

Limits of modularity maximization in community detectionAndrea Lancichinetti^{1,2} and Santo Fortunato^{1,3}¹*Complex Networks and Systems Lagrange Lab, Institute for Scientific Interchange, I-10133 Torino, Italy*²*Physics Department, Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino, Italy*³*Department of Biomedical Engineering and Computational Science, School of Science, Aalto University, P.O. Box 12200, FI-00076 Espoo, Finland*

(Received 6 July 2011; revised manuscript received 17 October 2011; published 27 December 2011)

Modularity maximization is the most popular technique for the detection of community structure in graphs. The resolution limit of the method is supposedly solvable with the introduction of modified versions of the measure, with tunable resolution parameters. We show that multiresolution modularity suffers from two opposite coexisting problems: the tendency to merge small subgraphs, which dominates when the resolution is low; the tendency to split large subgraphs, which dominates when the resolution is high. In benchmark networks with heterogeneous distributions of cluster sizes, the simultaneous elimination of both biases is not possible and multiresolution modularity is not capable to recover the planted community structure, not even when it is pronounced and easily detectable by other methods, for any value of the resolution parameter. This holds for other multiresolution techniques and it is likely to be a general problem of methods based on global optimization.

DOI: [10.1103/PhysRevE.84.066122](https://doi.org/10.1103/PhysRevE.84.066122)

PACS number(s): 89.75.Hc

I. INTRODUCTION

The detection and analysis of communities in graphs [1,2] is one of the most popular topics within the modern science of networks [3–10]. In recent years an increasing number of large networked datasets, including millions or even billions of vertices and edges, have become available, and a traditional analysis based on local network properties and their global statistics (e.g., degree distributions and the like) provides but a partial description of the system and its function. Communities (also called *clusters* or *modules*) are subgraphs including vertices with similar features or function, and their identification may disclose not only such similarities among vertices, which are often hidden, but also how the system is internally organized and works.

Vertices belonging to the same community have a considerably higher probability of being linked to each other than vertices belonging to different clusters. Therefore a community appears as a region of the network with a high density of internal links, much higher than the average link density of the graph. The most popular method to detect communities in graphs consists in the optimization of a quality function, the modularity introduced by Newman and Girvan [11,12]. Modularity quantifies the deviation of the internal link density of the clusters from the density one expects to find within the same groups of vertices in random graphs with the same expected degree sequence of the network at study. The idea is that vertices linked to each other in a random way should not form communities, as high values of the link density cannot be attained. Consequently, high values of modularity are supposed to indicate “suspiciously” high values of internal link densities for the subgraphs, which are then distinct from groups of randomly linked vertices and can be deemed as true communities. While this is actually not true [13,14], the optimization of the measure has been widely used in past years.

Recently, it has been pointed out that modularity optimization has a number of problems. In particular, it has a resolution limit [15] that leads to the systematic merger of

small clusters in larger modules, even when the clusters are well defined and loosely connected to each other. A more recent analysis of the resolution limit has led to the conclusion that the modularity landscape is “glassy” and includes an exponentially growing (with system size) number of local maxima whose values are very close to the absolute maximum of the measure, even if the corresponding partitions may be topologically quite different from each other [16]. This implies, on the one hand, that it is not too difficult to find a good approximation of the modularity maximum for many techniques; on the other hand, the maximum is essentially unreachable. A recent comparative analysis of community finding algorithms has indeed revealed that modularity fails to properly identify clusters on benchmark graphs with built-in community structure and that other methods are much more effective [17].

Nevertheless, modularity optimization is still being used. The main reason is the claim that the resolution limit can be removed by adopting suitable multiresolution versions of modularity, like those introduced by Reichardt and Bornholdt [18] and by Arenas, Fernández, and Gómez [19]. In these variations, a tunable resolution parameter enables one to set the size of the clusters to arbitrary values, from very large to very small. However, real networks are characterized by the coexistence of clusters of very different sizes, whose distributions are quite well described by power laws [20–22]. Therefore, there is no characteristic cluster size and tuning a resolution parameter may not help. Indeed, in this paper we show that multiresolution modularity is not capable of identifying the right partition of the network in realistic settings and that, therefore, it does not solve the problems of modularity maximization in practical applications. The problem is that modularity maximization is not only inclined to merge small clusters but also to break large clusters, and it seems basically impossible to avoid both biases simultaneously. This applies to other multiresolution methods as well and is probably a general feature of methods based on the optimization of a global measure.

The paper is structured as follows. In Sec. II we present a general analysis of some relevant mathematical properties of multiresolution modularity, with respect to the merger or split of subgraphs, leading to the identification of a range of values of the resolution parameter, where modularity should be safe from the above-mentioned problems. In Sec. III we test the result on realistic benchmark graphs with community structure, showing that it is often impossible to find a value of the resolution parameter that delivers the planted partition. Conclusions are reported in Sec. IV.

II. THE PROBLEM OF MERGING AND SPLITTING CLUSTERS

A. Multiresolution modularity

Our conclusions are not significantly affected by the specific modularity formula one chooses, as we will show in Sec. III. For the analytical discussion of this section we adopt the generalized modularity Q_λ proposed by Reichardt and Bornholdt [18], which reads

$$Q_\lambda = \sum_S \left[\frac{k_{in}^S}{2M} - \lambda \left(\frac{k_{tot}^S}{2M} \right)^2 \right], \quad (1)$$

where the sum runs over all the clusters, $2M$ is the total degree of the network, k_{tot}^S is the sum of the degrees of vertices in module S , and k_{in}^S is twice the number of internal edges in module S . So, we have $k_{tot}^S = k_{in}^S$ only if the module is disconnected from the rest of the graph. Here, λ works like a resolution parameter: high values of λ lead to smaller modules because the term $(k_{tot}^S/2M)^2$ in the sum of Eq. (1) becomes more important and its minimization, induced by the maximization of Q_λ , favors smaller clusters.

