Discovering Link Communities in Complex Networks by an Integer Programming Model and a Genetic Algorithm

Zhenping Li[1], Xiang-Sun Zhang[2], Rui-Sheng Wang[3], Hongwei Liu[1], Shihua Zhang[2]*

[1] School of Information, Beijing Wuzi University, Beijing, China, [2] National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Science, Beijing, China, [3] Department of Physics, The Pennsylvania State University, University Park, Pennsylvania, United States of America

Abstract

Identification of communities in complex networks is an important topic and issue in many fields such as sociology, biology, and computer science. Communities are often defined as groups of related nodes or links that correspond to functional subunits in the corresponding complex systems. While most conventional approaches have focused on discovering communities of nodes, some recent studies start partitioning links to find overlapping communities straightforwardly. In this paper, we propose a new quantity function for link community identification in complex networks. Based on this quantity function we formulate the link community partition problem into an integer programming model which allows us to partition a complex network into overlapping communities. We further propose a genetic algorithm for link community detection which can partition a network into overlapping communities without knowing the number of communities. We test our model and algorithm on both artificial networks and real-world networks. The results demonstrate that the model and algorithm are efficient in detecting overlapping community structure in complex networks.

Citation: Li Z, Zhang X-S, Wang R-S, Liu H, Zhang S (2013) Discovering Link Communities in Complex Networks by an Integer Programming Model and a Genetic Algorithm. PLoS ONE 8(12): e83739. doi:10.1371/journal.pone.0083739

Editor: Dongxiao Zhu, Wayne State University, United States of America

Received September 22, 2013; **Accepted** November 6, 2013; **Published** December 30, 2013

Funding: This work is supported by National Natural Science Foundation of China under grant number 11131009. It is also partially supported by the Funding Project for Academic Human Resources Development in Institutions of Higher Learning under the Jurisdiction of Beijing Municipality (No.PHR201006217, CIT&TCD20130327). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: zsh@amss.ac.cn

Introduction

In the past, it has been shown that many interesting systems can be represented as networks composed of nodes and links, such as the Internet, social and friendship networks, food webs, and citation networks [1–3]. An important topic of current interest in the area of networks has been the idea of communities and their detection. Detecting communities from a network is a universal problem in many disciplines from sociology, computer science to biology [4–6].

Typically there are two kinds of communities which are node communities and link communities respectively. A node community is a dense subgraph induced by a set of nodes, where nodes are densely connected within the subgraph, but sparsely connected with nodes outside of the subgraph. Most existing methods for community detection find a partition of network nodes, i.e. node communities. In this type of partition, each node is in one and only one community. A link community is a dense subgraph induced by a set of links where there are many links within the subgraph, but few links connecting the subgraph with the rest of the network. Detecting link communities in a partitioning way means to find a partition of network links. In this type of partition, each link is in one and only one community, but a node can belong to multiple communities, depending on the community membership of the links incident on it.

Community detection has many important applications in different fields. For example, in biology community detection has been applied to find protein functional modules [7] and predict protein functions [8]. In sociology, community structure is an important topological feature in considering vaccination interventions of infectious diseases in contact networks [9] and understanding viral propagation in social networks [10].

While most previous studies for community detection have focused on node communities, some recent works have started exploring link communities and cliques [11–15]. In some real-world networks, link communities could be more intuitive and informative than node communities, because a link is more likely to have a unique identity while a node often belong to multiple groups [16–21]. For example, most individuals in the society have multiple identities such as families, friends, and co-workers, whereas the link between two individuals usually exists for a dominant reason [11]. From the practical point of view, we can naturally detect the overlapping node communities by partitioning the links into communities [13,16,22–25], because the links connected to a node could belong to different link communities and consequently the node could be assigned to multiple communities of links.

In a recent study [11], the authors define the link density of a link community and the partition density to evaluate the quality of a link community partition. Given a network with M links and N nodes, $P=\{P_1,\cdots,P_C\}$ is a partition of the links into C subsets. The number of links in subset P_s is $m_s=|P_s|$. The number of induced nodes is $n_s=|\bigcup_{e_{ij}\in P_s}\{v_i,v_j\}|$. The link density D_s of community P_s is defined by

$$D_s = \frac{m_s - (n_s - 1)}{n_s(n_s-1)/2 - (n_s - 1)}.$$

The partition density D is defined as the average of D_s, i.e.,

$$D = \frac{2}{M}\sum_s m_s \frac{m_s - (n_s - 1)}{(n_s - 2)(n_s - 1)}.$$

We can see that the maximum value of D is 1 but it can take values less than 0. $D = 1$ when each community is a clique and $D = 0$ when each community is a tree. When a network is a tree, it cannot be partitioned into proper communities by maximizing D, because there are many different optimal partitions, and each partition has the same partition density $D = 0$. For example, the network in Figure 1 consists of two communities with one overlapping node, and each community is a star graph. If we want to partition the network into two communities by maximizing D, it is difficult to find the correct result shown in Figure 1A, because the partitions in Figure 1B and Figure 1C also have $D = 0$.

In most studies on link community partition, each link belongs to one and only one community. But in real-world networks, a link may represent more than one relation between two nodes. For example, two individuals from the same family are also co-workers in the same institute. Consequently two communities may have overlapping links as well. There are few results about how to partition a network into link communities with overlapping links. In this paper, we redefine the partition density of link communities, and formulate the link community partition problem into integer programming models. Then we design a genetic algorithm for solving the link community detection problem and conduct validations on some artificial and real-world networks.

Methods

Link Community Partition Density

Given a network with M links and N nodes, $P = \{P_1, \cdots, P_C\}$ is a partition of the links into C subsets. The number of links in community P_s is $m_s = |P_s|$. The number of induced nodes from community P_s is $n_s = |\bigcup_{e_{ij} \in P_s}\{v_i, v_j\}|$. The new link density H_s of community P_s is defined as follows:

$$H_s = \frac{m_s}{n_s(n_s - 1)/2}.$$

The new partition density H is the average of H_s:

$$H = \frac{1}{M}\sum_s m_s \cdot H_s = \frac{2}{M}\sum_s \frac{m_s^2}{n_s(n_s - 1)}.$$

We can see that the maximum value of H is 1 and the minimum value of H is 0. $H = 1$ when each community is a clique and $H = 0$ when each community is an empty graph. Given the number of communities, we can find the optimal link community partition by maximizing the value of H. For the network in Figure 1, the partition in Figure 1A has the maximum value of H, so we can easily find the optimal partition by maximizing H.

Integer Programming Model for Link Community Partition

Given a network $G = (V, E)$ with M links and N nodes, we assume that the number of link communities is K and find the optimal link community partition by maximizing the partition density H. This problem can be formulated into an integer programming model.

