SCIENTIFIC REPORTS

Correction: Publisher Correction

OPEN

Community Detection in Complex Networks via Clique Conductance

Zhenqi Lu[1], Johan Wahlström[2] & Arye Nehorai[1]

Received: 8 November 2017
Accepted: 20 March 2018
Published online: 13 April 2018

Network science plays a central role in understanding and modeling complex systems in many areas including physics, sociology, biology, computer science, economics, politics, and neuroscience. One of the most important features of networks is community structure, i.e., clustering of nodes that are locally densely interconnected. Communities reveal the hierarchical organization of nodes, and detecting communities is of great importance in the study of complex systems. Most existing community-detection methods consider low-order connection patterns at the level of individual links. But high-order connection patterns, at the level of small subnetworks, are generally not considered. In this paper, we develop a novel community-detection method based on cliques, i.e., local complete subnetworks. The proposed method overcomes the deficiencies of previous similar community-detection methods by considering the mathematical properties of cliques. We apply the proposed method to computer-generated graphs and real-world network datasets. When applied to networks with known community structure, the proposed method detects the structure with high fidelity and sensitivity. When applied to networks with no a priori information regarding community structure, the proposed method yields insightful results revealing the organization of these complex networks. We also show that the proposed method is guaranteed to detect near-optimal clusters in the bipartition case.

Networks are a standard representation of complex interactions among multiple objects, and network analysis has become a crucial part of understanding the features of a variety of complex systems[1–10]. One way to analyze networks is to identify *communities*, mesoscopic structures consisting of groups of nodes that are relatively densely connected to each other but sparsely connected to other dense groups in the network[11]. Communities, also called *clusters* or *modules*, mark groups of nodes which could, for example, share common properties, exchange information frequently, or have similar functions within the network[12]. The existence of communities is evident in many networked systems from a great many areas, including physics, sociology, biology, computer science, engineering, economics, politics, and neuroscience[13–20].

Community detection is important for many reasons. It allows classification of the functions of nodes in accordance with their structural positions in their communities[21–23]. It reveals the hierarchical organization that exists in many real-world networks[24]. Moreover, it improves the performance and efficiency of processing, analyzing, and storing networked data[25,26]. Communities also have concrete applications. In social networks, communities represent groups of individuals with mutual interests and backgrounds, and imply patterns of real social groupings[15]. In purchase networks, communities represent groups of customers with similar purchase habits, and can help establish efficient recommendation systems[26]. In citation networks, communities represent groups of related papers in one research direction, and identify scholars sharing research interests[27]. In brain networks, communities represent groups of nodes that are intricately interconnected and that could perform local computations, and they give insights into structural units of the brain[28].

The mathematical synonym of networks is *graphs*, and in the context of graph theory, one of the mathematical formalizations of community detection is *graph partitioning*. Guided by spectral graph theory[29], the method of spectral graph partitioning arose by relating network properties to the spectrum of the Laplacian matrix[30]. The earliest method in this category minimized connections between different communities[31,32]. In practice, this optimization problem can be efficiently solved, but it favors non-optimal solutions involving cutting a small part from the graph. One way to circumvent this drawback is to introduce balancing factors to the objective functions in order to enforce a reasonably large size for each community[33,34]. However, introducing balancing factors makes

[1]Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA. [2]Department of Computer Science, University of Oxford, Oxford, United Kingdom. Correspondence and requests for materials should be addressed to A.N. (email: nehorai@wustl.edu)

these optimization problems NP-hard[35]. Hence relaxed versions of these problems are solved by taking advantage of the properties of the Laplacian matrix.

Despite thousands of publications in the literature on spectral partitioning, these methods are constrained to conventional graph-based models. These models involve a set of vertices, which represent objects of interest, and a set of edges, which encode the existence or non-existence of a relationship between each pair of objects. However, in many real-world systems, the complex and rich nature of systems cannot be captured by such *dyadic* relationships. More importantly, recent computer innovations have greatly increased the size of the real networks that one can potentially handle. As a result, the way to process and understand graphs has been changed, and *polyadic* interactions are becoming more and more important. In particular, a community is intuitively a cohesive group of vertices that are "more densely" connected within the community than across communities[11]. The precise definition and characterization of "more densely" relies on polyadic interactions among multiple vertices. In order to quantitatively characterize polyadic structures, we employ the high-order structures of *cliques*, defined to be local complete subgraphs. In the context of networks, cliques are groups of objects that rapidly and effectively interact. This paper presents a graph-partitioning method that identifies clusters of cliques.

One line of related work is the method of k-clique percolation[36,37]. This method defines the k-clique community to be the union of "adjacent" k-cliques, which by definition share $k-1$ vertices, where k is any positive integer. However, this definition is too stringent because it rules out other possible communities that are not so well-connected. Its performance also relies heavily on the choice of k: A small k leads to a single giant community, and a large k leads to multiple small and possibly distant communities. In addition, this definition includes topological cavities[38], which enclose holes in networks and mark local lacks of connectivity. However, this feature is not an expected property of communities.

In a recent paper, Benson *et al*. devised a community-detection method based on high-order connectivity patterns called network motifs[39,40], and proposed a generalized framework for identifying clusters of network motifs[41]. Cliques are certainly one special kind of network motif, and Benson *et al*. provide numerical simulations for applying this framework to cliques. However, this framework has several drawbacks. First, the framework fails to consider the nested nature of cliques and so suffers from unnecessary computational cost, since it needs to take into consideration non-maximal cliques. Second, the method requires pre-specification of the sizes of the cliques involved, instead of considering all clique sizes occurring in the network. Third, the conductance function merely counts the number of cliques and ignores other properties influenced by partitions. Lastly, the performance guarantee works only for 3-cliques. We overcome all these drawbacks by designing a novel conductance function specifically for cliques.

In this paper, we propose a novel community-detection method that minimizes a new objective function, called the clique conductance function. We encode in this objective function the number and sizes of cliques, and the numbers of edges in the cliques. Finding a partition that exactly minimizes the clique conductance is computationally intractable. Thus we extend the spectral graph partitioning methodology, and devise a computationally tractable solution that approximately minimizes the clique conductance. In addition, we derive a performance guarantee for the bipartition case, showing that the resulting bipartition is near-optimal. Finally, we apply the proposed method to computer-generated graphs and real-world network datasets. When applied to networks with known community structure, the proposed method achieves excellent agreement with the ground-truth communities. When applied to networks with no a priori information regarding community structure, the proposed method yields insightful results that help us understand the structures embedded in these complex networks.

Methods

In this section, we describe our proposed graph-partitioning method. We begin by introducing several graph notations, and then state the formulation of our proposed graph-partitioning method based on clique-conductance minimization. We conclude this section by proposing a computationally efficient algorithm that approximately solves the optimization problem.

Graph Notations. An undirected weighted graph $\mathcal{G}$ is an ordered triplet $(\mathcal{V}, \mathcal{E}, \pi)$ consisting of a set of vertices $\mathcal{V} = \{v_1, \ldots, v_n\}$, a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ satisfying $(u, v) \in \mathcal{E}$ if and only if $(v, u) \in \mathcal{E}$ for all $u, v \in \mathcal{V}$, and a weight function π: $\mathcal{V} \times \mathcal{V} \to \mathbb{R}^+ \cup \{0\}$ satisfying $\pi(\mathcal{E}) > 0$, $\pi(\mathcal{V} \times \mathcal{V} - \mathcal{E}) = 0$, and $\pi(u, v) = \pi(v, u)$ for all $u, v \in \mathcal{V}$. If the weight function π in addition satisfies $\pi(\mathcal{E}) = 1$, then $\mathcal{G}$ is an undirected binary graph. The weighted adjacency matrix $\mathbf{W}$ of the graph is defined as $\mathbf{W}(i, j) := \pi(v_i, v_j)$. Since $\mathcal{G}$ is undirected, we have $\mathbf{W} = \mathbf{W}^{\mathrm{T}}$. The degree of a vertex v_i is defined as $d_i := \sum_{u \in \mathcal{V}} \pi(u, v_i)$, and the degree matrix $\mathbf{D}$ is a diagonal matrix with $d_1, \ldots, d_n$ as diagonal entries. The Laplacian matrix $\mathbf{L}$ of the graph is defined as $\mathbf{L} := \mathbf{D} - \mathbf{W}$. A graph $\mathcal{G}$ is said to have no loops if $\pi(u, u) = 0$ for all $u \in \mathcal{V}$.

Formally, a k-clique is a subgraph consisting of k nodes with all pairwise connections, where k is any positive integer. It naturally follows from the definition that any subgraph of a clique is also a clique, and such a subgraph is called a *face*. We call this feature the *nested nature* of cliques. A *maximal* clique is a clique that is not a face. Due to the nested nature of cliques, the maximal cliques of a graph contain all the clique information. The number of vertices constituting a clique σ is called the size of a clique and is denoted as $\omega(\sigma)$. In this paper, we use $\mathcal{M}_k$ to represent the collection of all maximal k-cliques, and $\mathcal{M} = \bigcup_k \mathcal{M}_k$ to represent the collection of all maximal cliques.

