Local Community Detection Based on Network Motifs

Yunlei Zhang, Bin Wu*, Yu Liu, and Jinna Lv

Abstract: Local community detection aims to find a cluster of nodes by exploring a small region of the network. Local community detection methods are faster than traditional global community detection methods because their runtime does not depend on the size of the entire network. However, most existing methods do not take the higher-order connectivity patterns crucial to the network into consideration. In this paper, we develop a new Local Community Detection method based on network Motif (LCD-Motif) which incorporates the higher-order network information. LCD-Motif adopts the local expansion of a seed set to identify the local community with minimal motif conductance, representing a generalization of the conductance metric for network motifs. In contrast to PageRank-like diffusion methods, LCD-Motif finds the community by seeking a sparse vector in the span of the local spectra, such that the seeds are in its support vector. We evaluate our approach using real-world datasets across various domains and synthetic networks. The experimental results show that LCD-Motif can achieve a higher performance than state-of-the-art methods.

Key words: community detection; network motifs; local spectral clustering; seed set expansion; random walk

1 Introduction

Network analysis is a fundamental tool for modeling complex systems in various domains. Moreover, community detection is an important task, which aims to identify clusters of densely connected nodes from only the given network[1]. The vast majority of algorithms optimize an objective function which measures the edge density of the community, for instance, conductance or modularity. Most of the methods for community detection are global and manage to assign all nodes of the network to communities. Local community detection, also known as targeted or seeded community detection, is a specific case of this problem that takes an additional input in the form of a seed set of nodes. The idea is to identify a single community nearby the seed set without ever exploring the entire network, which makes local community detection methods much faster than the traditional global community detection methods. The theories and algorithms for local approaches have been most well studied using conductance as measure for the community quality[2]. However, conductance is only defined for simple undirected networks. Current community quality measures simply count individual edges and do not consider how these edges connect to form small network substructures, called network motifs. Such network motifs (higher-order link patterns) are essential to revealing the fundamental structures that control and affect the behavior of many complex systems, such as biological networks[3], transcriptional regulation networks[4], social networks[5], brain connected networks[6], and traffic networks[7]. Network motifs (as shown in Fig. 1) recur throughout these kinds of networks much more
often than in random networks. We focus here on three-node motifs, which have often been suggested as being stronger signals of community structure compare to edges alone. For example, feed-forward loops (Fig. 1, M5) have been proven to be essential to understanding transcriptional regulation networks[4], triangle motifs (Fig. 1, M1 – M7) are common link patterns in social networks[5], open bidirectional wedges (Fig. 1, M13) are key to structural hubs in brain networks[6], and two-hop paths (Fig. 1, M8 – M13) are fundamental to understanding air traffic patterns[7]. It has been demonstrated that higher-order link patterns involving multiple nodes can improve the understanding of the underlying network structure[8]. However, most of the existing local community detection methods[9] detect local community at the level of individual nodes and edges, without considering how these edges connect in the network. Benson et al.[8] proposed a framework using a higher-order network structure to gain new insights into the organization of complex systems; Lim and Lee[10] proposed a method to cluster network based on motif embedding; and Yin et al.[11] proposed a method which detects local community by motif-approximate personalized PageRank. However, few local community detection methods are designed to account for higher-order network structures.

In many networks, we would like to identify the members of an interesting community from exemplar seeds; moreover, the community has specific interaction patterns between community members which can be modeled by network motifs. For example, in the Sina Weibo (https://weibo.com/) friendship network, it is necessary to identify the community by including some interesting users; moreover, members of the community are friends with each other.

In this paper, we propose a novel local community detection method based on network motifs, called Local Community Detection method based on network Motif (LCD-Motif). Firstly, we generate the motif adjacency matrix from the original network. Then, we consider the span of a few dimensions of vector after the short random walk on the motif adjacency matrix, and use this span as the approximate invariant subspace, which we refer to as local motif spectra. In contrast to the traditional spectral clustering methods, the local motif spectral method does not require the burdensome computation of a large number of singular vectors. Concretely, we mine the community by seeking a sparse approximate indicator vector in the span of the local motif spectra, such that the seeds are in its support vector. This can be mathematically achieved by solving an $l_1$-penalized linear programming problem.

