

Towards Online Multiresolution Community Detection in Large-Scale Networks

Jianbin Huang^{1*}, Heli Sun², Yaguang Liu¹, Qinbao Song², Tim Weneringer³

1 School of Software, Xidian University, Xi'an, China, **2** Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China, **3** Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, United States of America

Abstract

The investigation of community structure in networks has aroused great interest in multiple disciplines. One of the challenges is to find local communities from a starting vertex in a network without global information about the entire network. Many existing methods tend to be accurate depending on a priori assumptions of network properties and predefined parameters. In this paper, we introduce a new quality function of local community and present a fast local expansion algorithm for uncovering communities in large-scale networks. The proposed algorithm can detect multiresolution community from a source vertex or communities covering the whole network. Experimental results show that the proposed algorithm is efficient and well-behaved in both real-world and synthetic networks.

Citation: Huang J, Sun H, Liu Y, Song Q, Weneringer T (2011) Towards Online Multiresolution Community Detection in Large-Scale Networks. PLoS ONE 6(8): e23829. doi:10.1371/journal.pone.0023829

Editor: Yamir Moreno, University of Zaragoza, Spain

Received: March 31, 2011; **Accepted:** July 25, 2011; **Published:** August 24, 2011

Copyright: © 2011 Huang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: The authors have no support or funding to report.

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

* E-mail: jbhuan@xidian.edu.cn

Introduction

Nowadays, there are growing interests in analyzing the complex networks, such as the World Wide Web [1], citation networks [2], online social networks [3], and metabolic networks [4]. A network is always expressed as a graph consisting of vertices, representing entities, and edges, indicating relationships between entities. A common feature of these complex networks is community structure [5–7]. A network community is usually regarded as a group of vertices that are more densely connected within the group than the rest of the network. These communities usually indicate a meaningful topological relationship between entities within the network [5,8]. Community discovery within networks is an important problem with many applications in a number of disciplines ranging from social network analysis to image segmentation and from analyzing protein interaction networks to the circuit layout problem.

A large number of methods have been presented for detecting communities in complex networks [9,10]. For example, many community detection algorithms have been created utilizing greedy optimization of a modularity function Q [11,12]. However, any algorithm using Q must necessarily be a global method, requiring complete knowledge of the entire network. Meanwhile, it has been shown that modularity Q is not a scale-invariant measure, and hence, by relying on its maximization, detection of communities smaller than a certain size is impossible. It is well known as the resolution limit problem [13]. For many real-world networks, such as the Web graph and social networks, which are extremely huge and fast evolving, their global structures are almost impossible to consider. Therefore, existing global approaches have difficulty in handling these large, real-world networks.

Recently, researchers have proposed several methods for identifying communities based on the local link structure of a network [14]. P. Bagrow *et al.* [15] explored the local module of a source vertex through breadth first search. A local community is found until the expansion rate falls below some predefined threshold. This approach works well only if the source vertex is enclosed in the middle of the module. A. Clauset [16] proposed a new measure to detect communities by using a local optimization of a metric called *local modularity*, which only considers vertices within the boundary of a sub-graph. However, this method is sensitive to the join number parameter t , which must be specified by the user. The LWP algorithm proposed in [17] defines a different local modularity measure, which is closely related to the idea of a weak community [18]. X. Xu [19] proposed a density-based network clustering method for community detection. However, like other density-based clustering methods, it is sensitive to a minimum similarity threshold parameter, and it provides no automated way to find the parameter. Another important feature of complex networks is the intrinsic hierarchical community structure. So the revealed communities depend on the scale at which the network is examined [20–23]. A. Lancichinetti *et al.* [20] proposed a hierarchical and overlapping community detection method LFM based on a local fitness measure, which generates multiple communities to show hierarchies of the network by randomizing the starting vertices and varying a resolution parameter α .

To deal with the problems above, we present an approach for online multiresolution local community detection in large-scale networks. We introduce a similarity-based quality function of a community, called *tightness*, and present a algorithm LTE (Local Tightness Expansion), for revealing the natural community from a starting vertex or communities covering a network via local optimization of the tightness measure. Moreover, our method can

identify overlapping communities in multiresolution by adjusting an optional resolution parameter.

The rest of the paper is organized as follows. First, the tightness measure and vertex adoption criterion for local communities are formalized. Then the algorithm LTE is described in detail. The experimental results are reported. Finally, the last section summarizes the conclusions and suggests future work.

Methods

Measurement of local community

Usually, a network can be represented by a graph $G=(V,E)$, where V is the set of vertices and E is the set of edges. Community detection is the task of finding densely connected sub-graphs in G . Our algorithm is based on the assumption that a community containing a given vertex s is a sub-graph that can be locally expanded from s with a predefined criterion. In such local structures, a network is covered by distinct vertex groups. In the processing, we only deal with the vertices in a limited region, including the community C , the neighborhood N , and the unknown part U . The structure is shown in Fig. 1.

Generally, a community is measured by a specific property of the vertices within it. For this task, different community measurements have been proposed [16,20,24]. Here, we introduce a structural similarity-based metric. First, we formalize some notions of the local community.

Definition 1 (Neighborhood) Let $G=(V,E,w)$ be a weighted undirected network and $w(e)$ be the weight of the edge e . For a vertex $u \in V$, the structure neighborhood of vertex u is the set $\Gamma(u)$ containing u and its adjacent vertices which are incident with a common edge with u : $\Gamma(u) = \{v \in V | \{u,v\} \in E\} \cup \{u\}$.

Then we can use the similarity functions based on the common neighborhood to measure the similarity of any pair of adjacent vertices. In this paper, we adopt a structural similarity measure from the cosine similarity function used in [21] which effectively denotes the local connectivity density of any two adjacent vertices in a weighted network.

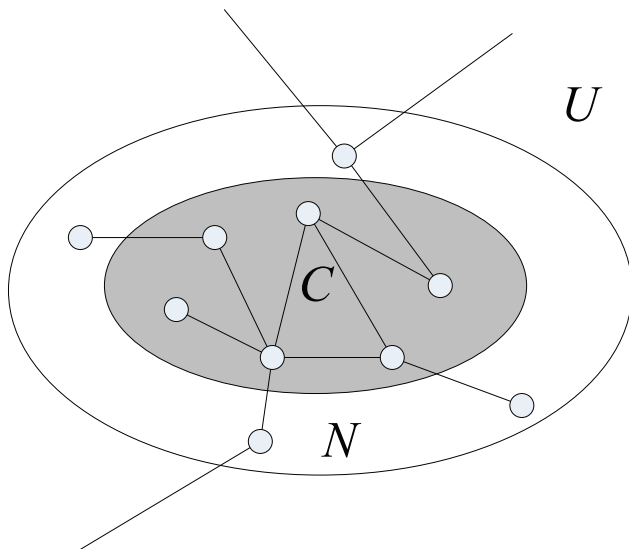


Figure 1. An illustration of the division of an abstract graph into the local community C , its neighborhood N , and the edges that connect N to the unknown vertices in U .
doi:10.1371/journal.pone.0023829.g001

Definition 2 (Structural Similarity) Given a weighted undirected network $G=(V,E,w)$, the structure similarity $s(u,v)$ between two adjacent vertices u and v is:

$$s(u,v) = \frac{\sum_{x \in \Gamma(u) \cap \Gamma(v)} w(u,x) \cdot w(v,x)}{\sqrt{\sum_{x \in \Gamma(u)} w^2(u,x)} \cdot \sqrt{\sum_{x \in \Gamma(v)} w^2(v,x)}}. \quad (1)$$

When we consider an unweighted graph, the weight $w(u,v)$ of any edge $\{u,v\} \in E$ can be set to 1 and the equation above can be transformed to

$$s(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{|\Gamma(u)| \cdot |\Gamma(v)|}}. \quad (2)$$

It corresponds to the so-called edge-clustering coefficient introduced by Radicchi et al. [18].