Clique Conductance Minimization. We now state the formulation of our proposed graph-partitioning method. Intuitively, the graph-partitioning problem based on cliques can be described as follows: We wish to

find a partition of the graph, such that cliques between different groups are few and have small sizes (which means that vertices in different clusters share few high-order connections), and cliques within each group have large sizes (which means that vertices within one cluster are connected in high-order fashion). Formally, suppose that $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \pi)$ is an undirected binary graph with no loops. Given a positive integer $m > 1$, we wish to find a partition $(A_1, \ldots, A_m)$ that satisfies $A_i \cap A_j = \varnothing$ for any $i \neq j$ and $\bigcup_i A_i = \mathcal{V}$, and that minimizes

$$\psi(A_1, \ldots, A_m) := \sum_{i=1}^{m} \mathrm{cut}(A_i, \bar{A}_i), \tag{1}$$

where

$$\mathrm{cut}(A, \bar{A}) := \sum_{\sigma \in \mathcal{M}} \omega(\sigma) \sum_{u,v \in \sigma} \mathbb{1}(u \in A, v \in \bar{A}), \tag{2}$$

where $\mathbb{1}$ is the truth-value indicator function. Conceptually, the cut function $\mathrm{cut}(A, \bar{A})$ measures how severely maximal cliques are influenced by the partition $(A, \bar{A})$. The cut function considers both the number and sizes of maximal cliques that are cut by the partition, and also the number of edges in each maximal clique that are cut by the partition. Unfortunately, in practice the solution of this approach often yields extreme cases separating the vertex with the lowest degree from the rest of the graph, similar to phenomena observed in minimizing conventional cut functions[31]. To circumvent this problem, we introduce a balancing factor

$$\mathrm{vol}(A) := \sum_{\sigma \in \mathcal{M}} \omega(\sigma) \sum_{u \in \sigma} \mathbb{1}\Big(u \in A), \tag{3}$$

which conceptually measures the size of a cluster A, and propose to minimize the clique conductance function defined as

$$\phi(A_1, \ldots, A_m) := \sum_{i=1}^{m} \frac{\mathrm{cut}(A_i, \bar{A}_i)}{\min(\mathrm{vol}(A_i), \mathrm{vol}(\bar{A}_i))}. \tag{4}$$

We note that this objective function is formulated in a similar way to normalized spectral partitioning[34]. However, introducing balancing factors causes the computationally tractable problem of minimizing equation (1) to become NP-hard[35]. Following the idea of spectral graph partitioning[30], we next reformulate our optimization problem and seek a computationally tractable solution.

Partitioning Algorithm. We introduce a new weighted graph, which we call the induced clique graph, to encode the maximal-clique information of $\mathcal{G}$. The induced clique graph of $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \pi)$ is an undirected weighted graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}, \pi_c)$, where the weight function π_c is defined as

$$\pi_c(u, v) := \sum_{\sigma \in \mathcal{M}} \sum_{u,v \in \sigma} \omega(\sigma). \tag{5}$$

By definition, $\pi_c(u, v)$ is the sum of the sizes of the maximal cliques that vertex u and vertex v both engage. Intuitively, π_c measures how densely two vertices are connected in $\mathcal{G}$. We denote by $\mathbf{W}_c$, $\mathbf{D}_c$, $\mathbf{L}_c$ the corresponding adjacency matrix, degree matrix, and Laplacian matrix, respectively. Following this spirit, the graph-partitioning problem on an undirected binary graph $\mathcal{G}$ can be transformed and implemented as a graph-partitioning problem on a weighted graph $\mathcal{G}_c$. Notice that a partition $(A_1, \ldots, A_m)$ on the original network $\mathcal{G}$ induces a partition on the induced clique graph $\mathcal{G}_c$. To measure conductance on this weighted graph, we recall the traditional conductance function on weighted graphs[30], defined as

$$\phi_c(A_1, \ldots, A_m) := \sum_{i=1}^{m} \frac{\mathrm{cut}_c(A_i, \bar{A}_i)}{\min(\mathrm{vol}_c(A_i), \mathrm{vol}_c(\bar{A}_i))}, \tag{6}$$

where

$$\mathrm{cut}_c(A, \bar{A}) := \sum_{u \in A, v \in \bar{A}} \pi_c(u, v) \tag{7}$$

is the total weight of edges cut, and

$$\mathrm{vol}_c(A) := \sum_{u \in A, v \in \mathcal{V}} \pi_c(u, v) \tag{8}$$

is the total connection from vertices in A to all vertices in the graph. The next proposition relates the traditional conductance function in equation (6) to the clique conductance function in equation (4).

Algorithm 1. Graph partitioning via clique conductance minimization.

Input : Adjacency matrix $\mathbf{W} \in \{0,1\}^{n\times n}$, number m of clusters to construct
Output : A partition of the network $(A_1, \ldots, A_m)$
Compute the maximal cliques from the adjacency matrix $\mathbf{W}$ using the Bron-Kerbosch algorithm;
Form the clique weight matrix $\mathbf{W}_c$ and the corresponding normalized Laplacian matrix $\mathcal{L}_c = \mathbf{D}_c^{-1/2}(\mathbf{D}_c - \mathbf{W}_c)\mathbf{D}_c^{-1/2}$;
if $m = 2$ **then**
    Compute the second eigenvector $\boldsymbol{h}$ of $\mathcal{L}_c$;
    For $1 \le i \le n$, let σ_i be the index of the i-th largest entry of $\boldsymbol{g} = \mathbf{D}_c^{-1/2}\boldsymbol{h}$;
    Set $A_1 = \arg\min_i \phi(S_i)$, where $S_i = \{v_{\sigma_1}, \ldots, v_{\sigma_i}\}$.
else
    Compute the first m eigenvectors of $\mathcal{L}_c$;
    Let $\mathbf{U} \in \mathbb{R}^{n\times m}$ be the matrix containing these eigenvectors as columns;
    Form $\mathbf{T}$ from $\mathbf{U}$ by normalizing all columns to norm 1;
    For $1 \le i \le n$, let $y_i \in \mathbb{R}^m$ be the i-th row of $\mathbf{T}$;
    Cluster the points $\{y_i\}_i$ using the k-means algorithm[48] into clusters $C_1, \ldots, C_m$;
    For $1 \le i \le m$, form cluster $A_i = \{v_j : y_j \in C_i\}$.
end

Proposition 1. *Given any undirected binary graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \pi)$, for any subset $A \subset \mathcal{V}$,* we have

$$\mathrm{cut}(A, \overline{A}) = \mathrm{cut}_c(A, \overline{A}), \tag{9}$$

$$\mathrm{vol}(A) = \mathrm{vol}_c(A). \tag{10}$$

The proof of Proposition 1 is given later. A straightforward consequence of Proposition 1 is that the conductance functions as shown in equations (4) and (6) are equal.

Corollary 2. *Given any undirected binary graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \pi)$, for any natural number $m > 1$ and any partition $(A_1, \ldots, A_m)$, we have*

$$\phi(A_1, \ldots, A_m) = \phi_c(A_1, \ldots, A_m). \tag{11}$$

Corollary 2 shows that the clique conductance minimization problem,

$$\underset{(A_1,\ldots,A_m)}{\text{minimize}}\ \phi(A_1, \ldots, A_m), \tag{12}$$

is equivalent to the conductance minimization problem on the induced weighted graph,

$$\underset{(A_1,\ldots,A_m)}{\text{minimize}}\ \phi_c(A_1, \ldots, A_m). \tag{13}$$

Solving this minimization problem directly can be computationally intractable[35]. One way to circumvent this issue is to solve a relaxed version of this problem by employing normalized spectral partitioning[30,34,42]. Thus our partitioning algorithm consists of three steps. First the maximal cliques are computed using the Bron-Kerbosch algorithm[43–46]. Then the induced clique graph $\mathcal{G}_c$ is formed. Finally, normalized spectral partitioning[42] is applied to achieve a partition of the graph $\mathcal{G}$. Our partitioning algorithm is stated in detail in Algorithm 1. As shown in Algorithm 1, we use two different clustering methods for $m = 2$ and $m > 2$ when applying normalized spectral partitioning, because for $m = 2$ the Cheeger inequality ensures that this clustering method produces a near-optimal partition, as shown later. For the general case of $m > 2$, there are no similar results providing performance guarantees. Among the several spectral partitioning methods[30], we choose normalized spectral partitioning[42] because of the construction of the clique conductance function. A recent work provides a performance guarantee for the general case, but the proof is constrained to regular binary graphs and is based on a new and untested clustering method[47]. We choose to keep using the k-means clustering method for its ease of implementation and successful empirical results.

Empirical Results

In this section we present a number of numerical experiments with the proposed method. We first perform experiments on computer-generated graphs, and then apply the proposed method to real-world networks with known community structures. In each case, we find that the proposed method almost perfectly detects community structures indicated by network connectivity.

Benchmarks. We use benchmarks to compare the proposed method to the motif-conductance method[41], the normalized spectral partitioning[34], and greedy methods, including the Louvain method[48], the Ravasz method[49], and the fast modularity maximization method[50–52]. Benchmarks are computer-generated graphs whose community structure is known. To compare two partitions $\mathcal{C}_1$, $\mathcal{C}_2$ of the same graph, we use the normalized mutual information[53,54], defined as

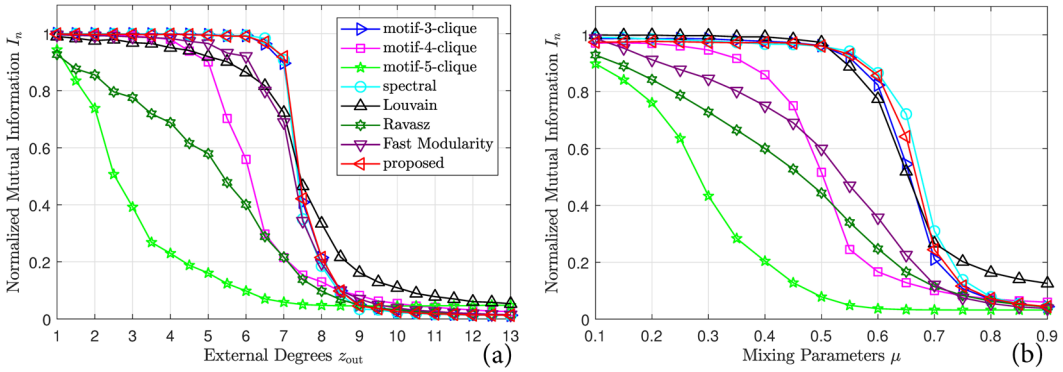

Figure 1. Normalized mutual information of different community-detection methods on (**a**) the Girvan-Newman benchmark, (**b**) the Lancichinetti-Fortunato-Radicchi benchmark (using the same legend as subfigure (a)).