In this paper, we incorporate motifs into the seed set expansion in support of detecting a local community. In summary, the contributions of our work are as follows:

- We propose a method that can find a local community with minimal motif conductance by incorporating higher-order network information captured by network motifs.
- Our proposed method generates a motif adjacency matrix from an original adjacency matrix, and then generates the local motif spectra from the motif adjacency matrix (avoiding the burdensome computation of a large number of singular vectors), and then seeks a sparse vector in the local motif spectra by solving an $l_1$-penalized linear programming problem.
- We demonstrate through experiments that the proposed method outperforms other state-of-the-art methods at finding a local community in large networks.

The rest of this paper is organized as follows. We introduce related work in Section 2, and some preliminaries are presented briefly in Section 3. Section 4 presents our proposed method, which can detect a local community based on higher-order interaction patterns. Section 5 shows experimental results on networks with and without ground truth. Finally, Section 6 concludes this study and describes some directions for future work.

2 Related Work

Many novel community detection methods have been developed recently, such as those based on dynamic...
Motif-based community detection. In Ref. [5], a theorem is presented which yields expectations and variances for social network model measures based on three-node network motifs. Milo et al. [3] found network motifs in biochemistry, neurobiology, ecology, and engineering networks, and that network motifs describe the basic building blocks of most networks. Because existing community detection methods typically rely on the edges between nodes in networks, Benson et al. [8] proposed a framework for partitioning networks based on network motifs. The framework uncovers motif clusters in a few steps, as follows: (1) form a motif adjacency matrix according to the given network $G$ and motif $M$ of interest; (2) compute the spectral ordering of the nodes from the normalized motif Laplacian matrix constructed via the motif adjacency matrix; and (3) find the prefix set of the spectral ordering of the nodes with the smallest motif conductance. These methods detect global communities effectively; however, they ignore the issue of how to detect local communities with higher-order structure information.

Seed set expansion based methods. In attempting to discover the community from a local viewpoint, LCD-Motif is in spirit similar to the seed set expansion methods in Refs. [2, 15–18]. In detail, Andersen and Lang [15] adapted lazy random walks to expand a set into a community with locally minimal conductance. Focusing on seeding strategies, Whang et al. [16] proposed several sophisticated methods to choose the seed set and then used a PageRank scheme to expand the seed set until a community with optimal conductance is detected. Kloumann and Kleinberg [17] provided a systematic understanding of variants of PageRank-based seed set expansions. The heat kernel method [18] improved on PageRank by combining it with a sophisticated diffusion method. LEMON [2] is a seed set expansion method based on a subspace derived by a short random walk. These methods can work well at detecting a local community; however, they do not consider higher-order network information.

3 Preliminaries

3.1 Motif adjacency matrix

Given an unweighted, directed network $G = (V, E)$ and a motif $M$ of interest, we define the motif adjacency matrix $W_M$ whose entries $(i, j)$ are the co-occurrence counts of nodes $i$ and $j$ in motif $M$. $(W_M)_{ij} = \text{number of motif instances in } M \text{ where nodes } i \text{ and } j \text{ participate in the motif}$. For example, in Fig. 2, the entry $(0, 1)$ is 3, because nodes 0 and 1 co-occur in three instances of motif $M_6$, including $\{2, 0, 1\}$, $\{4, 0, 1\}$, and $\{3, 0, 1\}$.

3.2 Problem statement

Given a directed network $G = (V, E)$, a motif $M$, and a seed set $S$ in the target community $C$, where $|C|\ll|V|$ and $|S|\ll|C|$, the task is to find the remaining members in $C$, containing as many instances of the given motif.

Fig. 2 Illustration of motif adjacency matrix and local community detection (the given motif type is $M_6$, $S:1$ denotes that node 1 is the seed node, the middle part is the motif adjacency matrix transformed from the original network, the right part is the local community which contains seed node 1 and three instances of motif $M_6$).
type as possible and avoiding cutting off instances of $M$. For example, we want to detect the local community containing seed node 1 with more instances of motif $M_6$ and less instances of motif $M_6$ being cut. Node 1 relates to two communities, community $\{0, 1, 2, 3, 4\}$ and community $\{1, 5, 6, 7, 8, 9\}$. The former community contains 3 instances of $M_6$, $\{0, 1, 2\}$, $\{0, 1, 4\}$, and $\{0, 1, 3\}$, and leads to 1 instance being cut, $\{1, 5, 6\}$. The latter contains 3 instances, $\{1, 5, 6\}$, $\{5, 7, 8\}$, and $\{7, 8, 9\}$, and leads to 3 instances being cut, $\{0, 1, 2\}$, $\{0, 1, 4\}$, and $\{0, 1, 3\}$. It is better to gather nodes $\{0, 1, 2, 3, 4\}$ as a community, because there are less instances cut. In Section 5.2, a motif-based evaluation metric of community is introduced. Generally speaking, we focus on addressing the question of how to accurately find a small community containing a given seed set of nodes and a given motif.

