Automatic detection of multilevel communities in complex networks with a scalable community fitness function

Kun Gao, Xuezao Ren, Lei Zhou and Junfang Zhu

School of Science, Southwest University of Science and Technology, Mianyang 621010, Sichuan Province, People’s Republic of China

Corresponding author: Xuezao Ren; Email: renxuezao@aliyun.com

Abstract

Community detection in complex networks has been held back by two obstacles: the resolution limit problem, which restrains simultaneous detection for communities of heterogeneous sizes, and the divergent outputs of heuristic algorithm, which are unfavorable for differentiating more relevant and significant results. In this paper, we propose a renewed method for community detection with a multiresolution and rescalable community fitness function. The scalability of the community fitness function on the one hand exempts our method from the resolution limit problem in heterogeneous networks, and on the other hand enables our method to detect multilevel community structures in deep hierarchical networks. Furthermore, we suggest a strict definition of “plateau,” with which we evaluate the stability of the outputs, and remove unstable and irrelevant ones automatically, without any artificial or arbitrary selection. As a result, our method outperforms most previous methods; it reproduces the expected community structures accurately for various classes of synthetic networks, as well as the ground truths for real-world networks.
1. Introduction

Community, also known as network cluster, is a mesoscopic structure ubiquitous in many systems whose topologies are customarily described by complex networks [1]. Since highlighted by Girvan and Newman in 2002 [2], community structure of network has deserved much attention from researchers, especially physicists and mathematicians [3, 4, 5]. Being distinct from microscopic (single-node level) or macroscopic (whole-network level) statistical properties of complex networks, community structure characteristically reveals functional, relational or even social information from groups of nodes [1]. Community detection aims at exploring optimized divisions of a network into groups of nodes, with the connections (edges) inside each group being denser than those between different groups [6]. It is expected that a community detection method may perfectly recover community structures that are already known from other studies, or make promising predictions on the structures or functions of a given network [2]. Ideally, if certain community structure within a network can be recognized even by visual inspection, then a good method of community detection should not fail to discover it [7].

Nevertheless, to detect communities from a network is always a challenge. Related studies can be traced back all along to graph partitioning in graph theory [8], or hierarchical clustering in sociology [9]. For large graphs, finding an exact solution to a partitioning task has been proven an NP-complete problem [6, 8]—and in the case of complex networks it is even more complicated: the total number of communities is usually unknown [6], the sizes of different communities may differ by orders of magnitude [10], and the overall structure of communities often shows multilevel or hierarchical [10, 11, 12]. Existing methods for community detection generally turn to heuristic algorithms for acceptably good solutions [6]: most typically, many methods use a quality function to assess the validity of each potential partition of the network. This quality function, is customarily formulated into either a modularity function in modularity-based methods [6, 13, 14, 15, 16], or a Hamiltonian of a first principle Potts model [17] in certain dynamical methods [18, 19, 20, 21, 22]. Heuristic algorithm is then implemented to optimize the quality function, i.e., to maximize the modularity, or minimize the Potts Hamiltonian.

Community detection method based on function optimization has been argued to be limited by two major drawbacks. The first one is the well-known resolution limit problem raised by Fortunato and Barthelemy in 2007 [23]. Depending on the numbers of intra connections of communities and the total number of connections within the whole network, a modularity optimization method tends to merge small communities (even if they are well-defined clusters as complete graphs) into larger but sparser ones. This reveals a fact that modularity optimization method can’t find communities of small sizes, like microscope can’t find microbes beyond its resolution range. For quality functions other than modularity, the same phenomenon has also been observed [24]. To overcome such a limit, a variety of “multiresolution” methods [3, 16, 19, 21] and “resolution-limit-free” [22, 25] methods have been suggested. The former uses a tunable parameter to alter the resolution, and enables the detection for communities of different levels in different resolution scales, while the latter refrains from using a “null model” to avoid the resolution limit. However, a further study by Lancichinetti and Fortunato [7] pointed out that the resolution limit problem is actually induced simultaneously by two opposite tendencies: the tendency of merging small clusters, and the tendency of breaking large ones. When communities in a network have very different sizes, it becomes impossible for an optimization procedure, no matter modularity-based or Hamiltonian-based, to avoid both biases simultaneously. Multiresolution and resolution-limit-free methods seem to have outperformed other methods only because the cluster sizes used in their tests are too close to one another, spanning less than one order of magnitude [7]. When cluster sizes vary over up to two orders of magnitude, as in
many real-world networks [10, 26], existing multiresolution and resolution-limit-free methods also fail to detect the expected community structures [7].

The second drawback of community detection is that finding an optimal partition for a given network is normally infeasible. It has been recognized that the modularity landscape of a network is often “glassy,” including an exponentially growing (with system size) numbers of local maxima [7, 27]. These local maxima may all be very close to the “global” maximum in terms of modularity values, but the corresponding partitions of the network can be topologically very different from one another [27]. This implies that not only an exactly optimal partition for a network is intrinsically unreachable [7], but the accessible solutions in practice can be largely unstable and inconsistent. Multiresolution methods furtherly aggravate the problem. Communities detected at “irrelevant” resolution scales are often messy: they are mostly incomprehensible and, for all practical purposes, uninformative. Although it has been argued that inquiring which is the “best” or “most relevant” scale of resolution is an ill posed question [3], many methods still manage to find out the most stable resolution scales within which persistent structures of communities can be detected. If an identical community structure can be repeatedly detected (as local maxima) at a series of different resolutions, this community structure is considered as a stable solution to the community detection task. The existence of such stable solutions is an “observed fact” [3]: they display strong “plateaus” against the resolution parameter [21]. It has become popular in previous literature to rank the relevance of different resolution scales by the strength of plateaus; stable solutions retrieved from strong plateaus have been demonstrated to be frequently consistent with the \textit{a priori} knowledge about the network [3, 16, 21].

Nonetheless, existing methods using the stability of plateaus are not yet satisfying. In principle, each plateau is supposed to reflect multiple times of convergence to a same topology of community structure at varying resolutions. However, no explicit comparisons among the topologies of different data points within a same plateau has been presented in previous literature, leaving a \textit{suspect} that these data points may not surely represent a same community structure at all. Some methods simply distinguish plateaus by the numbers of communities [3, 28, 29], while some others as in [16] by the values of the modularity function. According to our discussion above [27], none of such definitions can guarantee each plateau as defined represents a uniform and consistent community structure. On the other hand, some methods as that in [21] execute information-based quantitative comparisons among the topologies of network partitions detected on multiple “replicas” of the network at a \textit{fixed} resolution—yet replicas detected at different resolutions still remain not compared. Therefore, we suggest a strict definition of “plateau,” which involves a complete and thorough comparison among the topologies of all replicas detected at all resolutions, is urgently needed.

In this paper, we suggest a \textit{renewed} method to detect communities at multiple resolution scales. To avoid the resolution limit problem, we refrain from using the traditional modularity function [6] and its variations as in previous literature [3, 19]; instead we take a modified “fitness function” [16] as our community quality function. We run a heuristic algorithm to maximize this fitness function. Through an overall comparison among the topologies of all best solutions obtained at all resolutions, we remove unstable solutions and adopt only the most stable ones as our outputs for the community detection task.

2. Method

A typical community detection method of multiresolution includes three components: (1) a community quality function with tunable resolution parameters, (2) a heuristic algorithm to optimize
the community quality function, and (3) a strategy to filter out meaningful results from “messy” outputs. In this section, we describe our method. We choose a slightly modified “fitness function” [16] as our community quality function, and adopt a “Louvain” algorithm [30] to optimize it. Then we develop a strategy to filter the outputs and retrieve the most significant community structures by the stability of “plateaus,” where our definition for “plateau” will be more explicit and strict than those used in previous literature.

### 2.1 Community Fitness function

In this paper, we choose the “fitness function” proposed by Lancichinetti and Fortunato [16] as our community quality function. We choose this function because it is by design more “scalable” than traditional quality functions and seemed to be capable of avoiding the resolution limit problem (see our discussion in Section 4). The original form of the fitness function proposed in [16] is

$$F = \sum_{\mathcal{G}} f^{\mathcal{G}} = \sum_{\mathcal{G}} \frac{k_{in}^{\mathcal{G}}}{(k_{in}^{\mathcal{G}} + k_{out}^{\mathcal{G}})^{\alpha}}.$$  \hspace{1cm} \text{(Formula 1)}$$

Here $\mathcal{G}$ denotes a community given by a certain partition of the network, $f^{\mathcal{G}}$ quantifies the fitness (i.e., quality) of community $\mathcal{G}$, and $F$ assesses the fitness of the whole partition: larger values of $F$ indicate more reasonable community structures. $k_{in}^{\mathcal{G}}$ and $k_{out}^{\mathcal{G}}$ in the formula stand for the in-degree and out-degree of community $\mathcal{G}$, defined in the same way as those in previous literature such as [12]. $\alpha (\alpha>0)$ is a resolution parameter that tunes the resolution: large values of $\alpha$ yield small communities, while small values of $\alpha$ instead deliver large communities [16].

