Optimizing a hierarchical community structure of a complex network

François Queyroi

Optimizing a hierarchical community structure of a complex network

François Queyroi

Abstract 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 procedure to the algorithm proposed by Blondel *et al.* (2008) based on modularity maximization. In this context, a generalization of this quality measure in the multi-level case is introduced. We test our method on random graphs and real world examples.

1 Introduction

A central task of network analysis is the detection of a community structure[3]. Many graph clustering algorithms have been developed to fulfill this task (see [6] for a survey). These methods often rely on the maximization of a quality measure like modularity[12].

Previous works in human sciences[16, 14] suggest however the presence of a hierarchical structure in complex systems such as networks. Several strategies have been used to discover such hierarchies by iteratively grouping or splitting groups. Good examples are algorithms based on a similarity metric. At each iteration the two closest groups (in term of similarity) are merged leading to the construction of a hierarchy. However, the resulting hierarchy is barely relevant for an analyst because a level is the division of only one single group. In a recent paper [13], Pons and Latapy provide a procedure to simplify this kind of structure.

Other algorithms directly lead to workable hierarchies[10, 15]. Blondel *et al.*[2] introduced a flat clustering procedure that relies on the construction of a hierarchy

François Queyroi
Universiy of Bordeaux, CNRS, LaBRI – France, e-mail: francois.queyroi@labri.fr

of clusters. The identification of a hierarchy is not the final objective of the algorithm although many clues suggest that this hierarchy is meaningful to analyse the structure of the studied network.

This paper focuses on the validation of such hierarchies. We provide an optimization procedure allowing to filter out undesirable fusion of clusters. We enforce this post-procedure on the results produced by the algorithm of Blondel *et al.*[2]. For this purpose we introduce a generalisation of the modularity quality measure in order to quantify the quality of a hierarchical clustering.

The rest of the paper is organized as follows. In section 2, we introduce the approach used to evaluate the quality of a hierarchical community structure. In section 3, we provide an application of our approach to the algorithm of Blondel *et al.*. In section 4, we show that our procedure is efficient by describing some results obtained on a hierarchically clustered graph benchmark and on real world examples. We compare our results to those produced by two different state-of-the-art procedures[10, 15].

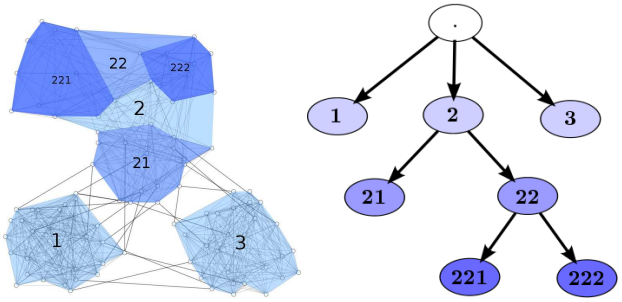
2 Optimizing a hierarchical community structure

2.1 Definitions

Given a graph $G = (V,E)$ where V is the set of vertices and E the set of edges. A flat clustering of G is a partition of the vertices V in several groups (also called *communities* when they are densely connected) defining a set of induced subgraphs of G. In the example provided in Figure 1(a), the vertices falling into the hulls labelled $\{1,2,3\}$ correspond to three subgraphs. A hierarchical clustering appears when some of these communities are recursively divided into subgroups. For example, the subgraph labelled 2 is divided into two subgraphs 21 and 22. The nesting between groups of vertices at different levels makes trees an efficient way to model hierarchical clusterings (see Figure 1(b)). We call *clustering trees* such structures.

Let T be a clustering tree of the vertices set V. It is a rooted tree where each node $t \in T$ can be either an *internal* node if its degree $d(t) \geq 2$ or a *leaf* node if $d(t) = 1$. The set of leaves of T is denoted $\mathscr{F}(T)$. In the previous example we have $\mathscr{F}(T) = \{1,21,221,222,3\}$. Each node $t \in T$ corresponds to a subset $V_t \subset V$. Let $p(t)$ be the direct ancestor of t and $\sigma(t)$ the set of direct successors of t. In the example, we have $p(22) = 2$ and $\sigma(22) = \{221,222\}$. These relations correspond to the following constraints : for each node t, we have $V_t \subseteq V_{p(t)}$ and $V_t = \bigcup_{c\in\sigma(t)} V_c$ if t is *internal*.