We ask when it is proficuous for modularity to keep two subgraphs together or separate. For this, we need to compute the difference $\Delta Q_\lambda = Q_\lambda$ (partition with merged subgraphs) $- Q_\lambda$ (partition with separated subgraphs): if $\Delta Q_\lambda > 0$ modularity would be higher for the partition where the subgraphs are merged; otherwise, the split would be more convenient.

We indicate with A and B the two subgraphs (see Fig. 1). Let Q_λ^{A-B} and Q_λ^{AUB} denote the value of modularity when A and B are kept separated and merged, respectively.

$$Q_\lambda^{A-B} = \left[\sum_{S \neq A, B} \dots \right] + \frac{k_{in}^A}{2M} + \frac{k_{in}^B}{2M} - \lambda \left(\frac{k_{in}^A + l + v}{2M} \right)^2 - \lambda \left(\frac{k_{in}^B + r + v}{2M} \right)^2, \quad (2)$$

where v denotes the number of links joining A with B , l the number of links joining A with the rest of the network (excluding B), and r is the equivalent of l for B . For Q_λ^{AUB} we have:

$$Q_\lambda^{AUB} = \left[\sum_{S \neq A, B} \dots \right] + \frac{k_{in}^A}{2M} + \frac{k_{in}^B}{2M} + \frac{2v}{2M} - \lambda \left(\frac{k_{in}^A + l + v + k_{in}^B + r + v}{2M} \right)^2. \quad (3)$$

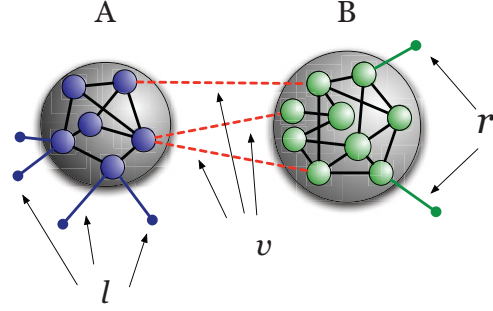


FIG. 1. (Color online) Schematic representation of the problem of merging versus splitting subgraphs. Here A and B are two subgraphs, the problem is whether one yields a higher value for modularity by merging them in a single subgraph or by keeping them separated. The parameters involved in the decision are the number of internal links in A and B (multiplied by 2), k_{in}^A and k_{in}^B , the number of links v between A and B (here $v = 3$), the number of links l between A and vertices belonging neither to A nor to B (here $l = 4$), and its equivalent r for B (here $r = 2$).

The difference $\Delta Q_\lambda = Q_\lambda^{AUB} - Q_\lambda^{A-B}$ reads

$$\Delta Q_\lambda = \frac{2v}{2M} - \lambda \frac{k_{in}^A k_{in}^B + l k_{in}^B + r k_{in}^A + l r}{2M^2} - \lambda \frac{v(k_{in}^A + k_{in}^B + l + r) + v^2}{2M^2}. \quad (4)$$

To simplify a little Eq. (4), we can define $\Delta = 2M \Delta Q_\lambda$

$$\Delta = 2v - \lambda \frac{k_{in}^A k_{in}^B + l k_{in}^B + r k_{in}^A + l r}{M} - \lambda \frac{v(k_{in}^A + k_{in}^B + l + r) + v^2}{M}. \quad (5)$$

Modularity is higher for A and B merged if and only if $\Delta > 0$.

Equation (5) is rather general, but we are just interested in testing modularity for some special cases, for which calculations are easy. Here in particular, we will consider the case $l = r = \eta$ and $k_{in}^A = k_{in}^B = \xi$. Equation (5) becomes

$$\Delta = 2v - \lambda \frac{(\xi + v + \eta)^2}{M}. \quad (6)$$

These results are essential to follow the discussion of the next subsections.

B. Splitting clusters

Despite the different approaches to the problem of detecting clusters in networks, there are some general ideas that are shared by most scholars. One of them is that a random graph has no communities, so it should not be split by an algorithm in smaller pieces, with the only exception of the trivial split in singletons, i.e., in groups containing each just a single vertex, which is still an acceptable answer. Another shared belief is that a complete graph (or clique), i.e., a graph whose vertices are all connected to each other, is a perfect community (due to the fact that the internal link density reaches the highest possible value of 1). So, if cliques are just loosely connected to each other, one would expect that a good method should detect them as separate clusters. We

would like to find the mathematical conditions, in particular the choice of the resolution parameter λ , that satisfy both requisites. In this subsection we search for the condition to avoid the splitting of random subgraphs, while the condition to avoid the merger of cliques will be given in the next subsection.

Let us consider a random subgraph S with total degree $2M_S$, which is part of a larger network with total degree $2M$. The goal is to check under which condition S is split by optimizing modularity. Here, for simplicity, we consider only bipartitions. The expected optimal modularity Q_2 for the bipartition of a random graph has been computed by Reichardt and Bornholdt [23],

$$Q_{RB} = 0.765 \frac{\langle \sqrt{k} \rangle_S}{\langle k \rangle_S}, \quad (7)$$

where the brackets indicate expectation values over the ensemble of random graphs with the same expected degree sequence of the subgraph at study.