Let $V = \{v_1, v_2, \cdots, v_N\}$ be the node set of G, and $E = \{e_1, e_2, \cdots, e_M\}$ be the edge set of G. We define $R = (r_{ij})_{N \times M}$ to be the incidence matrix of network G, where $r_{ij} = 1$ if link e_j is incident to node v_i, and $r_{ij} = 0$ otherwise. We also define binary variables x_{js} and y_{is} to represent the membership of link e_j and node v_i for link community P_s:

$$x_{js} = \begin{cases} 1, & \text{if } e_j \in P_s, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{is} = \begin{cases} 1, & \text{if } v_i \in P_s, \\ 0, & \text{otherwise.} \end{cases}$$

The link community partition problem can be formulated into the following integer programming model–Model-1.

$$\max H = \frac{2}{M}\sum\nolimits_{s=1}^{K} \frac{(\sum_{j=1}^{M} x_{js})^2}{(\sum_{i=1}^{N} y_{is})^2 - \sum_{i=1}^{N} y_{is}} \qquad (1)$$

$$s.t. \begin{cases} \sum_{s=1}^{K} x_{js} = 1 \; j = 1,2,\cdots,M & (2) \\ \sum_{j=1}^{M} r_{ij}x_{js} \le M\, y_{is} \; i = 1,2,\cdots,N; s = 1,2,\cdots K & (3) \\ y_{is} \le \sum_{j=1}^{M} r_{ij}x_{js} \; i = 1,2,\cdots,N; s = 1,2,\cdots K & (4) \\ x_{js} \in \{0,1\}; j = 1,2,\cdots,M; s = 1,2,\cdots,K & (5) \\ y_{is} \in \{0,1\}; i = 1,2,\cdots,N; s = 1,2,\cdots,K & (6) \end{cases}$$

The objective function (1) is to maximize the new link partition density H. Constraint (2) means that every link belongs to one community. Constraint (3) indicates that if there is one or more links in community P_s that are incident to node v_i, then node v_i must belong to community P_s. Constraint (4) denotes that if node v_i belongs to community P_s, then there is at least one link incident to node v_i that belongs to community P_s.

Since the constraint formulae are simple, we can solve the integer programming model by Lingo software for small networks to see if the model can find overlapping communities properly. Using the quantity function and the integer programming model, we are able to partition several networks into link communities, and obtain correct results. For example, for the network in Figure 2A, we can partition it into five overlapping communities {1, 2, 3, 4, 5}, {7, 8, 9, 10, 11}, {12, 13, 14, 15}, {16, 17, 18}, {1, 7, 12, 16}, and each community is a clique. Nodes 1, 7, 12, 16 are overlapping nodes. The partition density of this link community partition is the optimal objective function value 1. We can partition the network in Figure 2B into two communities with each being a clique. Node 1 and node 2 belong to the two communities and link (1, 2) belongs to the bigger community. The objective function value is less than 1 due to the unique community membership of link (1, 2).

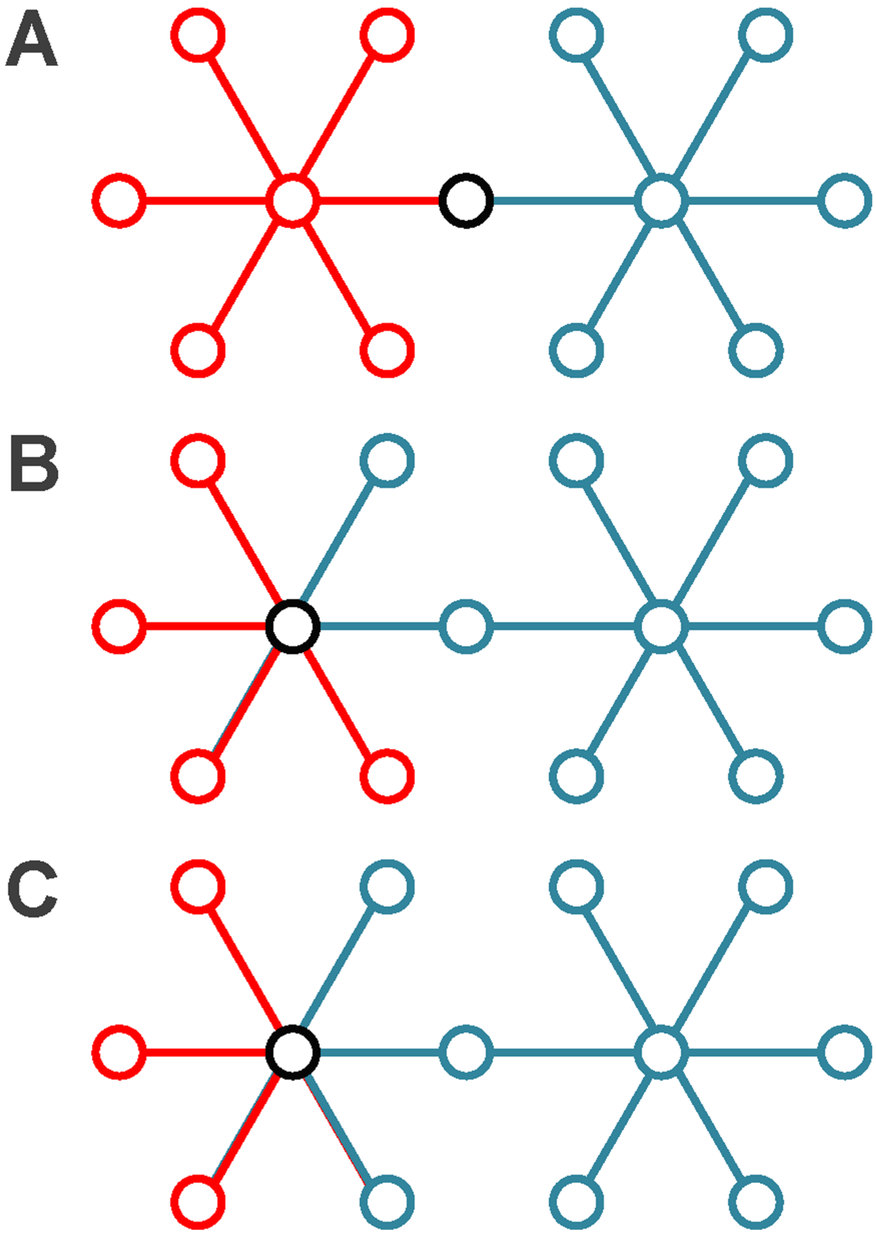

Figure 1. Three different partition results of a tree network. (A) Correct partition. (B,C) Two counter-intuitive partitions. The red links and their adjacent nodes constitute a community, the blue links and their adjacent nodes form another community. The black node is overlapped.
doi:10.1371/journal.pone.0083739.g001