We denote by T_t the subtree of T rooted in t and by G_t the subgraph induced by the vertices set V_t. In the example, G_1 is a graph which contains the vertices that fall into the hull labelled 1 and the edges having both extremities in the same



(a) A hierarchical clustered graph (b) A labelled clustering tree

Fig. 1 An example of hierarchical clustering of a graph (left) modelled using a clustering tree (right).

set. The *height* of a node t in T is the number of edges between the root of T and t. We denote by $N_i(T)$ the i-th level of T which is the set of *leaves* in the subtree $T \setminus \{t \in T, h(t) > i\}$. In the example given in Figure 1 we have $N_1(T) = \{1,2,3\}$, $N_2(T) = \{1,21,22,3\}$ et $N_3(T) = \mathscr{F}(T)$. Each level $N_i(T)$ of T is a flat clustering of the set V.

2.2 Evaluating a hierarchical community structure

To identify a community structure in a network, *quality measures* are often used in order to compare different flat clusterings of a graph. A *quality measure* Φ is a function having as domain the set of all flat clusterings and as range a real interval. Evaluating the quality of a hierarchical clustering is far more problematic because we have to take the nesting and the height of the clusters into account. To fulfill this task, Blanc *et al.*[1] introduced a recursively defined measure that generalized the Mancoridis criteria[11] to hierarchical clusterings. The same idea is used here for all measures respecting the additivity constraint[13].

Definition 1. A **quality measure** $\Phi(G,C)$ of a flat clustering $C = (C_1,\ldots,C_k)$ for the set of vertices V of a G is said to be **additive** if it can be written

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

where the function $\phi(G,C_i) \in [0,\frac{1}{k}]$ is called the *gain* of the community i.

Most of the existing quality measures are additive[13]. The idea underlying the extension of quality measures to hierarchical clusterings is the recursive call of an additive quality measure on each internal node of the clustering tree.

Definition 2. Given $\Phi(G,C)$ an additive quality measure, its extension to a hierarchical clustering tree T rooted in r is denoted $\Phi(G,T;q)$ and is defined as follows:

$$\Phi(G,T;q) = \begin{cases} \sum_{t\in\sigma(r)} \phi(G,V_t)\,(1+q\times\Phi(G_t,T_t;q)) & \text{if } \sigma(r)>0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

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

The measure $\Phi(G,T;q)$ is a polynomial with a variable $q \in [0,1]$. On one hand, the weight of an internal node at the bottom of the hierarchy increases when q is close to 1. On the other hand, we have $\Phi(G,T;q) = \Phi(G,N_1(T))$ for $q = 0$.

Note that the quality of a community (a node of T) is weighted by the product of the quality of its ancestors. This weight corresponds to the idea that a badly defined community (with an external density greater than its internal density for example) can only generate badly defined sub communities (see [1] for further details).

Definition 3. We denote by **hierarchical quality index** of a clustering tree T, the function $\Phi(G,T)$ which is the integral of the polynomial $\Phi(G,T;q)$ for $q \in [0,1]$:

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

The value of q to use is an open issue. When there is no reason to promote or penalize deep hierarchies, the criteria $\Phi(G,T)$ shall be used.

2.3 Hierarchy quality optimization

The formula 2 and 3 can be used to compare different hierarchical clusterings and select the best one for a given network. Therefore we are able to access the relevance of a modification applied on a hierarchical clustering. We can for example determine whether or not a given node in the clustering tree should be removed. The removal of a node t is the replacement of t by its successors $\sigma(t)$. We denote as $\Delta_t(T) = \Phi(G,T\setminus\{t\}) - \Phi(G,T)$ the quality variation due to this modification. Given an initial clustering tree T, our optimization procedure can be defined as the iterative suppression of an internal node t (if it exists) maximizing $\Delta_t(T)$ with $\Delta_t(T) > 0$.