4 Local Expansion via Minimizing One Norm

4.1 Algorithm overview

Spectral clustering based on network motifs[8] makes use of a small number of singular vectors proportional to the number of communities in the network. If a network has a lot of small communities, it is impractical to calculate a number of singular vectors greater than the number of communities. We propose a new method that could avoid the burdensome computation of a large number of singular vectors.

In spectral motif clustering methods[8], one finds the first few singular vectors of the motif Laplacian matrix of a network $G$ with $n$ nodes. Supposing the first $d$ singular vectors are obtained, one can form an $n \times d$ matrix as a latent space. Then each node is associated with a point in this latent space whose coordinates are given by the entries of the corresponding row of the matrix. Nodes are then clustered using some clustering method such as $k$-means. This method is not likely to work well if communities are small.

We make two fundamental changes to the spectral motif clustering method. The first change is to overcome the drawback of computing the singular vectors. Intuitively, the nodes around the seed members are more likely to be in the target community, thus a short random walk serves as a natural approach to reveal these potential members. We start a random walk from one or several known members of the target community and run it for a few steps. The number of random walk steps should be enough to reach out to the nodes in the target community, but not so long as to cover the entire network. Instead of considering a single probability vector, we consider the span of a few dimensions of vectors after the short random walks and use this as the approximate invariant subspace (local motif spectra). The second change is to the clustering method. Instead of using $k$-means to partition the points in the latent space into disjoint clusters, we look for the minimum 0-norm vector in the span of the invariant subspace obtained above, such that the seed members are in its support vector. We want to find rows of the invariant subspace that point in the same direction as seed members. We use a 1-norm vector as a proxy for minimum 0-norm vector since finding the 0-norm vector is an NP-hard problem.

Given the input of an seed set of nodes $S$ and a motif $M$, our algorithm outputs local community $C$ containing $S$ and with specific connected pattern characterized by motif type $M$ among members in $C$. The entire process is described in Algorithm 1.

Step 1. Generate the motif adjacency matrix.

Given a network $G$ and a motif type $M$ of interest, we generate the motif adjacency matrix $W_M$ whose entries $(i, j)$ are the co-occurrence counts of nodes $i$ and $j$ in the motif $M$: $(W_M)_{ij} = \text{numbers of instances of } M \text{ containing nodes } i \text{ and } j$.

In the process of generating the motif adjacency matrix, we should consider two classes of motifs: (1) triangles such as $M_1 - M_7$ in Fig. 1, and (2) wedges such as $M_6 - M_13$ in Fig. 1. For a directed network $G$, we can use the following algorithm to generate a triangle motif adjacency matrix: (1) form a new network $G_{\text{dir}}$ by removing the direction from all edges in $G$; (2) find all triangles in $G_{\text{dir}}$; then (3) for every triangle in $G_{\text{dir}}$, check which directed triangle motif it is in $G$. For wedge motifs, we can look at every pair of neighbors of every node to form the wedge motif adjacency matrix.

Step 2. Generate the local motif spectra.

Let $\overline{W}_M$ be the normalized motif adjacency matrix of the network. We define $\overline{W}_M$ as

$$\overline{W}_M = D^{-1/2}(\overline{W}_M + I)D^{-1/2}$$  \hspace{1cm} (1)

where $D$ is the diagonal matrix of node weight.

Consider a random walk starting from exemplary nodes in $S$. Let $p_0$ denote the initial probability vector where the total probability is evenly distributed among the seed members. Consider the span of $I$-dimensional
Algorithm 1 LCD-Motif algorithm

Input:
- network $G = (V, E)$, motif type $M$, seed set $S$, subspace dimension $l$, and random walk step $k$

Output:
- target community $C$

1: while Not finding the community with the first local minimum motif conductance during expanding seed set do
2: Generate subgraph $G_S$ containing seed set $S$ by using short random walk.
3: Generate the motif adjacency matrix $W_M$ using subgraph $G_S$ and motif type $M$.
4: Compute the local motif spectra according to Eq. (3).
5: Compute the sparse vector $\hat{y}$ according to Eq. (4).
6: Sort the sparse vector in non-ascending order.
7: while $\min \leq i < \max$ do
8: Truncate the sorted vector $\hat{y}$ at some point $i$ such that all the nodes corresponding to the elements no less than the $i$-th element in the vector are included in the detected community $C$.
9: Calculate the motif conductance of community $C$.
10: Select the community $C$ with the first local minimum motif conductance.
11: end while
12: Augment the seed set by adding the nodes corresponding to the top $t$ elements of the sparse vector $\hat{y}$, denoted by $S'$, where $t$ is increased by a constant expanding step $s$.
13: Assign $S'$ to $S$ as a new seed set to generate new subgraph.
14: end while
15: return community $C$.