$$I_n(\mathcal{C}_1, \mathcal{C}_2) := \frac{\sum_{c_1\in\mathcal{C}_1, c_2\in\mathcal{C}_2} p(c_1, c_2)\log\frac{p(c_1,c_2)}{p(c_1)p(c_2)}}{\frac{1}{2}\mathcal{H}(\mathcal{C}_1) + \frac{1}{2}\mathcal{H}(\mathcal{C}_2)}. \tag{14}$$

Here, $p(c)$ is the probability that a randomly chosen vertex belongs to community c, $p(c_1, c_2)$ is the probability that a randomly chosen vertex belongs to both community c_1 and community c_2. Also, $\mathcal{H}(\mathcal{C})$ is the Shannon entropy, defined as

$$\mathcal{H}(\mathcal{C}) := -\sum_{c\in\mathcal{C}} p(c)\log p(c). \tag{15}$$

Intuitively, the normalized mutual information measures the similarity between two partitions. If the two partitions $\mathcal{C}_1$, $\mathcal{C}_2$ are identical, then $I_n(\mathcal{C}_1, \mathcal{C}_2) = 1$, and if the two partitions are independent of each other, then $I_n(\mathcal{C}_1, \mathcal{C}_2) = 0$. In the following experiments, $\mathcal{C}_1$ is the ground-truth partition given by the benchmark, and $\mathcal{C}_2$ is the partition predicted by a community-detection method.

The first benchmark we use is the Girvan-Newman (GN) benchmark[55]. Here, each graph is composed of 128 vertices and is partitioned into 4 communities of size 32. Each vertex is connected to approximately 16 others. For each vertex, a fraction z_{out} of 16 connections is made to randomly chosen vertices of other communities, and the remaining connections are made to randomly chosen members of the same community. When z_{out} is a half-integer $k + \frac{1}{2}$, half of the vertices have k inter-community connections and the other half have $k + 1$ inter-community connections. The GN benchmark produces graphs with known community structures, which are essentially random in all other aspects.

The results of different community-detection methods compared against the GN benchmark are shown in Fig. 1a. Each curve is averaged over 1000 realizations. As can be seen, the proposed method achieves complete mutual information when $z_{out} \leq 7$, detecting virtually correct communities. The proposed method yields almost zero mutual information when $z_{out} \geq 9$, where each vertex has more inter-community connections than intra-community connections. The transition between these two regions is swift and sharp. In other words, the proposed method performs almost perfectly up to the point where each vertex has as many inter-community connections as intra-community connections. This performance is almost optimal, because the ground-truth community structure diminishes when each vertex has more inter-community connections than intra-community connections. In this situation, the community structure represented by graph connections deviates from the ground-truth community structure, and so these two sets of clusters share little mutual information. The normalized spectral partitioning and the motif-conductance method using 3-cliques as the network motif perform as well as the proposed method. But when 4-cliques and 5-cliques are chosen as network motifs, the performance of the motif-conductance method degrades severely. This degradation shows that the motif-conductance method heavily relies on the choice of, and prior knowledge about, which cliques are overexpressed in a graph. Finding this knowledge and determining this choice necessarily involve a brute-force search over all subgraphs of certain sizes. Among the greedy methods, the Louvain method and the fast modularity method offer the best performance, but compared to the proposed method, the accuracies of both methods are lower when $z_{out} \leq 7$.

The GN benchmark generates a random graph where all vertices have approximately same degrees and all communities have an identical size. However, many real-world networks are scale-free[56], with node degrees and community sizes following the power-law distribution. As a result, a community-detection method that performs well on the GN benchmark might fail on real-world networks. To ensure that the proposed method does not suffer from this limitation, we use the Lancichinetti-Fortunato-Radicchi (LFR) benchmark as a second benchmark[57], where both vertex degrees and ground-truth community sizes follow the power-law distribution. In this benchmark, each graph is composed of n vertices and is partitioned into m communities. Each vertex is given a degree following a power-law distribution with exponent γ, and each community is given a size following a power-law distribution with exponent β. The minimal and maximal values of degrees, k_{min}, k_{max}, and of community sizes,

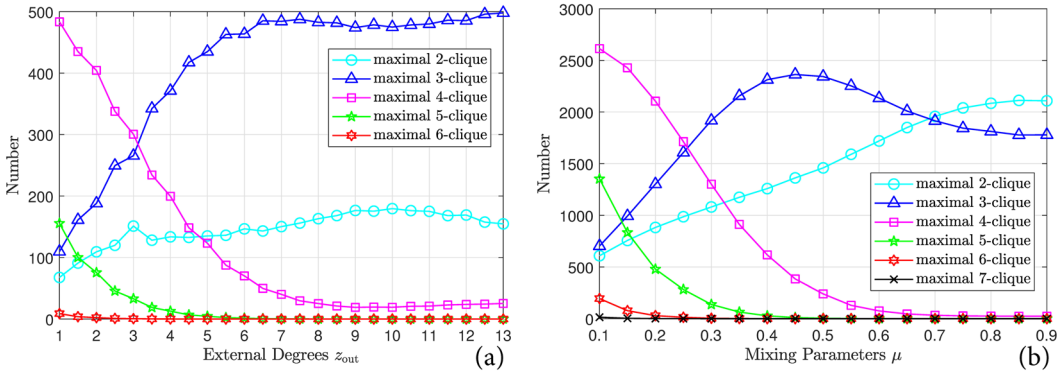

Figure 2. Distribution of sizes of maximal cliques in (**a**) the Girvan-Newman benchmark, (**b**) the Lancichinetti-Fortunato-Radicchi benchmark.

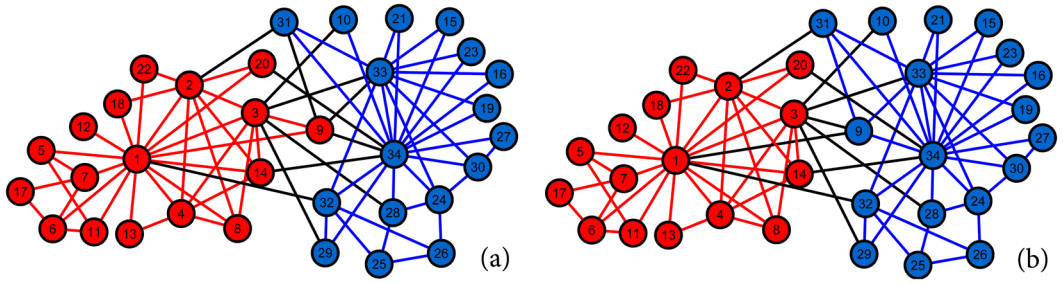

Figure 3. The friendship network from Zachary's karate club study. (**a**) The communities observed by Zachary. (**b**) The communities detected by the proposed method.

s_{min}, s_{max}, are chosen such that $k_{min} < s_{min}$ and $k_{max} < s_{max}$. For each vertex, a fraction $1 - \mu$ of its connections is made to randomly chosen members of the same community, and the remaining connections are made to randomly chosen members of other communities. A realization of this benchmark is constructed via the following steps. At the beginning, all vertices are homeless, i.e., they belong to no communities. Each vertex is assigned to a randomly chosen community with a size greater than the vertex degree. If the community is already full, a randomly chosen member of this community is kicked out. This procedure continues until each vertex is assigned to a community. Then connections are randomly generated while preserving the ratio between the external and internal degrees of each vertex.

The results of different community-detection methods compared against the LFR benchmark are shown in Fig. 1b, with parameters chosen as $n = 500$, $m = 10$, $k_{min} = 20$, $k_{max} = 80$, $\gamma = 2$, $s_{min} = 30$, $s_{max} = 100$, and $\beta = 1.1$. Each curve is averaged over 1000 realizations. The results are similar to those on the GN benchmark. The proposed method, the normalized spectral partitioning method, and the motif-conductance method using 3-cliques perform similarly: All closely approximate complete mutual information when $\mu \leq 0.5$, and yield nearly zero mutual information when $\mu \geq 0.8$. The performance of the motif-conductance method degrades severely when 4-cliques and 5-cliques are chosen as network motifs. The performances of the greedy methods are similar to their performances on the GN benchmark, except that the fast modularity method has a much lower accuracy when $\mu \leq 0.6$.

To further validate the advantage of the proposed method over the motif-conductance method, we depict in Fig. 2 the size distribution of maximal cliques in both benchmarks averaged over 1000 realizations. The distributions in both benchmarks are similar. When z_{out} and μ are small, the 4-cliques are the dominant maximal cliques and other maximal cliques generally have sizes of 2, 3, and 5. With increasing z_{out} and μ, the numbers of 4-cliques and 5-cliques decrease rapidly and are exceeded by the numbers of 2-cliques and 3-cliques when approximately 1/3 of the connections of each vertex are inter-community. In the GN benchmark, the number of 3-cliques keeps growing after this point and remains the most numerous maximal clique. But in the LFR benchmark, the number of 3-cliques is exceeded by the number of 2-cliques when $\mu > 0.7$. Given these patterns in the distributions, it is not surprising that the motif-conductance method performs poorly when 4-cliques and 5-cliques are chosen as network motifs. These distributions also further demonstrate the advantage of the proposed method. In practice, the distribution of cliques (and other network motifs) is mostly probably unavailable when one is processing observed network data. Collecting this information is computationally expensive. Since the maximal cliques contain all the clique information, the proposed method is able to process general networks with no prior knowledge of clique sizes and clique locations.

