Community identification in networks with unbalanced structure

Shuqin Zhang[*]
Center for Computational Systems Biology, School of Mathematical Sciences, Fudan University, Shanghai 200433, China

Hongyu Zhao[†]
Division of Biostatistics, Yale School of Public Health, New Haven, Connecticut 06520, USA

(Received 10 November 2011; revised manuscript received 18 April 2012; published 13 June 2012; publisher error corrected 20 June 2012)

Community (module) structure is a common and important property of many types of networks, such as social networks and biological networks. Several classes of algorithms have been proposed for community structure detection and identification, including clustering techniques, modularity optimization, and other methods. Among these methods, the modularity optimization method has attracted a great deal of attention and much related research has been published. However, the existing modularity optimization method does not perform well in the presence of unbalanced community structures. In this paper, we introduce a metric to characterize the community structure better than other metrics in this situation, and we propose a method to infer the number of communities, which may solve the resolution limit problem. We then develop an algorithm for community structure identification based on eigendecompositions, and we give both simulated and real data examples to illustrate the better performance of our approach.

DOI: 10.1103/PhysRevE.85.066114 PACS number(s): 89.75.Hc, 02.50.−r, 05.10.−a, 87.18.−h

I. INTRODUCTION

Network study has attracted a considerable amount of attention in recent years from researchers in different fields, including physics, computer science, statistics, and others. A network can be seen as a synonym for a mathematical graph. It is composed of vertices (nodes) and edges. The vertices represent the members in the network, while the edges represent the pair relations of the vertices. Many complex interaction systems can be described as networks, including biological systems, social systems, and the worldwide web.

Community (module) structure is a common feature of many networks. Since the seminal paper of Girvan and Newman [1], many related papers have been published on network community analysis [2–11]. Intuitively, a community is a subset of a network. The vertices in the same subnetwork are more likely to be connected with each other than those in different subnetworks. In general, members in the same community share some common properties or play similar roles. In a gene coexpression network, the vertices (which correspond to genes) in the same community may belong to the same functional category, such as lipid metabolism and acute-phase response, or they may be involved in the same pathway, such as a metabolic pathway or a ribosome [12,13]. In a collaboration network, the vertices (which correspond to researchers) in the same community likely share some common research interests [8].

There has been a concerted effort in recent years to develop mathematical tools and computer algorithms to identify and quantify community structure in networks [3–9,14]. Several recent review papers provide details of community identification methods [5,8,14]. Reference [14] compares the performance of several existing methods for both computation time and output. Reference [5] is a thorough, more recent discussion. Reference [8] contrasts different perspectives of the methods and sheds light on some important similarities of several methods. These community identification papers are mostly written by computer scientists, statisticians, and physicists, with physicists making the most contributions.

Earlier methods for community identification mainly arose from computer science. The communities are identified using graph partitioning methods or cluster-based methods. Graph partitioning methods require the sizes of the subgraphs as the input for network partitions, but little is known on the sizes in practice [15–18]. Cluster-based methods include hierarchical clustering, partitional clustering, and spectral partitioning. Hierarchical clustering has been shown to be effective since some networks do possess a hierarchical structure and the number of communities can be determined during the clustering process [3]. Partitional clustering is popular in data mining, but it may not be appropriate for community identification since the community structure describes the topological relations of the vertices, which may not be measured by Euclidean distance, correlations, and other distances usually used in partitional clustering. Spectral clustering can be applied to community identification [19], but this method tends to isolate some very small communities from the network instead of dividing the network into reasonably large subnetworks.

Two recent papers by statisticians considered the theoretical aspects of the community identification problem for dense networks [20,21]. In [20], the authors proposed a new modularity definition and provided sufficient conditions so that some modularity can give a consistent estimation of the community structure. Although the proposed modularity was shown to outperform other methods, it is time consuming to solve the optimization problem under this definition. In [21], the authors developed an algorithm for extracting the communities sequentially from a network when some vertices do not fit in with any of the communities. One limitation of the theoretical developments in these two papers is that they

[*] zhangs@fudan.edu.cn
[†] hongyu.zhao@yale.edu

both assume the network is dense. Although such networks do exist in reality, the networks encountered in many contexts are more likely to be sparse.

Most recent community identification methods arose from the physics field, with the divisive algorithms being one type of them [5]. The most popular algorithm within this class was proposed by Girvan and Newman [1,22], who introduced the community identification problem to physicists. In the divisive algorithm, the importance of each edge is estimated with betweenness. The edge with the largest betweenness is first removed and the betweenness for all the remaining vertices is recalculated. The algorithm is implemented iteratively until the expected number of communities is reached. Another important class of community identification methods arising from physics are the modularity-based optimization methods. Modularity, which is by far the best known and most commonly used method for community identification [6,7,20,23], measures the connectivities of vertices in the network. By maximizing the modularity, reasonable results for tested networks can be obtained. This method is also easily implementable.

Although many papers related to community identification have been published, this field has continued to attract a great deal of attention. A large portion of the literature has been focused on explaining the modularity and improving the computational methods [24–28]. One important problem with modularity optimization concerns the resolution limit [24]. Modularity optimization cannot resolve some communities smaller than a certain size, which depends on the size of the subnetworks to be divided and the interconnectedness of the subcommunities. Another problem with modularity optimization is its extreme degeneracies [25]. With the number of communities K in a network increasing, the number of distinct high-scoring solutions for maximizing modularity increases exponentially, making it unlikely to find the global maximum.

In the next section, we will first review the modularity optimization method and discuss its limitation as illustrated in our numerical tests. More specifically, when the communities in a network have very different sizes, this method will not perform as well as when the community structure is balanced. We also offer a possible explanation for its poor performance. We then introduce the metric in Sec. III to quantify the communities in a network, and we propose a method to infer the number of communities, which may solve the resolution limit problem. An algorithm to compute this metric based on eigendecompositions is presented in Sec. IV. The usefulness of our approach is illustrated through several examples with the computational complexity analysis in Sec. V, and concluding remarks are given at the end of the paper.