We now express Q_2 in terms of the number of edges v between the clusters of the bipartition with optimal modularity. We obtain

$$2M_S Q_2 = 2M_S - 2v - \frac{k_A^2 + k_B^2}{2M_S} = \frac{2k_A k_B}{2M_S} - 2v \quad (8)$$

where k_A (k_B) is the total degree of module A (B). Since modularity is optimal when the two modules are of about equal size, i.e., when $k_A \approx k_B \approx M_S$, we have

$$2M_S Q_2 = M_S - 2v, \quad (9)$$

from which we can derive v :

$$v = M_S \left(\frac{1}{2} - Q_2 \right). \quad (10)$$

For $Q_2 = 0$ we would have $v = M_S/2$, which is the expected average number of links joining two modules of equal size, arbitrarily chosen. Equation (10) implies that optimizing modularity decreases the number of expected links between the modules, with respect to arbitrary bipartitions, while it increases the internal density of links of the modules. One also sees that, for v to be positive, $Q_2 \leq 0.5$. Actually, in the calculation of Reichardt and Bornholdt, this holds only if $\langle k \rangle$ is big enough. To give an idea of the numbers that one could have, $Q_2 \approx 0.17$ when all vertices have degree $k = 20$, so $v \approx 0.33 \times M_S$, which is actually a not too bad approximation also for other degree distributions (for all vertices having degree $k = 10$, $v \approx 0.25 \times M_S$). Let us call α_S this proportionality factor between v and M_S ,

$$v = \alpha_S M_S \quad \text{and} \quad k_{in}^A = k_{in}^B = (1 - \alpha_S) M_S. \quad (11)$$

From Eqs. (7), (10), and (11) we get

$$\alpha_S = \frac{1}{2} - 0.765 \frac{\langle \sqrt{k} \rangle_S}{\langle k \rangle_S}. \quad (12)$$

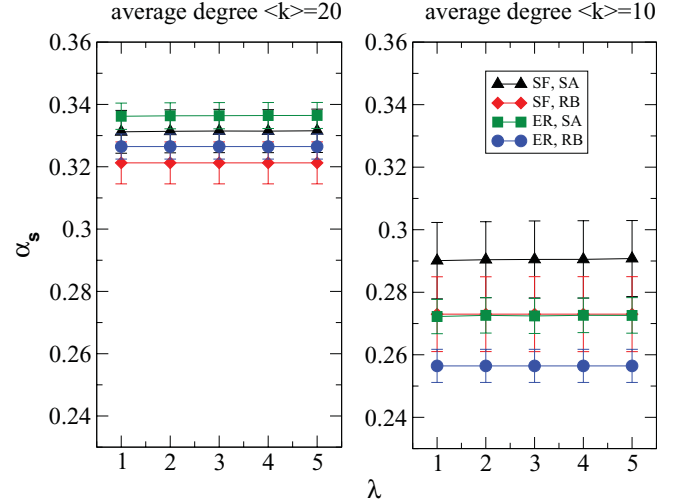


FIG. 2. (Color online) The plot shows α_S measured on Erdős-Rényi and scale-free graphs. For each type of graph, we plot the analytical estimate of Reichardt and Bornholdt (RB) and a numerical estimate obtained by optimizing modularity with simulated annealing (SA) [13]. The minimum cut $v = \alpha_S \times M_S$ was measured by optimizing modularity for different values of λ over the set of bipartitions. To optimize modularity, we are looking for small values of v and equal values of k^A and k^B , so tuning λ just controls the importance of either requirement. However, simulations show that the dependence on λ is quite weak, validating our approximation $k^A \approx k^B$.

In Fig. 2 we compare the values of α_S from Eq. (12) with numerical estimates derived by putting in Eq. (10) the maximum modularity Q_2 , derived with simulated annealing. The calculation of Q_2 is carried out for different values of λ , but the results seem to be essentially independent of λ . We consider both Erdős-Rényi (ER) and scale-free (SF) graphs, with 1000 vertices and average degree $\langle k \rangle = 20$ (left panel) and 10 (right panel). The SF graphs have degree exponent 2. As we can see from Fig. 2, the analytical estimate of Eq. (12) yields a good approximation of α_S .

Let us now consider our splitting-merging problem, considering A and B as candidates. We set $\eta = 1$, which means that only two links come out of S (ideally one from A , the other from B). In this case, we would like to have $\Delta > 0$, to avoid the split of the random subgraph S . From Eqs. (6) and (11) we get (remember that $\xi = k_{in}^A = k_{in}^B$):

$$2\alpha_S M_S > \frac{\lambda(M_S + 1)^2}{M}, \quad (13)$$

which implies

$$\lambda < \frac{2\alpha_S M}{M_S}. \quad (14)$$

Alternatively, we can incorporate the correction factor $[M_S/(M_S + 1)]^2 \approx 1$ in α_S , so that we call α_S what is actually $\alpha_S [M_S/(M_S + 1)]^2$. If the subgraph is a clique, $\alpha_S \approx 0.5$, and modularity can even split a clique when

$$\lambda \gtrsim \frac{M}{M_S}. \quad (15)$$

C. Merging clusters

Let us now consider two equal-sized subgraphs connected with one edge ($v = 1$ and $\eta = 1$) and let $k_{in}^A = k_{in}^B = \xi_C$. Equation (6) becomes

$$\Delta = 2 - \lambda \frac{(\xi_C + 2)^2}{M}. \quad (16)$$

In this case, we want $\Delta < 0$ (we wish to keep the two subgraphs separated), which implies

$$\lambda > \lambda_C = \frac{2M}{(\xi_C + 2)^2}. \quad (17)$$

If ξ_C is very small, λ has to be very big (for $\lambda_C > 1$ the subgraphs cannot be resolved by standard modularity, which corresponds to $\lambda = 1$, and we recover the resolution limit of Ref. [15]). On the other hand, if ξ_C is large, the subgraphs will be resolved for a large range of λ values.