In Model-1, since every link can belong to one and only one community, we might obtain the result that a pair of nodes belongs to the same two communities, but the link between them belong to only one of the communities. For example, in Figure 2B, link (1, 2) only belongs to the bigger community. In fact, node 1 and node 2 may have two different relations. For example, they can be classmates and sisters at the same time. So the link (1, 2) should belong to both classmate community and family community. To address this drawback, we can revise Model-1 and obtain the following model–Model-2.

$$\max H = \frac{2}{\sum_{s=1}^{K} \sum_{j=1}^{M} x_{js}} \sum_{s=1}^{K} \frac{(\sum_{j=1}^{M} x_{js})^2}{(\sum_{i=1}^{N} y_{is})^2 - \sum_{i=1}^{N} y_{is}} \quad (7)$$

$$s.t. \begin{cases} \sum_{s=1}^{K} x_{js} \geq 1 \; j=1,2,\cdots,M & (8) \\ \sum_{j=1}^{M} r_{ij} x_{js} \leq M \, y_{is} \; i=1,2,\cdots,N; s=1,2,\cdots K & (9) \\ y_{is} \leq \sum_{j=1}^{M} r_{ij} x_{js} \; i=1,2,\cdots,N; s=1,2,\cdots K & (10) \\ x_{js} \in \{0,1\} j=1,2,\cdots,M; s=1,2,\cdots,K & (11) \\ y_{is} \in \{0,1\}; i=1,2,\cdots,N; s=1,2,\cdots,K & (12) \end{cases}$$

In Model-2, the constraint (8) means that every link must belong to at least one community. The link belonging to more than one community is regarded as several links in the objective function (7). Using Model-2, we can partition the network in Figure 2B into the two communities, and link (1, 2) belongs to the two communities as

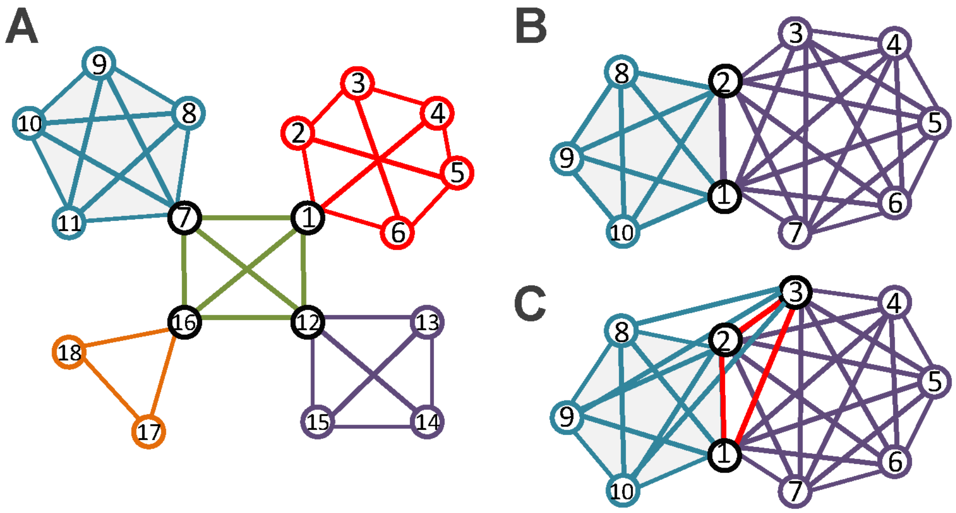

Figure 2. Link communities of three artifical networks. (A) The network consists of five overlapping communities. Nodes 1, 7, 12, 16 are overlapping nodes; (B) The network consists of two overlapping communities. Nodes 1 and 2 are overlapping nodes that belong to the two communities, and link (1, 2) belongs to the two communities as well; (C) The network consists of two overlapping cliques and the overlapped subgraph is a 3-clique.
doi:10.1371/journal.pone.0083739.g002

well. Each community is a clique, and the optimal objective function value that the partition corresponds is 1. Figure 2C is a network consisting of two cliques, which are overlapped with a 3-clique. This network can be partitioned into two communities, and each community is a clique. Two overlapping cliques are correctly identified as each link in the overlapping part (3-clique) belongs to the two communities at the same time. The optimal objective function value of the link partition is 1. Figure 3 is an example from reference [11]. In this network, the basketball team community consists of two part members: one part members are from junior community, and the other part members are from senior community. In other words, the basketball team group is completely subsumed in two other groups. Using Model-2, we can partition the network into three overlapping communities and correctly identify the multiple relationships in the basketball team community.

Model-2 can be used to partition sparse networks (e.g., tree-like networks) or even disconnected networks. It is easily to prove that, when a network is disconnected, it can be partitioned into several connected communities. The objective function value is between 0 and 1. Before using Model-2 to partition a network, the number of communities should be given. If the number of communities is unknown, we can use Model-1 to determine it. We can find the maximum partition density for every given number of communities, then compare all the partition densities and find the maximum one. The number of communities with the maximum partition density is the final number of communities.

Genetic Algorithm for Link Community Detection

Although we can solve Model-2 by Lingo software to partition small-scale networks into link communities, we cannot solve the integer programming model for large-scale networks which is an NP-hard problem. In addition, most of the algorithms for community detection need some *priori* knowledge about the community structure like the number of communities which is impossible to know in real-life networks.

In the following, we will design a genetic algorithm for link community detection. Genetic algorithm (GA) was proposed in [26]. It is a global optimization method in artificial intelligence. When the solution space of a problem is too large to allow exhaustive searching for exact optimal solutions, genetic algorithm can fast converge the problem to a relative smaller solution space, and produces approximately optimal solutions. In [27–29], the authors designed genetic algorithms for solving the node community detection problem in unipartite networks or bipartite networks. In this paper, we propose a link community detection algorithm based on the hybrid ideas of genetic algorithm and self-organizing mapping (SOM) algorithm, which aims to find the best link community structure by maximizing the link partition density. The algorithm does not need any *priori* knowledge about the number of communities, which makes the algorithm useful in real-world networks. The algorithm outputs the final link community structure and its corresponding overlapping nodes as the result and does not impose further processing on the output.