The original form of $F$ as shown in Formula 1 can be directly used in our method; actually we do use it in many of our calculations in the Result part of this paper. For networks having relatively simple structures, $F$ is sufficiently good. Yet for networks whose structures are even more complex, such as those having multilevel or hierarchical structures, it will be helpful to introduce an additional parameter to improve the “rescalability” of the fitness function. In this paper, we adopt the following modified form of the fitness function

$$F_\alpha^\beta = \sum_{\mathcal{G}} (F_\alpha^\beta)^\mathcal{G} = \sum_{\mathcal{G}} \frac{(k_{in}^{\mathcal{G}})^\beta}{(k_{in}^{\mathcal{G}} + k_{out}^{\mathcal{G}})^{\alpha}}.$$ \hspace{1cm} \text{(Formula 2)}$$

Here the power exponent $\beta (\beta \geq 1)$ is our newly introduced “scaling factor;” when $\beta = 1$, Formula 2 degrades to its original form as in Formula 1. The scaling factor $\beta$ amplifies the varying range of the resolution parameter $\alpha$: in Appendix, we estimate for each given value of $\beta$, the “relevant” range of $\alpha$ is $\beta-1 < \alpha < 2\beta-1$. Here the upper bound $2\beta-1$ prevents unexpected splitting of large communities, while the lower bound $\beta-1$ avoids inappropriate merging of small communities. For networks having only a single community level, scanning such a resolution range can be sufficient for the expected community structure being detected. But for networks having multilevel or hierarchical community structures, within the above range only the lowest community level (i.e., communities of smallest sizes which cannot be split further) can be detected. To obtain communities of higher levels, it is required to merge small communities into larger ones, thus the lower bound $\beta-1$ must be relaxed. In practice, we run our algorithm with the resolution parameter $\alpha$ varying between 0 and $2\beta-1$, which enables our detection on different levels of communities at different resolution scales.
2.2 Heuristic optimization algorithm

Since optimizing a community quality function has been proven an NP-complete problem [6, 8], heuristic algorithms are generally adopted to obtain the best accessible solutions. Early methods such as those in [2, 6] usually have heavy demands on computational resources, while more recently a number of faster algorithms have been proposed [13, 26, 30, 31, 32]. Among them, the “Louvain” algorithm raised by Blondel et al. [30] has been widely used due to its prominent efficiency and high accuracy. The label “Louvain” comes from the affiliation (UCLouvain) of the authors; alternatively, it is also called the “BGLL” algorithm by the authors’ initials. The original Louvain algorithm is a greedy algorithm to optimize the standard modularity function proposed by Newman [30]; similar algorithm has also been adopted to optimize a Potts Hamiltonian by other methods [21, 22].

We employ the Louvain algorithm to optimize our community fitness function (formula 2) in this paper. Here we briefly describe the steps of the algorithm; for more details, please refer to [30].

1. **Initialize communities.** At the very beginning, each node of the network is assigned to a different community. A network consisting of \( N \) nodes is then divided into \( N \) communities of size 1.

2. **Optimize communities of the lowest level.** Sequentially consider each node of the network and scan its neighboring communities (i.e., communities sharing at least one edge with the node in focus). Calculate the potential gains of the fitness function (formula 2) if the node in focus was moved from its original community to each of its neighboring communities. Place the node in the community which offers a maximum value of fitness.

3. **Iterate until convergence.** Repeat step (2) until a maximum value of formula 2 is reached where no more moves of any node may further increase this value. During this process, the sequence of orders of nodes is randomized every time a new round of iteration on step (2) is started.

4. **Merge communities into a higher level.** Consider each community obtained at the convergence of step (3) as a fixed module; hereafter all its members (nodes) must be moved together. Repeat steps (2) and (3) by moving each module in the same way as moving a node. During this process, connected modules gradually condense to form communities of a higher level, until a maximum value of formula 2 is reached.

5. **Iterate until convergence at the highest level.** Repeat step (4) and detect communities at all levels, until the highest level is reached where a further detection to an even higher level can no longer increase the fitness function.

6. **Output communities.** Communities of different levels detected by the above steps (1) to (5) form a hierarchical structure; each level can be independently outputted, if needed. In this paper, we only take the highest level as our output since at this level the fitness function finally converges to a maximum value among all levels.

As a heuristic method, the Louvain algorithm does not guarantee in multiple realizations, the same procedures implemented on a same network always converge to exactly the same solution. As a customary use of the algorithm, at each fixed resolution, we implement the above steps on multiple “replicas” of the given network [21]. Then by the strategy we propose in the next section, we adopt the replica having not only the highest fitness value but also a unique topology as the best solution obtained for this resolution.
2.3 Strategy of filtering the outputs

When the resolution parameter $\alpha$ varies, the Louvain algorithm optimizing our fitness function may converge to different solutions corresponding to different topologies of community structures. As a matter of fact, not all resolution scales are necessarily stable, or apparently yield any reasonable output. Previous methods customarily retrieve the most “relevant” resolution scales by the stability of “plateaus” [3, 16, 21, 27, 28, 29], only the conception of “plateau” was everywhere loosely defined (see our argument in the Introduction). In this paper, we suggest a much stricter definition for “plateau,” and correspondingly a stricter strategy to identify such plateaus. By our definition, a plateau is a continuous scale of resolution within which a heuristic optimization algorithm uniformly converges to a unique topological structure of community. To identify such plateaus, it is required to compare the topological structures of not only the solutions obtained at the same resolution, but also those obtained at different resolutions. In practice, we adopt the following strategy to discover such plateaus:

1. At each fixed resolution, with fixed values of parameters $\alpha$ and $\beta$, we implement the Louvain algorithm on multiple replicas of the given network. Among the outputs of all replicas, we adopt the one with the highest value of the fitness function as the best solution obtained at this resolution. In addition, we require the topology of this best solution must be unique: in case two or more solutions have equally highest values of the fitness function but represent different topological structures of community, these solutions will be all abandoned, and the corresponding resolution will be considered unstable and does not contribute to any potential plateau.

2. At different resolutions, with varying values of parameter $\alpha$ (during which the value of $\beta$ is still fixed), we run the above step (1) and obtain the best-and-unique solutions at all possible resolutions except those unstable ones. Then we compare the topologies of these best-and-unique solutions and organize their corresponding resolution scales into different plateaus: resolutions within a same plateau must correspond to an identical topology of community structure.

Plateaus defined and identified as above ensure all resolutions within a same plateau strictly yield an identical topology of community structure—only then can the stabilities (or significances) of plateaus be measured by their scales of persistence on the axis of the resolution parameter $\alpha$. Different plateaus usually represent different levels of communities. It has become a convention in previous literature [3, 16, 21, 27, 28, 29] to take the community levels revealed by large plateaus as the most significant results since they almost always have comprehensible meanings—but small plateaus can also be meaningful. In practice, we can adopt any community levels that fit our need.

For networks whose fitness landscape is really glassy, the fitness function often converges to unstable solutions. Sometimes small plateaus can be observed in unstable resolution scales, which reflect the transition between different community levels. These small plateaus are not really stable, but emerge only by chance. Then a further step may help us remove these small unstable plateaus. We run our method in multiple “ensembles,” which each ensemble involves multiple realizations of the same algorithm on multiple replicas of the given network. Within each ensemble we obtain an independent set of plateaus. We then take an “intersection” over all sets of plateaus obtained in different ensembles as the final output: if at a certain resolution, different ensembles yield different topologies of communities, this resolution will be considered unstable and knocked out from the plateaus in the final output. Namely, by such an intersection we not only require “one plateau one
topology”, but also require this topology be uniform across all ensembles. This further constraint effectively removes the small unstable plateaus, without making any artificial or arbitrary judgement.

3. Results

In this section, we demonstrate the effectiveness of our method on three synthetic networks and three real-world networks, which all have been widely used as benchmarks in previous literature. All networks selected have a priori structures of community: synthetic networks have prearranged communities during their constructions, while the real-world networks all have standard partitions given by previous studies. We then investigate the performance of our method in reproducing these a priori community structures for these networks.

The following calculations are implemented in the same way for all networks: with each fixed value of the scaling factor $\beta$, we vary the resolution parameter $\alpha$ from 0 to $2\beta-1$, with a stepwise increment $\Delta\alpha=0.01$. At each resolution $\alpha$ (with a given $\beta$), we carry out 20 ensembles of community detection each involving 1000 realizations of the Louvain algorithm implemented on 1000 replicas of the given network. Plateaus in the final output are then obtained through an intersection over the outputs of all 20 ensembles.

This section is organized as following: in section 3.1, we test our method’s ability of detecting multilevel community structures on hierarchical but homogeneous networks; in section 3.2, we test the robustness of our method on heterogeneous networks with communities of varying sizes; finally, in section 3.3, we inspect its performance on real-world networks.

3.1 On synthetic networks with hierarchical community structures

We firstly test our method with two hierarchical but homogeneous networks: the homogeneous in degree network (H13-4) proposed by Arenas et al. [1], and the scale-free hierarchical networks (RB) named after Ravasz and Barabasi [10]. Both these networks have hierarchical structures, in which communities of the same level have homogeneous sizes. We expect our method to reproduce the multilevel community structures therein at varying resolutions.

The H13-4 network includes 256 nodes and 2 prearranged community levels, whose topology is shown in figure 1 (a). Every node of the H13-4 network has 18 randomly connected edges, among which 13 are restricted within the same first-level community (see the red edges in figure 1 (a)), 4 stretch out of the first-level community but are still within the same second-level community (see the blue edges in figure 1 (a)), and only one edge reaches a different second-level community (see the brown edges in figure 1 (a)). The prearranged community levels within the H13-4 network are explicit: the first level includes 16 communities each containing 16 nodes, then every four first-level communities constitute a second-level community; besides, a third but trivial community level might be merging the whole network into one single community. We implement our method on the H13-4 network in 20 ensembles with $\beta=1$. Figure 1 (b) shows the output of one single ensemble: three major plateaus emerge, corresponding respectively to the three prearranged community levels (including the third and trivial level), and the related network partitions also perfectly match the prearranged structures of the H13-4 network. At very high resolutions (when $\alpha\geq 0.90$), a number of small plateaus emerge (see the “noises” denoted by red crosses in figure 1 (b)), dividing the network into irregular numbers of communities such as 25, 31, 32 and 71 (not shown). These small plateaus are unstable and noise-like: they all disappear in the intersection over 20 ensembles, while the three major plateaus all pass through the intersection without being affected (see figure 1 (c)).
In conclusion, our method has perfectly reproduced the expected community structures for the H13-4 network.