The probability vector which consists of probability vector in $l$ successive random walks:

$$P_{0,l} = [p_0, W_M^1p_0, \ldots, W_M^lp_0]$$

The initial invariant subspace is then obtained by calculating the orthonormal basis of the span $P_{0,l}$, denoted by $U_{0,l}$. We then use the following recurrence to iteratively calculate the $l$-dimensional orthonormal basis $U_{k,l}$ after $k$ steps of random walk:

$$U_{k,l}R_{k,l} = U_{k-1,l}W_M$$

where $R_{k,l} \in \mathbb{R}^{n \times l}$ is chosen such that $U_{k,l}$ is orthonormal. The orthonormal basis of $U_{k,l}$ is used as the local motif spectra. The details of generating local spectra are described in Algorithm 2.

Step 3. Seek sparse vector.

With the local motif spectra $U_{k,l}$, we solve the following linear programming problem to detect the community containing the seed set $S$ (http://www.gnu.org/software/glpk/).

$$\begin{align*}
\min & \quad e^Ty = \|y\|_1, \\
\text{s.t.} & \quad y = U_{k,l}x, \\
& \quad y \geq 0, \\
& \quad y(S) \geq 1
\end{align*}$$

Algorithm 2 LocalSpectra

Input:
- subgraph $G_S$, seed set $S$, subspace dimension $l$, and random walk step $k$

Output:
- local spectra $U_{k,l}$

1: Compute normalized motif adjacency matrix $W_M$ using Eq. (1).
2: Initialize $p_0$.
3: $U_{0,l} = [p_0, W_M^1p_0, \ldots, W_M^lp_0]$.
4: for $i = 1, \ldots, k$ do
5: $U_{i,l}R_{i,l} = U_{i-1,l}W_M$ ($R_{i,l} \in \mathbb{R}^{n \times l}$ is obtained by $QR$ factorization so that $U_{l,l}$ is orthonormal)
6: end for
7: return local spectra $U_{k,l}$

where $e$ is a vector of all ones, and both $x$ and $y$ are unknown vectors. The first constraint implies that $y$ is in the subspace $U_{k,l}$. The elements in $y$ indicate the likelihood of the corresponding nodes belonging to the target community, which is non-negative. The third constraint ensures that seeds are in the support of sparse vector $y$, indicating that seeds must be in the target community.

After vector $y$ is obtained, the elements in $y$ are sorted in non-ascending order and vector $\hat{y}$ is attained, the nodes corresponding to the top $|C|$ elements in $\hat{y}$ are returned as the detected community with respect to the seed set $S$.

Step 4. Expand the seed set.

In the expanding phase, we use the top $t$ nodes corresponding to the elements of the vector $\hat{y}$ as the augmented seed set, denoted as $S'$.

Then Steps 2 and 3 are repeated using the augmented seed set $S'$. The detection quality of the community can be improved through iterations via increasing $t$ by a constant number $s$ each time, where $s$ is defined as the seed expanding step, which can be used as a tunable parameter for adjusting the convergence rate. Usually, the larger expansion step would result in lower performance but a faster running speed with less iterations. In the experiments, we fix the seed expansion step to be 6. The number of iterations for the seed expansion is determined by the stop criteria (in Section 4.2).
4.2 Stop criteria

If there are ground truth communities available, the above algorithm is guaranteed to stop within a few iterations, since the seed set will no longer augment once its size exceeds that of the ground truth community. The algorithm would find the community with the highest F1 score (see Section 5.2.1) during the iterations as a result. However, in real cases, where ground truth communities are not available, it is difficult to stop expanding such that the detected community is a “good” community. It is necessary to solve two problems: (1) how to automatically determine the size of the community given a seed set $S$, and (2) when to stop augmenting the seed set during the expanding phase.