In summary, the proposed method achieves state-of-the-art performance on the homogeneous GN benchmark and on the scale-free LFR benchmark. In addition, the proposed method yields almost the optimal performance

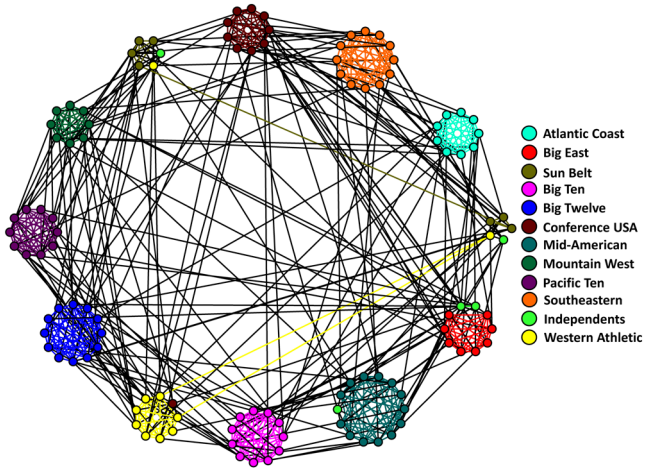

Figure 4. Communities of college football network, using colors for conferences and spatial clusterings for identified communities.

one could expect on these two benchmarks: The proposed method detects the pre-defined ground-truth community structure when it is well represented by connections, and deviates from it when the ground truth diminishes. This behavior explains why there is little improvement over the existing methods. As opposed to the motif-based method, the proposed method also benefits from the fact that it requires no pre-specification of clique sizes. As a result, the proposed method bypasses a computationally expensive search for the optimal choice of clique sizes.

Zachary's Karate Club. We apply our method to the network from the well-known karate club study by Zachary[58]. This study followed a social network composed of 34 members and 78 pairwise links observed over a period of three years. During the study, a political conflict arose between the club president (node 34) and the instructor (node 1). This political conflict later caused the club to split into two parts, each with half of the members. Zachary recorded a network of friendships among members of the club shortly before the fission, and a simplified unweighted version is shown in Fig. 3a. Different node colors are used in this figure to show the two factions of the fission after the political conflict.

Figure 3b shows the community structure detected by the proposed method. The identified communities almost perfectly reflect the two factions observed by Zachary, with only 1 (node 9) out of 34 nodes "incorrectly" assigned to the opposing faction. This exception can be explained by the conflict of interest faced by individual number 9. As recorded by Zachary, individual number 9 was a weak political supporter of the club president before the fission, but not solidly a member of either faction[58]. This ambivalence is revealed by the fact that node 9 is engaged in two maximal 3-cliques, on nodes {1, 3, 9} and on nodes {3, 9, 33}, and one maximal 4-clique on nodes {9, 31, 33, 34}, implying that node 9 is weakly more densely associated with members of the club president's faction. On the other hand, Zachary pointed out that individual number 9 had an overwhelming interest in staying associated with the instructor, which was not shared by any other member of the club. Individual number 9 was facing his black-belt exam in three weeks, and joining the club president's faction would result in renouncing his rank and starting over again[58]. In other words, individual number 9 would have joined the club president's faction, if this conflict of interest had not emerged. Therefore, the proposed method perfectly detected the social communities in an empirically observed network of friendships.

College Football Network. We then apply the proposed method to a more complex real-world network with known community structures. The network represents the schedule of United States football games between Division IA colleges during the regular season in Fall 2000[55]. The network is shown in Fig. 4, where the nodes represent teams, and the links represent regular season games between the two teams connected. The known communities are defined by conferences, each containing around 8 to 12 teams and marked with colors. Links representing intra-conference games are also marked with the same colors as the corresponding conferences. In principle, teams from one conference are more likely to play games with each other than with teams belonging to different conferences. There also exist some independent teams that do not belong to any conference, and these teams are marked with a light-green color.

The communities identified by the proposed method are represented by spatial clusterings in Fig. 4. In general, the proposed method correctly clusters teams from one conference. The independent teams are clustered with conferences with which they played games most frequently, because the independent teams seldom play games between themselves. The clusters detected by the proposed method deviate from the conference segmentation in several ways. First, the Sun Belt conference, marked with a brown color, is split into two parts, shown at the eleven o'clock and three o'clock directions, and each part is grouped with teams from the Western Athletic conference, marked with a yellow color, and independent teams. But this result is understandable given the fact that there was only one game involving teams from both these two parts. Second, one team from the Conference USA conference, marked with a dark red color, is clustered with teams from the Western Athletic conference. This team

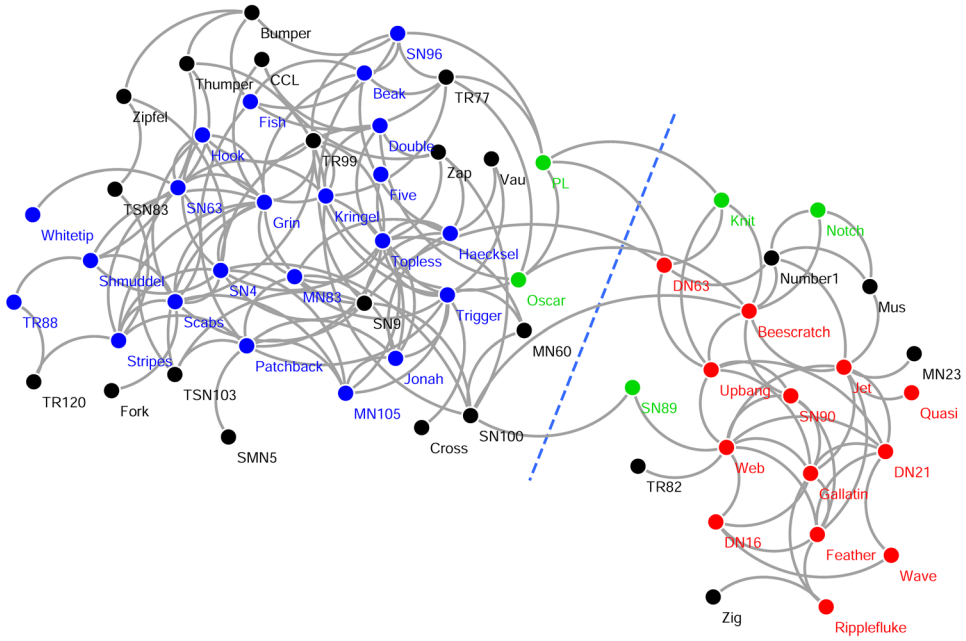

Figure 5. Social network of 62 bottlenose dolphins. The nodes are colored based on the groups observed in the study by Lusseau *et al.*[59]. The spatial clustering represents communities detected by the proposed method.

played no games with other teams from the Conference USA conference, but played games with every team from the Western Athletic conference. Third, two teams from the Western Athletic conference are isolated from other teams from this conference, and each is grouped with part of the Sun Belt conference. The team at eleven o'clock had no intra-conference game, and the team at three o'clock had only two intra-conference games, but they had inter-conference games with every member of the cluster that they are assigned to. In summary, the proposed method perfectly reflected the community structures established in regular-season-game association, and in addition detected the lack of intra-conference association that the known community structure fails to represent.

Applications to Complex Real-World Networks

In the previous section, we tested the proposed method on both computer-generated graphs and real-world networks for which the community structures are well-defined and known a priori. In this section, we apply the proposed method to complex real-world networks of which the community structures are not known, and show that the proposed method helps us understand these complex networks. For each application example, the number of communities is chosen based on prior information regarding the datasets.

Bottlenose Dolphin Social Network. Our first example is a social network composed of 62 bottlenose dolphins living in Doubtful Sound, New Zealand[59]. The social ties between dolphin pairs are established based on direct observations conducted during a period of seven years by Lusseau *et al.* The clustering analysis conducted by Lusseau *et al.* on 40 of these dolphins shows that three groups spent more time together than all individuals did on average, but group 1 is relatively weak in the sense that it is an artifact of the similar likelihood of encountering these individuals in the study area[59]. Figure 5 shows the social network of bottlenose dolphins, where nodes represent dolphins and links represent social ties. The three groups observed by Lusseau *et al.* are colored in green, red, and blue, respectively, and the dolphins not involved in the clustering analysis by Lusseau *et al.* are left in black. The dashed line denotes the community division found by the proposed method. As can be seen, the achieved division corresponds well with the observed groups, separating the red and blue groups into two communities. The green group (group 1) is split evenly between the two detected communities. This phenomenon is understandable, because group 1 is a weak group and is not well represented by the social network since most of its members share no social ties.

Food Web. Our second example is a food web representing the carbon exchange among 128 compartments (organisms and species) occurring during the wet and dry seasons in the Florida Bay ecosystem[60], as shown in Fig. 6. In this network, nodes represent compartments, and links represent energy flow (the link from node i to node j means that carbon is transferred from node i to node j). Part of the compartments are classified into a total of 13 groups (Part of the groups were compiled by Benson *et al.*[41]), as marked with different colors in Fig. 6. The remaining compartments are left in grey. This network is a directed network, and we apply the proposed method to a simplified version with each directed edge converted to an undirected edge.