If the subgraphs are two cliques of n_C nodes each, for instance, $\xi_C = n_C(n_C - 1)$.

D. Condition on the ineliminability of the bias

We now put together Eqs. (14) and (17). We have that

$$\lambda_2 < \lambda < \lambda_1, \quad (18)$$

where

$$\lambda_1 = \frac{2\alpha_S M}{M_S} \quad \text{and} \quad \lambda_2 = \frac{2M}{(\xi_C + 2)^2}. \quad (19)$$

Above λ_1 , modularity splits random subgraphs, below λ_2 it puts together subgraphs even if they are connected by just one link (even in the case in which they are cliques). In the range between λ_1 and λ_2 it should be possible to avoid both biases. However, if

$$\lambda_1 < \lambda_2, \quad (20)$$

the biases cannot be both simultaneously lifted. Equation (20) holds when, by setting $M_S/\alpha_S = \beta_S$,

$$(\xi_C + 2)^2 < \beta_S. \quad (21)$$

Note that Eq. (21) does not depend on the size of the whole network, either in terms of vertices or edges.

To be more concrete, we consider a simple example. We examine a network made out of two identical cliques of n_C vertices each and an internally random subgraph of n_S vertices and average degree $\langle k \rangle_S$. The three clusters are all connected to each other by one edge only (see Fig. 3). In Fig. 4 we plot the relation between n_C and n_S coming from the equality $\lambda_1 = \lambda_2$ [obtained turning the inequality of Eq. (21) to an equality] for some values of $\langle k \rangle_S$. We used Eq. (12) to evaluate α_S , with the approximation $\langle \sqrt{k} \rangle_S = \sqrt{\langle k \rangle_S}$ and the relations $\xi_C = n_C(n_C - 1)$ and $M_S = n_S \langle k \rangle_S / 2$. For any given value of $\langle k \rangle_S$, the inequality of Eq. (21) holds above the corresponding curve. In Fig. 5 we plot λ_1 and λ_2 as a function of n_S , for $n_C = 13$ and $\langle k \rangle_S = 100$. For λ_1 we show two curves, one corresponding to the exact function, determined numerically, while for the other we have used the theoretical approximation of α_S described above. The lines divide the $\lambda - n_S$ plane in four areas, characterized by the presence or absence of the two

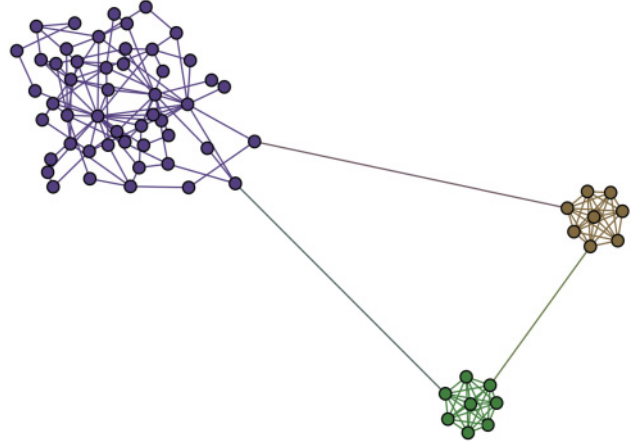


FIG. 3. (Color online) Schematic network with two cliques and a random subgraph, which are the natural communities of the network.

biases. As we can see, the portion of the plane in which both biases are simultaneously absent (gray area) is quite small.

One might still wonder that it could be possible to find a value of λ high enough that the random subgraph \mathcal{S} is split in n_S vertices and the two cliques are still correctly detected. Let us consider Eq. (5) when A consists of a single vertex, so that v is the internal degree of the vertex with respect to B and $l + v = k^A$ is the total degree of A . Recalling that $k_{in}^B + r + v = k_{tot}^B$, Eq. (5) becomes:

$$\Delta = 2v - \lambda \frac{k_{tot}^B k^A}{M}. \quad (22)$$

Therefore, A and B would be kept separated when

$$\lambda > \frac{2Mv}{k_{tot}^B k^A}. \quad (23)$$

By increasing λ , we can actually separate some vertices of \mathcal{S} and we would eventually split it in n_S clusters when $\lambda > \frac{2M}{x}$, where x is the minimum $k_i k_j$ over all the connected vertices

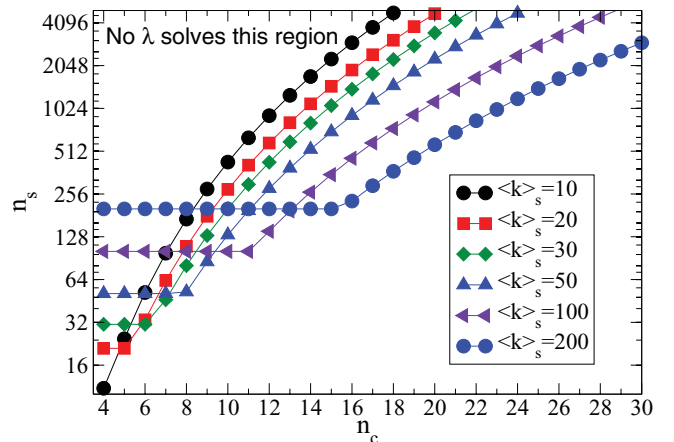


FIG. 4. (Color online) This plot shows Eq. (21) as a function of n_S and n_C for the simple network with the three clusters described in the legend of Fig. 3. Above the curves, modularity cannot find the right partition for any value of λ .