Definition 3 (Tightness) By employing the structural similarity, we introduce tightness, a new quality function of a local community C , which is given as follows:

$$T(C) = \frac{S_m^C}{S_m^C + S_{out}^C}, \quad (3)$$

where $S_m^C = \sum_{u \in C, v \in C, \{u,v\} \in E} s(u,v)$ is the internal similarity of the community

C which is equal to two times of the sum of similarities between any two adjacent vertices both inside the community C ; $S_{out}^C = \sum_{u \in C, v \in N, \{u,v\} \in E} s(u,v)$

is the external similarity of the community C which is equal to the sum of similarities between vertices inside the community C and vertices out of it.

The tightness measure is extended from the weak community definition proposed by F. Radicchi [18]. Similar to other community definitions [20,25], the tightness value of a community C , denoted by $T(C)$, will increase when sub-graph C has high internal similarity and low external similarity. The whole network without outward edges will achieve the maximal value 1, but the problem here is to find the local optimization of the measurement for each community.

Suppose a community C is detected from a certain vertex s . We explore the adjacent vertices in the neighborhood set N of C , as shown in Fig. 2. So the variant tightness of the community $C \cup \{a\}$ becomes

$$\begin{aligned} T(C \cup \{a\}) &= \frac{S_m^C + 2S_{in}^a}{(S_m^C + 2S_{in}^a) + (S_{out}^C - S_{in}^a + S_{out}^a)} \\ &= \frac{S_m^C + 2S_{in}^a}{S_m^C + S_{in}^a + S_{out}^C + S_{out}^a}, \end{aligned} \quad (4)$$

where $S_{in}^a = \sum_{\{v,a\} \in E \wedge v \in C} s(v,a)$ and $S_{out}^a = \sum_{\{a,u\} \in E \wedge u \notin C} s(a,u)$. Then the tightness increment of a vertex a joining in C is

$$\begin{aligned} \Delta T_C(a) &= T(C \cup \{a\}) - T(C) \\ &= \frac{S_m^C + 2S_{in}^a}{S_m^C + S_{in}^a + S_{out}^C + S_{out}^a} - \frac{S_m^C}{S_m^C + S_{out}^C} \\ &= \frac{2S_{in}^a \cdot S_{out}^C - S_m^C \cdot S_{out}^a + S_{in}^C \cdot S_{in}^a}{(S_m^C + S_{in}^a + S_{out}^C + S_{out}^a)(S_m^C + S_{out}^C)}. \end{aligned} \quad (5)$$

If $\Delta T_C(a) > 0$ holds, then $2S_{in}^a \cdot S_{out}^C - S_m^C \cdot S_{out}^a + S_{in}^C \cdot S_{in}^a > 0$ which is

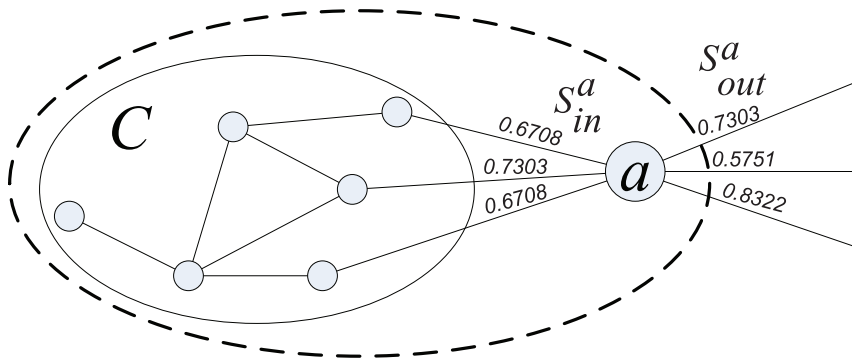


Figure 2. The tightness variant of a community C when a vertex a joins C .
doi:10.1371/journal.pone.0023829.g002

equivalent to $\frac{S_{out}^C}{S_{in}^C} - \frac{S_{out}^a - S_{in}^a}{2S_{in}^a} > 0$. So we define the tightness gain in the following.

Definition 4 (Tightness Gain) The tightness gain for the community C adopting a neighbor vertex a can be denoted as

$$\tau_C(a) = \frac{S_{out}^C}{S_{in}^C} - \frac{S_{out}^a - S_{in}^a}{2S_{in}^a}. \tag{6}$$

Let $\Delta S_{out}^a = S_{out}^a - S_{in}^a$ and $\Delta S_{in}^a = 2S_{in}^a$, the equation above is equivalent to

$$\tau_C(a) = \frac{S_{out}^C}{S_{in}^C} - \frac{\Delta S_{out}^a}{\Delta S_{in}^a}. \tag{7}$$

It means that the ratio of external similarity to internal similarity of community C is greater than the ratio of external similarity increment to internal similarity increment caused by adopting vertex a . Obviously, this case will result in the increase of the tightness value of community C . Therefore, $\tau_C(a)$ can be utilized as a criterion to determine whether the candidate vertex a should be included in the community C or not.

In the following, we introduce an optional resolution parameter α to control the scale at which we want to observe the communities in a network.

Definition 5 (Tunable Tightness Gain) The tunable tightness gain for the community C merging a neighbor vertex a can be denoted as

$$\tau_C^\alpha(a) = \frac{S_{out}^C}{S_{in}^C} - \frac{\alpha S_{out}^a - S_{in}^a}{2S_{in}^a} \tag{8}$$

A parameter $\alpha \in (0, \infty)$ is introduced as the coefficient of S_{out}^a which can increase or decrease the proportion of the external similarity of the candidate vertex a . Here, the criterion for accepting a vertex a is changed to $\tau_C^\alpha(a) > 0$. For $\alpha = 1$, the criteria is moderate and can be used in most normal cases. However, if we set $\alpha \in (0, 1)$, the value of S_{out}^a is reduced by this coefficient which increases the chance of a candidate vertex a joining C and bigger communities will be formed compared to the normal case with $\alpha = 1$. On the contrary, it will result in the formation of smaller communities in a network when we set $\alpha > 1$. Therefore, large values of α yield small communities, small values instead deliver large communities. In most cases, the whole network forms a

single community containing all the vertices with $\alpha < 0.01$. In contrast, several small groups containing two or more vertices are identified when we set $\alpha > 10$.

The Algorithm

Like other local community inferring methods [16,17,20], our local community detection algorithm is able to start from an arbitrary vertex s in a network G . To find the densely connected local community containing vertex s , our algorithm works with two iterative steps: update step and join step. First, the starting vertex s is added in C . In the update step, we refresh the neighborhood set N , and calculate the structural similarities between vertices in the community C and their neighbor vertices in N . In the joining step, we try to absorb a vertex in N having highest structural similarity with vertices in C into the community C . If $\tau_C^\alpha(a) > 0$, then the vertex a will be inserted into C . Otherwise, it will be removed from N and other vertices will be considered in the descending order of the structural similarity. The two procedures above will be repeated in turn until set N is empty. Then, the whole community C is discovered. Details of the local community detection algorithm are given in the following.

Step 1. Pick a vertex $s \in V$ as the starting vertex. Let $C = \{s\}$ and $N = \Gamma(s) - \{s\}$.

Step 2. Select the vertex $a \in N$ that possess the largest similarity with vertices in C .

Step 3. If $\tau_C^\alpha(a) > 0$, set $C = C \cup \{a\}$ and $N = N \cup \Gamma(a) - C$.