4.2.1 Determine the size of the community

It has already been shown that random walks produce communities with conductance guarantees and ensure a small boundary defining a natural community in locally based detection algorithms\textsuperscript{[8,15]}. The fact is that including irrelevant nodes into the target community would inevitably increase the conductance, and finding a lower-conductance community could ensure the closeness between the found members and the known seed set. In Refs. [2, 16], the authors adopted conductance as the metric for defining a good community found by the algorithm around a seed set. The local conductance of a small community contains valuable information and could be used as stop criteria for local community detection methods at the level of individual nodes and edges. In our method, we use motif conductance\textsuperscript{[8]} as our method’s stop criteria.

It is supposed that we have a rough estimation of the lower and upper bounds for the size of communities in a network, denoted by $min$ and $max$, respectively. We modify the original algorithm to automatically stop as follows.

At Step 3, after the sorted vector $\hat{y}$ is obtained, we truncate the sorted vector at some point $y_g$ such that all the nodes corresponding to the elements no less than $y_g$ are included in the detected community. To locate the best position of $y_g$, we denote $\Lambda_i$ as the set of nodes corresponding to the top $i$ elements in $\hat{y}$. We then sweep over the sets from $\Lambda_{min}$ to $\Lambda_{max}$ and calculate the corresponding motif conductance of each set. In practice, the value of motif conductance with respect to varying size would usually change in a non-monotonic pattern that decreases first and then increases later on. The first relative minimum motif conductance encountered on this curve is adopted as the estimated size of the community with respect to the seed set $S$, denoted by $\phi_{S}^{min}$.

4.2.2 Stop the expanding process

As we keep augmenting the seed set through expansion at Step 4, a different seed set would result in a different sparse vector $\hat{y}$ and thus lead to potentially different communities. Practically, one of these seed sets during the expanding process would find the best community. When to stop the expansion process needs to be determined, and it can be solved in a similar way as for determining community size. Actually, we track the value of $\phi_{S}^{min}$ for different seed sets during the expansion, and stop expanding the seed set when $\phi_{S}^{min}$ reaches a local minimum and starts to increase for the first time.

4.3 Complexity reduction by sampling

When finding a small community from a large network, it would be very costly to take all the nodes into account. We want to discover the target community accurately while keeping the number of examined nodes low. Sampling is an effective solution to solve the memory consumption issue, since it avoids the necessity to load the entire network into memory.

In practice, the remaining members in the target community are more likely to be around the seed members. This motivates us to reduce complexity by taking only a portion of the network into consideration. This partial network should contain as many nodes in the target community as possible, while maintaining a small size of the same scale as that of the target community.

To sample the network, we expand the seed set using a random walk. After a few steps of random walk, nodes with a large probability are more likely to be in the target community while nodes with a small probability are to be considered redundant. If the target community exists for seed members, according to Ref. [15], this targeted community would serve as a bottleneck for the probability to be spread out. It is noted that other expansion methods such as Breadth-First Search (BFS) would ignore the bottleneck defining the community and rapidly sample the entire network before the fraction of nodes in the community is finished. The subgraph returned by BFS always contains less nodes in the target community than the subgraph of the same
networks

5.1.1 Synthetic datasets

5.1 Datasets

5 Experiment Analysis

5.1.2 Real world datasets

4.4 Analysis of computational complexity

We now analyze the computational complexity of LCD-Motif. Overall, the complexity of the algorithm is governed by the generation of a motif adjacency matrix \( W_M \), the computation of a local motif spectra, the computation of a sparse vector, and the expansion of a seed set. For simplicity, we assume that we can access and modify matrix entries in \( O(1) \) time. Let \( m \) and \( n \) denote the number of edges and nodes in the network, respectively.

In Step 1, the computational time to generate \( W_M \) is bounded by the time to find all instances of the motif in the network. Theoretically, motif counting is efficient. Here we consider two classes of motifs: (1) triangles, and (2) wedges. For triangle motifs, an algorithm which has \( O(m^{1.5}) \) computational complexity\(^{[19]} \) can efficiently enumerate the triangles. Since changing a directed network to an undirected network is linear and we can check which directed triangle motif triangles are in \( O(1) \), the same \( O(m^{1.5}) \) complexity holds for generating the motif adjacency matrix. For wedge motifs, we can list them all by looking at every pair of neighbors of every node in computational complexity \( O(n d_{\text{max}}^2) \), where \( d_{\text{max}} \) is the maximum degree in the network. In Step 2, the computational time to generate the local motif spectra \( U_{k,l} \) is bounded by the time to orthogonal matrix \( U_{k,l} \) which has \( O(n'n^2) \) computational complexity, where \( m' \) and \( n' \) denote the number of edges and nodes in subgraph \( G_S \). In Step 3, we can find the spare vector in \( O(n') \) computational complexity\(^{[20]} \). In Step 4, the vector is sorted in \( O(n'\log n') \). The overall computational complexity of triangular motifs and wedge motifs is \( O(m^{1.5} + m'n^2 + n' + n'\log n') \) and \( O(m_{\text{max}}^2 + m'n^2 + n' + n'\log n') \), respectively. It is noted that, in large networks, \( n' \) is far smaller than \( n \).