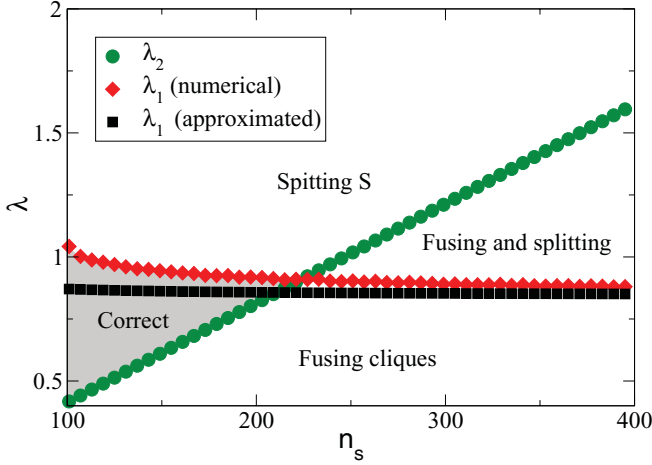


FIG. 5. (Color online) Threshold parameters λ_1 and λ_2 as a function of n_S ($n_C = 13$, $\langle k \rangle_S = 100$). The theoretical line for λ_1 is obtained by approximating α_S as described in the text. We see that $\lambda_1 > \lambda_2$, up to $n_S \approx 230$, so that no λ can eliminate the biases for bigger values of n_S . When $n_S < \approx 230$, the biases can be both eliminated only in the shadowed area between the curves.

(i, j) of \mathcal{S} . Similarly, the condition for the cliques not to be split reads:

$$\lambda < \frac{2M}{(n_C - 1)(n_C - 2)}, \quad (24)$$

since the denominator is the total degree of a clique of $n_C - 1$ vertices (we neglected r) and we considered $k^A = v$ (the vertex does not have external connections).

In conclusion, if there are two connected vertices in \mathcal{S} such that the product of their degrees is smaller than $(n_C - 1)(n_C - 2)$, no values of λ are suitable to guess the right answer(s). This is very likely to happen if the degree distribution of \mathcal{S} is broad, so that there are many low-degree vertices.

III. TESTS ON BENCHMARK GRAPHS

We want now to check the practical consequences of the limits of multiresolution modularity. For that we take the LFR benchmark, a model of graphs with built-in community structure that we have recently introduced [24]. It is an extension of the *planted ℓ -partition model* introduced by Condon and Karp [25]. Each graph has power law distributions of degree and community size, which are common features of real graphs with community structure. The degree of mixture between clusters is measured by the *mixing parameter* μ , expressing the ratio between the number of neighbors of a vertex outside its community and the total number of neighbors. So, $\mu = 0$ indicates that clusters are topologically disconnected from each other, as each vertex has neighbors within its community only, while $\mu = 1$ indicates that vertices are connected only to vertices outside their group, so the groups are not communities. Vertices are linked to each other at random, compatible with the constraints on the distributions of degree and community size and to the fact that μ has to be (approximately) the same for all vertices. So, the clusters are essentially random subgraphs.

We want to specialize Eq. (5) to the LFR benchmark graphs. Let us consider a cluster \mathcal{S} with n_b nodes, total degree $2m_b$, and internal degree $2M_{Sb}$. We split it into two equal-sized subgraphs, such that the internal degree of either part is the same: $k_{in}^A = k_{in}^B$. Moreover, for simplicity we assume that the split is done such to keep an equal number of edges between each of the subgraphs and the rest of the network: $l = r$. We have $M_{Sb} = (1 - \mu)m_b$, $l = r = \mu m_b$, $v = \alpha_{Sb}M_{Sb} = \alpha_{Sb}(1 - \mu)m_b$. The condition of non-splitting is

$$2v > \lambda \frac{(M_{Sb} + l)^2}{M}, \quad (25)$$

which is

$$2\alpha_{Sb}(1 - \mu)m_b > \lambda \frac{m_b^2}{M}. \quad (26)$$

So,

$$\lambda < \lambda_1 \quad \text{where} \quad \lambda_1 = 2\alpha_{Sb}(1 - \mu) \frac{M}{m_b}. \quad (27)$$

We now search for the condition that leads to the merger of two clusters of an LFR benchmark graph. For that we should know how many edges they share, which depends on the graph size and the number of clusters. We call v_{xy} the number of edges between modules x and y , and $2m_x$ and $2m_y$ are their total degrees. Equation (5) becomes

$$\Delta = 2v_{xy} - \lambda \frac{4m_x m_y}{M}. \quad (28)$$

The condition to keep the clusters separated is $\lambda > \lambda_2$, where

$$\lambda_2 = \frac{M v_{xy}}{2m_x m_y}. \quad (29)$$

So, the two biases can be simultaneously removed if $\lambda_1 > \lambda_2$, which amounts to

$$2\alpha_{Sb}(1 - \mu) \frac{M}{m_b} > \frac{M v_{xy}}{2m_x m_y}. \quad (30)$$

The inequality of Eq. (30) has to hold for all triples of clusters x , y , and b , and this is usually unlikely to happen. In order to show that, we check whether multiresolution modularity is able to deliver the planted partition of the LFR benchmark graphs for any value of the resolution parameter λ . The results are shown in Figs. 6 and 7. We plot the fraction of vertices that are incorrectly classified by modularity as a function of λ . We consider the misclassifications caused by merging (circles) or splitting (squares) the clusters of the planted partition of the graphs. We see that, for small values of λ , modularity merges many clusters and essentially splits none, whereas for large λ there is a dominance of splitting over merging. The plots clearly show that, for every value of λ , there will be some misclassification due to cluster merging, splitting, or both. The fraction of affected vertices does not go below 10% but it can be considerably larger. Figure 6 refers to graphs with 10 000 vertices, but the situation does not improve if we go to larger graph sizes (50 000 vertices for the benchmark graphs used for Fig. 7). We point out that we have chosen low values of the mixing parameter μ (0.1 and 0.3), corresponding to clusters that are well separated from each other. Modern algorithms for community detection (like Infomap [26] and OSLOM [27]) would easily find the correct partitions in the graphs we have