II. MODULARITY OPTIMIZATION METHOD

We begin by reviewing the modularity optimization method. Modularity measures the difference between the number of edges falling within groups in the network and the expected number of such edges in an equivalent network where the edges are placed at random [22]. We consider a network G with n vertices. The adjacency matrix is denoted as A, where each entry is 0 or 1. The degree of vertex i is defined as

$$d_i = \sum_{j=1}^{n} A_{ij}, \quad i = 1,2,\ldots,n.$$

For a particular partition of the network into two groups, we let $s_i = 1$ if vertex i belongs to group 1 and $s_i = -1$ if it belongs to group 2. Then $\frac{1}{2}(s_i s_j + 1)$ is equal to 1 if i,j are in the same group and 0 otherwise. If all the edges in the network are placed at random, the expected number of edges between vertices i and j is $P_{ij} = d_i d_j / 2m$, where m is the total number of edges in the network. Then modularity can be mathematically defined as

$$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - P_{ij}) s_i s_j.$$

Let $\mathbf{s}$ be a column vector with entries being s_i, $B_{ij} = A_{ij} - P_{ij}$. In matrix form, it is

$$Q = \frac{1}{4m} \mathbf{s}^T B \mathbf{s}.$$

When there are more than two communities in the network, we let S be an $n \times K$ matrix, where K is the number of communities in the network. The value of each entry S_{ik} in S is 1 if vertex i belongs to the kth community and 0 otherwise. Then

$$Q = \frac{1}{4m} \mathrm{Tr}(S^T B S).$$

By maximizing the modularity, we can infer the communities. The earliest proposed method for solving this optimization problem is based on the leading eigenvector of the matrix B. At each step, all the entries with the same sign in the leading eigenvector will be taken as belonging to the same community. The method is implemented iteratively until all the remaining communities are indivisible. This strategy can be easily implemented and runs fast. Although it cannot find the optimal solution in many cases, especially for the large networks with many communities, it appears to find divisions that are close to optimal. Refer to [6,22] for a detailed explanation of modularity.

A. Limitation for unbalanced community structure

Although various groups have improved the original modularity optimization method from different aspects, there is one limitation that has not been well addressed by the existing methods. Namely, when the sizes of the communities in a network are unbalanced, the modularity optimization method does not work as well as when the communities have comparable sizes. We give an example to illustrate this problem. Figure 1 shows a network with 197 vertices and 481 edges. From our visual inspection of the figure, there are two communities corresponding to the left and right sections of the graph with two edges connecting them. The modularity with this partition is 0.26. However, when we use the modularity optimization method, the large community is partitioned into two parts with one part linked with the small community on the right to form the second community. These two identified communities are represented by different colors and shapes in this figure. The value of the modularity

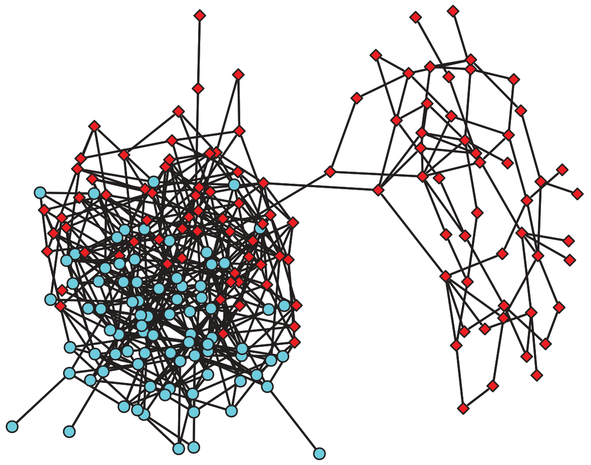


FIG. 1. (Color online) Example of an unbalanced community structure. Different shapes and colors of the vertices show the two communities identified with the modularity optimization method.

function is 0.32 for this partition, which is larger than that with our visually correct partition of the network. To see why the modularity optimization method does not work here, we plot the matrix B and the leading eigenvector of B in Fig. 2. In the figure on the left, different colors represent different values of matrix B. It is easy to see the community structure of the network in the figure. However, although we can see some pattern from the leading eigenvector of B, we cannot find a threshold to cut the eigenvector so that the network can be divided into the two desired communities. From our numerical analysis of many network examples, we found that this is a frequent phenomenon when the ratio between the sizes of two communities is less than a certain value, e.g., 0.25 depending on the inner structure of the communities.

To understand why the modularity optimization method does not work well in such unbalanced community networks, we need to consider the construction of the matrix B, which is defined as the difference between A and P. In matrix P, each entry $P_{ij} = \frac{d_i d_j}{2m}$ describes the expected number of edges between two vertices i and j and it is fully determined by their degrees. Now consider a network with two communities C_1 and C_2. For any vertex i from C_1 and j from C_2, we have $P_{ij} > 0$, and if d_i and d_j increase, P_{ij} will become greater. If the two communities have comparable size, the patterns of the submatrices corresponding to the two communities are likely to be similar. This pattern also holds for matrix B. The leading eigenvector of B captures the information for both communities and may partition the network into communities well. However, when the two communities have very different sizes, the degrees of the vertices in the small community are not as large as those in the large community. The range of the values in the submatrix of B corresponding to the small community will be quite different from that corresponding to the large community. Then the leading eigenvector of matrix B will mainly reflect the property of the large community instead of both communities, whereas the part corresponding to the small community will be close to zero, resulting in the poor identification of communities. For networks with more than two communities, this phenomenon also occurs. In addition, a large community in a network may be divided into several small groups instead of being kept together.

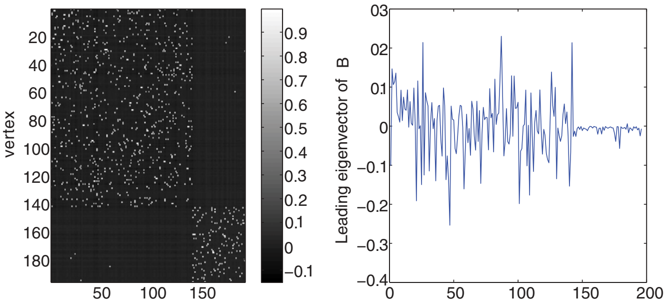


FIG. 2. (Color online) Pattern of matrix B for the network in Fig. 1 (left) and the leading eigenvector of matrix B (right).

III. COMMUNITY IDENTIFICATION METHOD

To overcome the problem arising from unbalanced community networks, we resort to the graph partition methods. In this section, we introduce a metric to quantify the communities in a network. We use the same notations as those used in the preceding section. Given a network structure, we suggest identifying the connected parts of the network first before we identify the communities. This step can be accomplished by standard spectral clustering, which can give the accurate partition. In the following, we will consider connected networks. To identify the communities in a network, we let $N(V_k)$ denote the number of vertices in subnetwork V_k, $L(V_k,V_k)$ denote twice the total number of edges in subnetwork V_k, and $L(V_k,V_l)$ denote the total number of connections between the subnetworks V_k and V_l, where $k,l = 1,2,\ldots,K$, and K is the total number of subnetworks (communities). In the following, we use N_k, L_{kl} to denote $N(V_k), L(V_k,V_l)$ for simplicity. Now, for any partition of the network $\mathbf{P}$, we define our metric for community identification $\Phi(\mathbf{P})$ as

$$\Phi(\mathbf{P}) = \Phi_1(\mathbf{P}) - \Phi_2(\mathbf{P}),$$

where

$$\Phi_1(\mathbf{P}) = \sum_{k=1}^{K} \frac{L_{kk}}{N_k}, \quad \Phi_2(\mathbf{P}) = \sum_{k=1}^{K} \sum_{l \neq k} \frac{L_{kl}}{N_k}.$$

In matrix form, if we let

$$S_{ik} = \begin{cases} 1 & \text{if vertex } i \in V_k, \\ 0 & \text{otherwise,} \end{cases} \qquad i = 1,2,\ldots,n,$$

then the value of $\Phi_1(\mathbf{P})$ is $\sum_{k=1}^{K} \frac{S_{.,k}^T A S_{.,k}}{S_{.,k}^T S_{.,k}}$, and the value of $\Phi_2(\mathbf{P})$ is $\sum_{k=1}^{K} \sum_{l \neq k} \frac{S_{.,k}^T A S_{.,l}}{S_{.,k}^T S_{.,k}}$. Thus the value of $\Phi(\mathbf{P})$ can be expressed as a function of S:

$$\begin{aligned} \Psi(S) &= \Psi_1(S) - \Psi_2(S) \\ &= \sum_{k=1}^{K} \frac{S_{.,k}^T A S_{.,k}}{S_{.,k}^T S_{.,k}} - \sum_{k=1}^{K} \sum_{l \neq k} \frac{S_{.,k}^T A S_{.,l}}{S_{.,k}^T S_{.,k}}, \end{aligned} \tag{1}$$

where $S_{.,k}$ denotes the kth column of matrix S.