The GA main functions. First of all, we need to design a chromosome representation encoding the solution for the link community detection problem. In our implementation, the chromosome is represented by a matrix $B=(b_{j,c})$, where $j=1,2,\cdots,M$, and $c=1,2,\cdots,K$. Each element $b_{j,c}$ is the strength with which a network link e_j belongs to a community P_c. Note that $b_{j,c}$ ranges in the interval [0.0, 1.0]. Each link of the network is subject to the following constraint:

$$\sum_{c=1}^{K} b_{j,c}=1. \tag{13}$$

Equation (13) is to normalize the membership strengths so that the strength sum of a link belonging to all the communities equals 1.

For each chromosome, we design a partition matrix $D=(d_{j,c})$, where $j=1,2,\cdots,M$, and $c=1,2,\cdots,K$. Each element $d_{j,c}$ is either 0 or 1. When $d_{j,c}=1$, the link e_j is assigned to community P_c, otherwise, link e_j is not assigned to community P_c. Matrix D can be calculated from matrix B according to the following equation:

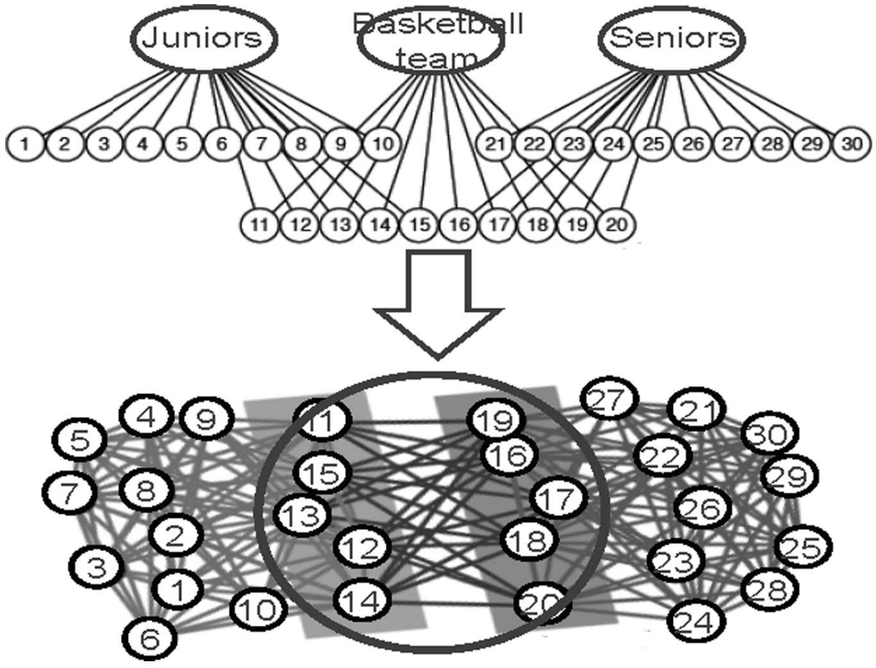

Figure 3. The network in Ref. [11] can be correctly partitioned into three communities by our model, and the objective function value is 1.
doi:10.1371/journal.pone.0083739.g003

$$d_{j,c} = \begin{cases} 1, & \text{if } b_{j,c} = \max_{1 \le s \le K} b_{j,s}, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The network is represented by incidence matrix R, link adjacency matrix A and weighted link adjacency matrix Q. The link adjacency matrix A can be calculated by the following equation: $A = R^T R$. In A, the diagonal elements are 2, and the off-diagonal elements take values in $\{0,1\}$ to represent whether two links have a common node or not. Let Z be a diagonal matrix whose diagonal elements are the inverse of nodes' degree. A node's degree is the number of links incident to it. In other words,

$$Z = \begin{pmatrix} \frac{1}{d(v_1)} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{d(v_2)} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{d(v_3)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{d(v_N)} \end{pmatrix}$$

The weighted link adjacency matrix Q is defined as $Q = R^T Z R$, which means the probability for a random walker going from one link to one of its adjacent links across their common node. This can be regarded as the possibility of two adjacent links belonging to the same community.

The GA Main Functions

- Input

Input the number of nodes N and the number of links M of the network, the maximum number of communities K. Calculate the incidence matrix R, the link adjacency matrix $A = R^T R$, and the weighted link adjacency matrix $Q = R^T Z R$. Give the number of individuals U, the maximum epoch T, mutation probability p, and SOM parameters α, β, θ.

- Output

Output the link partition matrix D^* and its fitness value H^* (i.e. link partition density value), the node partition matrix F. Partition the network into communities according to D^* and F.

- Initialization: $t = 0$

Randomly generate an initial population $B_1(t)$, $B_2(t), \cdots,$ $B_U(t)$, and give an initial values of D^* and H^*.

- Step 1. Population Fitness

For all individuals in the population $B_1(t)$, $B_2(t), \cdots, B_U(t)$, calculate the partition matrices $D_1(t)$, $D_2(t), \cdots, D_U(t)$, and their fitness values $H_1(t)$, $H_2(t), \cdots, H_U(t)$.

- Step 2. Population Sorting

Sort $B_1(t), B_2(t), \cdots, B_U(t)$ according to their fitness values in descending order. Suppose the sorted chromosomes are $B_1(t), B_2(t), \cdots, B_U(t)$, where $H_1(t) \ge H_2(t) \ge \cdots \ge H_U(t)$. If $H_1(t) > H^*$, then $D^* = D_1(t)$, $H^* = H_1(t)$. If $t = T$, then stop, output D^* and H^*, and calculate the corresponding node partition matrix F. Otherwise, go to Step 3.

- Step 3. Population Crossover

For $i=1,\ \ldots,\lfloor\frac{U}{2}\rfloor$, let $B_{\lfloor\frac{U}{2}\rfloor+i}(t)$ and cross over to produce two temporary individuals (matrices) $W_i(t)$ and $B_{\lfloor\frac{U}{2}\rfloor+i}(t)$. If U is an odd number, then let $W_U(t)=B_U(t)$.

- Step 4. Population Mutation

Randomly select pU temporary individuals (temporary matrices), and do mutation operation on each temporary individual.

- Step 5. Population SOM

For each temporary individual, do SOM operation on it.

- Step 6. Population Normalization

For each temporary individual, do normalization on it. Denote the normalized individuals by $B_1(t+1),B_2(t+1),\cdots,B_U(t+1)$. Let $t=t+1$, and go to Step 1.