Step 4. Repeat Step 2 and Step 3 until $N = \emptyset$.

Unlike methods proposed in [16,17,20], which calculate the quantitative metrics for every vertex in the neighbor sets and select the vertex who produces the greatest increment of the metric to join the community C , our method picks the neighbor vertex with the largest similarity as the candidate vertex and calculate $\tau_C^\alpha(a)$ to determine whether it should be added to the community C or not. The structural similarity reflects the local connectivity density of the graph. The larger the similarity between a vertex outside the community and a vertex inside it, the more common neighbors the two vertices share, and the more probability they are at the same community. Furthermore, unlike the quantitative metrics mentioned above, which need to be recalculated for each neighbors of a new vertex added in the community, the similarity of each pair of vertices and $\tau_C^\alpha(a)$ in our algorithm are both calculated only once by using a dynamical priority queue. So the execution of our algorithm is accelerated and the accuracy remains high.

The algorithm proposed above can be rerun from different vertices not yet belonging to any community, to form the communities covering the whole network. The process of our local tightness expansion algorithm LTE are given as follows.

- Step 1. Pick a vertex s that has not been assigned to any community as the starting vertex.
- Step 2. Detect the local community containing s by using algorithm described above.
- Step 3. Repeat from step 1 until all vertices in V are visited.

Actually, if we allow a new community to be able to absorb vertices possessed by other communities, then it will form overlapping communities. Otherwise, non-overlapping communities are formed. In a word, our algorithm can identify either the local community from a specified vertex or communities covering the whole network. A problem for our non-overlapping community detection method is that the result, to a certain extent, depends on the visiting order of vertices [11,16], which means that different starting vertices sequence may lead to different partition of the network. However, our experimental results show that the effect is slight, especially for networks with clear community structures.

The running time of our community detection algorithm LTE is mainly consumed in selecting the neighbor vertex with the largest similarity in the process of forming the local communities. We perform this function with a dynamic priority queue which is implemented with a Fibonacci heap. So the computational complexity for our non-overlapping local expansion method is almost the same as generating the minimal spanning tree of a graph with the Fibonacci heap, which has a computational complexity of $O(m+n\log n)$, where n is the number of vertices inferred, and m is the number of edges in the network. For scale free network, the running time complexity is $O(m\log n)$.

When the overlapping communities are considered, the running time is longer than the non-overlapping process because some vertices will be visited multiple times. However, the computational complexity is hard to estimate as it depends on the resolution parameter α and intrinsic link structure of the network.

To illustrate the running time of the proposed algorithm LTE and to compare it with other algorithms, we generate seven networks where the numbers of vertices n range from 1,000 to 500,000 and the numbers of edges m are always ten times n . The running times of our algorithm on the synthetic networks compared with other local community detection methods (e.g., Clauset's method, LFM and LWP) are plotted in Fig. 3. It shows that our algorithm LTE with $\alpha=1$ for detecting non-overlapping communities can process a network with 500,000 vertices within two minutes. We can observe that both the overlapping and non-overlapping version of our algorithm LTE run much faster than LFM, LWP and Clauset's method.

Results

In this section, we evaluate our method using some real-world datasets and computer-generated benchmark datasets. Our algorithm is implemented in ANSI C++. All the experiments were conducted on a PC with a 2.4 GHz Pentium IV processor and 2 GB of RAM.

Test On Real-World Networks

To evaluate the accuracy of our algorithm, we first conduct experiments on three popular real-world networks.

Zachary's karate club. Zachary's karate club is a well known social network that describes the friendship relations between 34 members of a karate club observed over two years by Zachary [26], as shown in Fig. 4. In that period, the club members split into two distinct groups due to a contrast between the owner (vertex 1) and the instructor (vertex 33) of the club.

We start from vertices 1 and 33 respectively to detect the overlapping communities, and then detect communities from other vertices which have not been absorbed in communities. When we set $\alpha \in [0.78, 1.36]$, four communities are discovered in this network by our algorithm, as shown in Fig. 4. It is noteworthy that vertex 10 is shared by two communities. It is absorbed by these two communities since it connects the two communities in the same way and the positive tightness gains are harvested when it is merged by both of these two communities.

When we set the parameter $\alpha > 1.36$, the number of detected community also increases along with the increase of the parameter. Finally, we get a stable result in which all communities are made up of two or a little more vertices. This is because the tightness gain of the first absorbed vertex starting from an arbitrary vertex must be positive, no matter how large the parameter α is. If there are neighbors only having links to the above two vertices, they are also enclosed in this community. We find that in any networks, when the value of parameter α is large enough, our algorithm always finds covers of networks as that described above. When we set $\alpha \in [0.50, 0.60]$, our algorithm detects two communities in which vertices 5, 6, 7, 11, 17 form a community and the remaining vertices are enclosed in the other community. When $\alpha < 0.5$, the whole network is identified as a single community.

The algorithm LFM detects the same four communities in this network with parameter $\alpha=1$. The Clauset's method also finds four communities in this network. The only difference is that three vertices 3, 9, 10 are identified as shared overlapping vertices by the owner's community and instructor's community. Though the result of our algorithm on Karate network does not match the ground truth of the dataset, it is similar to results often found by other methods, which means that it is topologically meaningful.

NCAA college-football network. The NCAA college-football network is a representation of the schedule of American Division I college football games for the 2000 season [1]. As shown in Fig. 5, 115 vertices in the network represent teams (identified by their college names), which are divided into eleven communities (or conferences) and five independent teams (Utah State, Navy, Notre Dame, Connecticut and Central Florida). 616 edges represent regular season games between the two teams they connect. The question is to automatically recover the conferences within the network.

First, we use our algorithm LTE to detect non-overlapping communities in this network. The visiting sequence of the vertices is in the ascending order of the vertex number in the dataset. When we set $\alpha \in [0.79, 1.10]$, our algorithm identifies thirteen communities within this network, as shown in Fig. 5. Among them, seven conferences (e.g., Atlantic Coast, Mountain West, Big 10, Big 12, Pac 10, SEC and Conference USA) are correctly identified. Interestingly, the Mid-American conference is divided into two small divisions in real-world: MAC East and MAC West, which are correctly identified as communities respectively by our algorithm. The remaining two communities closely resemble the Sunbelt and Western Athletic conferences. Five independent teams that do not belong to any conference tend to be grouped with the conference with which they are most closely associated. There are also a few failure cases for our algorithm. For example, the team Boise State in the Western Athletic conference is