The function $\Phi_1(\mathbf{P})$ defines the sum of the average degrees in each subnetwork and $\Phi_2(\mathbf{P})$ defines the sum of the average number of connections between one subnetwork and other subnetworks. It is easy to see that for community identification, our goal is to both maximize Φ_1 and minimize Φ_2. Although these two metrics may seem to be achieving the same overall objective detailed in (1), they may lead to inconsistent results when applied separately. For example, in a network constructed with a very small community and a very large community with no edges connecting the two, the partition which maximizes Φ_1 tends to partition the network into two communities, with one consisting of one section of the large community and another consisting of the remaining section of the large community and the small community because this partition will maximize the sum of average degrees. In very sparse networks, the partition which minimizes Φ_2 will tend to divide the network into some very small communities, while the partition which maximizes Φ_1 will try to keep some communities together. In networks in which the community structures are very clear and the sizes of the communities are fairly balanced, maximizing Φ_1 and minimizing Φ_2 may give the same partition of the network. This measure was also introduced in [10]. By maximizing $\Phi(\mathbf{P})$, we expect to achieve a good balance and make a correct inference on the communities. Therefore, we formulate our community identification problem as

$$\begin{aligned}&\max \quad \Psi(S)\\ &\text{subject to} \quad S_{i,j} \in \{0,1\} \quad \text{for} \quad i,j = 1,2,\ldots,K,\\ &\sum_{k=1}^{K} S_{\cdot,k} = \mathbf{1}.\end{aligned} \tag{2}$$

Here $\mathbf{1}$ is a vector with all elements being 1.

Now we discuss the choice of K, the number of communities, which is often unknown to researchers. We first introduce some notations. Consider a subnetwork V_k of G; its complementary subnetwork is denoted as $\bar{V}_k$. Then the degree of the vertex $i \in V_k$ can be written as

$$d_i = d_i(V_k) + d_i(\bar{V}_k),$$

where

$$d_i(V_k) = \sum_{j\in V_k} A_{ij},$$

$$d_i(\bar{V}_k) = \sum_{j\in \bar{V}_k} A_{ij},$$

where $d_i(V_k)$ defines the connections that vertex i has in the subnetwork V_k. More generally, we can define $d_i(V_l)$ as the total number of connections between vertex i and all the vertices in subnetwork V_l.

Several methods have been proposed to determine the community structure independently instead of embedding it in the model and algorithm design. In [9], the authors gave two definitions of a community based on the degrees. The subgraph V_k of G is defined as a community in a strong sense if

$$d_i(V_k) > d_i(\bar{V}_k), \quad \forall\, i \in V_k,$$

and as a community in a weak sense if

$$\sum_{i\in V_k} d_i(V_k) > \sum_{i\in V_k} d_i(\bar{V}_k).$$

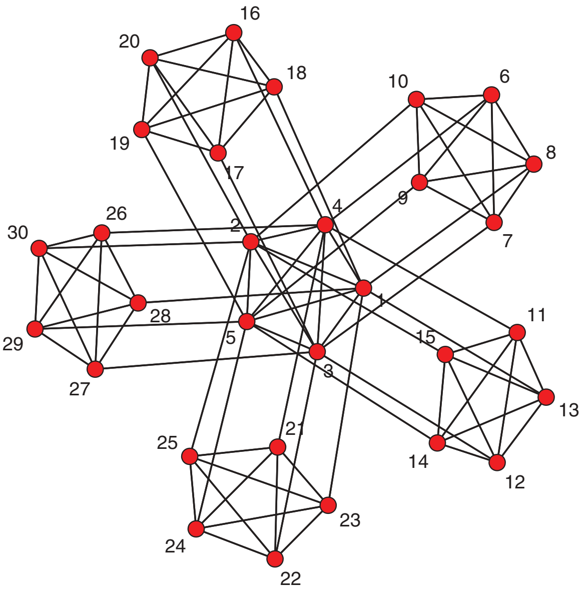


FIG. 3. (Color online) A network constructed from six cliques.

It is easy to see that if a subgraph satisfies the condition in the strong sense definition, it will satisfy the condition in the weak sense definition. However, for some networks, both definitions may not determine a community. Figure 3 shows a network with 30 vertices constructed from six cliques, with each having five vertices. One clique is connected to all the other five cliques with a one-to-one vertex connection. Although neither the strong nor the weak definition is satisfied for the central clique, it is intuitive that it forms a community.

We modify the above method for determining a community by considering the average degrees. Suppose a network is well partitioned into distinct communities. We expect that the average connectivity within a community is larger than that between communities, i.e.,

$$\frac{\sum_{i\in V_k} d_i(V_k)}{N_k} > \frac{\sum_{i\in V_k} d_i(V_l)}{N_k}, \quad l \neq k. \tag{3}$$

Alternatively, it can also be written as

$$L_{kk} > L_{kl}$$

if we multiply both sides with $N(V_k)$. This criterion uses the whole community structure of the network to determine all communities.

To determine the number of communities in a network, we compute the maximum value of $\Phi(\mathbf{P})$ for different numbers of communities K. We select K where the inequality (3) holds. If for any $K > 1$ the inequality (3) does not hold, the network will have no subcommunities, and it is determined as one community. With such a choice of K, the network is divided into subnetworks satisfying the inequality (3). We hope the resolution limit problem of the modularity optimization method can be avoided as shown in the numerical results.

IV. NUMERICAL ALGORITHM FOR IDENTIFYING THE COMMUNITIES

To find the best partition that maximizes $\Psi(S)$, the computational complexity is exponential if we enumerate all the possible partitions for a network with size n. Therefore, we propose to use an approximate method for solving the optimization problem (2).