Figure 1. Communities in the H13-4 network. (a) Topology of the H13-4 network. With inner edges shorter than outer edges, the network is naturally displayed as 16 communities on the first level, or 4 communities on the second level. There are three types of edges in the network: red edges within the same first-level community, brown edges between different second-level communities, and blue edges between different first-level communities but still within the same second-level community; the proportions of these three types of edges are 13:1:4. (b) Plateaus in the output of one single ensemble of our community detection with the scaling factor \( \beta = 1 \). There are three major plateaus (with texts indicating the numbers of communities) and a few noise-like small plateaus (denoted by red crosses). (c) Intersection over all plateaus identified in 20 ensembles, which retains all the stable plateaus and removes the noises.

It should be noted that although we exhibited different plateaus by their corresponding numbers of communities in the vertical axes of figure 1 (b) and (c), our plateaus are identified not simply by their numbers of communities, but through a comparison on the topologies of all solutions obtained from all replicas, ensuring all data points within a same plateau strictly represent exactly the same community structure. On the other hand, since the H13-4 network has relatively simple community structure, with \( \beta = 1 \) our result is already good enough; \( \beta > 1 \) simply yields the same result.

Next we detect communities for the Ravasz-Barabasi (RB) networks [10]. The smallest RB network is RB5, which is a complete graph consisting of 5 nodes and 10 edges. RB5 is a basic unit to constitute higher levels of RB networks. As shown in figure 2 (a), we call node 0 the central node of the RB5 unit, and all other nodes peripheral nodes. Five RB5 units, one in the center and four on the periphery, constitute an RB25 network, as shown in figure 2 (b) and (c). These RB5 units are connected in such a way that every peripheral node of the peripheral units is connected to the central node of the central unit, but the peripheral units themselves are not connected to one another. Note
that in figure 2 (b) and (c), for convenience we did not shift the central node of each RB5 unit slightly off the center as in figure 2 (a); each RB5 unit in figure 2 (b) and (c) is still a complete graph containing 10 edges, only the diagonal edges become invisible by overlapping with other edges. In the same way, five RB25 networks constitute an RB125 network, as shown in figure 3. Obviously, the RB network is a fractal-like hierarchical network, which can grow infinitely. We implement our community detection method on the RB networks with two different scaling factors, $\beta=1$ and $\beta=2$. On small networks such as RB25 and RB125, our method does not produce any unstable plateaus even in single ensembles; community structures and plateaus detected for these networks are shown in figures 2 and 3. For large RB networks such as RB625, RB3125 and RB15625, we still take an intersection over multiple ensembles to remove the small and unstable plateaus; plateaus detected for these networks can be seen in supplementary figures 1 and 2.

![Network Structures](image)

**Figure 2.** Communities in the RB25 network. (a) An RB5 network is a complete graph consisting of 5 nodes and 10 edges. We call node 0 the central node of the RB5 network, and all other nodes peripheral nodes. (b) An RB25 network is composed of five RB5 units, one in the center and four on the periphery. Every peripheral node of the peripheral unit is connected to the central node of the central unit, but different peripheral units are not connected to one another. Ideally an RB25 network is expected to be divided into 5 communities: each RB5 unit makes a community. (c) A slightly different partition to (b), which divides the RB25 network into 6 communities. Four peripheral RB5 units make four communities, and the central unit is divided into two communities: the central node makes one community and all other nodes make the other. (d) and (e): Plateaus obtained from single ensembles of our community detection method with $\beta=1$ and $\beta=2$. 
Figure 3. Communities in the RB125 network. (a)–(d): Four different community structures detected by our method within the RB125 network. In each subfigure, we exhibit different communities with different colors and shapes of nodes (note triangles in different directions also represent different communities). (a) shows the “natural” partition of the network: each RB5 unit makes a community; (b) shows a most stable partition of the network into 30 communities on the first level: each RB25 unit splits to 6 communities as in figure 2 (c); (c) shows a stable partition on the second level of the network into 10 communities; and (d) shows a partition to 26 communities on the first level but with a “relaxed stringency:” except the RB5 unit in the very center of the whole network is divided into 2 communities (see the hollow red circle and red squares in the figure), all other RB5 units are kept intact. (e) and (f): Plateaus obtained by our method in single ensembles with $\beta=1$ and $\beta=2$. In (e), with $\beta=1$ only the first community level can be detected, but there emerge three different partitions.
(plateaus) for it: 30 communities (as in (b)), 25 communities (the “natural” partition as in (a)) and 26 communities (as in (d)). In (f), with $\beta=2$ both community levels are detected. On the first level, only the 30-community structure is detected since it is most stable, and on the second level, the 10-community structure (as in (c)) is rightly detected.

The prearranged community structure within an RB network is obvious: an RB network with $5^n$ nodes (hereafter we call it an “RB$5^n$ network”) is expected to be naturally divided into $5^{n+1}$ RB5 units on the lowest community level, or $5^{n-2}$ RB25 units on a higher level, and so on, forming a hierarchical structure of $n$ levels. However, such “natural” partitions should not be taken for granted. One problem is, the central node of a central unit (see the hollow circles in figure 2 (c) and figure 3 (b)) is designed to be connected to every peripheral node in peripheral units: this central node has a large out-degree, which would seriously dilute the fitness of the central community. As a result, the central community is apt to break down, separating its central node as an independent community. On the lowest community level, each RB25 unit splits to 6 communities: four peripheral RB5 units make four of them, while the central RB5 unit breaks down into two communities, as in figure 2 (c). This would divide an RB$5^n$ network ($n \geq 2$) into $6 \times 5^{n-2}$ communities: figure 3 (b) shows an example that an RB125 network is divided into 30 communities; comparing to the “natural” partition shown in figure 3 (a), this partition is more stable and is discovered by our most persistent plateaus in figure 3 (e) and (f). Similarly, on a higher level, each RB125 unit splits to 10 communities as in figure 3 (c): four peripheral RB25 units make four communities, and the central RB25 unit splits to six. On this level, an RB$5^n$ network would be divided into $10 \times 5^{n-3}$ communities. Following this regulation, on the $m$-th community level (here $m=1$ stands for the lowest level), an RB$5^n$ network will be divided into $(4m+2) \times 5^{n-m-1}$ communities: it firstly splits to $5^{n-m-1}$ RB$5^{m+1}$ units, then each RB$5^{m+1}$ unit splits to $4m+2$ communities including four RB$m$ units, four RB$5^{m-1}$ units, and so on, till finally the central RB5 unit breaks down into 2 communities as in figure 2 (c); supplementary figure 1 (a) visualizes such a topology for an RB625 network being divided into 14 communities on the third community level. Table 1 exhibits a full list of the numbers of communities on different community levels of different RB networks; next we demonstrate these community structures are most stable, and will be rightly discovered by our method.

| RB$5^n$ Networks | Community Levels ($m$) |
|------------------|-----------------------|
|                  | 1 (Lowest) | 2 | 3 | 4 | 5 | 6 |
| RB25 ($n=2$)     | 6          | / | / | / | / |
| RB125 ($n=3$)    | 30         | 10| 1 | / | / |
| RB625 ($n=4$)    | 150        | 50| 14| 1 | / |
| RB3125 ($n=5$)   | 750        | 250| 70| 18| 1 |
| RB15625 ($n=6$)  | 3750       | 1250| 350| 90| 22 |

Table 1. Numbers of communities in the most stable partitions for an RB$5^n$ network on the $m$-th community level. Here $m=1$ stands for the lowest level, which divides the network into smallest communities that cannot split further; $m$ is restricted to be no larger than $n$. Numbers in this table all follow such a formula: $(4m+2) \times 5^{n-m-1}$. Our method detects almost all these structures, except for the second and fifth levels of the RB15625 network, which divides the network into 1250 and 22 communities respectively.
On the RB networks, our method detects different community structures with different scaling factors $\beta$. For all networks, with $\beta=1$ we detect only two community levels: the highest (but trivial) level that merges the whole network into one community, and the lowest level that splits the network all along to the smallest RB5 units; intermediate levels, if exist, are all omitted. On the other hand, with $\beta=1$ we detect different structures for the lowest community level. Except the stable partitions as described above, whose numbers of communities have been listed in the first column of table 1, our method also discovers the prearranged communities (i.e., the “natural” partitions), but only for small networks such as RB25 and RB125 (see figure 2 (b) and figure 3 (a)). These natural partitions are not really persistent: even for RB125, the corresponding plateau is already very small. Besides, with $\beta=1$ an RB125 network can alternatively be divided into 26 communities—this can be done by splitting only the central RB5 unit of each RB125 module to 2 communities, but keeping all other RB5 units intact (see figure 3 (d)). Comparing to the stable partitions described above, these variant structures can be considered as products of “relaxed stringencies.” Following this regulation, an RB$^n$ network ($n \geq 3$) can be divided into $26 \times 5^{n-3}$ communities on the first community level; this explains the plateaus for 26, 130, 650 and 3250 communities in the RB125, RB625, RB3125 and RB15625 networks respectively shown in figure 3 (e) and supplementary figures 1 (b), 2 (a) and 2 (c). Moreover, an RB625 network can be divided into 126 communities with a further relaxed stringency, in which only the central RB5 unit of each RB625 module is divided to 2 communities. Yet such a stringency has been too relaxed that in our result it only produces a very unrecognized plateau for the RB625 network (see supplementary figure 1 (b)), but hasn’t been observed anywhere else.