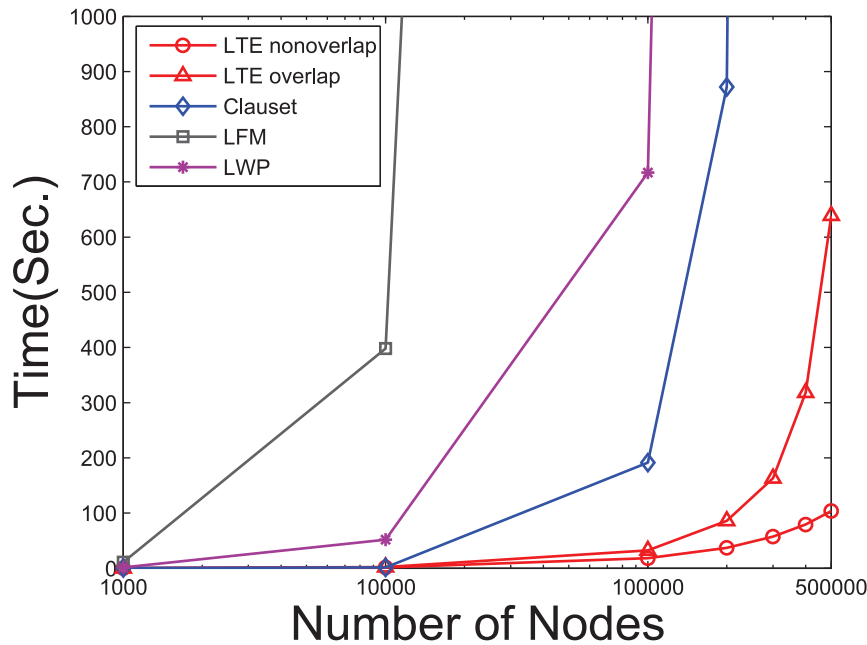


Figure 3. Running time comparison for our algorithm LTE with other state-of-the-art local community detection algorithms.
doi:10.1371/journal.pone.0023829.g003

incorrectly merged into the Sunbelt conference. We found that Boise State played more games against Sunbelt teams than teams in its own conference, which likely explains the inaccuracy. In general, our algorithm is inaccurate in cases like this where the network structure genuinely does not correspond to the real-world organization of conferences. In other respects, our algorithm performs remarkably well.

When we increase the value of α from 1.10 gradually, the number of detected community increases and the size of each community decreases correspondingly. Finally, we get the same stable result described above. When we set $\alpha \in [0.27, 0.29]$ and $\alpha \in [0.19, 0.24]$, our algorithm can detect six and two communities respectively in this network. The whole network forms a single community when $\alpha < 0.19$.

Based on the intrinsic conferences in the football network, we compare the accuracy of our algorithm with other local methods by calculating precision and recall scores for each conference [27]. We take every vertex in the network as the starting vertex, and execute the algorithm LTE with $\alpha=1$ to infer the local community. Finally, the overall precision and recall scores of a conference is the average of the detected communities starting from each vertex in the conference. As shown in Table 1, our algorithm LTE achieve much higher accuracy than the Clauset's method and LWP method in every conference. The result of algorithm LFM is comparable to that of our method. Both LTE algorithm and LFM algorithm correctly detect the conferences Atlantic Coast, Big East, Mountain West, Pac 10, and SEC in this network. In conferences Big 10, USA, Mid-American and Western

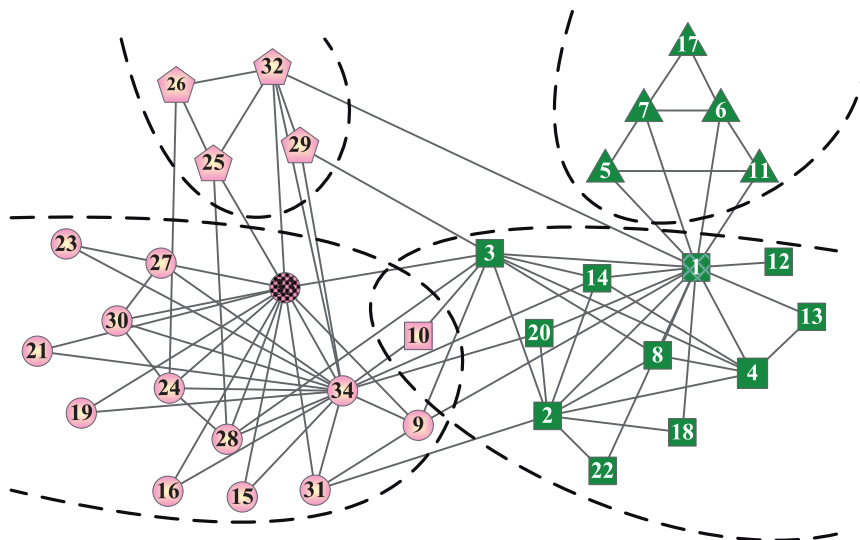


Figure 4. The Zachary's karate network and four communities discovered by our algorithm on it.
doi:10.1371/journal.pone.0023829.g004

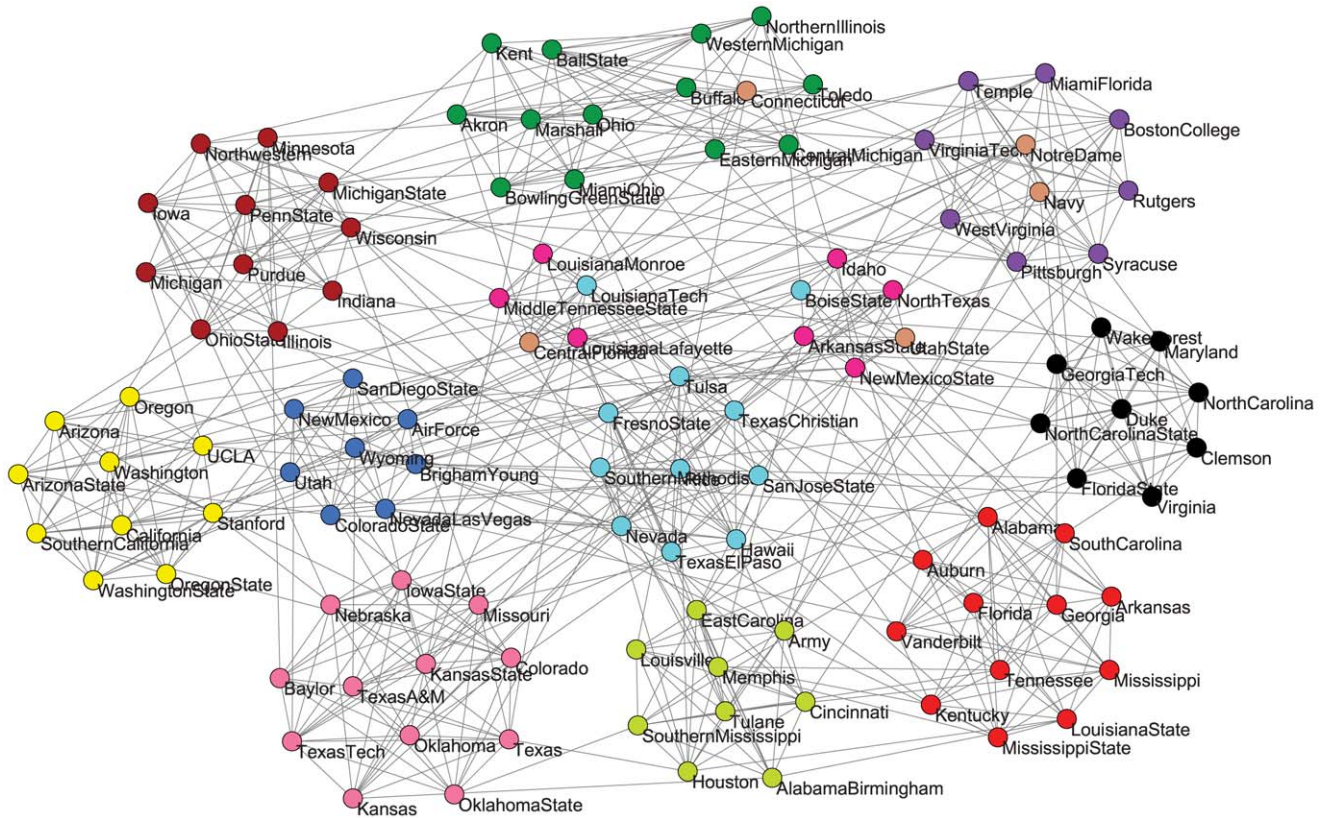


Figure 5. The NCAA college-football network and thirteen communities obtained by our algorithm on it with $\alpha \in [0.79, 1.10]$. The conferences are represented by the vertex groups and the detected communities are distinguished by different colors.
doi:10.1371/journal.pone.0023829.g005

Athletic, the precision and recall values of LFM are a little higher than that of our method. But our algorithm LTE obtains better result on conferences Big 12 and Subelt. The LFM algorithm performs well in this network because it using a dynamic process of vertex selection for detecting a local community. When a new vertex is absorbed in a community, the fitness variant of all the vertices in the community will be recalculated and the vertices

having negative fitness contribution will be get out of the community which will result in more densely connected communities in sparse real-world networks.