Let $\tilde{S}_{.,k} = \frac{S_{.,k}}{\|S_{.,k}\|_2}$. The function $\Psi_1(S)$ can be relaxed to a new function that we denote as $\tilde{\Psi}_1(\tilde{S})$, which is defined as

$$\tilde{\Psi}_1(\tilde{S}) = \sum_{k=1}^{K} \tilde{S}_{.,k}^T A \tilde{S}_{.,k} = \mathrm{Tr}(\tilde{S}^T A \tilde{S}).$$

Similarly, the function $\Psi_2(S)$ can be relaxed to a new function $\tilde{\Psi}_2(\tilde{S})$, which is

$$\tilde{\Psi}_2(\tilde{S}) = \mathrm{Tr}(\tilde{S}^T L \tilde{S}).$$

Here, L is the so called Laplacian matrix, which is defined as

$$L = D - A,$$

where D is a diagonal matrix with each entry being the degree of the corresponding vertex.

Now, $\Psi(S)$ can be approximated as $\mathrm{Tr}(\tilde{S}^T A \tilde{S}) - \mathrm{Tr}(\tilde{S}^T L \tilde{S}) = \mathrm{Tr}[\tilde{S}^T(2A - D)\tilde{S}]$, and we aim to solve the optimization problem:

$$\max \tilde{\Psi}(\tilde{S}) = \mathrm{Tr}[\tilde{S}^T(2A - D)\tilde{S}] \quad \text{subject to} \quad \tilde{S}^T \tilde{S} = I.$$

The problem of maximizing $\tilde{\Psi}(\tilde{S})$ is the standard form of a trace optimization problem. Its solution can be obtained from the Rayleigh-Ritz theorem. We list the eigenvectors according to the eigenvalues in ascending order. The solution can be approximated by the eigenvectors corresponding to the last K (largest K) eigenvalues of the matrix $2A - D$. Notice that minimization of $\Psi_2(S)$ is in fact the ratio-cut problem [29]. Although it itself can be applied for community identification, it often cuts a very small part from the network as a community by the approximation method as noted in our previous discussion.

To obtain a binary matrix S, which defines the partition of the network, we apply the K eigenvectors to do the k-means clustering for community assignments. By adding maximization of the sum of the average degrees in each subnetwork, the network can be divided into comparatively large communities, which avoids the problem that occurs when spectral clustering is applied alone. Overall, the algorithm is summarized as follows:

Algorithm

Input: adjacency matrix $A_{n\times n}$, and K, which is the number of communities.

(1) Compute the matrix $2A - D$.

(2) Compute the last K eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K$ of matrix $2A - D$.

(3) Construct a new matrix $T \in R^{n\times K}$, with columns $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K$.

(4) Cluster the points constructed from each row of matrix T with k-means clustering into communities $C_1, C_2, \ldots, C_K$.

Output: Index of vertices in each community.

To illustrate our method, we show the second eigenvector of the matrix L, which corresponds to the spectral clustering method, and the leading eigenvector of $2A - D$, which corresponds to our proposed method in Fig. 4 for the network shown in Fig. 1. Compared to the leading eigenvector of matrix B, it is easy to see that our method performs better in this case.

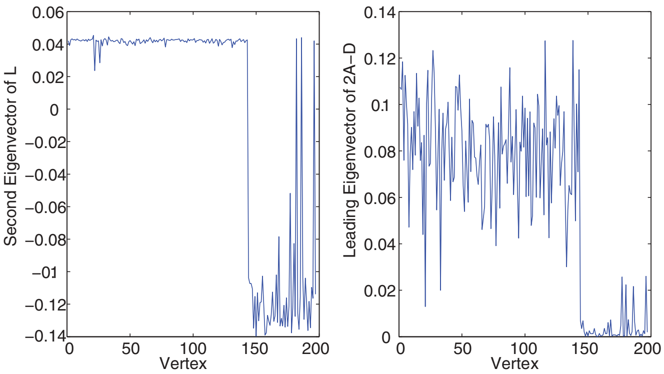


FIG. 4. (Color online) The second eigenvector of the matrix L (left) in the example in Sec. II and the leading eigenvector of the matrix $2A - D$ (right).

V. NUMERICAL EXPERIMENTS

In this section, we evaluate the performance of our proposed method through its application to several examples. We first start with several artificial networks having comparatively clear community structures. The size of the networks considered ranges from tens of vertices to thousands of vertices. We then apply our method to some real networks to evaluate its performance. The first real network is the well-known karate club network and the second is the dolphin social network. These two networks have been studied by many researchers for community identification. The third real network focuses on an American college football network, the fourth focuses on a gene regulation network from E. Coli, and the last focuses on a collaboration network. These examples show that the proposed metric is able to capture the community structure quite well.

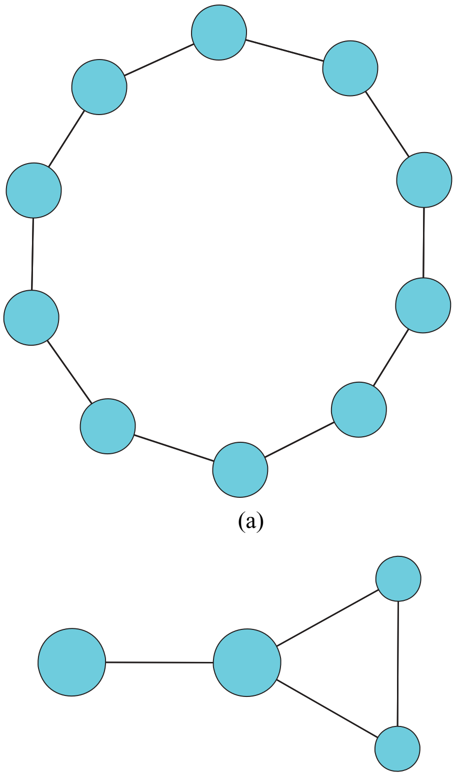
We compare our method with several popular methods, including Newman Modularity optimization (NM), Spectral Partitioning (SP), the Edge-betweenness Based method (EB) [1], and the information-theoretic method (infomod) [32]. For NM and EB, we use the programs developed in the R package "igraph" directly. For infomod, we use the package "infomod" developed by the authors. NM and infomod can select the number of communities. We use our proposed criterion shown in Eq. (3) to determine the number of communities for SP and our proposed Graph Partitioning method GP. For EB, the number of communities depends on how to cut the hierarchical tree. Here, we only consider its identification accuracy assuming the number of communities K is known.

A. Results for artificial networks

1. *Networks composed of cliques*

To investigate the resolution limit problem and evaluate the criterion for determining a community, we consider the networks composed of cliques because the structure of such networks can be analyzed theoretically.

(a) The first example is the network shown in Fig. 3, which is composed of six cliques with one of them connecting to all the others. With NM, this network will be one community without any subcommunity. We vary the value of K with K


(a)

(b)

FIG. 5. (Color online) (a) A network composed of identical cliques connected by single links as shown in Fig. 3(a) in [24]. (b) A network with two pairwise identical cliques as shown in Fig. 3(b) in [24]. Larger circles denote the larger cliques.

increasing from 1 until condition (3) does not hold. With our approach, the network is divided exactly into six communities, with each community being one clique. With our criterion, SP can identify all six communities. infomod can also find all the communities.