The communities detected by the proposed method are divided by the dashed lines. The division corresponds quite closely with the division of groups of compartments. The clustering reveals four known aquatic layers: macroinvertebrates and microbial microfauna (left), sediment organism microfauna (bottom), pelagic fishes and

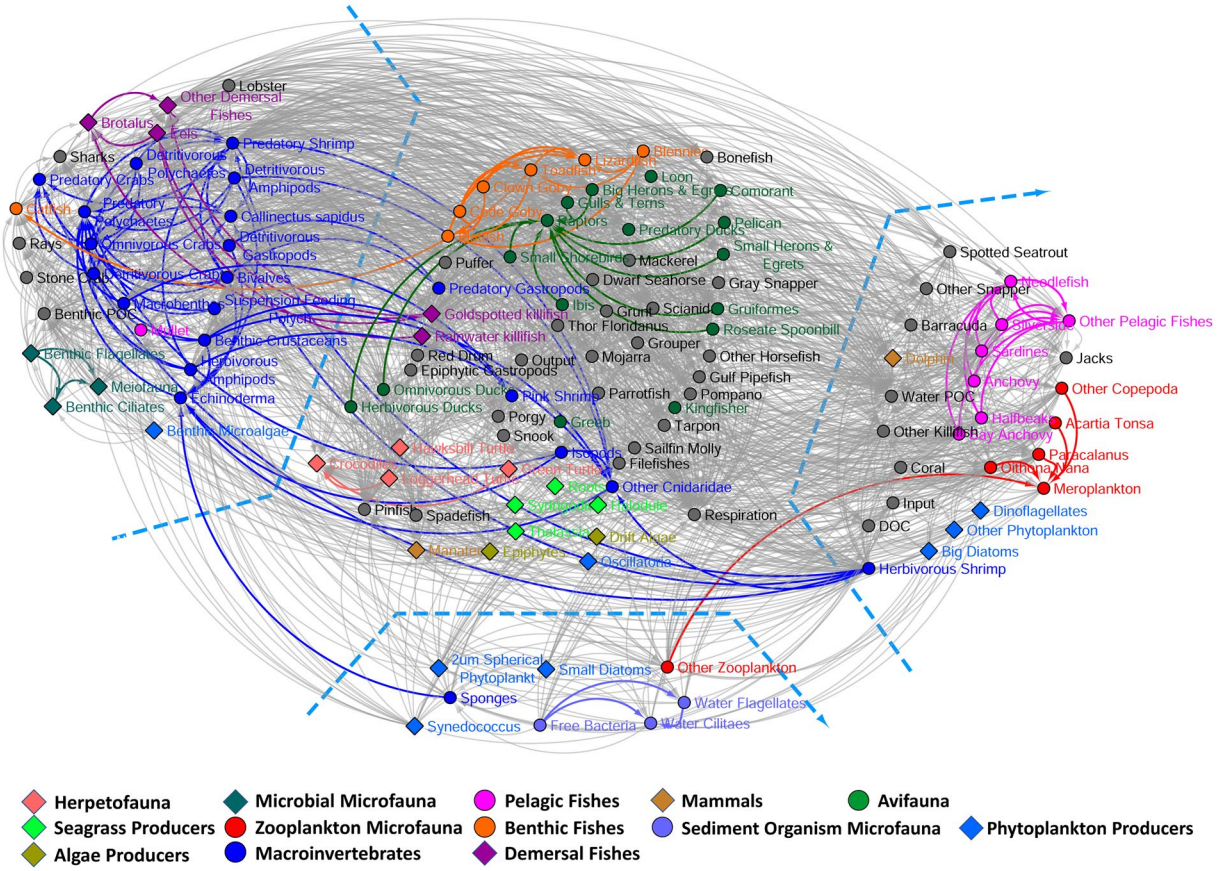

Figure 6. Food web in the Florida Bay. The nodes are colored based on the group classification given in the original research report[60]. The spatial clustering represents communities detected by the proposed method.

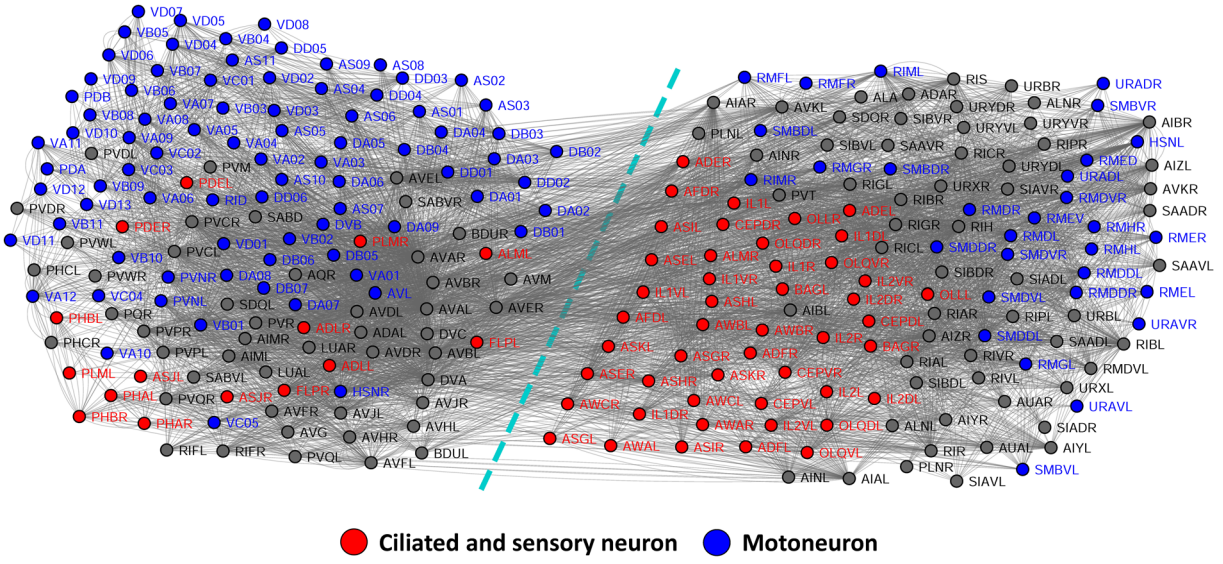

Figure 7. Neural network of the nematode *Caenorhabditis elegans*. The nodes are colored based on neuron categories described in the original research report[61]. The spatial clustering represents communities detected by the proposed method.

zooplankton microfauna (right), and algae producers, avifauna, benthic fishes, herpetofauna, and seagrass producers (middle). Interestingly, some groups are evenly distributed in multiple communities, like mammals, demersal fishes, and phytoplankton producers, while some other groups have a few members clustered into different communities, like benthic fishes, macroinvertebrates, and pelagic fishes. This phenomenon presumably indicates

that the roles of these species in the carbon exchange cannot be derived from the traditional divisions in a trivial manner. For example, though both are mammals, the manatee and the dolphin have very diverse diets. The manatee feeds on submergent aquatic vegetation, and the dolphin feeds on small fishes and shrimps. Consequently, one would expect that the manatee and the dolphin play different roles in the carbon exchange. Thus the simple traditional divisions of taxa, for example, into benthic, demersal, and pelagic organisms, or into fishes, aves, herptiles, and mammals, may not ideally reflect their roles in the carbon exchange.

Neural Network. Our third example is the nervous system of the soil nematode *Caenorhabditis elegans*[61], the only organism whose connectome has been completely mapped so far. The nervous system of *C. elegans* is represented by a neural network consisting of 280 nonpharyngeal neurons and covering 6393 chemical synapses, 890 electrical junctions, and 1410 neuromuscular junctions[62,63], as shown in Fig. 7. In this network, nodes represent neurons and links represent the existence of any of the three neural interactions. The original network is directed and contains multi-edges and loops, and we apply the proposed method to the simplified undirected version, with each directed edge converted to an undirected edge, multi-edges merged, and loops deleted. We have labeled part of the neurons as ciliated/sensory neuron or motoneuron based on descriptions in the original research[61], and these labeled neurons are colored in Fig. 7. The remaining neurons are left in grey. In general, ciliated/sensory neurons are neurons that are part of sensilla (groups of sense organs) or directly associated with sensilla, and motoneurons are neurons that innervate muscles. The neurons left in grey are mostly interneurons that create neural circuits among other neurons.

The dashed line denotes the community division found by the proposed method. As can be seen, the achieved division yields an approximate distinction between ciliated/sensory neurons and motoneurons. This distinction is not perfect: A small number of ciliated/sensory neurons find their way into the motoneuron community (left), and several motoneurons are clustered into the ciliated/sensory-neuron community (right). This "incorrect" clustering of motoneurons is understandable. The families of motoneurons clustered into the ciliated/sensory-neuron community (RIM, RMD, RME, RMF, RMG, RMH, SMB, SMD, URA) are motoneurons that innervate head muscles and are located near the head, where the major sensilla are also located. Thus one would expect these motoneurons to frequently interact with ciliated/sensory neurons that are also located in the head. On the other hand, part of the families of ciliated/sensory neurons clustered into the motoneuron community (PHB, PHA, PDE, PLM) are ciliated/sensory neurons that are connected to sensilla located at the posterior body, where motoneurons are densely located to control body movements. As a result, one would expect these ciliated/sensory neurons to be more associated with local motoneurons than with ciliated/sensory neurons in the head. However, the other four families of incorrectly clustered ciliated/sensory neurons (ADL, ASJ, ALM, FLP) cannot be explained by this theory, because they are located near the head, and in addition some of them are connected to major sensilla in the head. This anomaly might arise because our simplification of the neural network (ignoring interaction directions, merging multi-edges, deleting loops, and regarding all kinds of neural interactions as equivalent) could only approximately represent neural associations, and some information is lost after the simplification.

Conclusion and Discussion

In this paper, we developed a novel community-detection method on the basis of cliques, i.e., local complete subnetworks. The proposed method overcomes the deficiencies of previous similar community-detection methods by considering the nested nature of cliques and encoding the size of cliques into the optimization objective function. In addition, it does not require any pre-specification of the type or size of the subnetworks considered in partitioning. To verify the effectiveness of the proposed method, numerical experiments were conducted using both well-established benchmarks and real-world networks with known communities. In all cases, the community structure detected by the proposed method either achieves state-of-the-art performance or aligns well with ground-truth communities. Finally, we applied the clique-based community-detection method to real-world networks with no a priori information regarding community structure. Specifically, the detected community structure provides insights into the social groupings of bottlenose dolphins, the roles of compartments in ecological carbon exchange, and the functions of neurons in the connectome of the model organism *Caenorhabditis elegans*. We also presented a theoretical analysis of the performance of the proposed method. Specifically, we showed that our method was guaranteed to yield near-optimal performance in the bipartition case, and analyzed the computational complexity of our method.