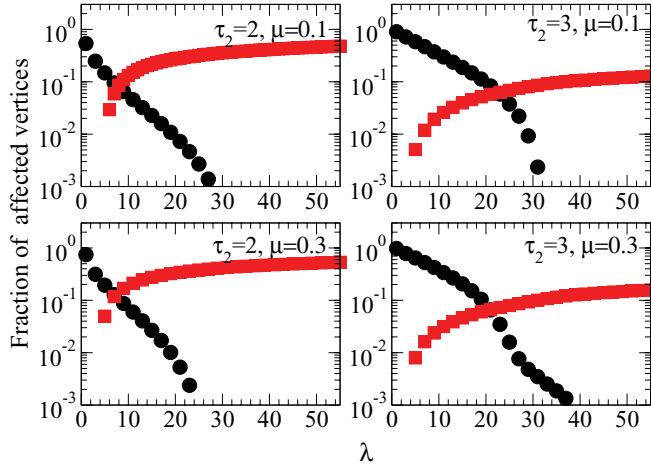


FIG. 6. (Color online) Test of multiresolution modularity on LFR benchmark graphs. Each panel shows the fraction of misclassified vertices due to artificial mergers (circles) and splits (squares) of clusters, as a function of the resolution parameter λ . The panels correspond to different choices of the exponent τ_2 of the cluster size distribution of the graph and of the mixing parameter μ . Each point represents an average over 100 benchmark graphs. All graphs have 10 000 vertices. The other parameters are: average degree $\langle k \rangle = 20$; maximum degree $k_{\max} = 100$; minimum cluster size $c_{\min} = 10$; maximum cluster size $c_{\max} = 1000$; degree exponent $\tau_1 = 2$.

used for the tests of Figs. 6 and 7 (see Ref. [17]). One may object that our estimate of the modularity maximum for each graph is just an approximation of the actual result, whose search is an NP-complete problem [28]. However, we have checked in each case that the partitions found have a higher modularity than the planted partition of the benchmark graphs.

Finally, we check how general our results are. We have focused on the multiresolution modularity proposed by Reichardt and Bornholdt in Ref. [18]. In this paper, however, the authors had proposed a general ansatz for the quality function, and their multiresolution modularity was just a specific case of it. In a recent work [29], Traag *et al.* have shown that this ansatz

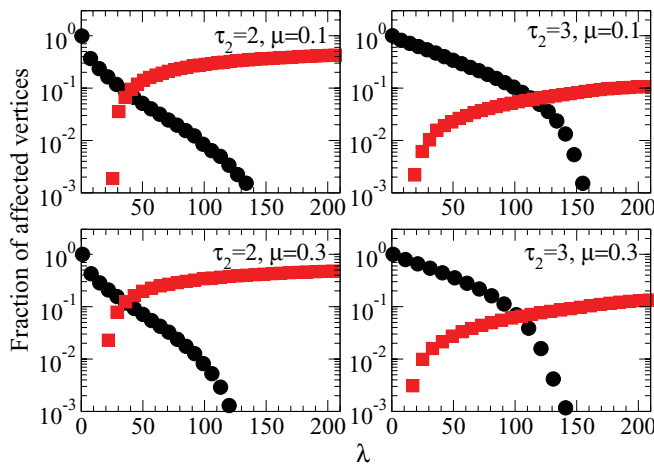


FIG. 7. (Color online) Same as Fig. 6, but for LFR benchmark graphs of 50 000 vertices. All other parameters are the same as for the graphs used in Fig. 6.

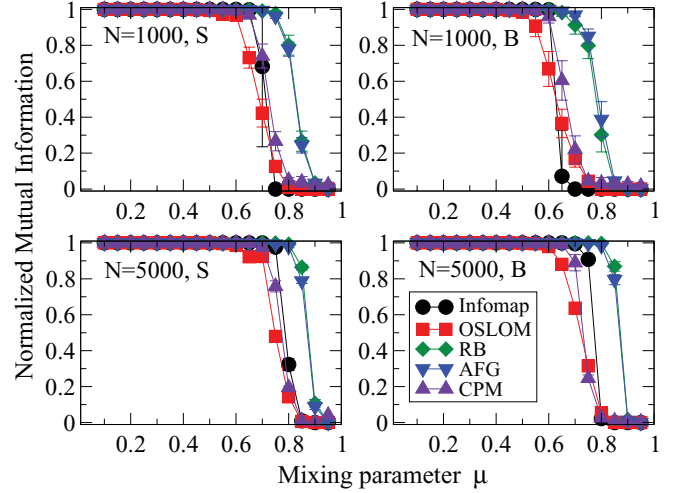


FIG. 8. (Color online) Comparative analysis of several multiresolution techniques on the LFR benchmark. The graphs are made of 1000 and 5000 vertices, the exponent of the degree distribution $\tau_1 = 2$, the exponent of the clusters size distribution $\tau_2 = 1$, the average degree $\langle k \rangle = 20$, the maximum degree $k_{\max} = 50$, the cluster size ranges are $S = [10, 50]$ and $B = [20, 100]$.