(b) We consider networks composed of identical cliques connected by single links shown in Fig. 5(a), which is the same as Fig. 3(a) in [24]. Suppose there are n_0 vertices in each clique and there are a total of K cliques. Then when K is greater than about the square root of the total number of links $[Kn_0(n_0-1)/2+K]$, NM will combine two or more cliques together. When the number of communities is set to be K, our proposed method can find all the communities because the objective function will be maximized under the correct partition. When the number of communities is greater than K, some cliques will be divided into two or more parts. In this case, the condition (3) for determining communities does not hold. Thus the total number of inferred communities will be K. We tried different values of K, and the results are consistent with our analysis. SP can get the same results as our method in this case. However, infomod cannot determine the number of communities correctly. For example, in a network composed of 25 identical cliques with five vertices each, it will identify six communities, with five of them having 20 vertices and one having 25 vertices.

(c) We consider networks consisting of cliques with different sizes, as shown in Fig. 5(b), which is the same as Fig. 3(b) in [24]. For a network composed of two pairwise identical cliques, with each pair having the same size, if the size of the two larger cliques is large enough compared to the two smaller cliques, NM will merge the two smaller cliques into one community. Similar to the analysis in (b), our method can correctly find the communities. Numerical tests also show that both our proposed method and SP can find all the communities. For this network structure, infomod will give the same results as NM in all our tests.

(d) In this example, we construct a network with comparatively large size. This network is composed of 10 cliques with sizes 5, 5, 10, 20, 40, 80, 160, 320, 640, and 1280. There are a total of 2560 vertices. Two cliques are connected by a single link in order of increasing size, e.g., the clique with 80 vertices connects the cliques with sizes 40 and 160. In addition, the largest clique is connected to the smallest clique with a single link. Finally, these cliques constitute a ring. The sizes of the communities in this network are quite unbalanced, and the increase of the size is not smooth. Due to the extensive computational time, we do not consider EB. NM identifies seven communities in the network. It combines the network with sizes 5, 5, 10, and 20 together. infomod has the same result as NM. Both SP and our proposed GP can make a correct inference of the communities.

(e) We consider a network with the same size and the same communities (cliques) as that in (d). This time one single link between any two communities is generated. In this case, NM identifies six communities, with the vertices in the two smallest cliques separated and assigned to larger communities. The cliques with sizes 10, 20, and 40 are combined into one community. All the other five cliques can be identified. infomod combines the cliques with sizes 5, 5, 10, and 20 together, with the remaining communities correctly inferred. Both SP and our proposed method GP can find all the communities correctly.

The above tests show that infomod also has the resolution limit problem in the identification of network communities in that it has difficulty in finding the communities with very small size. This problem also persists in our following examples. Because all the networks have clearly defined community structures and all the vertices have several links in one community, SP performs as well as our proposed method.

2. *Random networks with community structure*

In this subsection, we apply our proposed method and the other methods to randomly generated networks having community structure. Here, to see the identification accuracy, we assume the number of communities is known. We consider different settings for generating the random networks in the following.

(a) In this example, we let the size of the networks be 200 or 400. For the networks with 200 vertices, we assume that there are four communities in the network, with the community sizes being 10, 45, 45, and 100. For the network with 400 vertices, we assume there are six communities in the network, with sizes 10, 25, 45, 70, 100, and 150. We first partition all the vertices into four or six communities. We then connect each vertex with

TABLE I. Comparison of identification accuracy for networks constructed based on the degree of each vertex.

n	deg_{in}	r_{out}	SP	NM	EB	GP	infomod
200	2	0.3	**0.96**	0.62	0.70	**0.96**	0.98(3, 97%)
		0.5	0.64	0.53	0.52	**0.83**	–
	3	0.5	**1.00**	0.65	0.99	**1.00**	1.00(3, 100%)
		0.8	0.89	0.54	0.83	**0.99**	1.00(3, 100%)
400	2	0.3	0.97	0.53	0.60	**0.98**	0.99(4, 96%)
		0.5	0.82	0.46	0.40	**0.93**	1.00(5, 80%)
	3	0.5	**1.00**	0.56	0.98	**1.00**	1.00(5, 100%)
		0.8	0.98	0.51	0.87	**1.00**	1.00(5, 86%)

TABLE II. Comparison of identification accuracy for networks constructed based on the average degree of the vertices in one community.

n	deg_{in}	r_{out}	SP	NM	EB	GP	infomod
200	4	0.1	**0.97**	0.69	0.95	**0.97**	0.97(3, 82%)
		0.3	0.75	0.64	0.69	**0.94**	0.95(3, 92%)
	5	0.3	0.92	0.72	0.89	**0.98**	0.97(3, 99%)
		0.5	0.67	0.67	0.71	**0.96**	0.96(3, 98%)
400	4	0.1	0.96	0.61	0.87	**0.97**	0.94(6, 27%)
		0.3	0.80	0.57	0.53	**0.94**	0.94(4, 85%)
	5	0.3	0.89	0.59	0.79	**0.98**	0.97(5, 90%)
		0.5	0.62	0.55	0.56	**0.97**	0.96(5, 63%)

a given number of vertices in the same community (deg_{in}). Finally, we connect one community with other communities with a number of links. This number is proportional to its community size. We denote this ratio by r_{out}. For each case, we generated 100 networks. The identification accuracy is defined as the average accuracy from these 100 networks. We apply NM, SP, GP, EB, and infomod to these simulated networks. For infomod, we list the identification accuracy, the number of communities inferred, and the proportion of times that the method identified this number of communities. The accuracy of this method is based on the number of inferred communities, not the true number of communities. We merge the corresponding communities together to compute the identification accuracy. A dash means that only one community is inferred in the simulated networks. For all the other methods, we assume that the number of communities is known. In Table I, we highlight the highest identification accuracy with bold type except for infomod. Among these methods, our proposed method achieves the highest identification accuracy.

(b) In this example, we assume the sizes of the network and the communities are the same as above. We first partition the vertices into different communities. The vertices in one community are connected with a probability such that the community will have a given average degree. We remove the vertices that have no connections. Since the connections are randomly generated, the average degree will not be exactly our given number. By removal of the singleton vertices, the average degree of the community becomes larger. Finally, we connect the communities using the same method as that in the above example. For each case, we generate 100 networks. We also apply NM, SP, GP, EB, and infomod to these tests. The results are summarized in Table II. Again, we highlight the method with the highest identification accuracy with bold type. It is easy to see that our proposed method achieves the best performance.

These two examples suggest that our method performs well compared to other methods. Since the networks have very unbalanced structure, NM does not identify the communities well. In our analysis, we found that EB focuses more on the local structure of the network, and a good cutoff of the hierarchical tree is critical for the performance of this method. When a vertex connects several vertices within its community but very few vertices beyond its community, SP performs similarly to our proposed method. This is because maximizing the average degree within each community and minimizing the average number of connections between different communities leads to similar results in this case. In Tables I and II, SP and GP have similar performance under these conditions. However, when the connections within the communities are not dense, GP performs better than SP.