The above results for $\beta=1$ is still not satisfying: one major problem is, the highest and lowest community levels have occupied almost all the resolution scales; intermediate community levels are seriously compressed and cannot be detected at all. To fix this problem, we employ a larger scaling factor $\beta$, which would amplify the varying range of the resolution parameter $\alpha$ from $(0, 1)$ to $(0, 2\beta-1)$. It turns out that with $\beta \approx 2$, our method detects all community levels including the intermediate levels for the RB25–RB3125 networks. On each community level, it only outputs the most stable community structure as listed in table 1, but discards all the variants due to relaxed stringencies. We show in figures 2 (e), 3 (f) and supplementary figures 1 (c) and 2 (b) the plateaus obtained with $\beta=2$; with $\beta>2$ we simply get similar results. These results prove that a larger scaling factor $\beta$ does rescale the resolutions of intermediate community levels, and facilitate our detection on them effectively.

Nevertheless, the effectiveness of the scaling factor $\beta$ as in formula 2 still has a limit. It enables us to detect up to five community levels on the RB25–RB3125 networks, but when the network becomes even larger, such as on an RB15625 network, our method starts to fail in detecting certain community levels. For example, in supplementary figure 2 (d), the plateaus for the second and fifth community levels (corresponding to 1250 and 22 communities) are missing, and the corresponding resolution scales within which such community levels are expected to emerge turn out to be unstable. Actually, the scaling factor $\beta$ does not rescale the resolutions evenly for all community levels: the highest and lowest levels always occupy most of the resolution scales. For example, when $\beta=2$, for RB networks of different sizes, intermediate community levels are always limited roughly within $\alpha=1-1.6$. When the number of community levels increases with the network size, some intermediate levels will be compressed and become difficult to be detected. Further increasing the value of $\beta$ does not necessarily solve the problem. As in supplementary figure 3, when $\beta$ increases, the resolution scale for all intermediate community levels does not expand much. Within the scope of this paper, the limit of our method is to detect up to five community levels on an RB network. To detect more community levels, an improved method should be worth studying in the future.
3.2 On synthetic networks with heterogeneous community sizes

In [7], the authors argue that in heterogeneous networks whose community sizes vary vastly, multiresolution methods lose their effectiveness; in this section we test our method in heterogeneous networks. We employ the famous LFR network proposed by Lancichinetti, Fortunato, and Radicchi [33], which has been widely used as a benchmark network to access the effectiveness of community detection methods. The LFR network is heterogeneous: both the degrees of nodes and the sizes of communities follow power-law distributions, thus communities in an LFR network may have very different sizes. Each node in the network is preassigned to a community by construction: it shares a fraction of $1-\mu$ of its connections with the other nodes assigned to the same community, and the rest fraction $\mu$ with nodes in other parts of the network; $\mu$ is called a mixing parameter. Obviously, the value of $\mu$ determines the level of significance of the preassigned community structure; large values of $\mu$ obscure such a structure, or even make it suspicious. Especially, when $\mu \geq 0.5$, the preassigned communities in an LFR network no longer fit the definition of community in either a strong sense or even a weak sense [34], which respectively require $k_{in} > k_{out}$ for every node, or $\Sigma k_{in} > \Sigma k_{out}$ for every community. However, since outer connections can be diluted by a large number of communities, while inner connections are more concentrative, the preassigned community structure, although violates the customary definitions of community under large values of $\mu$, may still be a best available solution. A good community detection method is expected to reproduce the preassigned community structure for an LFR network at least under small values of $\mu$; some methods even do so when $\mu$ is fairly large. Since the LFR network is constructed by a random process, a normalized mutual information (NMI) [35] is frequently employed to evaluate the consistency between the detected community structures and the preassigned ones; NMI equals to 1 means that they agree with each other perfectly.

In [7], the authors tested the effectiveness of a bunch of community detection methods on the LFR network; they argued that multiresolution methods outperform other methods only if the sizes of different communities are “too close to one another” (as in the networks shown in figure 4 (b)-(e)). In large LFR networks, when the community sizes vary from 10 to 1000 (as in in figure 4 (f)-(i)), multiresolution methods fail to detect the expected community structures even when the mixing parameter $\mu$ is very small. Figure 4 shows the effectiveness of our method on the same LFR networks constructed with exactly the same parameters as in [7]. Since the LFR network contains only a single community level, for each network we only run a single ensemble of our method with $\beta=1; \beta=2$ simply yields similar results. We show in the figure the values of our best NMIs with the preassigned communities at different values of $\mu$. $\mu_1$ indicates a threshold below which (i.e., when $\mu \leq \mu_1$) our best-fit communities (i.e., those having the best NMIs with the preassigned ones) are discovered by our strongest plateaus, and the corresponding data points are shown in black squares. On the other hand, when $\mu > \mu_1$, the best-fit communities are suggested by the second strongest plateaus (data points shown in red crosses), while the strongest plateaus suggest a trivial solution that merges the whole network into one community. $\mu_2$ indicates another threshold at which our communities start to deviate from the preassigned ones; when $\mu \leq \mu_2$, our best NMIs always equal to 1. Multiresolution methods tested in [7] mostly perform well on the networks shown in subfigures (b)-(e), which are small and relatively homogeneous—but they all perform much worse on the networks in subfigures (f)-(i), which are more heterogeneous. In contrast, our method, being also a multiresolution method, reproduces the expected communities perfectly for all networks, at least when the mixing parameter $\mu$ is below 0.5. This indicates that our method deals with heterogeneous networks much better than previous methods; it can be expected to be immune from the resolution limit problem due to network heterogeneity as described in [7].
Figure 4. Community detection on the LFR networks. (a) Topology of an LFR network: different communities have different sizes. (b)-(i) Normalized mutual information (NMI) between the communities detected by our method and the preassigned ones for the LFR networks with different parameters. Parameters of the LFR networks include: the size of the network $N$, the average degree $\langle k \rangle$ and the maximum degree $k_{\text{max}}$ of nodes, the power-law exponents of the degree distribution $\tau_1$ and the community size distribution $\tau_2$, and the minimal and maximal sizes of communities $s_{\text{min}}$ and $s_{\text{max}}$. In all subfigures, $\langle k \rangle=20$ and $\tau_1=2$; other parameters are labelled in each subfigure.
3.3 On real-world networks

In this section, we investigate three real-world networks: Zachary’s karate club network [36], Lusseau et al.’s dolphins social network [37], and the American college football network [2]. We apply our method on these networks to detect their communities, and compare our results with the standard divisions given by previous literature.

**Figure 5.** Communities in the karate club network. (a) Structure of the network. Here we show three representative divisions of the network: we show our division by different fill colors (blue, maroon, green and turquoise), Newman and Girvan’s division [6] by different boarder colors (red, yellow, cyan and pink), and Medus et al.’s division [38] by different node shapes (circle, square, hexagon and octagon). Obviously, all these divisions split the network into four communities. The dashed line in the figure divides the network into two parts, corresponding to a fission which had actually happened to the karate club. (b)-(c) Plateaus obtained through the intersection over 20 ensembles each involving 1000 realizations of the Louvain algorithm at each resolution with $\beta=1$ and $\beta=2$. The largest plateau among all levels except the 1-community level is specifically exhibited by red crosses, and numbers above certain plateaus indicate the corresponding numbers of communities. Below for figure 6 (c)-(d) and figure 7 (b)-(c), the same also holds.
Figure 5 shows the community structures within the karate club network. This network consists of 34 nodes representing 34 members of a karate club; connections between nodes imply consistent interactions between the corresponding members outside the club. Due to a disagreement between the club president (John A.) and a part-time instructor (Mr. Hi), the club later split into two clubs, the officers’ and Mr. Hi’s; members of the club also diverge to follow their own favorite leader (see the communities split by the dashed line in figure 5 (a)). Community detection methods in previous literature have attempted to recreate such a fission by mathematical models; among them the results given by Newman and Girvan [6], and by Medus et al. [38], are largely accepted as two standard divisions for the karate club network.

Being different to the synthetic networks we investigated in section 3.2, which display “discrete” community levels, real-world networks generally have more “continuous” levels of communities. For example, within the karate club network we detected 11 different community levels with β=1 and 2, which are all stable and unique at certain resolution scales, see figure 5 (b)-(c) for the plateaus, and supplementary figure 4 for the network divisions. These divisions roughly form a hierarchical structure: communities of a lower level are in most cases subsets of communities of a higher level. Reassembling of communities is minor: it occurs only between few neighboring community levels; for the karate club network it occurs only when 4 communities split to 5, and 5 communities split to 6. Among all levels, the most representative division should be the 4-community structure; it can be detected with all values of β (including β>2, which is not shown since it only yields similar results as β=2). In figure 5 (a), we visualize such a division by nodes filled with different colors (blue, maroon, green and turquoise). In contrast, we denote the division given by Newman and Girvan [6] through their random walk betweenness method by nodes having different boarders (red, yellow, cyan and pink), and the division proposed by Medus et al. [38] by nodes in different shapes (circle, square, hexagon and octagon). It can be observed that our division is highly consistent with Newman and Girvan’s—the only difference is the classification for node 10. Node 10 has only two neighbors, node 3 and node 34; in our division it is classified to the same community as the former (i.e., node 3), while in Newman and Girvan’s it is classified to that of the latter (node 34). Medus et al.’s classification for node 10 agrees with Newman and Girvan’s. Besides, Medus et al.’s classification for nodes 24 and 28 is peculiar: they are classified to the community of square nodes rather than hexagonal nodes. But the fact is, except the connection between nodes 24 and 28 themselves, node 24 has three connections to hexagons but only one connection to a square (node 26), while node 28 has one connection to a square (node 25), and one connection to a hexagon (node 34). By Medus et al.’s classification, node 24 would have a higher out-degree than its in-degree, and node 28 would have equal out-degree and in-degree. In contrast, our classification ensures both these nodes have higher out-degrees than in-degrees, thus should be considered as being more reasonable than Medus et al.’s.