The proposed method emphasizes the power of maximal cliques in community detection. In networks with community structure, nodes within each community tend to be densely interconnected and may potentially form multiple cliques with large sizes, whereas nodes from different communities are sparsely connected and so are unlikely to form high-order cliques. It would in general be unfair to assume that the sizes of these cliques are above some certain threshold, though most existing methods involving cliques have made such assumptions. Maximal cliques allow algorithms to operate without such assumptions by adaptively encoding all clique information based on whatever clique sizes are available. Though the computational complexity of the proposed method makes it unsuitable for large-scale networks, considering maximal cliques could be useful in devising more computationally efficient methods. For example, some greedy methods may converge faster without losing much accuracy by treating local maximal cliques as a whole. By requiring only information of local maximal cliques, it is possible to bypass the collection of global maximal-clique information, which is computationally expensive.

Theoretical Analysis

In this section, we present the theoretical analysis of the proposed method. We begin by analyzing the performance of the proposed method for a special case. We then discuss the computational complexity of the proposed method, and conclude this section by proving the key theoretical results in this paper.

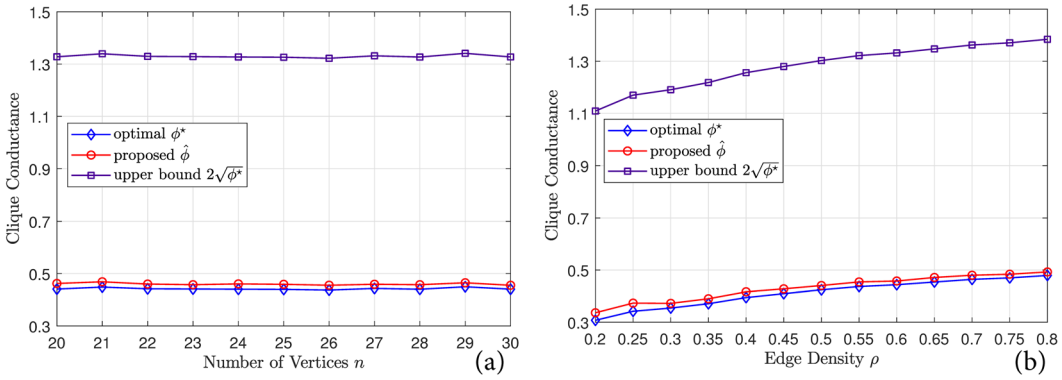

Figure 8. Comparisons of clique conductance of the proposed method and the performance bounds in Theorem 4. (**a**) Varying n and fixed $\rho = 0.6$. (**b**) Varying ρ and fixed $n = 30$.

Performance Guarantee for Graph Bipartition. For the case $m = 2$, the graph-partitioning problem becomes a graph-bipartition problem. For this special case, spectral graph theory provides guidance on measuring the goodness of approximation to the clique conductance minimization[64–66]. One way is through an expanded version of the Cheeger inequality that characterizes the performance of spectral graph partitioning[67]. We follow a similar approach in the remainder of this subsection. Next we introduce terminology necessary to present our result. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \pi)$ be a connected undirected binary graph with no loops. For a subset $A \subset \mathcal{V}$, the Cheeger ratio of A is defined as

$$h(A) := \frac{\text{cut}(A, \bar{A})}{\min(\text{vol}(A), \text{vol}(\bar{A}))}, \tag{16}$$

and the Cheeger constant of $\mathcal{G}$ is defined as

$$h_{\mathcal{G}} := \min_{A} h(A). \tag{17}$$

Let $\alpha_{\mathcal{G}}$ be the Cheeger ratio of the output of Algorithm 1. Chung proved an expanded version of the Cheeger inequality, relating these values for spectral bipartition on connected binary graphs[67]. However, in our setting, $\mathcal{G}_c$ is defined to be a weighted graph. Thus our first step is to generalize Chung's result to connected weighted graphs.

Lemma 3. (Expanded Cheeger inequality). *Let $\mathcal{G}$ be a connected undirected binary graph and $\mathcal{G}_c$ be the induced clique graph with a normalized Laplacian matrix $\mathcal{L}_c$. Let $\lambda_{\mathcal{G}}$ be the second smallest eigenvalue of $\mathcal{L}_c$, and $h_{\mathcal{G}}$ be the Cheeger constant of $\mathcal{G}$. Then*

$$2h_{\mathcal{G}} \geq \lambda_{\mathcal{G}} \geq \frac{\alpha_{\mathcal{G}}^2}{2} \geq \frac{h_{\mathcal{G}}^2}{2}, \tag{18}$$

where $\alpha_{\mathcal{G}}$ is the Cheeger ratio of the output of Algorithm 1.

The proof of Lemma 3 is given later in this section. In our setting, the Cheeger constant $h_{\mathcal{G}}$ is equal to $\phi^\star$, which is the optimal value of the clique conductance optimization (12), and the Cheeger ratio $\alpha_{\mathcal{G}}$ is equal to $\hat{\phi}$, which is the clique conductance of the output of Algorithm 1. Therefore, combining Proposition 1 and Lemma 3 yields Theorem 4.

Theorem 4. *Let $\mathcal{G}$ be a connected undirected binary graph. Let $\phi^\star$ denote the optimal clique-conductance value of* (12) *and $\hat{\phi}$ be the clique-conductance value of output of Algorithm 1 for the case $m = 2$. Then*

$$\phi^\star \leq \hat{\phi} \leq 2\sqrt{\phi^\star}. \tag{19}$$

Theorem 4 shows that our optimization algorithm finds a bipartition that is bounded within the optimal bipartition by a quadratic factor. Therefore our algorithm is mathematically guaranteed to achieve a near-optimal partition.

Performance Guarantee Verification. To verify the performance guarantee of the proposed method, given in Theorem 4, we apply it to a set of randomly generated graphs. Each graph is composed of n vertices, each of which is assigned a random point in $[0,1]^{100}$. An undirected weighted graph is generated by computing the negative Euclidean distances between each pair of these vertices, and then an undirected binary graph is generated by preserving a percentage ρ of the edges with the largest weights. This process produces graphs that reflect the degradation of correlation with distance, which is a common assumption in many network models, and that are essentially random in other aspects.

	Initialization	Clustering
Proposed	$O(3^{n/3})$	$O(n^3)$
Motif[41]	$O(2^n)$	$O(n^3)$
Spectral[34]	—	$O(n^3)$
Louvain[48]	—	$O(nlogn)$
Ravasz[49]	—	$O(n^2)$
Fast Modularity[50–52]	—	$O(n(logn)^2)$

Table 1. Computational complexity of the community-detection methods.

We apply the proposed method to each graph and partition it into two parts. We also enumerate all possible bipartitions and find the bipartition with the minimal clique conductance. In Fig. 8a, we show comparisons of clique conductance of the bipartitions achieved by the proposed method and the optimal bipartitions, with n varying from 20 to 30 and $\rho = 0.6$. In Fig. 8b, we repeat the experiments with $n = 30$ and ρ varying from 0.2 to 0.8. Each curve is averaged over 50 independent trials. As can be seen, the proposed method follows the optimal performance curve closely in general, and is well bounded by the upper bound in Theorem 4. In other words, the proposed method performs almost perfectly and always finds a near-optimal bipartition.

Computational Complexity. Finding all maximal cliques in an arbitrary graph requires $O(3^{n/3})$ computations[45], which is optimal as a function of n because any n-vertex graph has up to $3^{n/3}$ maximal cliques[68]. After forming the clique weight matrix, computing the first m eigenvectors requires an eigenvalue decomposition of the clique weight matrix, for which the computational complexity is $O(n^3)$[69]. The k-means clustering algorithm needs $O(nm^2i)$ computations[70], where i is the number of iterations needed to achieve convergence. Since m is much less than n and i is very small in practice, we conclude that the number of required computations in the clustering scales as $O(n^3)$.

In Table 1, we summarize the computational complexity of the proposed method, the motif-conductance method, and other community-detection methods discussed in the Empirical Results section. As can be seen, the greedy methods are much faster than the proposed method, but the proposed method exhibits better performance on benchmarks (see Fig. 1). The motif-conductance method suffers from the high computational complexity of the brute-force search for the optimal clique size before clustering. By focusing on maximal cliques, the proposed method decreases the computational complexity of this step from $O(2^n)$ to $O(3^{n/3})$. However, the exponential complexity of the proposed method still makes it unsuitable for large networks.