(c) We randomly generate a network with 2560 vertices partitioned into eight communities. The sizes of these communities are 20, 20, 40, 80, 160, 320, 640, and 1280. The construction of the network is similar to that in (b). Here, we let the average degree in each community be 5, and r_{out} is 0.5. Then we remove the singleton vertices, which results in a network with 2544 vertices. The structure of the network is shown in Fig. 6(a). We apply the above-tested methods to this network. Figures 6(b), 6(c), and 6(d) show the eight most important eigenvectors of the methods GP, SP, and NM. It can be seen that the eigenvectors of the GP method have a much clearer pattern of the communities. The identification accuracy of these methods is 0.98, 0.88, and 0.77, respectively. For the EB method, we cut its hierarchical structure into eight communities, where there are four communities of size 2. This also suggests that EB puts more emphasis on the local structure of the network. infomod only identifies four communities, where the communities with fewer than 320 vertices are grouped together.

B. Results for empirical networks

1. *Karate club network*

We consider Zachary's network of karate club members [33] in this example. There are 34 vertices in this network corresponding to the members in a karate club. This dataset has been applied as a benchmark to test many community identification algorithms since the true communities are known in this network. The people in the club were observed for a period of three years. The edges represent connections of the individuals outside the activities of the club. At some point, the administrator and the instructor of the club broke up due to a conflict between them. The club was separated into two groups supporting the administrator and the instructor. The question is whether it is possible to infer the composition of the two groups from the original network structure recorded during the three years. Figure 7 shows the network. Different shades of the vertices distinguish the two groups. The two communities have 16 vertices and 18 vertices, respectively, which can be seen as a very balanced community structure.

Among the community identification algorithms applied to this network, the major difference of the identification

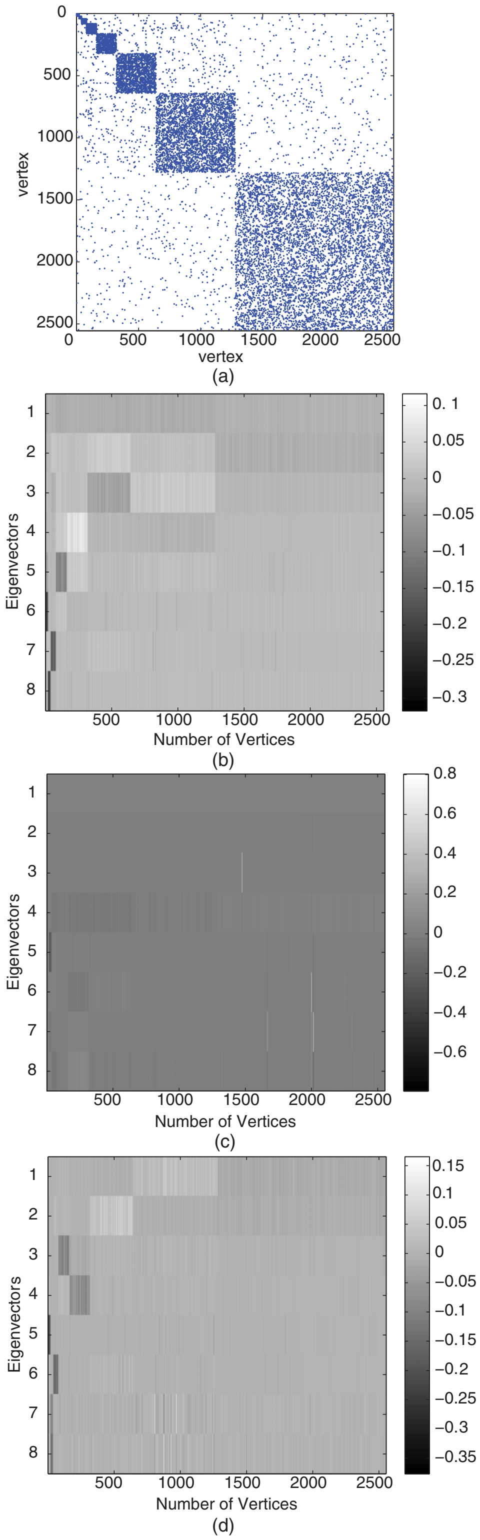


FIG. 6. (Color online) Pattern of the adjacency matrix A for a randomly generated network with 2544 vertices (a) and the eight most important eigenvectors for the three methods: (b) our proposed method; (c) spectral partitioning; and (d) modularity optimization method.

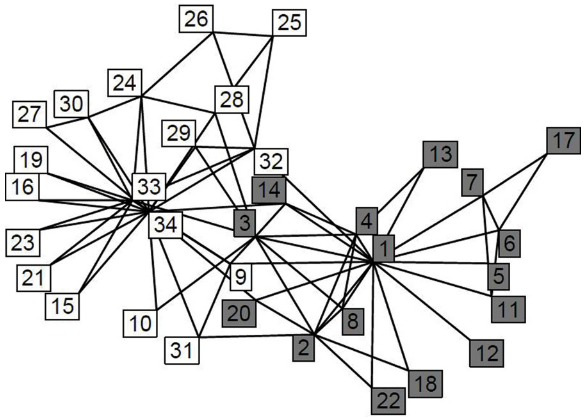


FIG. 7. The friendship network from Zachary's karate club study. The two communities are shown as either open or filled rectangles.

results is vertex 3. If we denote the community with the white background of the vertex notation in the figure as C_1, and the other community as C_2, then vertex 3 is in community C_1 with the method proposed in [1], while this vertex is in community C_2 with the methods proposed in [3,4,7]. With our proposed method, we can achieve the true partition of the two groups, and the objective value is 6.49.

2. Dolphin social network

The dolphin social network in this subsection consists of 62 bottlenose dolphins living in Doubtful Sound, New Zealand. The associations between different dolphins were observed over several years. There are a total of 159 connections in this network. Researchers found that this network can be divided into two small groups following some key members' departure in the population, and the structure of the network appears stable. Figure 8 shows the structure of the network. Different colors and shapes of the vertices show the two groups that the researcher found. The triangle vertex (vertex 37) is the vertex that departed. The sizes of the two communities are 20 and 42, which is somewhat unbalanced.

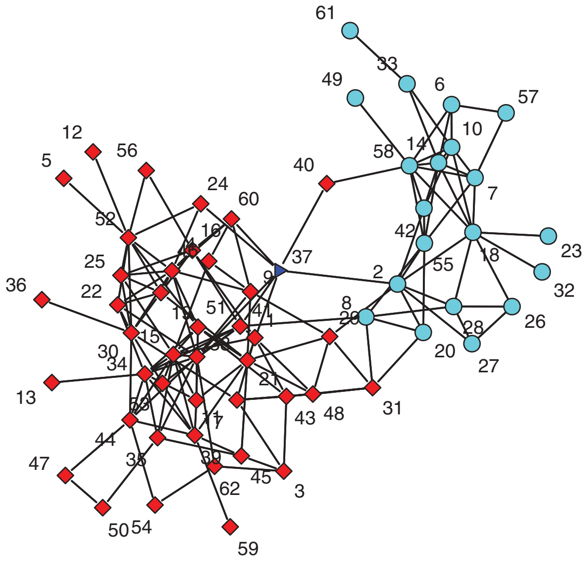


FIG. 8. (Color online) The dolphin social network of Lusseau *et al.* [34]. Different shapes show the different groups, with the triangle vertex belonging to the group with square vertices.