Amazon co-purchase network. We also use our method to study a large-scale co-purchase network from Amazon.com. This network was collected in January of 2006, and has been studied in [16,17]. In this network there are 585,283 vertices and 4,566,749

Table 1. Accuracy comparison of local community detection methods for each conference in NCAA college-football network.

| NCAA college-football network | | Clusset | | LWP | | LFM | | LTE | |
|-------------------------------|------|-----------|--------|-----------|--------|---------------|---------------|---------------|---------------|
| conference name | size | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| Atlantic Coast | 9 | 0.7037 | 0.7054 | 1 | 1 | 1 | 1 | 1 | 1 |
| Big 10 | 12 | 0.3889 | 0.39 | 0.8472 | 0.9226 | 0.9167 | 1 | 0.8819 | 0.9881 |
| Big 12 | 12 | 0.9236 | 0.8251 | 1 | 1 | 1 | 0.8571 | 1 | 1 |
| Big East | 8 | 0.5625 | 0.4895 | 0.1719 | 0.1033 | 1 | 1 | 1 | 1 |
| Conference USA | 10 | 0.67 | 0.6686 | 0.1444 | 0.119 | 0.9 | 1 | 0.89 | 0.9889 |
| Mid-American | 12 | 0.9236 | 0.753 | 1 | 0.8571 | 1 | 0.8571 | 0.5 | 0.8571 |
| Mountain West | 8 | 0.8906 | 0.8026 | 0.1563 | 0.1169 | 1 | 1 | 1 | 1 |
| Pac 10 | 10 | 0.82 | 0.7322 | 1 | 1 | 1 | 1 | 1 | 1 |
| SEC | 12 | 0.9236 | 0.7874 | 1 | 1 | 1 | 1 | 1 | 1 |
| Sunbelt | 7 | 0.5306 | 0.5365 | 0.1429 | 0.0897 | 0.4286 | 0.5 | 0.5102 | 0.6381 |
| Western Athletic | 10 | 0.62 | 0.5874 | 0.6091 | 0.6717 | 0.8 | 0.8889 | 0.74 | 0.8378 |

doi:10.1371/journal.pone.0023829.t001

Table 2. Source vertices and sizes for illustrative examples of detected communities using our algorithm LTE in Amazon co-purchase network.

| source vertex | vertex number | type | degree | community size |
|--|---------------|------|--------|----------------|
| Cirque Reinvente by Cirque du Soleil | B00000F3V8 | DVD | 15 | 18 |
| Alegria by Cirque du Soleil | B000003FRF | CD | 11 | 25 |
| Love Wants to Dance by Maria Muldaur | B0002M5TB6 | CD | 14 | 20 |
| Toddlers Sing by Music for Little People | B00000C41T | CD | 10 | 6 |
| Preschool Songs by Cedarmont Kids | B000008UPJ | CD | 17 | 23 |
| Molecular Biology of Cell by Bruce Alberts | 0815332181 | book | 222 | 47 |
| An Introduction to Bioinformatics Algorithms by Neil C. Jones et al. | 0262101068 | book | 89 | 43 |
| Molecular Biology of the Gene, Fifth Edition by James D. Watson | 080534635X | book | 24 | 9 |
| Small Worlds by Duncan Watts | 0691117047 | book | 15 | 27 |
| Harry Potter and the Order of the Phoenix by J. K. Rowling | 0439358078 | book | 34 | 37 |

doi:10.1371/journal.pone.0023829.t002

edges with an average degree of 7.80. The vertices represent items such as books, CDs and DVDs sold on Amazon’s Web site and the edges connect items that are frequently purchased together by customers.

As illustrative examples, we choose five digital media (four CDs and one DVD) and five books as source vertices, as shown in Table 2.

These items were adopted in [17]. The CD “Alegria” and the books “Small Worlds” and “Harry Potter” were used in [16].

We find a community from each of the starting vertices above. On the whole, the vertices in the communities are mostly works by the same singer or author. Due to space limitations, we present only five identified communities below. Fig. 6(A) shows the local

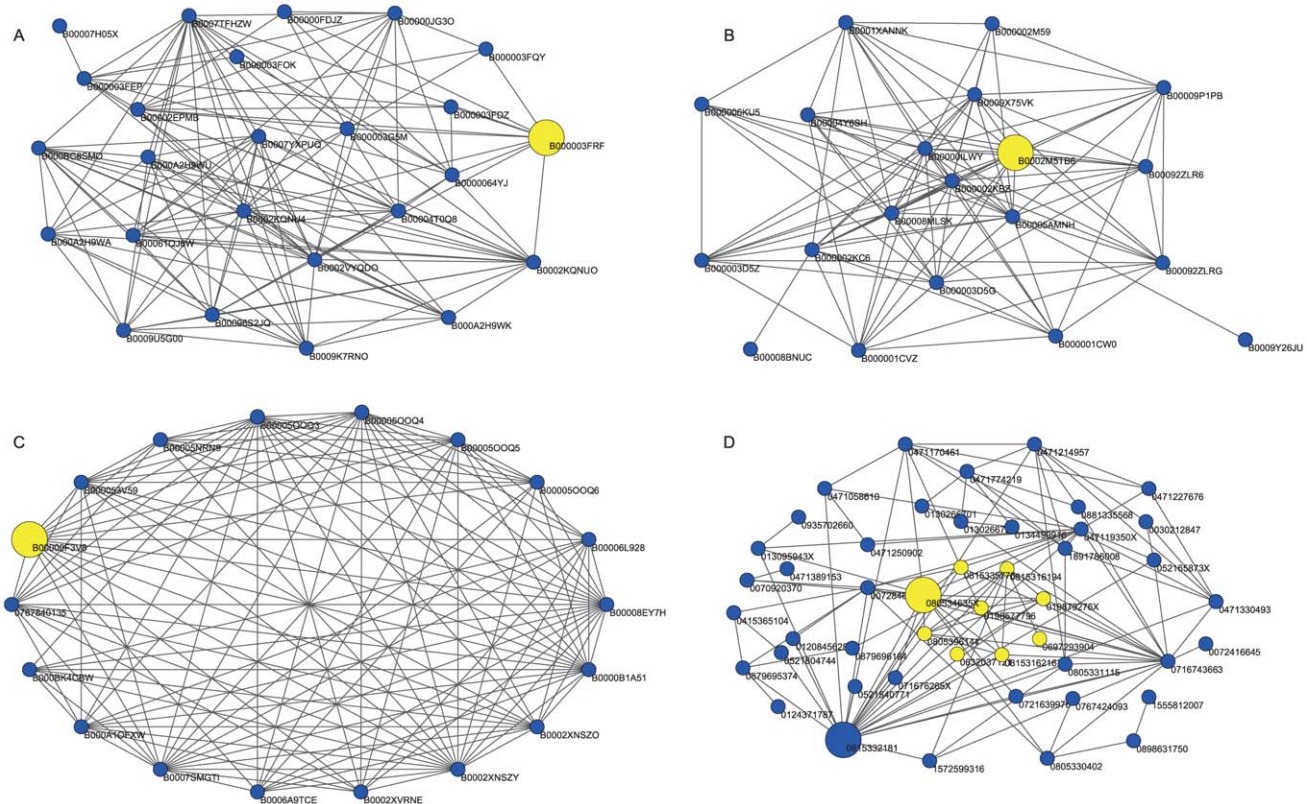


Figure 6. Identified local communities from different starting vertex in the Amazon co-purchase network. (A) Local community for the compact disc Alegria (ID B000003FRF), (B) local community for the DVD Cirque Reinvente (ID B00000F3V8), (C) local community for the compact disc Love Wants to Dance (ID B0002M5TB6), and (D) local community for the book Molecular Biology of Cell (ID 0815332181) and the book Molecular Biology of the Gene (ID 080534635X)(marked in yellow).