Partition matrix and fitness evaluation. For each individual B_i, calculate the partition matrix D_i according to the formula (14). For each community P_s, $1\leq s\leq K$, let $D_i(\,:,s)$ be the s-th column of matrix D_i. Then $E_i(s)=R\cdot D_i(\,:,s)$ is a column vector whose elements are non-negative integers. A non-zero element in $E_i(s)$ represents that the corresponding node belongs to community P_s. Let $F_i(s)$ be a 0–1 vector, and $f_i(j,s)=1$ whenever $e_i(j,s)\geq 1$. $f_i(j,s)=1$ means that node v_j belongs to community P_s. The fitness of individual B_i can be calculated by the following equation:

$$H_i=\frac{2}{\sum_{s=1}^{K}\sum_{j=1}^{M}D_i(j,s)}\sum_{s=1}^{K}\frac{(\sum_{j=1}^{M}D_i(j,s))^2}{(\sum_{v=1}^{N}F_i(v,s))^2-(\sum_{v=1}^{N}F_i(v,s))}.$$

Since there is often one maximum value in each row of matrix B, by formula (14), we often partition a link into one and only one community. When a link is an overlapping link of two communities, it cannot be detected by formula (14) directly. To identify the overlapping link correctly, we can replace formula (14) by the following formula (15).

$$d_{j,c}=\begin{cases}1, & \text{if } \dfrac{b_{j,c}}{\max\limits_{1\leq s\leq K} b_{j,s}}\geq 0.5,\\ 0, & \text{otherwise.}\end{cases} \tag{15}$$

Using formula (15), an overlapping link can be partitioned into more than one communities.

Population sorting. Sort $B_1(t),B_2(t),\cdots,B_U(t)$ according to their fitness values in descending order. Suppose the sorted chromosomes are $B_1(t),B_2(t),\cdots,B_U(t)$, where $H_1(t)\geq H_2(t)\geq\cdots\geq H_U(t)$. If $H_1(t)>H^*$, then $D^*=D_1(t)$, $H^*=H_1(t)$.

Population crossover. For $i=1,2,\ \ldots,\lfloor\frac{U}{2}\rfloor$, do crossover operation on $B_i(t)$ and $B_{\lfloor\frac{U}{2}\rfloor+i}(t)$ by the following rules: randomly select a column s, revise the s-th column of $B_{\lfloor\frac{U}{2}\rfloor+i}(t)$ by the s-th column of $B_i(t)$, and obtain two new temporary individuals $W_i(t)$ and $W_{\lfloor\frac{U}{2}\rfloor+i}(t)$. Let $W_i(t)=B_i(t)$. We revise the s-th column of $B_{\lfloor\frac{U}{2}\rfloor+i}(t)$ by adding a fraction of the s-th column of $D_i(t)$ (where $D_i(t)$ is the partition matrix corresponding to $B_i(t)$), that is,

$$W_{\lfloor\frac{U}{2}\rfloor+i}(t)(\,:,c)=\begin{cases}B_{\lfloor\frac{U}{2}\rfloor+i}(:,s)+0.1*D_i(:,s) & \text{if } c=s.\\ B_{\lfloor\frac{U}{2}\rfloor+i}(:,c) & \text{if } c\neq s.\end{cases}$$

Population mutation. According to the mutation probability p, randomly select pU temporary individuals, do mutation operation on each selected individual. For each selected temporary individual $W_i(t)$, randomly select two parameters j,s, $1\leq j,s\leq M$. There are three mutation rules that can be used in this genetic algorithm, i.e. exchange the j-th row and the s-th row in $W_i(t)$, or replace the j-th row by the s-th row in $W_i(t)$, or replace the elements of the j-th row with randomly selected numbers in [0.0,1.0]. Three rules lead to insignificant difference in this genetic algorithm. In the following simulation, we replace the j-th row with the s-th row in $W_i(t)$. The other elements in $W_i(t)$ remain unchanged.

Population SOM. The Self-Organizing Mapping (SOM) process analyzes the link community ID variance of each link. If the community ID variance of a link is larger than a threshold value, then increase the membership strength of this link for community P_s and that of its all neighbor links belonging to the same community. Meanwhile, decrease the membership strengths of all non-neighbor links for community P_s. If the community ID variance of a link is smaller than the threshold value, the membership strength of the link and all neighbor links belonging to the same community decreases. This process can improve the quality of the partition by eliminating wrongly placed links due to the behaviors of the algorithm.

For $i=1,\cdots,N$, do SOM operations on individual (chromosome) W_i as follows:

- Calculate its partition matrix $D_i^{'}$ from the matrix W_i according to the formula (14);
- For $j=1,\cdots,M$, do the following operation on link e_j.
- Find the community ID of link e_j which corresponds to the maximum element in the j-th row of $D_i^{'}$ (the maximum element must be 1). Suppose the maximum element in the j-th row of $D_i^{'}$ is in the s-th column, which is $D_i^{'}(j,s)$. This means that link e_j belongs to community P_s.
- Calculate the total number $TN(e_j)$ of adjacent links of e_j (including edge e_j), and the number of adjacent links in $TN(e_j)$ belonging to community P_s (denoted by $IN(e_j)$). $TN(e_j)$ is equal to the sum of elements in the j-th row of matrix A, which can be expressed by $TN(e_j)=A(j,\,:\,)\cdot I$, where $I=(1,1,\cdots,1)^T$, and $IN(e_j)$ can be obtained by the equation $IN(e_j)=A(j,\,:\,)\cdot D_i^{'}(\,:,s)$.
- Calculate the community ID variance $CV(e_j)$ of link e_j by the following equation.

$$CV(e_j)=\frac{IN(e_j)}{TN(e_j)}.$$

- If $CV(e_j)\geq\theta$, then

$$W_i(\,:,s) = W_i(\,:,s) + Q(\,:,j)\cdot\alpha - (I - A(\,:,j))\cdot\beta,$$

otherwise,

$$W_i(\,:,s) = W_i(\,:,s) - Q(\,:,j)\cdot\beta.$$

where α and β are adjustable parameters that decrease with the step t (In this paper, we let $\alpha = \alpha - \frac{t}{T}(\alpha - 0.1)$, $\beta = \beta - \frac{t}{T}(\beta - 0.05)$). In the above equations, if an element is negative, then we set it to be 0.01.

Normalization. Since the sum of row elements in temporary matrix W_i might not be 1, we should do normalization on each row of matrix W_i. For $i = 1,2,\cdots,U$, do normalization on each row of temporary matrix W_i through dividing it by the sum of row elements.

Complexity of the genetic algorithm. The running time of the genetic algorithm is mainly determined by the running time of Step 1 and Step 5. The complexity of Step 1 is at most $O(MKN)$, and the complexity of Step 5 is at most $O(MKN)$. So the complexity of the genetic algorithm is $O(MKNT)$.

Results

In this section, we apply the genetic algorithm to a class of artificial networks and several real-world networks, and analyze the results in terms of classification accuracy and ability of detecting meaningful communities. The algorithm is implemented by Matlab version 7.1.