Comparing to the actual fission of the club, our 2-community structure shown in supplementary figure 4 fails to fit such a ground truth. Yet except the 2-community level, communities of any other levels can be merged to recreate the clubs after the fission quite well. For instance, in figure 5 (a), the blue nodes and turquoise nodes roughly recreate Mr. Hi’s club, while the maroon nodes and green nodes recreate the officers’; among them only nodes 9 and 10 are misclassified. As we have discussed above, node 10 has only two neighbors: node 3 joined Mr. Hi’s club, and node 34 joined the officers’. With our method, it is not sufficient to determine which club node 10 should join based on its own network connections only. In principle for node 10 itself, neither choice is better than the other; in previous literature such as [36], node 10 has also been identified as a member having no faction (see Table 1 in [36]). We chose to classify node 10 into Mr. Hi’s club only to maximize the global fitness. As for node 9, it evidently has more connections to nodes in the officers’ club than
to nodes in Mr. Hi’s. According to [36], node 9 is a weak supporter of the president, John A., but he chose to join Mr. Hi’s club only for his black belt; all methods including Zachary’s own model had misclassified the club for node 9. Therefore, our classification can be accepted as it has largely recreated the ground truth.

Figure 6. Communities within the dolphins’ social network. (a) Newman and Girvan’s divisions for the dolphins’ network. Rectangles and ellipses differentiate two major groups, in which ellipses can be further divided into four smaller communities, indicated by different colors (cyan, red, purple and turquoise). (b) Our divisions for the dolphins’ network. It can be divided into two communities by different shapes of nodes (rectangle or ellipse), or seven communities by different colors of nodes (blue, orange, cyan, red, turquoise, green and purple). The dashed line in both (a) and (b) indicates an allegedly known division of the dolphins’ society [6] due to a temporary leave of a “central” individual, SN100. (c)-(d) The same as figure 5 (b)-(c).

As for the dolphins’ social network (hereafter we call it the “dolphins’ network” for short), it was compiled by Lusseau and his collaborators from seven years of filed studies on a bottlenose dolphins’ society living in Doubtful Sound, New Zealand [37, 39]. To our knowledge, the first version of this network was established in [39], including 40 individuals of dolphins; an extended version including 62 vertexes and 159 edges was published in [37] right after. Within the dolphins’ network, vertexes represent individuals of dolphins, while edges imply associations between
dolphin pairs occurring more often than expected by chance [40]. Newman and Girvan firstly split this network into two communities in [6], which allegedly correspond to a “known division” of the dolphins’ society. The larger one of these two communities can be further divided into four smaller communities. In figure 6 (a), we visualize the structure the dolphins’ network, as well as Newman and Girvan’s divisions of this network to 2 or 5 communities [6]; such divisions have later been cited by both Newman and Lusseau [40, 41]. In [40], the smallest community containing only two nodes (Zipfel and TSN83, see the purple nodes in figure 6 (a)) was merged into a larger community (i.e., the community shown in red in figure 6 (a)) so that the total number of communities decreased to four.

In the same way as on the karate club network, we apply our method on the dolphins’ network with $\beta =1$ and 2; plateaus obtained by the intersection over 20 ensambles are exhibited in figure 6 (c) and (d). Similar to the karate club network, the dolphins’ network also contains multiple levels of communities: the number of communities of different levels varies from 1 to 21, roughly building a hierarchical structure, with minor reassembling of communities between few of the neighboring levels. Among all community levels, the 2-community structure has been detected with all values of $\beta$; in figure 6 (b) we exhibit such a structure by different shapes of nodes (rectangle and ellipse). Comparing to the division given by Newman and Girvan [6], our division only misclassified one node, SN89, among all 62 nodes. At higher resolutions, our method divides the network into more communities, among which the 7-community structure is the most comparable to the division to 5 communities given by Newman and Girvan. In figure 6 (a) and (b), we visualize these 5- and 7-community structures by different colors of nodes. Obviously, two of our smallest communities (in orange and green) can be merged into the communities in blue and red respectively to recreate the 5-community structure of Newman and Girvan’s perfectly. Note that at on our 7-community level, at this resolution our previously misclassified node, SN89, has been reclassified to the “right” group.

For SN89, we notice that this node has only two connections: one to SN100, the “central” node that has the highest betweenness, and the other to Web, an individual in the other group. It was said that the actual division between the two groups of dolphins was due to a temporary leave of SN100: the interactions between the two groups were restricted while SN100 was away and became more common when it reappeared [6]. We argue that when SN100 was away, presumably the association between SN89 and the group in ellipse should also be cut off, however its association with the group in rectangle can be maintained through its connection to Web. Therefore, at the resolution of our 2-community level, our classification for SN89 to a different group than Newman and Girvan’s, also fits the ground truth.

The American college football network is constructed from the schedule of Division I games for the 2000 season of United States college football [2]. It consists of 115 nodes representing 115 college football teams, identified by their college names. Among these teams, 110 are affiliated with 11 different conferences each containing 8 to 12 teams, and 5 teams belong to no conferences and are identified as “IA independents” (nodes 37, 43, 81, 83 and 91, see the red hexagons in figure 7 (a)). Edges of the network represent scheduled games between the connected teams, which turn out to be much more frequent between teams of the same conference than between teams of different conferences. Inter-conference games also exist: they are observed to be highly correlated with the geographical distances between the related teams: teams that are geographically close to one another are more likely to play games than those separated by large geographic distances [2]. Girvan and Newman applied their community detection algorithm on the American college football network, and had correctly recreated the conference affiliations for most teams [2]. With our method, we also
detect multilevel community structures within this network, see figure 7 (b) and (c). Specifically, our strongest plateau suggests a 12-community structure as shown in figure 7 (a); this structure turns out to be well consistent with the community structure proposed by Girvan and Newman [2], which performs remarkably well in recreating the conference affiliations for the teams.

(a) Communities in the American football network.

Figure 7. Communities within the American college football network. (a) Structure of the network: 115 nodes represent 115 college football teams; edges denote scheduled games between these teams. Our method detected 12 communities out of the network, shown as 12 densely connected circles in the figure. We differentiate the conferences that these football teams belong to with different node colors, and annotate beside each community the name of the conference with which most teams within the community are affiliated. (b)-(c) The same as figure 5 (b)-(c).
Communities detected by our method are naturally displayed in figure 7 (a) with edges inside communities being shorter than those between different communities. Text beside each community denotes the name of the conference that most teams within that community are affiliated with, while the conference affiliation for every single node is differentiated by different node colors. Obviously, most conferences have been perfectly recreated by our communities, with only minor differences as listed below: (1) “IA independents” is not a conference, nor did its five teams play games frequently with one another. These teams are assigned to four different communities whose members did play games with them. (2) Conference Sunbelt (see the black circles in figure 7 (a)) is divided into two communities; members of different communities seldom played games with one another. (3) Node 111 (Texas Christian) belongs to Conference USA, but it did not play even a single game with any other teams of the same conference, instead it did play a number of games with teams in conference Western Athletic and is then assigned to the community of the latter. For the same reason, node 29 (Boise State) is assigned to one of the communities of Sunbelt instead of Western Athletic. Girvan and Newman’s result (data not shown, please refer to [2]) agrees with ours on all the above (1)–(3); the only difference lies in the assignment for node 37 (Central Florida); in [2] it is assigned to the same community as conference Mid American, but in our result it belongs to one of the communities of Sunbelt. Actually node 37 has three connections to Mid American, but two to Sunbelt; we assign it to Sunbelt only because it maximizes the global fitness function.