The removal of a node $t \in T$ results in several modifications in the multilevel quality measure computation. First, the weights of all nodes in the subtree T_t are greater because the depth of the clusters this subtree contains are now smaller in T. Secondly, the nodes of the set $\sigma(t)$ do not longer correspond to a flat clustering of the subgraph G_t but are new parts of the flat clustering of the subgraph $G_{p(t)}$. Looking at the previous example in Figure 1, after the deletion of the node 22, the nodes 221 and 222 are now direct successors of the node 2. Therefore, the number of edges leaving 221 and 222 increases because the edges between these clusters

and the cluster 21 are added.

The complexity of our procedure is $O(|T|^3)$ where $|T|$ is the number of nodes in the clustering tree T. First, we assume that the gain function ϕ can be computed in constant time. This can be achieved by keeping some information into memory (the number of internal/external edges for example). Secondly, the function Φ is computed in $O(|T|)$ as a simple depth-first search over the clustering tree. Finally, the procedure described above lies in the family of greedy algorithms.

3 Application to modularity maximization

In this section, we present the algorithm of Blondel *et al.*[2] which produces a hierarchical clustering of a graph. We then illustrate the fact that the hierarchies produced may contain some irrelevant groups. These observations justify the use of our method.

3.1 Algorithm description

The algorithm of Blondel *et al.* is a modularity maximization heuristic. The modularity can be defined as follows:

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

where e_t is the number of edges having both ends in the cluster t, d_t is the sum of the degrees of nodes belonging to the cluster t and M is the number of edges in G. We can easily prove that $Q(G,C)$ is additive. The gain $\phi(G,V_t)$ is here the difference between the observed proportion of internal edges in V_t and its theoretical value in a random graph with the same degree distribution.

At the beginning of the algorithm, each vertex corresponds to a single community. The algorithm has two major phases. First, we seek for each vertex the communities that lie in its direct neighbourhood and compute the potential increase of modularity resulting of assigning the vertex to each of them. The vertex is then assigned to the community that maximize the gain (ties are broken randomly). This phase is repeated as long as an increase of the modularity is possible and results in a flat clustering of the graph. Secondly, we replace the previous graph by the quotient graph computed using the previous clustering. These two phases are iteratively repeated as long as the modularity increases.

3.2 Discussion on the resulting hierarchy

The algorithm of Blondel *et al.* produces a hierarchy T by iteratively applying a flat clustering procedure (the first phase described above) to the quotient graph created at the previous iteration. Each level of T can be seen as a local maximum of the modularity. The authors suggest that the last level found $N_1(T)$ is the most meaningful since it corresponds to the highest modularity reached.

This algorithm is very popular in social network analysis because it can be applied on very large graphs while providing clusterings with high modularity values. We can however hardly determine whether or not the hierarchy is meaningful to analysis a given network. We provide here two major issues.

First, the hierarchy may contain irrelevant intermediate clusters. Note that the first phase of the algorithm is nondeterministic because the resulting flat clustering depends on the order in which the vertices are taken. We illustrate this issue using the example given in Figure 1. The first iteration of the algorithm leads to the detection of the communities $\{1,21,221,222,3\}$ (the last level of the final hierarchy). At the second iteration the communities 221 and 222 are grouped leaving the others isolated even if grouping 221, 222 and 21 would lead to a greater modularity. Looking at the final clustering tree, we could say that the cluster 22 is just a building step and is therefore irrelevant.

Secondly, the direct optimization of modularity can lead to the excessive aggregation of several communities. This issue is called the *resolution limit* (see a description in [7] and experimental illustrations in [8]). Blondel *et al.* actually discussed the fact that the hierarchy can be seen as an alternative to this issue. The excessive aggregations occur at the first levels of the hierarchy. While the flat modularity gain may be small (but still positive) remember that the multilevel quality measure we provide takes the whole hierarchy into account. Top level clusters can therefore be removed if their contribution is not strong enough to justify an additional level.

These issues illustrate the usefulness of our method when applied to the hierarchy produced by the algorithm described in this section. We therefore use the procedure described in Section 2.3 using in Eq. 3

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

as the *gain* of the community indexed by t in T.

4 Results

In this section, we discuss the results of several experiments. First, we show that our procedure is able to detect irrelevant intermediate clusters using benchmark graphs where a two level hierarchical clustering is known. Secondly, we provide results of

our procedure when it is applied on real world networks. These results seem reasonable when compared to other state-of-the-art hierarchical clustering algorithms.

