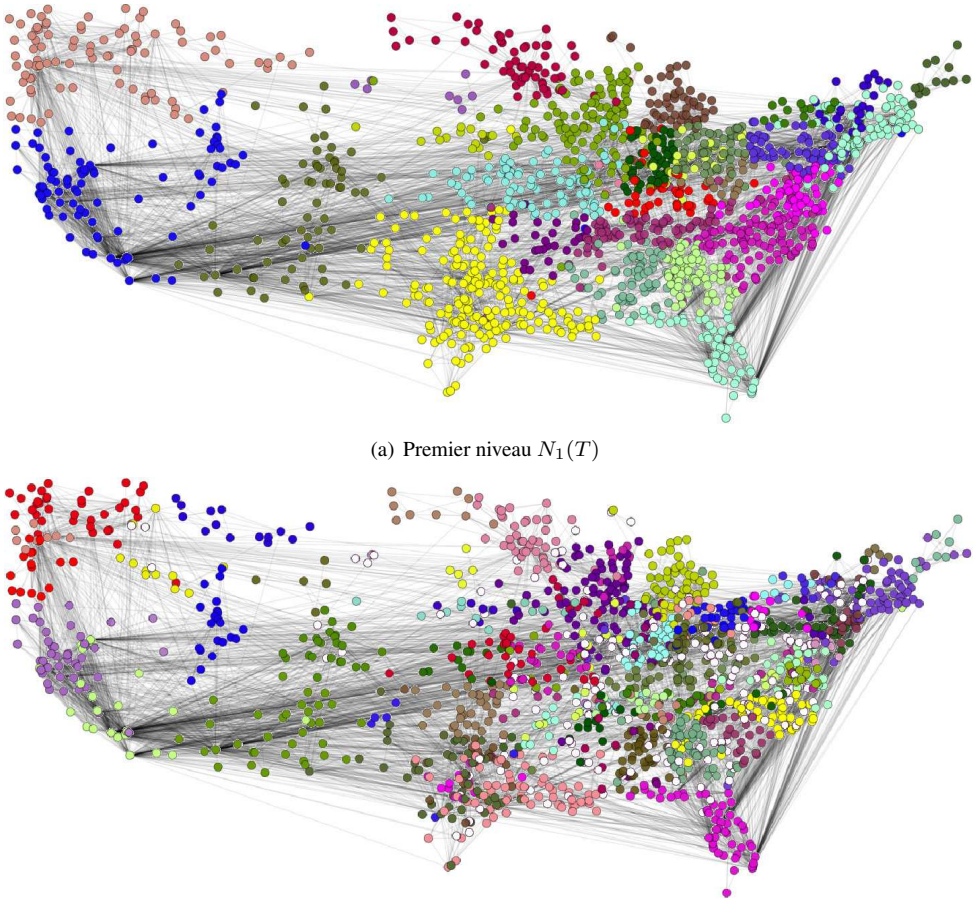


FIG. 3 – ***Résultats sur le réseau de collaboration scientifique.*** *Les enveloppes bleues (respectivement grises) correspondent aux communautés du niveau $N_1(T)$ ($N_2(T)$ respectivement).*



(a) Premier niveau $N_1(T)$

(b) Second $N_2(T)$

FIG. 4 – ***Résultats sur le réseau de migrations (États-Unis).*** *La position relative des sommets correspond aux coordonnées géographiques des comtés qu'ils représentent. Les sommets de même couleur appartiennent à la même classe. Les sommets blanc sont des sommets isolés.*