doi:10.1371/journal.pone.0023829.g006

Table 3. The detailed parameters of the generated benchmark networks.

| Network | n | k | $maxk$ | μ | $t1$ | $t2$ | $minc$ | $maxc$ | on | om |
|---------|--------|-----|--------|---------|------|------|--------|--------|-------|------|
| 10000S | 10000 | 20 | 50 | 0.1–0.8 | 2 | 1 | 10 | 50 | 0–0.3 | 2 |
| 10000B | 10000 | 20 | 50 | 0.1–0.8 | 2 | 1 | 20 | 100 | 0–0.3 | 2 |
| 100000S | 100000 | 40 | 100 | 0.1–0.8 | 2 | 1 | 50 | 100 | 0–0.3 | 2 |
| 100000B | 100000 | 40 | 100 | 0.1–0.8 | 2 | 1 | 100 | 200 | 0–0.3 | 2 |

doi:10.1371/journal.pone.0023829.t003

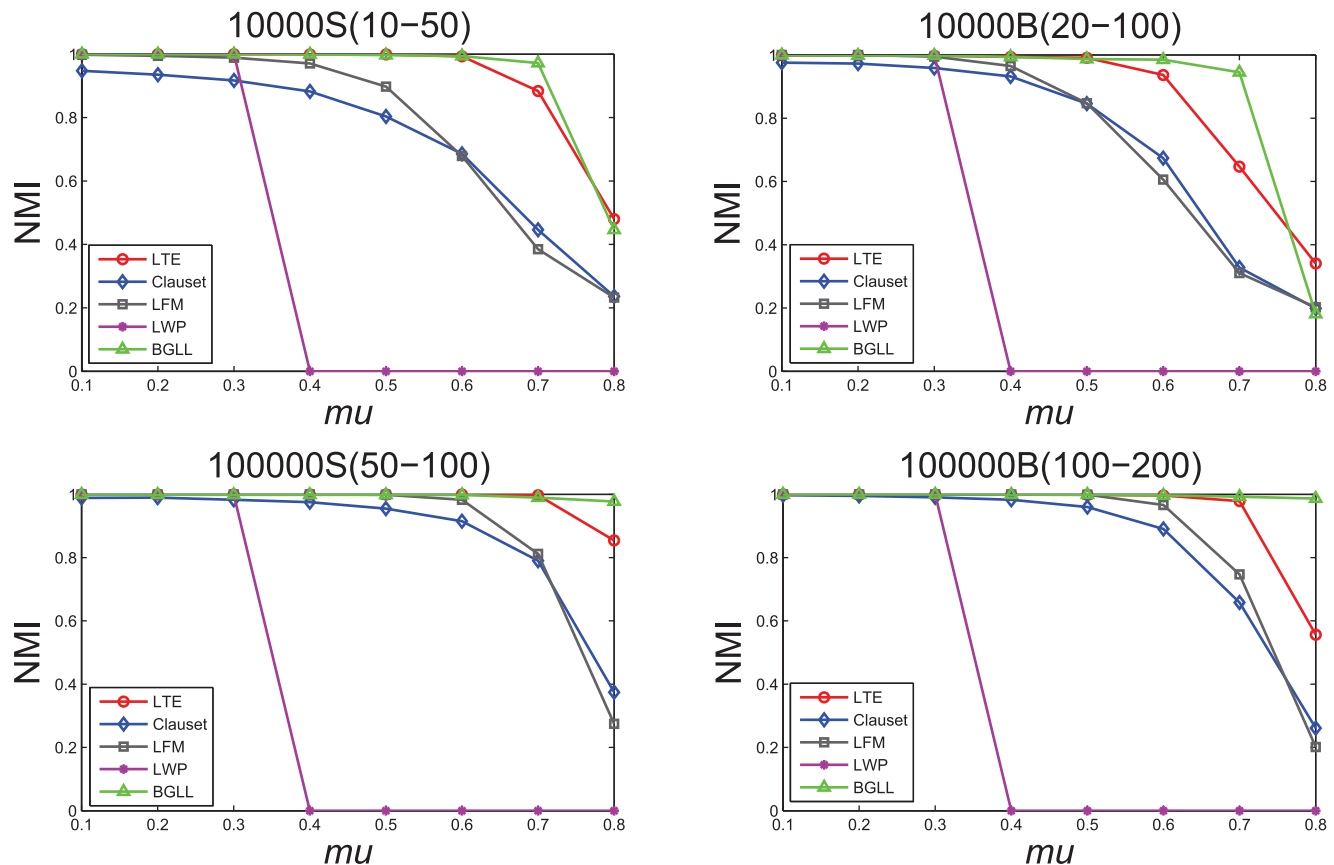
community for the CD “Alegria”. All 25 items found in this community are compact disks by Cirque du Soleil, the same as the results found in [17]. Fig. 6(B) shows the local community for the DVD “Cirque Reinvente”. Compared with the result found by in [17], the community found by our method is more accurate because it contains all 18 DVDs produced by Cirque du Soleil and excludes the books by Cirque du Soleil. Fig. 6(C) shows the local community for the compact disk “Love Wants to Dance”. All 20 items are compact disks of songs by Maria Muldaur, which covers the items found in [17]. The compact disk “Sweet Lovin Ol Soul” by Maria Muldaur is only identified by our method. Actually, it should be included in the community because it has more links with the vertices in the community than with the vertices outside the community.

Note that unlike the Clauset’s algorithm proposed in [16], which may identify multiple communities as explores the network, our algorithm only identifies one community for each source

vertex with a certain parameter. For example, starting from the same vertex: the compact disc “Alegria” by Cirque du Soleil, the Clauset’s algorithm will identify multiple communities including a community of compact discs by Cirque du Soleil that corresponds to our result, a community of DVD by Cirque du Soleil that corresponds to the community we detect using the DVD “Cirque Reinvente” by Cirque du Soleil as the source vertex, and another community containing books and movies by Cirque du Soleil.

It is also discovered that some communities that start from certain vertices will be totally included in or share some vertices with other communities starting from other vertices. As is shown in Fig. 6(D), the community starting from the vertex “Molecular Biology of the Gene” (ID 080534635X) is wholly enclosed in the community starting from the vertex “Molecular Biology of Cell” (ID 0815332181).