4.1 Validation on random graphs with a known hierarchical clustering

In order to validate our method we use the LFR-*benchmark*[9] extended to hierarchical clustering[10]. This benchmark is used in order to evaluate the effectiveness of clustering algorithms (see [15] for example).

We generate graphs with a power law degree distribution and a two-level community structure. These levels are denoted *micro*-communities and *macro*-communities. We can decide how well the communities are defined (in term of density). This is achieved by using two parameters μ_1 and μ_2 which correspond to the proportion of edges between *macro*-communities and the proportion of edges between *micro*-communities lying in the same *macro*-communities respectively. The graphs have 10000 vertices with an average degree of 20 and a maximum degree of 100. The size of *macro*-communities and *micro*-communities are in the range $[400, 4000]$ and $[10, 100]$ respectively.

We evaluate how well the given multilevel structure is identified by using the normalized mutual information[5]. This measure is used to access the similarity between two partitions of the same set. The result is a score between 0 (the partitions are completely different) and 1 (the partitions are the same).

The results are given in Figure 2. The x-axis corresponds to the value $\mu_1 + \mu_2$ which is the proportion of edges outside *micro*-communities. For four different values of μ_1, we compare the different clusterings for $\mu_2 \in [\mu_1, 1 - \mu_1]$. The y-axis corresponds to the normalized mutual information between the compared clusterings. We compare the real *micro*-communities to the clustering given by $N_2(T)$ and $\mathscr{F}(T)$ (orange and red curves respectively) and the real *macro*-communities to the clustering given by $N_1(T)$ (blue curves). The results reported here correspond to an average on one hundred samples.

First, we analyze the results obtained using the algorithm without optimization (left column in Figure 2). The *micro*-communities seem to be identified when they are well defined theoretically. This situation occurs when $\mu_2 < 0.5$. *Macro*-communities are also identified when the proportion of edges between *micro*-communities is superior to the proportion of edges between *micro*-communities. We can however observe that the clustering trees produced by the algorithm contain additional levels. Indeed the flat clustering $N_2(T)$ should be equal to $\mathscr{F}(T)$ (the *micro*-communities) but it is obviously not always the case here. This last observation confirms the risks outlined in Section 3.2.

Looking now at the results obtained using our method (right column in the Figure 2), we can see that the intermediate levels are removed and that the flat clustering $N_2(T)$ is almost always equal to the *micro*-communities. Moreover, the similarities