4. Discussion

In Results, we have demonstrated the effectiveness of our method on three classes of networks: hierarchical, heterogeneous and real-world. The outperformance of this method can be attributed to two properties: (1) (re)scalability of the community fitness function, and (2) stability of the outputs. The original form of the community fitness function introduced in [16] is by design more “scalable” than traditional community quality functions, for example, the modularity function $Q$ proposed by Newman [13], which is generally reformulated as [3, 7, 12, 23]

$$Q = \sum_{G} Q^{G} = \sum_{G} \left[ \frac{k^{G}_{in}}{2m} - \left( \frac{k^{G}_{out}}{2m} \right)^{2} \right].$$

Here $m$ is the total number of edges within the whole network, $Q^{G}$ and $k^{G}_{in} = k_{in}^{G} + k_{out}^{G}$ represent the modularity and “total” degree of community $G$. Obviously, the value of $Q^{G}$ depends heavily on the community size. For example, in an LFR network, since $k_{in}^{G} / k_{out}^{G} = 1 - \mu$ ($\mu$ is the mixing parameter),

$$Q^{G} = \frac{k_{in}^{G}}{2m} \left( \frac{k_{out}^{G}}{2m} \right)^{2} = \frac{k_{in}^{G}}{2m} \left[ 1 - \frac{k^{G}_{out}}{2m(1 - \mu)} \right].$$

If the network is large, and contains many communities, $2m$ is expected to be much greater than $k^{G}$, then $\frac{k^{G}_{out}}{2m(1 - \mu)} \approx 0$, so that $Q^{G} \approx \frac{k_{in}^{G}}{2m} \propto k_{in}^{G}$. This reflects a fact that the modularity $Q^{G}$ increases almost linearly with the community size; here without loss of generality, we measure the community size by the in-degree rather than the number of nodes. Large gaps of $Q^{G}$ exist between communities of very different sizes, which inhibits small and large communities being detected simultaneously. In contrast, the community fitness function (formula 1)
\[ f^{\gamma} = \frac{k_{in}^{\gamma}}{(k^{\gamma})^\alpha} = (1 - \mu)^\alpha (k_{in}^{\gamma})^{1-\alpha} \propto (k_{in}^{\gamma})^{1-\alpha}. \]

Since \( \alpha > 0 \), apparently \( f^{\gamma} \) increases more slowly with the community size than \( Q^{\gamma} \): it tends to narrow the gap between large and small communities. As a result, in a heterogeneous network, communities having close densities of inner connections but far different sizes can be simultaneously identified by the community fitness function \( F \), but not by the modularity function \( Q \). The generalized version of \( Q \) proposed by Reichardt and Bornholdt [19] does not solve the problem: it introduces a resolution parameter \( \gamma \) in the following way

\[ Q_\gamma = \sum_\sigma \left[ \frac{k_{in}^{\gamma}}{2m} - \gamma \left( \frac{k^{\gamma}}{2m} \right)^2 \right]. \]

However, since \( \left( \frac{k^{\gamma}}{2m} \right)^2 \) plays a minor role in the formula, \( \gamma \) is not able to rescale \( Q \), and overcome the resolution limit problem effectively [7]. Similar problem also holds for a majority of the previous multiresolution methods, including the popular Hamiltonian-based Potts models [18, 19, 20, 21, 22]. That explains why previous multiresolution methods perform poorly on heterogeneous networks, as argued by [7], while our method in this paper turns out to have outperformed all of them (see Section 3.2).

Furthermore, by introducing a scaling factor \( \beta \) in formula 2, we improve the “rescalability” of the fitness function. The fitness function originally introduced in [16] has \( \beta \) fixed to 1 (as in formula 1); correspondingly, the varying range of the resolution parameter \( \alpha \) is between 0 and 1. According to our estimation in Appendix, such a varying range is a “relevant” scale of resolution, within which both the merging of small communities and the splitting of large ones are supposed to be minor, so that for multilevel networks, only the lowest community level can be detected. Although we stressed that our estimation was made merely by necessary but not sufficient conditions—in principle it is still possible for higher levels of communities being detected therein, in practice with \( \beta = 1 \) we truly detected only the highest (but trivial) and the lowest community levels, but omitted all intermediate levels. In contrast, when \( \beta > 1 \), it rescales the whole resolution range by a multiple of \( 2/\beta - 1 \), and expands the resolution scales for different community levels by varying degrees. This effectively facilitates our detection for all levels of communities. As a result, for RB networks with hierarchical structures, our method successfully detects up to five levels of communities with \( \beta = 2 \), which to our knowledge, hasn’t been done by any other methods in previous literature.

Nevertheless, our scaling factor \( \beta \) in formula 2 rescales the resolutions unevenly: comparing to the highest and lowest community levels, expansions for the resolution scales of intermediate levels are always minor. As a result, for networks having too many community levels, our method fails to detect some of them (for example, in supplementary figure 2 (d) the second and fifth community levels of the RB15625 network are omitted). On the other hand, when the network size is too large, it becomes difficult for the original Louvain/BGLL algorithm to work out. Improved methods and algorithms are still worth studying in the future.

The second property of our method, stability of the outputs, is mainly due to our strict definition of “plateau.” It has been popular to rank the “relevance” of outputs by the persistence of plateaus in previous literature. However, as we have argued in the Introduction, if the term “plateau” was only loosely defined, and cannot ensure all data points within a same plateau represent a same topological
structure of community, such plateaus are then unreliable and suspicious. Although we believe most plateaus in previous literature did have identical topological structures, it has not been guaranteed at all by either the definition or the working procedures. In this paper, we not only require “one plateau one topology” strictly to remove the above suspicion, but also require each data point in the plateaus represents a best-and-unique solution at the corresponding resolution; the latter eliminates unstable resolution scales out of our plateaus, which has significantly improved the stability of our outputs. One proof is, on synthetic networks, such as the H13-4 network and the RB networks, all plateaus in our outputs are relevant and interpretable with respect to the prearranged community structures (see figures 1-3 and supplementary figures 1-2). In contrast, previous methods did not require uniqueness; they output all best solutions at all resolution scales. Consequently, plateaus for the same networks exhibited in previous literature are not as clear as ours: various small plateaus emerge in the transitional regions between large plateaus; see figure 1 in [3], figure 2 in [21], figure 3 in [22], figure 2&3 in [29], and so on. These small plateaus emerge in unstable scales of resolution and represent relatively insignificant community structures; here we raise just one example instead of all: for an RB25 network as in figure 2 (c), one can merge the central node (i.e., the red hollow circle) to a randomly chosen peripheral RB5 unit (for example, the green left-pointing triangles on the bottom right) and form a community of 6 nodes. Then an RB25 network can be divided into 5 communities: one of 6 nodes, three of 5 nodes, and one of 4 nodes; similar structures can be inferred by analogy in larger RB networks. Here the problem is, if the central node can be merged to one of the peripheral units for a best solution, it can also be merged to any one of the other peripheral units for an equally good solution, since the network is symmetrical: there are four different partitions of equally good qualities, while the corresponding small plateau just represent one of them by chance. To our viewpoint, such uncertain outputs should be less informative than those more definite ones, and besides, they are indeed “noise-like” by representing different partitions in different realizations. In our method, we require our outputs being best-and-unique, which automatically filters out such uncertain results, without any artificial or arbitrary selection, and keeps only the stable and relevant plateaus as we expected.

But when applied to real-world networks, plateaus outputted by our method also start to look “messy”: various small plateaus emerge and fill up almost all the resolution scales, and the numbers of communities continuously take every possible value from 1 to maximum (see figures 5-7). Being different to the small and unstable plateaus for synthetic networks, small plateaus for the real-world networks are stable: they remain consistent across different realizations, or even different ensembles. This suggests that real-world networks may have “continuous” levels of communities. Customarily, all the real-world networks we investigated in this paper, karate club, dolphin, and American college football game, are not considered as multilevel networks—in most cases they are dealt with just as single-level. Yet with continuously varying resolutions, these networks display multiple levels of communities, forming a rough hierarchical structure (see supplementary figure 4). Inquiring which scale of resolution is the “best,” or which community level is the “most relevant,” as argued by [3], are both natural but ill posed questions. Community detection in complex networks is purely based on the network topology, while the “relevance” of detected community structures depends on our a priori knowledge about the network (community structures by network construction, or empirically known partitions of the network)—whether our a priori knowledge can be well reconstructed from the network topology, is a posteriori. For example, whether the interactions between the members being an essential indication for the fission of the karate club, or whether the football games between certain teams being so positively correlated with their affiliations of conferences, are not necessarily definitive. It has been observed that communities suggested by our strongest plateaus often coincide with the ground truths, implying information about these ground truths can be well reflected by the
network topology—but it does not mean at all communities detected by other plateaus are irrelevant or uninformative. As proposed by [3], all structures detected are embedded in the network topology, and should have their own particular information.

Finally, we discuss the computational complexity of our method. We adopt a classical BGLL (i.e., Louvain) algorithm for our optimization, which has been reported to have linear complexity on typical and sparse data [30]. Complexity introduced by our method is mainly due to calculations in multiple realizations or multiple ensembles. (1) For each group of fixed parameters \((\alpha, \beta)\), we run the BGLL algorithm in 1000 realizations to make sure it converges to a best solution. In practice, usually it is not necessary to run so many realizations; 100 realizations will be sufficiently enough in most cases. (2) With each fixed value of \(\beta\), we vary \(\alpha\) from 0 to \(2\beta-1\), with a stepwise increment 0.01, to search every resolution scales and discover all possible plateaus. In practice, one can firstly use a larger value of increment for a global search, and then use a relatively small value for searches in focused regions, to reduce unnecessary computations. (3) For networks having relatively complex structure, we run multiple “ensembles;” each ensemble involves multiple realizations of community detection with varying parameters. This is to remove the “noise-like” outputs that emerge randomly, which cannot be eliminated by single ensembles of realizations (see figure 1 (b) and (c)). In practice, depending on the purpose, this is not always necessary, especially for networks whose community structures are unambiguous. Actually, for the real-world networks we investigated in Section 3.3, single ensembles of detection already work sufficiently well; we run multiple ensembles for these networks in Section 3.3 only to make sure all plateaus detected are stable. To summarize, based on a classical BGLL/Louvain algorithm, our method can be efficiently implemented on most complex networks.

5. Conclusion

In this paper, we introduced a scaling factor \(\beta\) to the community fitness function \(F\) [16] and raised a renewed method for community detection in complex networks. Our renewed community fitness function inherits the property of scalability from the original version of \(F\), which ensures our method performs excellently on heterogeneous networks whose sizes of communities span a few orders of magnitude. Besides, our community fitness function is also “rescalable;” a properly larger value of \(\beta\) will amplify the scales of resolution and enable our detection on multilevel community structures; on a hierarchical RB network, our method successfully detects communities within up to five levels. Moreover, we suggested a strict definition for “plateau,” which has on the one hand corrected the inexplicit use of the term in previous literature, and on the other hand improved the stability of our outputs significantly. Consequently, our method automatically filters out unwanted results such as uncertain partitions of the network which emerge totally due to randomness. On real-world networks, our method discovers different levels of communities, among which the most stable levels are well consistent with ground truths for these networks. Finally, our method can be carried out by fast heuristic algorithms to improve the computation efficiency; in this paper, we implement it with a fast Louvain/BGLL algorithm, which facilitates our community detection on both synthetic and real-world complex networks.