We first do validations on the networks described in Figure 2. By setting the parameters as described in Table 1, we can find all the optimal partitions. Then we conduct validation experiments on several types of overlapping networks with special structures and several real-world networks.

Ring Networks Consisting of Cliques

We test our algorithm on a type of exemplar networks, that is, rings of cliques, which is not the same as in [30–32]. This network consists of many heterogeneous cliques, connected through single nodes (Figure 4A). Each clique C_i $(i = 1,2,\cdots,K)$ is a complete graph. The network has a clear link modular structure where each community corresponds to a single clique, thus the optimal partition density is 1. Using our genetic algorithm, we can easily detect the optimal partition and identify the overlapping nodes. Figure 4A demonstrates a network consisting of two 4-cliques and three 5-cliques. Our method can obtain the optimal partition and identify the overlapping nodes correctly.

Table 1. The parameters used in the GA algorithm for solving the link community detection problem on networks in Figure 2.

network	K	N	p	θ	α	β	T
A	5	40	0.3	0.2	1.0	0.2	2000
B	2	40	0.3	0.3	1.0	0.2	600
C	2,3,4,5	40	0.3	0.2	1.4	0.1	600

doi:10.1371/journal.pone.0083739.t001

We also test our algorithm on an overlapping ring network of cliques. The network consists of many heterogeneous cliques, and two adjacent cliques are overlapped by several nodes and links (these overlapping nodes and links form a small clique) (Figure 4B). The overlapping ring of clique network can be partitioned into multiple communities by our genetic algorithm, and each community is a clique. The overlapping small cliques connecting pairs of large cliques can also be correctly identified.

We further validate our algorithm on a tree network of cliques. This network consists of multiple cliques connected by overlapping nodes. Many subnetworks of metabolic networks are similar to a tree of cliques. The network we test consists of five cliques depicted in Figure 4C. Using our genetic algorithm, the network can be partitioned into the five cliques, and the fitness (partition density) of the partition is 1.

Applications on Real-world Networks

In this subsection, we validate our method on three real-world networks.

The karate club network. The first example we consider is the famous karate club network analyzed by Zachary [33]. It has also been analyzed by many community detection studies. It consists of 34 members of a karate club as nodes and 78 edges representing friendship between members of the club which was observed over a period of two years. We apply our method to the karate club network using the parameters $K = 3$, $N = 600$, $p = 0.2$, $\theta = 0.2$, $\alpha = 0.6$, $\beta = 0.2$, $T = 1000$. The result is illustrated in Figure 5A. The average link density is 0.3349. The colors of the links indicate the link communities detected by our genetic algorithm, and the colors of the nodes indicate the node communities deduced from link communities. In this karate club network, our link communities show that node 1 belongs to three communities, and nodes 2 and 3 belong to two communities. The overlapping part is a 3-clique which was not identified by previous methods.

Word association network. The word association network is picked from the South Florida Free Association norm list (http://www.usf.edu/FreeAssociation/). In the South Florida Free Association norm list, the weight of a directed link from one word to another indicates the frequency with which the people in the survey associate the end point of the link with its starting point. The word "play" association network has been replaced with an undirected one and tested in [34–36]. This network has 53 nodes representing different words and 197 association edges. Using the genetic algorithm with parameters $K = 3$, $U = 40$, $p = 0.2$, $\theta = 0.2$, $\alpha = 1.0$, $\beta = 0.2$, $T = 10000$, we can partition this network into three overlapping communities with the fitness (objective function) value 0.3396. The result is described in Figure 5B. From the partition results, we can see that words with frequent associations are in the same communities. In this network, the word "play" is strongly associated with most words, so it is an overlapping node. This result has also been obtained by a graph-theoretical method for node community detection [35].

The co-appearance network. The co-appearance network contains 77 characters in the novel Les Misérables by Victor Hugo. There are 77 nodes and 254 links in the co-appearance network. The nodes represent 77 characters and the links connect any pair of characters that appear in the same chapter of the book. This network was compiled by Knuth [37] based on the list of characters' appearance by scene. In this paper, we use the unweighted network. Figure 5C shows the partition obtained by our genetic algorithm, which divides the network into seven overlapping communities. The resulting partition agrees reasonably well with the social divisions and subplots in the plot-line of