Therefore, the local communities identified by our local expansion algorithm LTE are more uniform and reasonable,

**Figure 7.** Test of the accuracy of different community detection algorithms on LFR benchmark networks.

doi:10.1371/journal.pone.0023829.g007

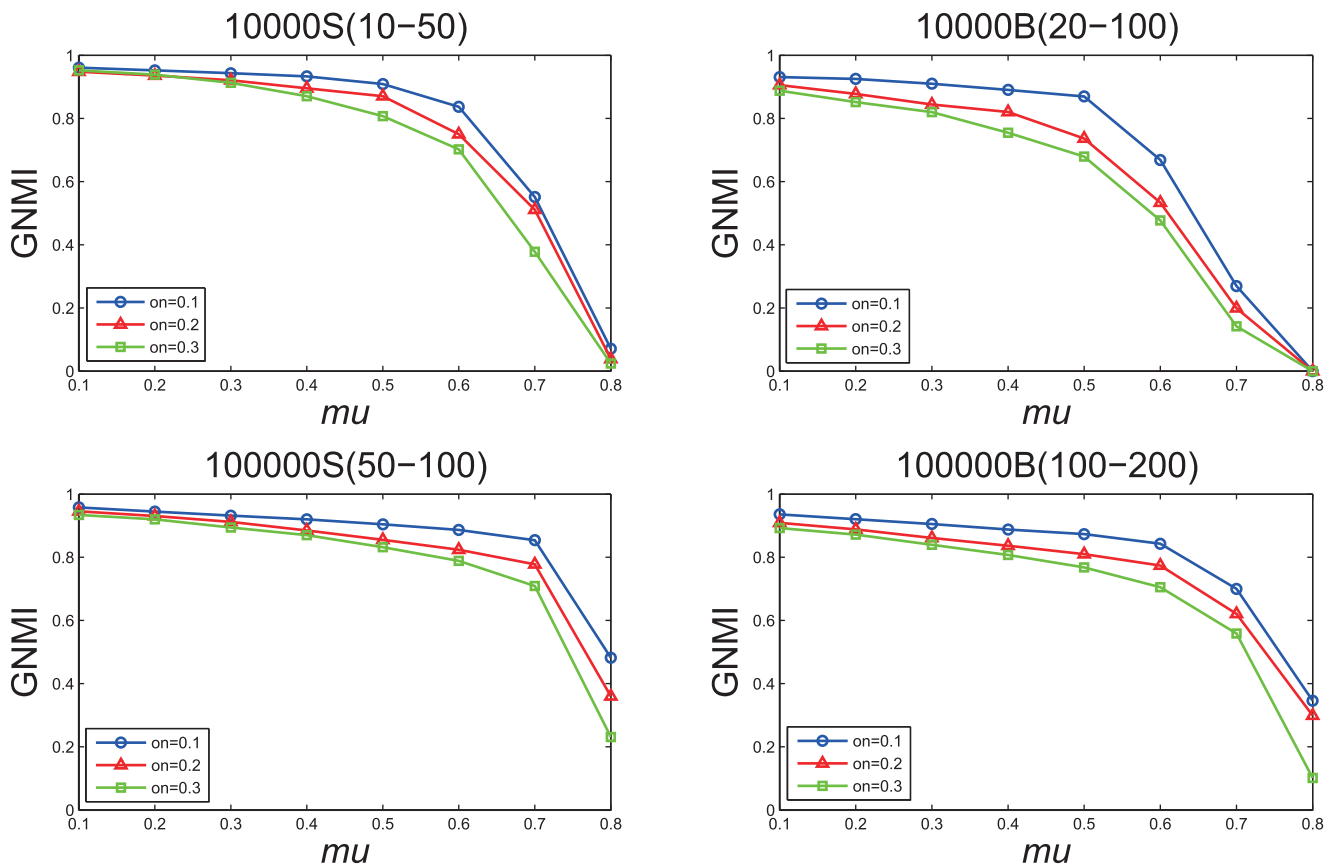


Figure 8. Test of the accuracy of our algorithm on LFR overlapping benchmark networks.
doi:10.1371/journal.pone.0023829.g008

and will therefore provide more accurate information for future recommendation.

Test On Synthetic Networks

So far, we have presented the experimental results of our algorithm on several real-world networks. In this section, we use some synthetic datasets to evaluate the performance of our algorithm.

LFR benchmark graphs. We use the Lancichinetti-Fortunato-Radicchi (LFR) benchmark graphs [24,28] to evaluate the accuracy of our algorithm. Two kinds of networks are generated with different ranges of community size, where S means that the sizes of the communities in the dataset are relatively small and B means that the sizes of the communities are relatively large. The detailed parameters of the benchmark networks are given in Table 3.

Some important parameters of the benchmark networks are: n : number of vertices; k : average degree of the vertices; $maxk$: maximum degree; μ : mixing parameter, i.e., each vertex shares a fraction of its edges with vertices in other communities (the higher the mixing parameter of a network is, the more difficult it is to reveal the community structure); $t1$: minus exponent for the degree sequence; $t2$: minus exponent for the community size distribution; $minc$: minimum for the community size; $maxc$: maximum for the community sizes; on : percent of the overlapping vertices; om : number of memberships of the overlapping vertices.

First, we set the overlapping parameter $on=0$, and generate eight networks for each type of dataset with mixing parameter μ ranging from 0.1 to 0.8. When $on=0$, communities in the generated networks are non-overlapped. We adopt the *normalized mutual information*(NMI) to evaluate the quality of detected

communities which is currently widely used in measuring the performance of graph clustering algorithms [24]. The accuracy of our method is compared with other state-of-the-art community detection methods: Clauset's local modularity [16], LWP [17], LFM [20] and BGLL [19]. For dealing with the possible effects of the vertex visiting order, we execute our method ten times with randomized vertex sequences and take the average as the final result. We select the result with maximum NMI for the LFM method which tries the parameter α by itself and shows hierarchical module structure of the whole network.

The NMI scores of the four methods are plotted in Fig. 7. Our algorithm gets $NMI = 1$ when $\mu \leq 0.5$ on the two S-networks and gets $NMI = 1$ when $\mu \leq 0.6$ on the two B-networks, which means a perfect match with the original network structure. We can see that the accuracy of our method is higher than other three local community detection algorithms (e.g., Clauset's method, LFM and LWP) on the generated networks, because our method use a similarity-based quality function of local community and a proper vertex adoption criterion. Note that the NMI values of the results obtained by algorithm LWP on networks with $\mu \geq 0.4$ are all 0. This is because the community definition of the algorithm LWP is that the number of edges within the community should be more than the number of edges between vertices in the community and vertices outside it. However, none of the subgraphs can satisfy this definition in the network with $\mu \geq 0.4$. We can also observe that the obtained NMI values of our method LTE on the networks with $\mu \leq 0.5$ are almost the same as that of the method BGLL. But the accuracy of our algorithm is lower than the method BGLL on the networks with $\mu > 0.5$ which shows that our algorithm has

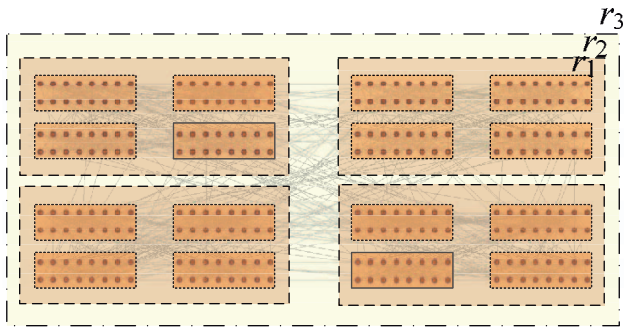


Figure 9. Schematic networks with hierarchical structures. Each of the four large clusters is made out of 64 vertices and has an internal subdivision in four clusters with 16 vertices.
doi:10.1371/journal.pone.0023829.g009

difficulty to deal with the network with indistinct community structure. BGLL is actually a global method by optimizing the modularity function locally and is known as an efficient and effective community detection method [13].