Our work in this paper has revealed the advantages of one class of scalable community quality functions. Their outperformance on both heterogeneous and hierarchical networks notwithstanding, for community detection on even larger and more complex networks, community quality functions with better properties and heuristic algorithms of higher efficiency are definitely worth pursuing in future work within this research filed.
Acknowledgement

This work was supported by the “One Thousand Talents Program” of Sichuan Province with Grant No. 17QR003, P.R. China.

Availability of tools and data

All tools and data used in this paper are publicly available.

Freely available code for the Louvain/BGLL algorithm can be downloaded from the webpage of Vincent Blondel: https://perso.uclouvain.be/vincent.blondel/research/louvain.html. We modified its source code slightly to substitute our community fitness function for the original modularity function.

Source code to create the LFR benchmark networks can be downloaded from: https://sites.google.com/site/santofortunato/inthepress2.

Real-world network data can be downloaded from the webpage of Mark Newman: http://www-personal.umich.edu/~mejn/netdata/.

References

[1] Arenas A, Diaz-Guilera A, Perez-Vicente C J. Synchronization Reveals Topological Scales in Complex Networks. Physical Review Letters, 2006, 96(11): 114102.1-114102.4.

[2] Girvan M, Newman M E. Community structure in social and biological networks. Proceedings of the National Academy of the Sciences of the United States of America, 2002, 99(12): 7821-7826.

[3] Arenas A, A Fernández, S Gómez. Analysis of the structure of complex networks at different resolution levels. New Journal of Physics, 2007, 10(5): 053039.

[4] Newman M E J. Detecting community structure in networks. European Physical Journal B, 2004, 38(2): 321-330.

[5] Arenas A, Danon L, Díaz-Guilera A, Gleiser P M, Guimera R. Community analysis in social networks. The European Physical Journal B, 2004, 38(2): 373-380.

[6] Newman M, Girvan M. Finding and Evaluating Community Structure in Networks. Physical Review E, 2004, 69(2): 026113.

[7] Lancichinetti A, Fortunato S. Limits of modularity maximization in community detection. Physical Review E, 2011, 84(6): 066122.

[8] Garey M R and Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, San Francisco (1979).

[9] Scott J. Social Network Analysis: A Handbook. Sage Publications, London, 2nd edition (2000).

[10] Ravasz E and Barabasi A. Hierarchical organization in complex networks. Physical Review E, 2003, 67(2): 026112.
Leskovec J, Lang K J, Dasgupta A, and Mahoney M W. Statistical properties of community structure in large social and information networks. In *Proceeding of the 17th international conference on World Wide Web*, 2008, pages 695-704, New York, NY, USA. ACM.

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

Newman M. Analysis of weighted networks. *Physical Review E*, 2004, 70: 056131.

Clauset A. Finding local community structure in networks. *Physical Review E*, 2005, 72(2): 026132.

Luo F, Wang J Z and Promislow E. Exploring Local Community Structures in Large Networks. *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings) (WI'06)* 0-7695-2747-7/06.

Lancichinetti A, Fortunato S, Kertész J. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics*, 2009, 11(3): 033015.

Reichardt J, Bornholdt S. Partitioning and modularity of graphs with arbitrary degree distribution. *Physical Review E*, 2007, 76(1): 015102.

Reichardt J, Bornholdt S. Detecting Fuzzy Community Structures in Complex Networks with a Potts Model. *Physical Review Letters*, 2004, 93(21): 218701.

Reichardt J, Bornholdt S. Statistical mechanics of community detection. *Physical Review E*, 2006, 74(2): 016110.

Hastings M B. Community Detection as an Inference Problem. *Physical Review E*, 2006, 74(2): 035102.

Ronhovde P, Nussinov Z. Multiresolution community detection for megascale networks by information-based replica correlations. *Physical Review E*, 2009, 80(1): 016109.

Ronhovde P, Nussinov Z. Local resolution-limit-free Potts model for community detection. *Physical Review E*, 2008, 81(2): 046114.

Fortunato S and Barthelemy M. Resolution limit in community detection. *Proceedings of the National Academy of the Sciences of the United States of America*, 2007, 104(1): 36-41.

Kumpula J M, Saramaki J, Kaski K, Kertesz J. Resolution limit in complex network community detection with Potts model approach. *The European Physical Journal B*, 2007, 56: 41-45.

Traag V A, Dooren P V, Nesterov Y. Narrow scope for resolution-limit-free community detection. *Physical Review E*, 2011, 84(1): 016114.

Clauset A, Newman M E J, Moore C. Finding community structure in very large networks. *Physical Review E*, 2004, 70: 066111.

Good B H, Montjoye Y D, Clauset A. Performance of modularity maximization in practical contexts. *Physical Review E*, 2009, 81(2): 046106.

Le Martelot E, Hankin C. Multi-scale Community Detection using Stability Optimisation. *International Journal of Web Based Communities*, 2013, 9(3): 323-348.

DOI: 10.1504/IJWBC.2013.054907
[29] Xiang J, Tang Y N, Gao Y Y, Zhang Y, Deng K, Xu X K, Hu K. Multi-resolution community detection based on generalized self-loop rescaling strategy. Physica A: Statistical Mechanics & Its Applications, 2015, 432:127-139.

[30] Blondel V D, Guillaume J L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics Theory & Experiment, 2008: P10008.

[31] Newman M. Fast algorithm for detecting community structure in networks. Physical Review E, 2004, 69: 066113.

[32] Saha S, Ghrera S P. Nearest Neighbor search in Complex Network for Community Detection. Information (Switzerland), 2015, 7(1): 17.

[33] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. Physical Review E, 2008, 78: 046110.

[34] Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. Proceedings of the National Academy of the Sciences of the United States of America, 2004, 101: 2658–2663.

[35] Fred A L N, Jain A K. Robust data clustering. in Proceedings of the IEEE Computer Society Conference on Computer Vision Pattern Recognition (Computer Society, Toronto, 2003), Vol. 2, page 128.

[36] Zachary W W. An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research, 1977, 33(4): 452-473.

[37] Lusseau D. The emergent properties of a dolphin social network. Proceedings of the Royal Society B Biological Sciences (suppl.), 2003, 270: S186-S188. DOI 10.1098/rsbl.2003.0057.

[38] Medus A, AcunA G, Dorso C O. Detection of community structures in networks via global optimization. Physica A: Statistical Mechanics and its Applications, 2005, 358(2-4): 593-604.

[39] Lusseau D, Schneider K, Boisseau O J, Haase P, Slooten E, Dawson S M. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behavioral Ecology & Sociobiology, 2003, 54(4): 396-405.

[40] Lusseau D, Newman M. Identifying the role that animals play in their social networks. Proceedings of the Royal Society B Biological Sciences, 2004, 271 (Suppl. 6): S477-S481.

[41] Lusseau D. Evidence for social role in a dolphin social network. Evolutionary Ecology, 2007, 21(3):357-366.
Appendix: relevant range of the resolution parameter $\alpha$

In this appendix, we make a rough estimation on the relevant range of the resolution parameter $\alpha$ for each fixed scaling factor $\beta$ in the community fitness function

$$F_\alpha^\beta = \sum_{\varphi'} \frac{(k_{\text{in}}^\varphi)^{\alpha}}{(k_{\text{in}}^\varphi + k_{\text{out}}^\varphi)^{\alpha}}$$  \hspace{1cm} (Formula 2 in the main text)

It has been discussed in [25] that the resolution limit problem in community detection is due to two opposite tendencies (or biases): the tendency of merging small communities into larger ones, and the tendency of breaking large ones into small pieces. These tendencies/biases can often occur simultaneously. A strict deduction for a “relevant” resolution range, within which both biases can be avoided, is not straightforward—nor is it necessary for our purpose in this paper. In this appendix, we investigate each bias separately, and then give a rough estimation on the bounds of the relevant range for the resolution parameter $\alpha$ in formula 2. It should be noted that our purpose is very simple: all we need is a rough range for $\alpha$ to vary in. Therefore, we only investigate necessary conditions, rather than sufficient or necessary-and-sufficient conditions.

I. Upper bound of $\alpha$: splitting a random graph

With a fixed value of $\beta$, since large values of $\alpha$ deliver small communities, the upper bound of $\alpha$ can be estimated by the limit at which the fitness function $F_\alpha^\beta$ starts to split a graph inappropriately into smaller parts. For such an estimation, one useful reference is the random graph: since a random graph is believed to have no communities, by any algorithm it shouldn’t be split into smaller pieces [25].

Suppose we have a random graph consisting of $N$ nodes, with probability $p$ each pair of nodes shares an edge between them. Consider splitting the graph into two parts: subgraph $\mathcal{M}$ contains $m$ nodes ($0 \leq m \leq N$), while subgraph $\mathcal{N}$ contains $N-m$ nodes. Both $\mathcal{M}$ and $\mathcal{N}$ are random graphs with identical connection probability $p$.