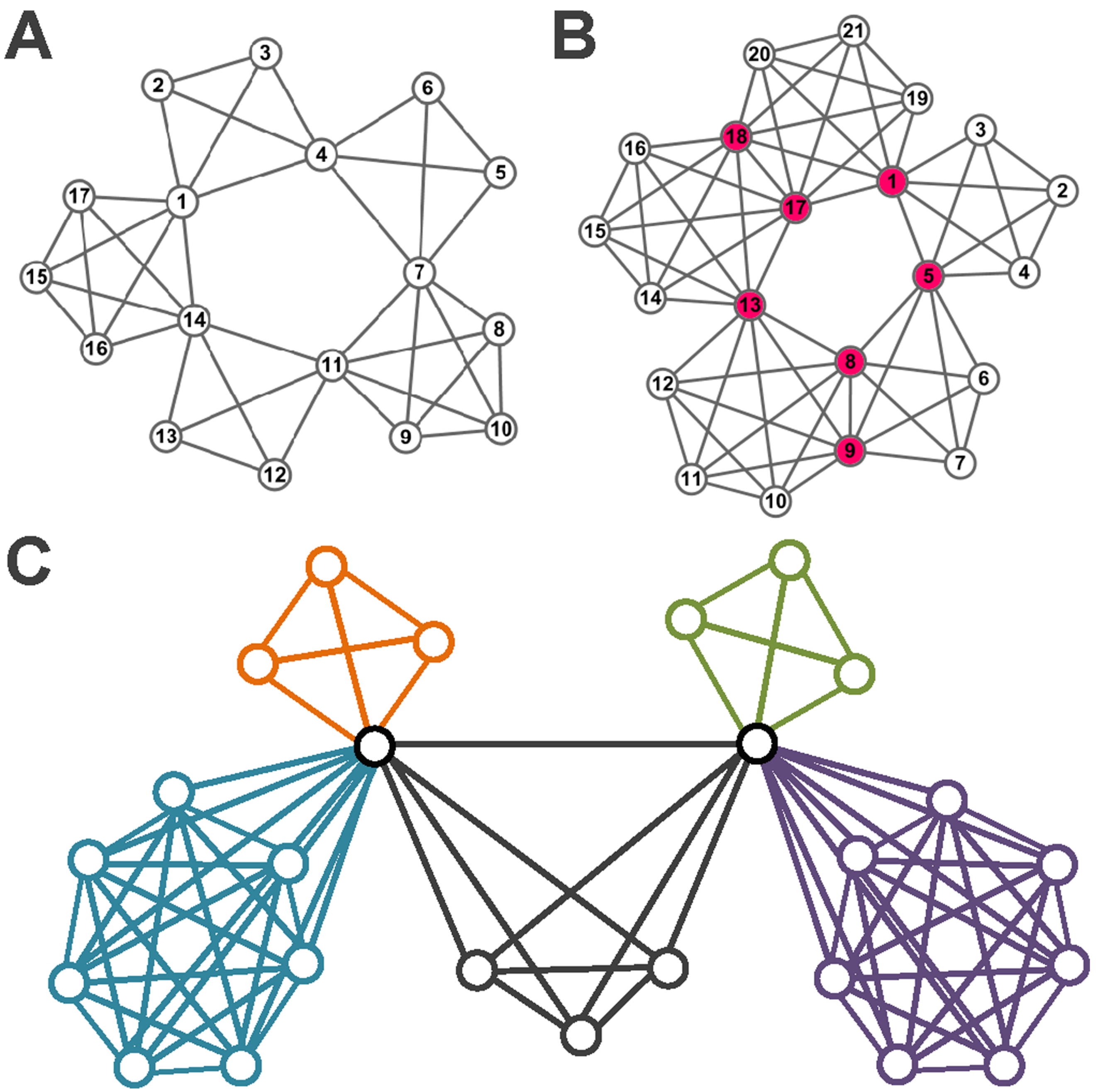

Figure 4. Link communities of three networks of heterogeneous cliques. (A) The ring network of heterogeneous cliques. Each community is a clique, and two adjacent communities are connected by one node. (B) The ring network of overlapping heterogeneous cliques. Each community is a clique, and two adjacent communities are connected by one node or one link. (C) The tree network of heterogeneous cliques. Each community is a clique, and two adjacent communities are overlapped by one node [11].
doi:10.1371/journal.pone.0083739.g004

the novel. In [16], the network is partitioned into five communities.

From the results, we can see that this network contains some highly connected nodes, some of which (nodes 11, 16, 23, 29, 41, 48, 55, 58) are overlapping nodes and can connect to multiple communities of the network. These nodes can cause serious problems if we want to partition the network by conventional node community schemes because they do not fit adequately to any community. No matter which community we place a highly connected node in, its outside links are more than its inside links. In contrast, link community schemes can provide an elegant solution to this problem because they allow a node to belong to multiple communities. As shown in Figure 5C, our algorithm properly places nodes 11, 16, 23, 29, 41, 48, 55, 58 into more than one community. These nodes correspond to the major characters in the novel. In addition, our algorithm also classifies the major characters of the novel into their proper communities. For example, node 48 corresponds to Gavroche, who is assigned to three communities, corresponding to his family members, friends, and the people with battle respectively.

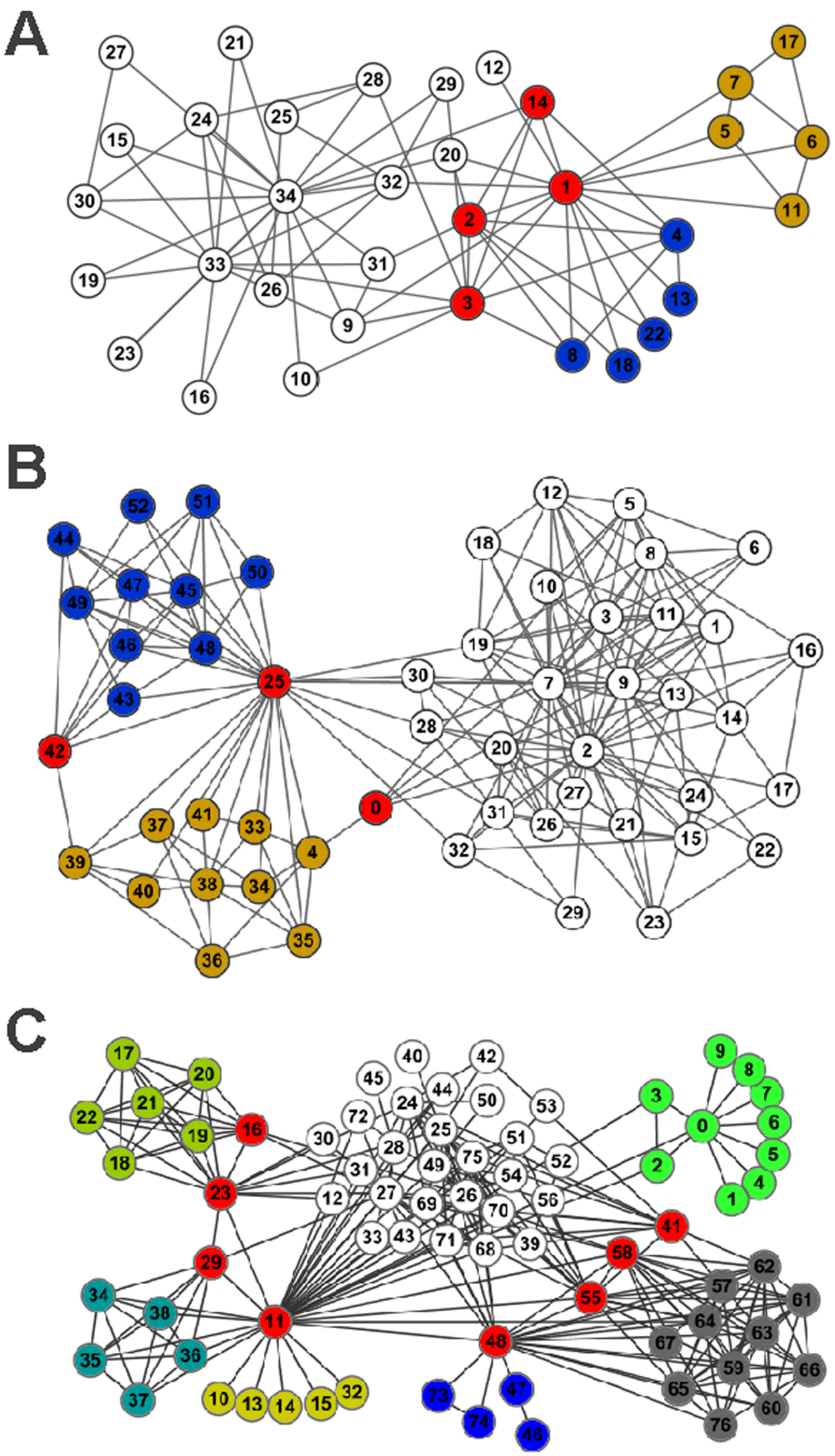

Figure 5. Link communities of some real-world networks. (A) The Karate club network; (B) The word association network; (C) The co-appearance network.

doi:10.1371/journal.pone.0083739.g005

Discussion and Conclusion

Community structure is one of the main characteristics of complex networks and detecting community structure is very helpful for understanding the functions of these networks. In this paper, we investigate the link community detection problem and propose a new quantity function for link community detection. We formulate the link community identification problem into an integer nonlinear programming model based on the proposed quantity function. Furthermore, we design a GA algorithm for solving the link community detection problem and conduct validation experiments on some artificial and real-world networks.