We also test the accuracy of our method with $\alpha = 1$ in detecting overlapping communities in the networks with parameter $on = 0.1, 0.2, 0.3$, respectively. In this experiment, we adopt *Generalized Normalized Mutual Information* (GNMI) proposed in [20] to evaluate the quality of detected overlapping communities. Fig. 8 shows the evaluation results on the benchmark overlapping graphs. We can see that our algorithm gets high precision. However, the GNMI values decrease along with the increase of the parameter on , because there are more and more vertices belonging to multiple communities in the network when the value of parameter on increases, which is increasingly difficult to be correctly identified.

Hierarchical benchmark networks. Here, we also test our method on synthetic networks with built-in hierarchical community structure. A benchmark network adopted by A. Arenes *et al.* [29] is used. This benchmark network is a hierarchical version of the classical benchmark proposed by Girvan and Newman [5]. There are 256 vertices in the network, split into 16 groups with 16 vertices on the first level. The 16 groups are ordered into four super-groups at the second level, as shown in Fig. 9. The internal degree of vertices at first level Z_{in1} and the internal degree of vertices at second level Z_{in2} keep an average degree $Z_{in1} + Z_{in2} + Z_{out} = 18$. The networks are indicated as H $Z_{in1} - Z_{in2}$, and we test the performance of our method on the H13-4 and H15-2 networks respectively. Our algorithm detects sixteen communities in both networks when we set $\alpha = 1$, which matches exactly with the original network's structure. When we decrease the value of parameter α , our algorithm can detect four communities and at last one community in the two networks, respectively. The parameters α for three different resolutions r_1, r_2 and r_3 are given in Table 4.

Random networks. For evaluating the performance of our method on random networks, we first test on the Erdos-Renyi random graph. Graphs with 100, 200, 500, 1000 vertices are generated where the connectivity probability p between each pair of

Table 4. The detailed parameters α for three different resolutions r_1, r_2 , and r_3 of the communities detected in the benchmark hierarchical networks H13-4 and H15-2, where c is the number of discovered communities.

| Network | $r_1(c=16)$ | $r_2(c=4)$ | $r_3(c=1)$ |
|---------|--------------|--------------|------------|
| H13-4 | [0.22, 2.74] | [0.08, 0.14] | [0, 0.05] |
| H15-2 | [0.15, 3.24] | [0.04, 0.05] | [0, 0.02] |

doi:10.1371/journal.pone.0023829.t004

vertices ranges from 0.01 to 0.8. We run our algorithm with $\alpha = 1$ for detecting overlapping communities. We find that our method discover only one community in the random networks with parameter $p \geq 0.05$. In the networks with $p < 0.05$, our method always finds covers of the whole networks consisting of a big community as well as several small ones. We also test our method on random scale-free graphs generated by the configuration model (<http://snap.stanford.edu/>). After testing our algorithm on several networks with different degree sequences, we find that our method always identifies a big community containing over 90% of the vertices along with several small communities in these networks. The above results show that our algorithm tends to find a big community as well as many small communities in the sparse random graph, while only one community can be detected in the random graphs with dense connections.

Discussion

In this paper, we present a novel local community quality criterion, called similarity-based tightness, and design a greedy algorithm LTE to infer the local communities in large-scale undirected networks. The proposed method is able to identify communities both overlapping and non-overlapping, and we show that it is fast and scalable in large-scale networks. Moreover, we describe that our algorithm is a multiresolution solution that can be used freely to acquire communities at any resolution. Experimental results on the real-world and synthetic datasets show that our algorithm achieves good performance. In the future, we believe it is beneficial to apply this method when analyzing large-scale online networks like mobile communication networks and/or Web graphs.

Acknowledgments

The authors would like to thank Jame P. Bagrow for his thoughtful comments on this paper. We thank Dr. A. Clauset, Dr. A. Lancichinetti and Dr. F. Luo for providing the source codes of their proposed algorithms. We also thank anonymous reviewers greatly for their valuable comments which make this paper more interesting and informative.

Author Contributions

Conceived and designed the experiments: JH HS QS. Performed the experiments: YL. Analyzed the data: JH HS QS. Contributed reagents/materials/analysis tools: YL HS. Wrote the paper: JH YL TW.

References

- Kleinberg JM, Kumar SR, Raghavan P, Rajagopalan S, Tomkins A (1999) The web as a graph: Measurements, models and methods. In: COCOON'99: Proceedings of the International Conference on Combinatorics and Computing. Berlin, Heidelberg, Germany: Springer-Verlag. pp 1–18.
- Hajra K, Sen P (2005) Aging in citation networks. *Physica A* 346: 44–48.
- Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. *Nature* 393: 440–442.
- Guimera R, Amaral LAN (2004) Functional cartography of complex metabolic networks. *Nature* 433: 895–900.
- Girvan M, Newman MEJ (2002) Community structure in social and biological networks. *PNAS* 99: 7821–7826.
- Newman MEJ (2004) Detecting community structure in networks. *Eur Phys J* 38: 321–330.
- Lancichinetti A, Kivela M, Saramaki J, Fortunato S (2010) Characterizing the community structure of complex networks. *PLoS ONE* 5: e11976.
- Palla G, Dernyi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435: 814–818.

9. Fortunato S (2010) Community detection in graphs. *Physics Reports* 486: 75–174.
10. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. *Phys Rev E* 69: 026113.
11. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *JSTAT* 2008: P10008.
12. Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. *Phys Rev E* 70: 066111.
13. Fortunato S, Barthélemy M (2007) Resolution limit in community detection. *PNAS* 104: 36–41.
14. Bagrow JP (2008) Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Application* 2008: P05001.
15. Bagrow JP, Bolt EM (2008) Local method for detecting communities. *Phys Rev E* 72: 046108.
16. Clauset A (2005) Finding local community structure in networks. *Phys Rev E* 72: 026132.
17. Luo F, Wang JZ, Promislow E (2008) Exploring local community structures in large networks. *Web Intelligence and Agent Systems* 6: 387–400.
18. Radicchi F, Castellano C, Ceconi F, Loreto V, Parisi D (2004) Defining and identifying communities in networks. *Proc Natl Acad Sci USA* 101: 2658–2663.
19. Xu X, Yuruk N, Feng Z, Schweiger T (2007) Scan: a structural clustering algorithm for networks. In: *KDD'07: Proceedings of the 13rd international conference on knowledge discovery and data mining*. San FranciscoCalifornia, USA: ACM. pp 49–60.
20. Lancichinetti A, Fortunato S, Kertesz J (2009) Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* 11: 033015.
21. Huang J, Sun H, Han J (2011) Density-based shrinkage for revealing hierarchical and overlapping communities in networks. *Physica A* 390: 2160–2171.
22. Arenas A, Fernández A, Gómez S (2008) Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics* 10: 053039.
23. Ronhovde P, Nussinov Z (2009) Multiresolution community detection for megascale networks by information-based replica correlations. *Phys Rev E* 80: 016109.
24. Lancichinetti A, Fortunato S (2009) Community detection algorithms: A comparative analysis. *Phys Rev E* 80: 056117.
25. Leskovec J, Lang K, Mahoney M (2010) Empirical comparison of algorithms for network community detection. In: *WWW '10: Proceedings of the 19th international conference on World wide web*. New York NY, USA: ACM. doi:10.1145/1772690.1772755.
26. Zachary WW (1977) An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* 33: 452–473.
27. Chen J, Zaiane OR, Goebel R (2009) Local community identification in social networks. In: *ASONAM'09: Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*. Washington, DC, USA: IEEE Computer Society. pp 237–242.
28. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys Rev E* 78: 046110.
29. Arenas A, Diaz-Guilera A, Perez-Vicente C (2006) Synchronization reveals topological scales in complex networks. *Phys Rev Lett* 96: 114102.