In [6], Newman compared the community identification results with NM and SP. With NM, three vertices—numbered 20, 40, and 48 in the network—were assigned to the wrong group, which outperformed SP. With our proposed method, both vertices 20 and 48 can be classified correctly. The only wrongly assigned vertex is 40, which has the same number of connections to both groups. In this network, the modularity obtained with our proposed method and NM is 0.38 and 0.39, respectively. The larger modularity does not correspond to a better community identification. infomod arrives at the same results as our method.

3. Collage football network

To evaluate the performance of our proposed method GP in a network with more than two communities, we turn to the U.S. college football network [1]. This network describes the American football games between Division IA colleges during the regular season in 2000. There were 115 teams and 613 games. The vertices in the network correspond to the teams, while the edges represent the games between any two teams connected. All the teams are divided into "conferences" containing about 8 to 12 teams, with teams in the same conference having more games among each other. In this network, there are 12 communities corresponding to the "conferences."

We apply the methods NM, SP, GP, EB, and infomod to identify the communities in this network. GP identifies 104 vertices correctly and SP identifies two more vertices correctly. The major differences among SP, GP, and the original true partition are mainly from the teams that do not belong to any conference (five members in all). These teams tend to be clustered with the conference with which they are most closely associated. EB achieves the same results as our method.

The modularity of NM, SP, and GP is 0.45, 0.59, and 0.60, respectively. If we use our proposed method in Eq. (3) to infer the number of communities, we will get a total of 13 communities, with the mid-American conference being divided into two. This division can be easily seen in [1], where EB is applied. infomod partitions the network into 10 communities with 99 vertices being correctly identified. Among these 10 communities, 7 communities are the same as those identified with our proposed method. The modularity of it is 0.60.

4. E. Coli transcriptional regulation network

In this section, we apply our proposed method to a transcriptional regulation network of E. Coli. The data set was downloaded from [http://www.weizmann.ac.il/mcb/UriAlon/]. It is a sparse network with 423 nodes and 519 edges. Since we can use some simple methods to find the separate parts of a network, we only consider the largest connected part of the network, which includes 328 nodes and 456 edges. We notice that when the modularity optimization is applied for community identification for the whole network, 27 communities are identified [24]. However, there are 35 unconnected subnetworks in the whole network, which also shows that it is better to detect the unconnected parts first.

Since we need to determine the cutoff for EB to determine the number of communities, we only apply NM, SP, GP, and infomod to this network, and we record the number of communities with size greater than 5. GP identifies 20 communities, SP identifies 14 communities, NM identifies 15 communities, and infomod identifies 5 communities. The modularity of these methods is 0.68, 0.66, 0.65, and 0.22, respectively, and among them, GP achieves the maximum value of modularity.

Since genes in the same community are more likely to have similar biological functions, we perform the enrichment analysis by using the KEGG pathways [35], with the R package GOSTATS from Bioconductor. For each community, the statistically most significant pathways are analyzed. We compare the enrichment results of the four methods and record the communities in which there are at least two genes in the same pathway. There are 9, 8, 8, and 3 communities enriched with GP, SP, NM, and infomod, respectively. GP identifies one more community than NM, which is enriched for the pathway "pyrimidine metabolism, purine metabolism, and metabolic pathways." SP identifies one different community from NM and GP, which is enriched for the pathway "two-component system." infomod enriches quite different pathways from the other three methods, with the largest community having a size of 328, which enriched the pathway "two-component system." The other two enriched pathways are "nucleotide excision repair" and "phenylalanine, tyrosine, and tryptophan biosynthesis."

5. General Relativity and Quantum Cosmology collaboration network

In this section, we consider a comparatively large network with 5242 vertices and 28 980 edges. This network is from the e-print arXiv and describes the scientific collaborations between authors with papers submitted to the General Relativity and Quantum Cosmology (arXiv GR-QC) category in the period from January 1993 to April 2003 (124 months). It begins within a few months of the inception of the arXiv, and thus represents essentially the complete history of its GR-QC section. In the network, if two authors coauthored a paper, there is an edge between them. If one paper was coauthored by t authors, a completely connected (sub)network on t vertices was generated. The data set was downloaded from [http://snap.stanford.edu/data/ca-GrQc.html].

Due to the extensive computation, we only apply the methods NM, SP, GP, and infomod (not EB) to this network. In this example, we mainly look at the structures of the identified communities. For NM, we determine the number of communities by maximizing the modularity for different numbers K. It identifies 258 communities in the network, and the modularity is 0.77. For SP, after calculation of its eigenvalues, we identify 355 unconnected parts. We use the k-means algorithm to cluster the eigenvectors corresponding to these eigenvalues. The modularity is 0.66. Due to the clustering efficiency, it may not result in 355 unconnected parts in reality. With infomod, 359 communities are identified and 8 communities have size greater than 10. The modularity is 0.72. For our proposed method, we set the number to be 359, which is the maximum number of communities among all these tested methods, and it satisfies the inequality (3). The modularity is 0.73.

TABLE III. The five communities with the highest average degree identified by the methods SP, NM, GP, and infomod.

SP	21.96(53)	16.11(19)	15.19(47)	15.00(16)	14.15(133)
NM	28.37(33)	21.88(43)	17.70(54)	15.00(82)	14.44(43)
GP	44.78(46)	37.49(43)	34.00(35)	23.00(24)	22.33(24)
infomod	44.78(46)	37.49(43)	32.47(38)	10.19(469)	6.00(7)

For each of these methods, we record the five communities with the highest average degrees, which are shown in Table III. Our method and infomod identify the same two communities, which have the highest average degrees. In addition to these two communities, our method identifies more communities with average degree higher than those identified by other methods. We have tried different numbers of K, from 30 to 400 for our method, and these five communities are always consistent. Such dense subnetworks are, in fact, what we aim to find in community identification.