The extensive computational results demonstrate that our model and algorithm can detect overlapping communities effectively. It will be promising to apply and test our method onto real large-scale networks. Generally, note that the real large-scale networks are very sparse. According to the computational complexity analyzed before, it will be feasible to apply it onto sparse networks with about 10000 nodes. This method can be easily extended to detect the communities of both directed networks and bipartite networks, which will be further explored in our future study.

Author Contributions

Conceived and designed the experiments: ZL RSW SZ. Performed the experiments: ZL. Analyzed the data: ZL XSZ RSW HL SZ. Contributed reagents/materials/analysis tools: ZL HL SZ. Wrote the paper: ZL RSW SZ.

References

1. Albert R, Barabási AL (2002) Statistical mechanics of complex networks. Rev Mod Phys 74: 47–97.
2. Newman MEJ (2003) The structure and function of complex networks. SIAM Rev 45: 167–256.
3. Hu Y, Chen H, Zhang P, Li M, Di Z, et al. (2008) Comparative definition of community and corresponding identifying algorithm. Phys Rev E 78: 026121.
4. Fortunato S (2010) Community detection in graph. Physics Reports 486: 75–174.
5. Newman MEJ (2012) Communities, modules and large-scale structure in networks. Nature Physics 8: 25–31.
6. Zhang S, Jin G, Zhang XS, Chen L (2007) Discovering functions and revealing mechanisms at molecular level from biological networks. Proteomics 7: 2856–2869.
7. Spirin V, Mirny LA (2003) Protein complexes and functional modules in molecular networks. Proc. Natl. Acad. Sci. U.S.A. 100: 12123–12128.
8. Lee J, Gross SP, Lee J (2013) Improved network community structure improves function prediction. Scientific Reports 3: 2197.
9. Salath M, Jones JH (2010) Dynamics and control of diseases in networks with community structure. PLoS Comput Biol. 6: e1000736.
10. Weng L, Menczer F, Ahn YY (2013) Virality prediction and community structure in social networks. Scientific Reports 3: 2522.
11. Ahn YY, Bagrow JP, Lehmann S (2010) Link communities reveal multi-scale complexity in networks. Nature 466: 761–764.
12. Evans TS, Lambiotte R (2009) Line graphs, link partitions and overlapping communities. Phys Rev E 80: 016105.
13. Evans TS (2010) Clique graphs and overlapping communities. J Stat Mech: P12037.
14. Evans TS, Lambiotte R (2010) Line graphs of weighted networks for overlapping communities. Eur Phys J B 77: 265–272.
15. Zhang S, Liu HW, Ning XM, Zhang XS (2009) A hybrid graph-theoretic method for mining overlapping functional modules in large sparse protein interaction networks. International Journal of Data Mining and Bioinformatics 3, 68–84.
16. He DX, Liu D, Zhang W, Jin D, Yang B (2012) Discovering link communities in complex networks by exploiting link dynamics. J Stat Mech: P10015.
17. Zhang S, Wang RS, Zhang XS (2007) Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physica A 374: 483–490.
18. Szalay-Bekő M, Palotai R, Szappanos B, Kovács IA, Papp B, et al. (2012) ModuLand plug-in for Cytoscape: extensively overlapping network modules, community centrality and their use in biological networks. Bioin-formatics 28: 2202–2204.
19. Shen HW, Cheng XQ, Guo JF (2009) Quantifying and identifying the overlapping community structure in networks. J Stat Mech: P07042.
20. Zhang S, Wang RS, Zhang XS (2007) Uncovering fuzzy community structure in complex networks. Phys Rev E 76: 046103.
21. Li K, Gong X, Guan S, Lai CH (2012) Efficient algorithm based on neighborhood overlap for community identification in complex networks. Physica A 391: 1788–1796.
22. Nepusz T, Petróczi A, Négyessy L, Bazsó F (2008) Fuzzy communities and the concept of bridgeness in complex networks. Phys Rev E 77: 016107.
23. Gregory S (2011) Fuzzy overlapping communities in networks. J Stat Mech, P02017.
24. Kovacs IA, Palotai R, Szalay MS, Csermely P (2010) Community landscapes: an integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. PLOS ONE 5: e12528.
25. Esquivel AV, Rosvall M (2011) Compression of flow can reveal overlapping-module organization in networks. Phys Rev X 1: 021025.
26. Holland JH (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan.
27. Nicosia V, Mangioni G, Carchiolo V, Malgeri M (2009) Extending the definition of modularity to directed graphs with overlapping communities. J Stat Mech: P03024.
28. Tasgin M, Bingol H (2006) Community detection in complex networks using genetic algorithm. http://arxiv.org/abs/0711.0491v1.
29. Zan W, Zhang Z, Guan J, Zhou S (2011) Evolutionary method for finding communities in bipartite networks. Phys Rev E 83: 066120.
30. Li Z, Zhang S, Wang R, Zhang XS, Chen L (2008) Quantitative function for community detection. Physical Review E 77: 36109.
31. Fortunato S, Barthelemy M (2007) Resolution limit in community detection. Proc. Natl. Acad. Sci. U.S.A. 104: 36–41.
32. Zhang XS, Wang RS, Wang Y, Wang JG, Qiu YQ, et al. (2009) Modularity optimization in community identification of complex networks. EPL 87: 38002.
33. Zachary WW (1977) An informal flow model for conflict and fission in small groups. J Anthropol Res 33: 452–473.
34. Vicsek T (2007) Phase transitions and overlapping modules in complex networks. Physica A 378: 20–32.
35. Wang RS, Zhang S, Zhang XS, Chen L (2007) Identifying modules in complex networks by a graph- theoretical method and its application in protein interaction networks. Lecture Notes in Computer Science 4682: 1090–1101.
36. Pallal G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435: 814–818.
37. Knuth DE (1993) The Stanford GraphBase: a platform for combinatorial computing (Reading, MA: Addison-Wesley).