can be specialized to include other known measures, such as the multiresolution modularity by Arenas *et al.* [19] and the quality function adopted by Ronhovde and Nussinov [30], which is characterized by not having a null model term, in contrast to modularity. In fact, Traag *et al.* derived another model from the general class of functions of Reichardt and Bornholdt, which they called Constant Potts Model (CPM), which allegedly has no resolution limit. In Fig. 8 we reproduce the results of the comparative analysis performed by Traag *et al.* on the LFR benchmark. Here, we compare five methods: Infomap, OSLOM, the optimization of the multiresolution modularities of Reichardt and Bornholdt (RB), and Arenas *et al.* (AFG), and the CPM by Traag *et al.* For each selected value of the mixing parameter μ , we generated 100 realizations of the LFR benchmark and averaged on them the values of the similarity between the detected and the planted partition. As a similarity measure, we took the normalized mutual information (NMI) [31], which has become a standard in this kind of evaluation. In our computations, we used a modified version of the measure of Ref. [32], recently introduced by the authors of this paper, which is able to estimate the similarity of partitions as well as the similarity of covers, i.e., of divisions of a network into overlapping communities. We have used this version of the NMI in our comparative analysis of community detection algorithms [17], so we stick to it for consistency. We stress, however, that the clusters of the graphs we considered are not overlapping.

As found in Ref. [29], it is possible to find values of the resolution parameter for RB and CPM, which make these methods outperform both Infomap and OSLOM. This holds for AFG as well, whose performance is essentially identical as RB. However, this is due to the fact that the cluster sizes are too close to each other, spanning less than one order of magnitude. This is demonstrated by Fig. 9, in which we take LFR benchmark graphs with the same parameters as those used for Fig. 6. Now we have 10 000 vertices and cluster sizes

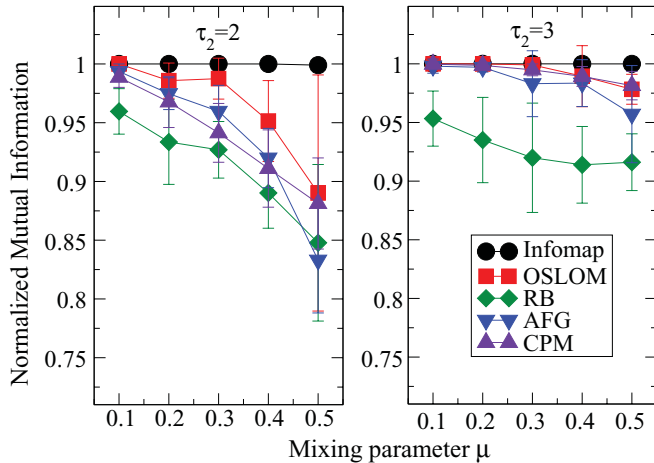


FIG. 9. (Color online) Comparative analysis of several multiresolution techniques on the LFR benchmark. The network parameters are now the same as for the graphs used in Fig. 6. In particular, the network size is 10 000 and the cluster size spans two orders of magnitude. The two panels correspond to $\tau_2 = 2$ (left) and $\tau_2 = 3$ (right).

vary from 10 to 1000 vertices. Again, for the multiresolution methods we use the values of the resolution parameters that give the best results. The figure shows that the multiresolution methods fail to detect the planted partition even for very low values of the mixing parameter μ , especially when the cluster size distribution is broader ($\tau_2 = 2$). This is consistent with the results of Figs. 6 and 7. Infomap and OSLOM, on the other hand, have a clearly better performance, despite the fact that they do not have a tunable resolution parameter. In particular, Infomap always detects the right partition, for the range of μ explored here. Most networks of current interest have many more than 10 000 vertices, and accordingly, community sizes span much broader ranges of values. Figure 9 suggests that in such cases the performance of multiresolution methods might become far worse.

IV. CONCLUSIONS

We have shown that multiresolution modularity maximization is characterized by two concurrent biases: the tendency to merge small clusters and to split large ones. We have seen that it is usually very difficult, and often impossible, to tune the resolution such to avoid both biases simultaneously. Tests on artificial benchmark graphs with community structure indeed show that a considerable fraction of vertices is misclassified, for any value of the resolution parameter, even when clusters are well separated and easily identified by other methods. Since, in practical applications, one knows very little about the community structure of the graphs at study, it is impossible *a priori* to quantify the systematic error induced by the use of modularity. Moreover, it is not easy to implement a way to “heal” the partition delivered by modularity, just because there are two sources of errors. If modularity simply combined smaller clusters in larger ones, as people have been thinking until now, one could hope to recover the real partition by looking inside the clusters delivered by modularity. Instead, since clusters can be both split and merged, the real partition

must be recovered by splitting some clusters and merging others, and it is very difficult to understand which clusters contain smaller ones and which others are parts of larger clusters instead. This would require a careful exploration of groups of clusters.

Our results hold for various types of quality functions, including the recently introduced Constant Potts Model by Traag *et al.* [29]. One could argue that, after all, multiresolution methods have a remarkable performance in some cases (see Fig. 8) and a poor one in others (see Fig. 9), just like any method, including Infomap and OSLOM (from the same figures). This objection is, however, not sustainable, since we believe that, when clusters are so weakly connected to each other that one could even distinguish them by visual inspection, a good method cannot fail to detect them. While this is a shared view among scholars, it is still unclear where to set the limit of fuzziness between subgraphs that separates a regime in which they are clusters from one in which they are not. This problem has attracted some attention lately [33,34]. So, in the tests we reported (Figs. 8 and 9), it is not clear up to which value of the mixing parameter μ the subgraphs of the benchmark graphs are still “significant” clusters, beyond random fluctuations. But there is no doubt that they are cluster for very small values of the mixing parameter μ .