C. Computational complexity analysis

Our proposed method includes two steps: the computation of the eigenvectors, and k-means clustering. We use the Lanczos algorithm shown in [30] to compute the last K eigenvectors of the matrix $2A - D$. This method first transforms the original matrix into a $K \times K$ tridiagonal matrix T_K. The transformation requires $O(Kn^2)$ for a dense matrix. For a sparse matrix, the computational complexity is about $O(Kn)$. Then the eigenvalues and eigenvectors of the matrix T_K are calculated with a QR algorithm. For a tridiagonal matrix, one QR decomposition requires $O(K)$ operations. If we set the number of iterations for QR to be q, then the computation of eigenvectors will require $O[K(q+n)]$ for a sparse matrix. We are now working on a fast algorithm for computing the eigenvectors by the randomized Lanczos method [31]. k-means clustering has been studied for a long time. It is an NP-hard problem in the general Euclidean space. When the clusters are comparatively clear, it converges very fast. Here we prespecify the total number of iterations n_{iter}. Then the leading term of the computational cost for Lloyd's algorithm is $O(nK^2 n_{\text{iter}})$. The total computational cost for our proposed method under our setting will be about $O[nK^2 n_{\text{iter}} + (q+n^2)K]$ for a dense network and $O[nK^2 n_{\text{iter}} + (q+n)K]$ for a sparse network. SP has the same computational cost if the same methods are applied for eigenvector calculation and k-means clustering. The computational complexity of EB is about $O(n^3)$ for sparse networks [1]. The computation of infomod is parameter-dependent [5]. The computational complexity for NM by using the leading eigenvector requires about $O(n^2 \log n)$. Overall, although our method is not the fastest, it is computationally competitive compared to other methods, as well as most methods presented in [5].

VI. CONCLUDING REMARKS

Research on community structure in networks has experienced an exponential growth in the past decades due to its importance in understanding various networks. Many methods have been proposed to identify communities from the observed data, and some of these methods have proven more effective than others in revealing network structures. Despite such rapid progress in methodology, a rigorous and functionally useful benchmark for comparing the community structure identification methods remains an open issue. In this paper, we have proposed a metric to identify network communities, which can work well even in unbalanced community networks. The computational method is very fast and easily implementable. Compared to popular methods in the literature, our method performed better both for simulated networks and some benchmark networks. It also led to biologically more specific results for an E. Coli gene regulation network that was considered.

As described in Ref. [36], networks may have topological scales, which means that the same network may have different community structures at different topological scales. Depending on the measure of communities, the vertices in the same community at one scale may be separated at other scales. For example, in a network composed of human acquaintances, at some scale the communities may be formed by families, and at some scale the communities may be formed by different countries. Since in a family people may be from different countries, the communities of different countries cannot be the combination of families directly. Thus it is not easy to construct the hierarchical structure of communities when the measures of defining communities have overlaps. Determining how to overcome such problems and construct the hierarchical structure of communities remains a major challenge.

Although many methods have been proposed for the identification of communities, and some methods estimate the number of communities, further studies are needed to better infer the number of communities. Also, due to the hierarchical structure of communities, the number of communities may vary for the same network. With our method, if we only use our proposed criterion to determine the number of communities, all the communities will be at the lowest level. According to the hierarchical structure of communities in a network, it may be more realistic to give several possible choices of the number of communities. This is also left for future work.

For the practical question of choosing a specific method for community identification, we suggest the users of these methods conduct a brief analysis of the network, such as dividing the network into unconnected parts and analyzing the degree distributions in order to choose a more appropriate method. In great contrast to the computational algorithms developed for module identification, there is a lack of theoretical analysis on the properties of these methods. Moreover, there is also a lack of literature on the mechanism with which these communities are generated in the first place. Progress in these two critical areas will undoubtedly shed light on the relative performance of different methods and also lead to better approaches to this fascinating problem.

ACKNOWLEDGMENTS

We thank the two anonymous reviewers for their constructive comments and suggestions. We thank Can Yang for his comments and suggestions. This work was supported in part by NSF Grant No. DMS 1106738, NIH Grants No. R01 GM59507 and No. P01 CA154295, and NSFC Grants No. 10901042, No. 10971075, and No. 91130032.

[1] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).
[2] A. Arenas, J. Borge-Holthoefer, S. Gómez, and G. Zamora, New J. Phys. **12**, 053009 (2010).
[3] J. Dong and S. Horvath, BMC Syst. Biol. **1**, 24 (2007).
[4] E. Estrada and N. Hatano, Phys. Rev. E **77**, 036111 (2008).
[5] S. Fortunato, Phys. Rep. **486**, 75 (2010).
[6] M. E. J. Newman, Phys. Rev. E **74**, 036104 (2006).
[7] M. E. J. Newman, Proc. Natl. Acad. Sci. USA **103**, 8577 (2006).
[8] M. A. Porter, J.-P. Onnela, and P. J. Mucha, Not. AMS **56**, 1082 (2010).
[9] F. Radicchi *et al.*, Proc. Natl. Acad. Sci. USA **101**, 2658 (2004).
[10] Z. Li, S. Zhang, R. S. Wang, X. S. Zhang, and L. Chen, Phys. Rev. E **77**, 036109 (2008).
[11] J. Zhang *et al.*, New J. Phys. **11**, 113003 (2009).
[12] R. Guimerà and L. A. N. Amaral, Nature (London) **433**, 895 (2005).
[13] X. Yang *et al.*, Genome Res. **20**, 1020 (2010).
[14] L. Danon *et al.*, J. Stat. Mech.: Theory Exp. (2005) P09008.
[15] E. R. Barnes, SIAM. J. Algebraic Discrete Meth. **3**, 541 (1982).
[16] L. R. Ford and D. R. Fulkerson, Canad. J. Math. **8**, 399 (1956).
[17] B. W. Kernighan and S. Lin, Bell Syst. Tech. J. **49**, 291 (1970).
[18] A. Pothen, Tech. Rep., Old Dominion University, Norfolk, VA, 1997.
[19] U. von Luxburg, Tech. Rep. 149, Max Planck Institute for Biological Cybernetics, 2006.
[20] P. Bickel and A. Chen, Proc. Natl. Acad. Sci. USA **106**, 21068 (2009).
[21] Y. Zhao *et al.*, Proc. Natl. Acad. Sci. USA **108**, 7321 (2011).
[22] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[23] M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).
[24] S. Fortunato and M. Barthélemy, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).
[25] B. H. Good, Y.-A. de Montjoye, and A. Clauset, Phys. Rev. E **81**, 046106 (2010).
[26] A. Khadivi, A. Ajdari Rad, and M. Hasler, Phys. Rev. E **83**, 046104 (2011).
[27] T. Richardson, P. J. Mucha, and M. A. Porter, Phys. Rev. E **80**, 036111 (2009).
[28] J. Ruan and W. Zhang, Phys. Rev. E **77**, 016104 (2008).
[29] L. Hagen and A. Kahng, IEEE Trans. Computer-aided Des. **11**, 1074 (1992).
[30] G. H. Golub and C. F. V. Loan, *Matrix Computation* (The John Hopkins University Press, Baltimore, MD, 1996).
[31] N. Halko, P. G. Martinsson, and J. A. Tropp, SIAM Rev. **53**, 217 (2011).
[32] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **104**, 7327 (2007).
[33] W. W. Zachary, J. Anthropol. Res. **33**, 452 (1977).
[34] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behav. Ecol. Sociobiol. **54**, 396 (2003).
[35] M. Kanehisa *et al.*, Nucl. Acid Res. **32**, D277 (2004).
[36] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, Phys. Rev. Lett. **96**, 114102 (2006).