Subgraph $\mathcal{M}$ (containing $m$ nodes) is expected to have $\frac{m(m-1)}{2} p$ intra-connections, thus it has a total in-degree $k_{\text{in}}^\mathcal{M} = m(m-1) p$. Each node of $\mathcal{M}$ shares with each node of $\mathcal{N}$ a connection with probability $p$, thus the out-degrees $k_{\text{out}}^\mathcal{M} = k_{\text{out}}^\mathcal{N} = m(N-m) p$. Then the fitness of community $\mathcal{M}$ can be calculated as

$$\left( F_\alpha^\beta \right)^\mathcal{M} = \frac{[m(m-1)p]^\alpha}{[m(m-1)p + m(N-m)p]^\alpha} = \frac{m^\beta \alpha (m-1)^{\alpha} p^{\beta-\alpha}}{(N-1)^{\alpha}}$$  \hspace{1cm} (1)

Similarly, the fitness of community $\mathcal{N}$ can be calculated as

$$\left( F_\alpha^\beta \right)^\mathcal{N} = \frac{(N-m)^{\beta-\alpha} (N-m-1)^{\alpha} p^{\beta-\alpha}}{(N-1)^{\alpha}}$$  \hspace{1cm} (2)

Then the fitness of the whole network (with respect to a division to $\mathcal{M}$ and $\mathcal{N}$) is
\[
\left( F_\beta^\alpha \right)^{M+V} + \left( F_\beta^\alpha \right)^V = [m^{\beta-\alpha} (m-1)^\beta + (N-m)^{\beta-\alpha} (N-m-1)^\beta] p^{\beta-\alpha} (N-1)^\alpha \tag{3}
\]

In comparison, if the whole network is recognized as one community (indicated by \( M+N \)), its fitness can be calculated as

\[
\left( F_\beta^\alpha \right)^{M+V} = N^{\beta-\alpha} (N-1)^\beta p^{\beta-\alpha} (N-1)^\alpha \tag{4}
\]

We do not hope the random graph \( M+N \) be split into subgraphs \( M \) and \( N \), which requires

\[
\left( F_\beta^\alpha \right)^{M+V} > \left( F_\beta^\alpha \right)^M + \left( F_\beta^\alpha \right)^V \tag{5}
\]

Substitute (3) and (4) into (5), we get

\[
N^{\beta-\alpha} (N-1)^\beta > m^{\beta-\alpha} (m-1)^\beta + (N-m)^{\beta-\alpha} (N-m-1)^\beta, \text{ for any } 0 < m < N \tag{6}
\]

Inequality (6) is equivalent to the following statement, i.e., the maximum of the function

\[
f_\beta^\alpha (m) = m^{\beta-\alpha} (m-1)^\beta + (N-m)^{\beta-\alpha} (N-m-1)^\beta \tag{7}
\]

should be reached at \( m=0 \) or \( m=N \).

(7) is a \((2\beta-\alpha)\)-order polynomial on variable \( m \); a full set of its extrema (either maxima or minima) is difficult to solve. Here we simply make a very rough estimation on the solution of (6): we notice that due to symmetry, \( f_\beta^\alpha (m) \) has an extremum at \( m=N/2 \) (we do not care if it’s a maximum or a minimum). As a necessary condition, (6) at least requires

\[
f_\beta^\alpha (0) = f_\beta^\alpha (N) > f_\beta^\alpha \left( \frac{N}{2} \right) \tag{8}
\]

Substitute (7) into (8) we get

\[
N^{\beta-\alpha} (N-1)^\beta > 2^{\alpha-2\beta+1} N^{\beta-\alpha} (N-2)^\beta \tag{9}
\]

i.e.,

\[
2^{\alpha-2\beta+1} < \left( 1 + \frac{1}{N-2} \right)^\beta \tag{10}
\]

Obviously, when \( \alpha \leq 2\beta-1 \), (10) can be always satisfied. In other words, when \( \alpha \leq 2\beta-1 \), at least a random graph would not be split into two subgraphs of the same size. This makes a necessary condition for avoiding the first bias (inappropriate splitting of large communities); in this paper, we simply take \( 2\beta-1 \) as the upper bound of the resolution parameter \( \alpha \).

II. Lower bound of \( \alpha \): merging complete graphs

Merging small and dense communities into larger but sparser ones, reflects the resolution limit problem at the other end of the resolution scale: small values of \( \alpha \) may cause this problem. Suppose we have a couple of complete graphs: \( M \) consisting of \( m \) nodes and \( N \) consisting of \( n \) nodes. If \( M \) and \( N \) are identified as two independent communities, it is straightforward to calculate their in-degrees: \( k_{in}^M = m(m-1) \), \( k_{in}^N = n(n-1) \). As for their out-degrees, for simplicity we assume \( M \)
and \( N \) are disconnected, i.e., \( k_{out}^M = k_{out}^N = 0 \). Then the fitness of the network with \( M \) and \( N \) identified as separate communities can be calculated as

\[
(F_{\alpha}^\beta)^M + (F_{\alpha}^\beta)^N = m^{\beta - \alpha} (m-1)^{\beta - \alpha} + n^{\beta - \alpha} (n-1)^{\beta - \alpha}
\]

(11)

For convenience, we denote \( m(m-1) = a, \ n(n-1) = b, \ \beta - \alpha = k \), then (11) becomes

\[
(F_{\alpha}^\beta)^M + (F_{\alpha}^\beta)^N = a^k + b^k
\]

(12)

On the other hand, if \( M \) and \( N \) are merged into one large community (indicated by \( M+N \)), the in-degree, out-degree and fitness of this large community can be calculated as

\[
k_{in}^{M+N} = m(m-1) + n(n-1) = a + b,
\]

\[
k_{out}^{M+N} = 0,
\]

\[
(F_{\alpha}^\beta)^{M+N} = (m(m-1) + n(n-2))^{\beta - \alpha} = (a + b)^k
\]

(13)

We expect \( M \) and \( N \) being identified as two independent communities, which at least requires

\[
(F_{\alpha}^\beta)^M + (F_{\alpha}^\beta)^N \geq (F_{\alpha}^\beta)^{M+N}
\]

(14)

i.e.,

\[
a^k + b^k \geq (a + b)^k
\]

(15)

Consider a function of \( k \): \( f(k) = a^k + b^k - (a + b)^k \), (15) is equivalent to finding out a range of \( k \) within which \( f(k) \geq 0 \).

Since \( \frac{df}{dk} = a^k \ln a + b^k \ln b - (a + b)^k \ln(a + b) \), without loss of generality, we assume \( m \leq n \), so that \( a \leq b \), then

\[
\frac{df}{dk} \leq a^k \ln b + b^k \ln b - (a + b)^k \ln b = [a^k + b^k - (a + b)^k] \ln b = f(k) \ln b
\]

Since \( f(1) = 0 \), then \( \frac{df}{dk} \bigg|_{k=1} \leq f(1) \ln b = 0 \), which indicates that \( f(k) \) is not an increasing function in the neighborhood of \( k=1 \). Therefore, when \( k < 1 \), i.e., \( a > \beta - 1 \), presumably \( f(k) \geq 0 \), so that (15) would be satisfied. In this paper, we take \( \beta - 1 \) as the lower bound of \( \alpha \).

Combining the above I and II, we estimate a “relevant range” for the resolution parameter \( \alpha \) with a fixed value of the scaling factor \( \beta \): \( \beta - 1 < \alpha < 2\beta - 1 \). It should be noted that this range of \( \alpha \) was roughly estimated through necessary conditions rather than sufficient or necessary-and-sufficient conditions: the “real” relevant scale of resolution can be expected to fall in this range, as a proper sub-region probably—but resolution limit problem can still exist in the rest part of this range since a sufficient condition is not guaranteed here.
Supplementary Figure 1. Communities in the RB625 network. (a) Topology of the RB625 network. Here by different shapes and colors of nodes, we show a structure of the third community level (14 communities). (b) and (c): Plateaus detected for the RB625 network by our method with $\beta=1$ and 2; these plateaus are obtained through an intersection over the outputs of 20 ensembles, each involving 1000 realizations of the Louvain algorithm at each resolution.
Supplementary figure 2. Plateaus detected for the RB3125 and RB15625 networks with $\beta=1$ and 2. In each subfigure, plateaus are obtained through an intersection over the outputs of 20 ensembles, each involving 1000 realizations of the Louvain algorithm at each resolution. Subfigures (a)-(c) are perfectly consistent with our discussions in the main text. However, in (d), for the RB15625 network, our detection starts to deviate from perfection: the second and fifth community levels (1250 and 22 communities) are missing, and the small plateau for 142 communities turns out to be a hybrid of different levels of communities in different parts of the network: each of the four peripheral RB3125 units is divided into 18 communities (as in the fourth community level of RB3125), while the central RB3125 unit is divided into 70 communities (as in the third community level of RB3125).
Supplementary figure 3. Plateaus for the RB15625 network detected with $\beta=2$–$5$. For each subfigure, we run 20 ensembles, each involving 1000 realizations at each resolution. Plateaus in the figure are obtained through an intersection over the outputs of all 20 ensembles. $\alpha_1$ and $\alpha_2$ in the figure indicate the upper bound and lower bound of the resolution scales of the highest and lowest community levels; all intermediate community levels should be detected within the scale $\alpha_1<\alpha<\alpha_2$ (not shown). It can be observed that the gap between $\alpha_1$ and $\alpha_2$ does not increase linearly with $\beta$, thus increasing the value of $\beta$ does not necessarily help to detect more levels of communities in large RB networks.
Supplementary figure 4. Multi-level community structures from 2 to 12 communities detected by our method for the karate club network. Different community levels roughly make a hierarchical structure: communities of a lower level are in most cases subsets of communities of a higher level. Reassembling of communities only occurs when 4 communities split to 5 communities, and 5 split to 6.