We want to stress here that we are not advocating the superiority of some methods over others. The problems that we point out in this paper are probably common to many other methods. Infomap, for instance, is a method based on the optimization of a global measure, like modularity, and is likely to have a resolution limit as well, although it probably emerges only on large networks. In addition, it may also break random subgraphs, although its performance is perfect for well-separated communities in all tests we have performed. OSLOM could be also improved, since it occasionally fails to detect the right partition for small μ . Still, at variance with multiresolution methods, neither Infomap nor OSLOM have a tunable resolution parameter, so their performance is quite remarkable.

We conjecture that the tendency to simultaneously merge and split clusters is an inevitable feature of methods based on global optimization and that it could be more easily circumvented by local approaches. Global optimization techniques work well when clusters are approximately of the same size; if clusters span a broad range of sizes, which is likely to happen on very large networks, such techniques get confused and may fail to detect some of the clusters, even when they are clearly identifiable. Resolution parameters improve things, but they do not (cannot?) solve the problem.

We hope that the scientific community working on the problem of community detection will address this issue in the future and that general structural limits of classes of methods will be identified and, possibly, removed. In this way, it will be possible to define safe guidelines to design new methods that do not suffer from such problems and that, therefore, could be more reliable in practical applications.

ACKNOWLEDGMENTS

We gratefully acknowledge ICTeCollective Grant No. 238597 of the European Commission.

- [1] M. Girvan and M. E. Newman, *Proc. Natl. Acad. Sci. USA* **99**, 7821 (2002).
- [2] S. Fortunato, *Phys. Rep.* **486**, 75 (2010).
- [3] R. Albert and A.-L. Barabási, *Rev. Mod. Phys.* **74**, 47 (2002).
- [4] S. N. Dorogovtsev and J. F. F. Mendes, *Adv. Phys.* **51**, 1079 (2002).
- [5] M. E. J. Newman, *SIAM Rev.* **45**, 167 (2003).
- [6] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach* (Cambridge University Press, New York, 2004).
- [7] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang, *Phys. Rep.* **424**, 175 (2006).
- [8] G. Caldarelli, *Scale-Free Networks* (Oxford University Press, Oxford, 2007).
- [9] A. Barrat, M. Barthélemy, and A. Vespignani, *Dynamical Processes on Complex Networks* (Cambridge University Press, Cambridge, 2008).
- [10] R. Cohen and S. Havlin, *Complex Networks: Structure, Robustness and Function* (Cambridge University Press, Cambridge, 2010).
- [11] M. E. J. Newman and M. Girvan, *Phys. Rev. E* **69**, 026113 (2004).
- [12] M. E. J. Newman, *Proc. Natl. Acad. Sci. USA* **103**, 8577 (2006).
- [13] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, *Phys. Rev. E* **70**, 025101(R) (2004).
- [14] J. Reichardt and S. Bornholdt, *Physica D* **224**, 20 (2006).
- [15] S. Fortunato and M. Barthélemy, *Proc. Natl. Acad. Sci. USA* **104**, 36 (2007).
- [16] B. H. Good, Y.-A. de Montjoye, and A. Clauset, *Phys. Rev. E* **81**, 046106 (2010).
- [17] A. Lancichinetti and S. Fortunato, *Phys. Rev. E* **80**, 056117 (2009).
- [18] J. Reichardt and S. Bornholdt, *Phys. Rev. E* **74**, 016110 (2006).
- [19] A. Arenas, A. Fernández, and S. Gómez, *New J. Phys.* **10**, 053039 (2008).
- [20] A. Clauset, M. E. J. Newman, and C. Moore, *Phys. Rev. E* **70**, 066111 (2004).
- [21] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, *Nature (London)* **435**, 814 (2005).
- [22] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, *Proc. Natl. Acad. Sci. USA* **101**, 2658 (2004).
- [23] J. Reichardt and S. Bornholdt, *Phys. Rev. E* **76**, 015102(R) (2007).
- [24] A. Lancichinetti, S. Fortunato, and F. Radicchi, *Phys. Rev. E* **78**, 046110 (2008).
- [25] A. Condon and R. M. Karp, *Random Struct. Algor.* **18**, 116 (2001).
- [26] M. Rosvall and C. T. Bergstrom, *Proc. Natl. Acad. Sci. USA* **105**, 1118 (2008).
- [27] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, *PLoS ONE* **6**, e18961 (2011).
- [28] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikolski, and D. Wagner (2006), <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/3255>.
- [29] V. A. Traag, P. Van Dooren, and Y. Nesterov, *Phys. Rev. E* **84**, 016114 (2011).
- [30] P. Ronhovde and Z. Nussinov, *Phys. Rev. E* **81**, 046114 (2010).
- [31] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, *J. Stat. Mech.* (2005) P09008.
- [32] A. Lancichinetti, S. Fortunato, and J. Kertész, *New J. Phys.* **11**, 033015 (2009).
- [33] G. Bianconi, P. Pin, and M. Marsili, *Proc. Natl. Acad. Sci. USA* **106**, 11433 (2009).
- [34] A. Lancichinetti, F. Radicchi, and J. J. Ramasco, *Phys. Rev. E* **81**, 046110 (2010).