Proof of Proposition 1

Proof. Let $\boldsymbol{z} \in \{0, 1\}^n$ be a vector such that $\boldsymbol{z}(i) = 1$ if $v_i \in A$ and $\boldsymbol{z}(i) = 0$ if $v_i \in \overline{A}$. Further let $\mathbf{W}_{c,k}$ be an adjacency matrix defined as

$$\mathbf{W}_{c,k}(i, j) := \sum_{\sigma \in \mathcal{M}_k v_i, v_j \in \sigma} \sum \omega(\sigma),$$

let $\mathbf{D}_{c,k}$ be the corresponding degree matrix, and let $\mathbf{L}_{c,k}$ be the corresponding Laplacian matrix. Then

$$\begin{aligned}
\mathrm{cut}(A, \bar{A}) &= \sum_{k\geq 1} \sum_{\sigma \in \mathcal{M}_k} \omega(\sigma) \sum_{v_i, v_j \in \sigma} \mathbb{1}(z(i) = 1, z(j) = 0) \\
&= \frac{1}{2} \sum_{k\geq 1} \sum_{\sigma \in \mathcal{M}_k} \omega(\sigma) \sum_{\{v_i, v_j\} \subset \sigma} \mathbb{1}(z(i) \neq z(j)) \\
&= \frac{1}{2} \sum_{k\geq 1} \sum_{\sigma \in \mathcal{M}_k} \omega(\sigma) \sum_{\{v_i, v_j\} \subset \sigma} (z(i) - z(j))^2 \\
&= \sum_{k\geq 1} z^{\mathrm{T}} \mathbf{L}_{c,k} z \\
&= z^{\mathrm{T}} \mathbf{L}_c z \\
&= \mathrm{cut}_c(A, \overline{A}),
\end{aligned}$$

where the fourth and sixth equalities make use of the standard properties of Laplacian matrices[30], and the fifth equality follows $\mathbf{L}_c = \sum_k \mathbf{L}_{c,k}$. In addition,

$$\begin{aligned}
\mathrm{vol}(A) &= \sum_{k\geq 1} \sum_{\sigma \in \mathcal{M}_k} \omega(\sigma) \sum_{v_i \in \sigma} z(i) \\
&= \sum_{k\geq 1} z^{\mathrm{T}} \mathbf{D}_{c,k} z \\
&= z^{\mathrm{T}} \mathbf{D}_{c,k} z \\
&= \mathrm{vol}_c(A),
\end{aligned}$$

where the third equality follows from $\mathbf{D}_c = \sum_k \mathbf{D}_{c,k}$. This concludes the proof. □

Proof of Lemma 3

Proof. This proof extends Chung's proof to connected weighted graphs[67]. The second smallest eigenvalue $\lambda_{\mathcal{G}}$ of $\mathcal{L}_c$ can be expressed as the infimum of the Rayleigh quotient

$$\lambda_{\mathcal{G}} = \inf_{\boldsymbol{y}} R(\boldsymbol{y}) = \inf_{\boldsymbol{y}} \frac{\sum_{u\sim v}(\boldsymbol{y}(u) - \boldsymbol{y}(v))^2 \pi_c(u, v)}{\sum_{v\in\mathcal{V}} \boldsymbol{y}(v)^2 d_v}, \tag{20}$$

where $u \sim v$ means $\{u, v\}$ is a connected pair of vertices, and $\boldsymbol{y}$ satisfies $\sum_{v\in\mathcal{V}} \boldsymbol{y}(v) d_v = 0$. Suppose the Cheeger constant, $h_{\mathcal{G}}$, is achieved by a set S. Let χ_S be the vectorized indicator function of S, defined as

$$\chi_S(u) = \begin{cases} 1 & \text{if } u \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Consider $\boldsymbol{y} = \chi_S - \mathrm{vol}(S)/\mathrm{vol}(\mathcal{V})\mathbf{1}$, and it follows that

$$\lambda_{\mathcal{G}} \le R(\boldsymbol{y}) \le 2h_{\mathcal{G}}. \tag{21}$$

Thus the remainder of this proof focuses on deriving a lower bound for $\lambda_{\mathcal{G}}$ in terms of Cheeger ratios.

Let $\boldsymbol{g}$ be an eigenvector achieving $\lambda_{\mathcal{G}}$, namely,

$$\boldsymbol{g} = \arg\min_{\boldsymbol{y}^{\mathrm{T}}\mathbf{D}_c\mathbf{1}=0} \frac{\boldsymbol{y}^{\mathrm{T}}(\mathbf{D}_c - \mathbf{W}_c)\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}\mathbf{D}_c\boldsymbol{y}}. \tag{22}$$

Reorder the vertices such that

$$\boldsymbol{g}(v_1) \ge \boldsymbol{g}(v_2) \ge \cdots \ge \boldsymbol{g}(v_n),$$

and set $S_i = \{v_1, \ldots, v_i\}$. It follows that

$$\alpha_{\mathcal{G}} = \min_i h(S_i). \tag{23}$$

Let r denote the largest integer such that $\mathrm{vol}(S_r) \le \mathrm{vol}(\mathcal{V})/2$. Since $\boldsymbol{g}^{\mathrm{T}}\mathbf{D}_c\mathbf{1} = 0$,

$$\sum_{i=1}^{n} \boldsymbol{g}(i)^2 d = \min_c \sum_{i=1}^{n} (\boldsymbol{g}(i) - c)^2 d_i \le \sum_{i=1}^{n} (\boldsymbol{g}(i) - \boldsymbol{g}(s_r))^2 d_i,$$

where $d_i := \mathbf{D}_c(i, i)$ for any i. Denote by $\boldsymbol{g}_+$ and $\boldsymbol{g}_-$ the positive and negative parts of $\boldsymbol{g} - \boldsymbol{g}(s_r)$, respectively, defined as

$$\begin{aligned} \boldsymbol{g}_+(i) &= \begin{cases} \boldsymbol{g}(i) - \boldsymbol{g}(s_r) & \text{if } \boldsymbol{g}(i) \ge \boldsymbol{g}(s_r), \\ 0 & \text{otherwise,} \end{cases} \\ \boldsymbol{g}_-(i) &= \begin{cases} \boldsymbol{g}(s_r) - \boldsymbol{g}(i) & \text{if } \boldsymbol{g}(i) \le \boldsymbol{g}(s_r), \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

By the Rayleigh-Ritz theorem[71],

$$\begin{aligned} \lambda_{\mathcal{G}} &= R(\boldsymbol{g}) \\ &= \frac{\sum_{u\sim v}(\boldsymbol{g}(u) - \boldsymbol{g}(v))^2 \pi_c(u, v)}{\sum_{v\in\mathcal{V}} \boldsymbol{g}(v)^2 d_v} \\ &\ge \frac{\sum_{u\sim v}(\boldsymbol{g}(u) - \boldsymbol{g}(v))^2 \pi_c(u, v)}{\sum_{v\in\mathcal{V}} (\boldsymbol{g}(v) - \boldsymbol{g}(v_r))^2 d_v} \\ &\ge \frac{\sum_{u\sim v}((\boldsymbol{g}_+(u) - \boldsymbol{g}_+(v))^2 + (\boldsymbol{g}_-(u) - \boldsymbol{g}_-(v))^2)\pi_c(u, v)}{\sum_{v\in\mathcal{V}} (\boldsymbol{g}_+(v)^2 + \boldsymbol{g}_-(v)^2) d_v}. \end{aligned}$$

Without loss of generality, we may assume $R(\boldsymbol{g}_+) \le R(\boldsymbol{g}_-)$, and then we have $\lambda_{\mathcal{G}} \ge R(\boldsymbol{g}_+)$ because

$$\frac{a+b}{c+d} \ge \min\left(\frac{a}{c}, \frac{b}{d}\right)$$

if $a, b, c, d > 0$. For ease of presentation, we use the notation $\mathrm{vol}^\dagger(S) := \min(\mathrm{vol}(S), \mathrm{vol}(\bar{S}))$. Then we have

$$\begin{aligned}
\lambda_{\mathcal{G}} &\ge R(\boldsymbol{g}_+) \\
&= \frac{\sum_{u\sim v}(\boldsymbol{g}_+(u) - \boldsymbol{g}_+(v))^2 \pi_e(u, v)}{\sum_{v\in\mathcal{V}} \boldsymbol{g}_+(v)^2 d_v} \\
&= \frac{(\sum_{u\sim v}(\boldsymbol{g}_+(u) - \boldsymbol{g}_+(v))^2 \pi_e(u, v))\,(\sum_{u\sim v}(\boldsymbol{g}_+(u) + \boldsymbol{g}_+(v))^2 \pi_e(u, v))}{\sum_{v\in\mathcal{V}} \boldsymbol{g}_+(v)^2 d_v \sum_{u\sim v}(\boldsymbol{g}_+(u) + \boldsymbol{g}_+(v))^2 \pi_e(u, v)} \\
&\ge \frac{(\sum_{u\sim v}(\boldsymbol{g}_+(u)^2 - \boldsymbol{g}_+(v)^2)\pi_e(u, v))^2}{2(\sum_{v\in\mathcal{V}} \boldsymbol{g}_+(v)^2 d_v)^2} \\
&= \frac{(\sum_{1\le i\le n-1} |\boldsymbol{g}_+(v_i)^2 - \boldsymbol{g}_+(v_{i+1})^2| \mathrm{cut}(S_i, \bar{S}_i))^2}{2(\sum_{v\in\mathcal{V}} \boldsymbol{g}_+(v)^2 d_v)^2} \\
&\ge \frac{(\sum_{1\le i\le n-1} |\boldsymbol{g}_+(v_i)^2 - \boldsymbol{g}_+(v_{i+1})^2| a_{\mathcal{G}} \mathrm{vol}^\dagger(\boldsymbol{S}_i))^2}{2(\sum_{v\in\mathcal{V}} \boldsymbol{g}_+(v)^2 d_v)^2} \\
&= \frac{\alpha_{\mathcal{G}}^2}{2} \frac{(\sum_{1\le i\le n} \boldsymbol{g}_+(v_i)^2 |\mathrm{vol}^\dagger(S_i) - \mathrm{vol}^\dagger(S_{i-1})|)^2}{(\sum_{v\in\mathcal{V}} \boldsymbol{g}_+(v)^2 d_v)^2} \\
&= \frac{\alpha_{\mathcal{G}}^2}{2} \frac{(\sum_{1\le i\le n} \boldsymbol{g}_+(v_i)^2 d_{v_i})^2}{(\sum_{v\in\mathcal{V}} \boldsymbol{g}_+(v)^2 d_v)^2} \\
&= \frac{\alpha_{\mathcal{G}}^2}{2}
\end{aligned}$$

where the second inequality is by the Cauchy-Schwarz inequality and the arithmetic-geometric-mean inequality, and the third inequality is by definition of $\alpha_{\mathcal{G}}$. This concludes the proof. □