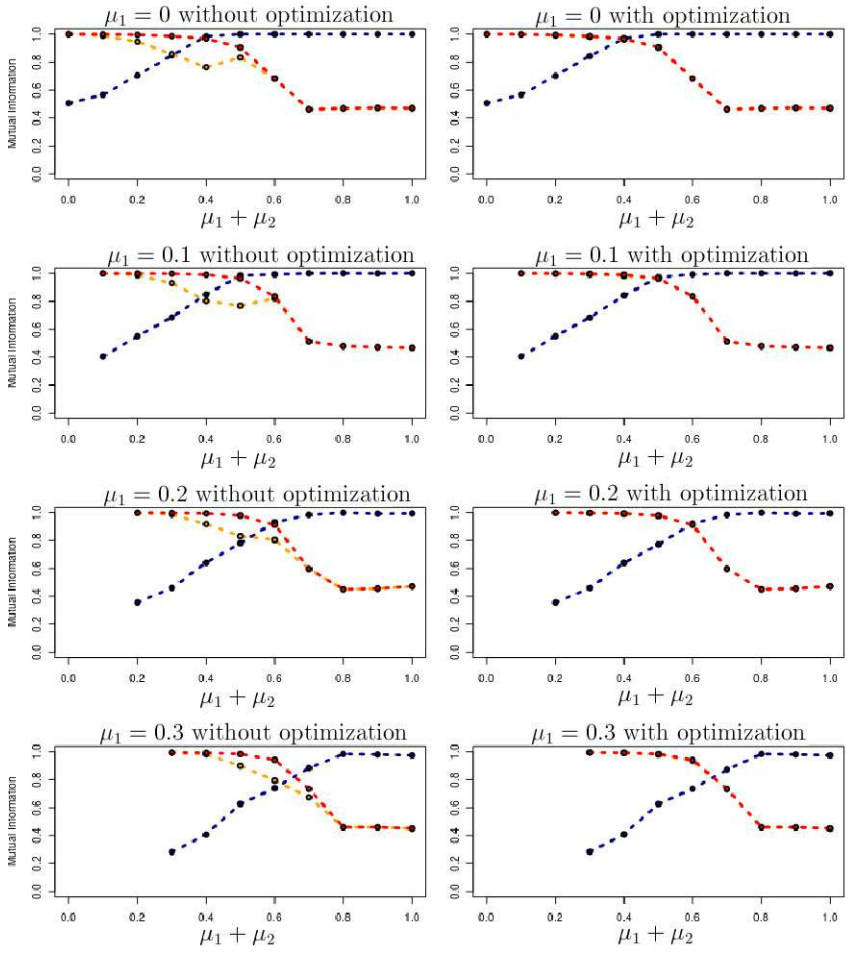


Fig. 2 Evaluation on the multilevel LFR Benchmark for different values of μ_1 and μ_2. The blue line corresponds to the mutual information between $N_1(T)$ and the real *macro*-communities. The red one between $\mathscr{F}(T)$ and the real *micro*-communities. Finally, the orange one between $N_2(T)$ and the real *micro*-communities

between $N_1(T)$/*macro*-communities and between $\mathscr{F}(T)$/*micro*-communities do not change. It means that we do not remove wrongly some clusters.

4.2 Real world examples

We now present some results of our method when applied to real world networks. We compare our resulting hierarchical clustering to the hierarchical clusterings produced by the *Oslom*[10] and *Infomap*[15] algorithms. Note that these algorithms are also nondeterministic.

4.2.1 Co-publication network

We first look at a co-publication network in social network analysis (see [6] for more details). The graph contains 515 authors (vertices), two authors are linked when they are co-authors in at least one paper. The graph contains 1318 edges.

This kind of network is conductive to the presence of some hierarchical community structure. Indeed, the top level of such hierarchy could correspond to people in the same university/institute while the bottom level could correspond to groups formed by Professors/Ph.D. students.

The *Oslom* and *Infomap* algorithms detect big clusters at the top level (with over a hundred people). While these clusters can be easily separated from the rest of the network by removing a couple of edges, they are not densely connected. In particular they contain a lot of biconnected components.

The results obtained using the algorithm of [2] without our method provides similar results. Using our procedure, the first level is removed and contains subgraphs with a small graph diameter that may correspond to close collaboration within same research teams. A visualization of the results is given in Figure 3. The clusterings $N_1(T)$ and $N_2(T)$ are drawn using blue and grey concave hulls respectively.

4.2.2 Migration Network

We now investigate the hierarchical structure which can be found in a migration network. The graph models migration flows in USA (see [4] for details on this dataset). The 1650 vertices represent American counties. For each couple of counties we know the number of person who moved from one to the other between 1995 and 2000. There is a total of 6500 positive relations in this network which are represented as weighted directed edges.

A visualisation of the results is provided in Figure 4 where counties are geolocalized. The two first hierarchical levels are drawn using a color mapping. Both levels illustrate the following observation: geographically close counties are more likely to be part of the same clusters. This observation can be also found in the results obtained using *Oslom* and *Infomap* algorithms.

The *Infomap* algorithm does not find any hierarchical structure in this network. The biggest identified communities correspond to California, Texas and the East of the country. On the opposite, the *Oslom* algorithm provides a deep hierarchical clustering tree. The first level contains mostly two very big communities corresponding

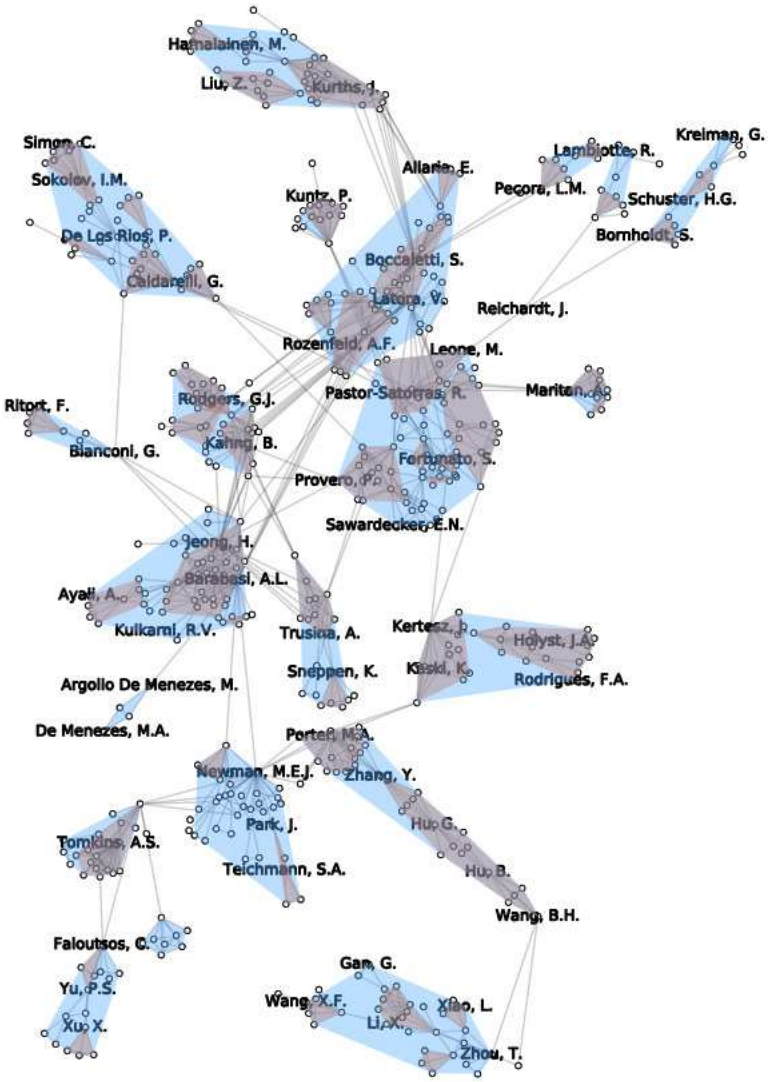
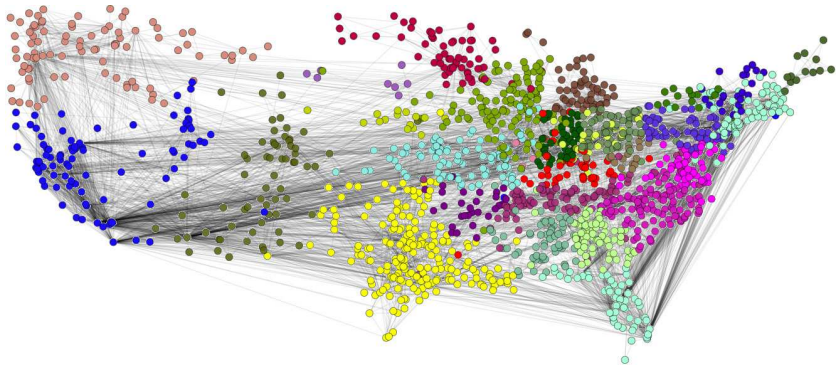


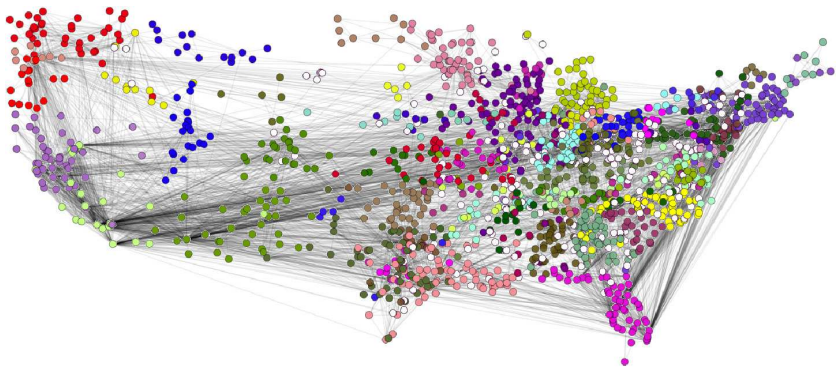
Fig. 3 Results on a scientific co-publication network. The blue and grey hulls correspond to the flat clusterings $N_1(T)$ and $N_2(T)$ respectively.

to the West/Mid-West area and the East. These clusters are then divided over two additional levels. The bottom clustering corresponds to the clustering provided by the *Infomap* algorithm.