References

1. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. Complex networks: Structure and dynamics. *Physics Reports* **424**, 175–308 (2006).
2. Caldarelli, G. *Scale-free networks: Complex webs in nature and technology* (Oxford University Press 2007).
3. Newman, M. E. The structure and function of complex networks. *SIAM Review* **45**, 167–256 (2003).
4. Newman, M. The physics of networks. *Physics Today* **61**, 33–38 (2008).
5. Strogatz, S. H. Exploring complex networks. *Nature* **410**, 268–276 (2001).
6. Wasserman, S. & Faust, K. *Social Network Analysis: Methods and Applications* (Cambridge University Press 1994).
7. Wahlström, J., Skog, I., Rosa, P. S. L., Händel, P. & Nehorai, A. The β-model-maximum likelihood, Cramér-Rao bounds, and hypothesis testing. *IEEE Transactions on Signal Processing* **65**, 3234–3246 (2017).
8. Yang, P., Tang, G. & Nehorai, A. Optimal time-of-use electricity pricing using game theory. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3081–3084 (Kyoto, Japan 2012).
9. Yang, P., Tang, G. & Nehorai, A. A game-theoretic approach for optimal time-of-use electricity pricing. *IEEE Transactions on Power Systems* **28**, 884–892 (2013).
10. Chavali, P. & Nehorai, A. Distributed power system state estimation using factor graphs. *IEEE Transactions on Signal Processing* **63**, 2864–2876 (2015).
11. Porter, M. A., Onnela, J.-P. & Mucha, P. J. Communities in networks. *Notices of the AMS* **56**, 1082–1097 (2009).
12. Fortunato, S. Community detection in graphs. *Physics Reports* **486**, 75–174 (2010).
13. Coleman, J. S. *et al. Introduction to mathematical sociology*. (Collier-Macmillan, London, UK, 1964).
14. Borgatti, S. P., Mehra, A., Brass, D. J. & Labianca, G. Network analysis in the social sciences. *Science* **323**, 892–895 (2009).
15. Moody, J. & White, D. R. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review* 103–127 (2003).
16. Rives, A. W. & Galitski, T. Modular organization of cellular networks. *Proceedings of the National Academy of Sciences* **100**, 1128–1133 (2003).
17. Spirin, V. & Mirny, L. A. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences* **100**, 12123–12128 (2003).
18. Chen, J. & Yuan, B. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics* **22**, 2283–2290 (2006).
19. Flake, G. W., Lawrence, S., Giles, C. L. & Coetzee, F. M. Self-organization and identification of web communities. *Computer* **35**, 66–70 (2002).
20. Dourisboure, Y., Geraci, F. & Pellegrini, M. Extraction and classification of dense communities in the web. In *Proceedings of 16th International Conference on World Wide Web*, 461–470 (Banff, Alberta, Canada 2007).
21. Granovetter, M. S. The strength of weak ties. *American Journal of Sociology* **78**, 1360–1380 (1973).
22. Burt, R. S. Positions in networks. *Social Forces* **55**, 93–122 (1976).
23. Freeman, L. C. A set of measures of centrality based on betweenness. *Sociometry* **40**, 35–41 (1977).

24. Simon, H. A. The architecture of complexity. In *Facets of Systems Science*, 457–476 (Springer 1991).
25. Krishnamurthy, B. & Wang, J. On network-aware clustering of web clients. In *Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 97–110 (Stockholm, Sweden 2000).
26. Reddy, P. K., Kitsuregawa, M., Sreekanth, P. & Rao, S. S. A graph based approach to extract a neighborhood customer community for collaborative filtering. In *International Workshop on Databases in Networked Information Syst.*, 188–200 (Springer, Aizu, Japan 2002).
27. Redner, S. How popular is your paper? An empirical study of the citation distribution. *The European Physical Journal of B-Condensed Matter and Complex Systems* **4**, 131–134 (1998).
28. Sizemore, A., Giusti, C., Betzel, R. F. & Bassett, D. S. Closures and cavities in the human connectome. *arXiv preprint arXiv:1608.03520* (2016).
29. Chung, F. R. *Spectral Graph Theory*. 92 (American Mathematical Society 1997).
30. Von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing* **17**, 395–416 (2007).
31. Wu, Z. & Leahy, R. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**, 1101–1113 (1993).
32. Stoer, M. & Wagner, F. A simple min-cut algorithm. *Journal of the ACM* **44**, 585–591 (1997).
33. Hagen, L. & Kahng, A. B. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design Integrated Circuits Systems* **11**, 1074–1085 (1992).
34. Shi, J. & Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 888–905 (2000).
35. Wagner, D. & Wagner, F. Between min cut and graph bisection. *Mathematical Foundations of Computer Science* 744–750 (1993).
36. Palla, G., Derényi, I., Farkas, I. & Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814–818 (2005).
37. Derényi, I., Palla, G. & Vicsek, T. Clique percolation in random networks. *Physical Review Letters* **94**, 160202 (2005).
38. Hatcher, A. *Algebraic Topology* (Cambridge University Press, 2002).
39. Milo, R. *et al*. Network motifs: Simple building blocks of complex networks. *Science* **298**, 824–827 (2002).
40. Yaveroğlu, Ö. N. *et al*. Revealing the hidden language of complex networks. *Scientific Reports* **4** (2014).
41. Benson, A. R., Gleich, D. F. & Leskovec, J. Higher-order organization of complex networks. *Science* **353**, 163–166 (2016).
42. Ng, A. Y., Jordan, M. I. & Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Proceedings of 14th International Conference on Neural Information Processing Systems*, 849–856 (Vancouver, British Columbia, Canada 2001).
43. Bron, C. & Kerbosch, J. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM* **16**, 575–577 (1973).
44. Koch, I. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science* **250**, 1–30 (2001).
45. Tomita, E., Tanaka, A. & Takahashi, H. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science* **363**, 28–42 (2006).
46. Cazals, F. & Karande, C. A note on the problem of reporting maximal cliques. *Theoretical Computer Science* **407**, 564–568 (2008).
47. Lee, J. R., Gharan, S. O. & Trevisan, L. Multiway spectral partitioning and higher-order Cheeger inequalities. *Journal of ACM* **61**, 37:1–37:30 (2014).
48. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008**, P10008 (2008).
49. Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N. & Barabási, A.-L. Hierarchical organization of modularity in metabolic networks. *Science* **297**, 1551–1555 (2002).
50. Newman, M. E. Fast algorithm for detecting community structure in networks. *Physical review E* **69**, 066133 (2004).
51. Clauset, A., Newman, M. E. & Moore, C. Finding community structure in very large networks. *Physical review E* **70**, 066111 (2004).
52. Good, B. H., de Montjoye, Y.-A. & Clauset, A. Performance of modularity maximization in practical contexts. *Physical Review E* **81**, 046106 (2010).
53. Danon, L., Diaz-Guilera, A., Duch, J. & Arenas, A. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* **2005**, P09008 (2005).
54. Barabási, A.-L. *Network Science* (Cambridge university press, 2016).
55. Girvan, M. & Newman, M. E. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* **99**, 7821–7826 (2002).
56. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).
57. Lancichinetti, A., Fortunato, S. & Radicchi, F. Benchmark graphs for testing community detection algorithms. *Physical Review E* **78**, 046110 (2008).
58. Zachary, W. W. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **33**, 452–473 (1977).
59. Lusseau, D. *et al*. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology* **54**, 396–405 (2003).
60. Ulanowicz, R. E. & DeAngelis, D. L. Network analysis of trophic dynamics in South Florida ecosystems-the Florida Bay ecosystem: Annual report to the U.S. geological survey. *U.S. Geological Survey Program on the South Florida Ecosystem* 114–115 (1999).
61. White, J., Southgate, E., Thomson, J. & Brenner, S. The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* **314**, 1–340 (1986).
62. Chen, B. L., Hall, D. H. & Chklovskii, D. B. Wiring optimization can relate neuronal structure and function. *Proceedings of the National Academy of Sciences* **103**, 4723–4728 (2006).
63. Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H. & Chklovskii, D. B. Structural properties of the *Caenorhabditis elegans* neuronal network. *PLOS Computational Biology* **7**, 1–21 (2011).
64. Cheeger, J. A lower bound for the smallest eigenvalue of the laplacian. In *Proceedings of Princeton Conference in honor of Professor S. Bochner*, 195–199 (Princeton University Press 1970).
65. Donath, W. E. & Hoffman, A. J. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development* **17**, 420–425 (1973).
66. Fiedler, M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal* **25**, 619–633 (1975).
67. Chung, F. Four Cheeger-type inequalities for graph partitioning algorithms. *Proceedings of ICCM, II* 751–772 (2007).
68. Moon, J. W. & Moser, L. On cliques in graphs. *Israel journal of Mathematics* **3**, 23–28 (1965).
69. Jacobi, C. G. Über ein leichtes verfahren, die in der theorie der säkularstörangen vorkommenden gleichungen numerisch aufzuloösen, crelle's journal 30 (1846) 51. *Crelle's Journal* **30**, 51–94 (1846).
70. Lloyd, S. Least squares quantization in pcm. *IEEE Transactions on Information Theory* **28**, 129–137 (1982).
71. Trefethen, L. N. & Bau, D. III *Numerical Linear Algebra* (SIAM, 1997).

Acknowledgements

We thank Mark Newman for compiling and sharing the Zachary's karate club and the college football network. We thank Mark Newman and David Lusseau for compiling and sharing the bottlenose dolphin network. We thank Jure Leskovec and Robert E. Ulanowicz for compiling and sharing the Florida bay food network. We thank

the contributors to WORMATLAS for compiling and sharing the connectome of *Caenorhabditis elegans*. We thank Vincent D. Blondel for sharing the MATLAB implementation of the Louvain method.

Author Contributions

Z.L., J.W. and A.N. designed research; Z.L. performed research and analyzed data; Z.L. and J.W. discussed the results and wrote the manuscript; all authors reviewed the manuscript; A.N. supervised the project.

Additional Information

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.