The results of our method are a good compromise. The first level (see Figure 4(a)) contains relatively large clusters. The second level (see Figure 4(b)) is similar to the clustering provided by the *Infomap* algorithm.



(a) First level $N_1(T)$



(b) Second level $N_2(T)$

Fig. 4 Results on a migration network (United-States). Vertices position corresponds to the geographical coordinates of the corresponding county. Two vertices belong to the same cluster if they have the same color. White coloured vertices are isolated (cluster of size 1).

5 Conclusion

We introduced a post-processing procedure to improve the quality of a hierarchical clustering of a network. This is achieved by iteratively removing the internal clusters that decrease a multilevel quality measure. Our method was applied to the algorithm of Blondel *et al.*[2] by using a generalization of the modularity metric to hierarchical clusterings. The experiments run on random graphs clearly show that the hierarchies we provide are very close to the ground truth hierarchies. Results obtained on real networks are also meaningful.

Note that our method does not allow to know whether or not the leaves of the clustering tree should be removed. We can reduce this problem to the following one: is a given clustering better than no clustering at all ? One way to overcome this problem is to use a minimal threshold for modularity. However dealing with this

kind of clusters is less problematic. Indeed, from an analysis perspective, the first levels of a hierarchical clustering are the most relevant.

As future work, we plan to test the effectiveness of our post-processing procedure when applied with different hierarchical clustering algorithms. The greedy removal of internal clusters is a fast and intuitive method but adding internal clusters to the hierarchy is also a possible modification. We need to investigate the way of combining these basic operations to explore the space of hierarchical clusterings using a hierarchical quality measure as objective function.

References

1. Blanc, C., Delest, M., Fédou, J.M., Mélançon, G., Queyroi, F.: Évaluer la qualité d'une fragmentation de graphe multi-niveaux. In: Journées MARAMI 2010. Toulouse, France (2010). URL http://hal.archives-ouvertes.fr/hal-00542484/en/
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**, P10,008 (2008). URL http://arxiv.org/pdf/0803.0476
3. Cook, D.J., Holder, L.B.: Mining Graph Data. Wiley (2006)
4. Cui, W., Zhou, H., Qu, H., Wong, P., Li, X.: Geometry-based edge clustering for graph visualization. IEEE Transactions on Visualization and Computer Graphics **14**(6), 1277–1284 (2008)
5. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment **2005**, P09,008 (2005)
6. Fortunato, S.: Community detection in graphs. Physics Reports **486**(3-5), 75–174 (2010). URL http://arxiv.org/pdf/0906.0612
7. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. Proceedings of the National Academy of Sciences **104**(1), 36 (2007)
8. Good, B., De Montjoye, Y., Clauset, A.: Performance of modularity maximization in practical contexts. Physical Review E **81**(4), 46,106 (2010). URL http://arxiv.org/pdf/0910.0165
9. Lancichinetti, A., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Physical Review E **78**(4), 046,110 (2008)
10. Lancichinetti, A., Radicchi, F., Ramasco, J.: Finding statistically significant communities in networks. PloS one **6**(4), e18,961 (2011). URL http://arxiv.org/pdf/1012.2363v2
11. Mancoridis, S., Mitchell, B., Rorres, C., Chen, Y., Gansner, E.: 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 (1998)
12. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences, USA **103**, 8577–8582 (2006). URL http://www.pnas.org/content/103/23/8577.long
13. Pons, P., Latapy, M.: Post-processing hierarchical community structures: Quality improvements and multi-scale view. Theoretical Computer Science **412**, 892–900 (2010)
14. Pumain, D. (ed.): Hierarchy in Natural and Social Sciences, *Methodos Series*, vol. 3. Springer (2006)
15. Rosvall, M., Bergstrom, C.: Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. PloS one **6**(4), e18,209 (2011). URL http://www.tp.umu.se/ rosvall/publications/RosvallBergstromPLoSONE2011.pdf
16. Simon, H.: The architecture of complexity. Proceedings of the American Philosophical Society **106**(6), 467–482 (1962)