5 Experiment Analysis

5.1 Datasets

5.1.1 Synthetic datasets

Lancichinetti-Fortunato-Radicchi (LFR) benchmark networks\(^{[21]} \) have been widely used for checking the performance of community detection algorithms. LFR networks are generated with a nested community structure. It is convenient to generate networks with different topological features by tuning different parameters, including the number of nodes \( n \), the average degree \( k \), the maximum degree \( k_{\text{max}} \), the minimum and maximum community sizes \( |C|_{\text{min}} \) and \( |C|_{\text{max}} \), and the mixing parameter \( \mu \). Among these parameters, the mixing parameter \( \mu \) is the most significant, controlling the fraction of connections for each node that cross to a community with which the node is not associated. Usually, a larger \( \mu \) would result in lower performance.

To make the performance evaluation of our algorithm, we generate eight networks with mixing parameters \( \mu \) varying from 0.1 to 0.8. Table 1 lists the value of the parameters we have used for generating the LFR datasets.

5.1.2 Real world datasets

For the purpose of testing on realworld networks, we include five datasets from the Stanford Network Analysis Project (http://snap.stanford.edu/data) and Uri AlonLab (https://www.weizmann.ac.il/mcb/UriAlon/homepage). The statistical information of the datasets is summarized in Table 2. These datasets span various domains of network applications, including a food chain network (FloridaBay), a biology network (E.Coli), a citation network (HepPh), a social network (Slashdot), and a WWW network (WebStanford).

In the FloridaBay food chain network, each node represents a compartment (roughly, a organism and

| n | Number of nodes | 2000 |
| \( \mu \) | Mixing parameter | 0.1, 0.2, 0.3, \ldots, 0.8 |
| k | Average degree | 10 |
| \( k_{\text{max}} \) | Maximum degree | 50 |
| \( |C|_{\text{min}} \) | Minimum community size | 20 |
| \( |C|_{\text{max}} \) | Maximum community size | 50 |

Table 2 Statistics for the networks. n: number of nodes; m: number of edges; CC: clustering coefficient of the network; GT: ground truth.

| Domain     | Dataset         | n  | m  | CC  | GT |
|------------|-----------------|----|----|-----|----|
| Food chain | FloridaBay      | 128| 2106| 0.3346 | Y |
| Biology    | E.Coli          | 418| 519 | 0.0865 | Y |
| Citation   | HepPh           | 34546| 421578| 0.2848 | N |
| Social     | Slashdot        | 77360| 905468| 0.0555 | N |
| WWW        | WebStanford     | 281903| 2312497| 0.5976 | N |
a species) and each edge represents directed carbon exchanges from one species to another species (in many cases, this means that species $j$ eats species $i$). We labeled each node according to its ecological classification. In the E.Coli transcription network, each node represents an operon, and each edge is directed from an operon that encodes a transcription factor to an operon that it directly regulates. Each node is labeled according to its specific function\cite{4}. The HepPh citation network is derived from the e-print arXiv; each node represents a paper, and each edge represents a citation in a later paper to an earlier one. In the Slashdot social network, each node represents a user, and each edge represents the friendship between two users. In the WebStanford WWW network, each node represents a page in a WWW network, and a directed edge represents a hyperlink from a source page to a destination page.

5.2 Evaluation metrics

In this paper, we adopt two metrics to evaluate the performance of methods on networks with and without ground truth, respectively. (1) F1 score is adopted to evaluate the similarity between the algorithm community and the ground truth community, and (2) motif conductance measures the quality of the community detected from networks without ground truth.

5.2.1 F1 score

For the evaluation metric, we adopt F1 score to quantify the similarity between the algorithm community $C$ and the ground truth community $C^*$. A higher F1 score means better performance. The F1 score for each pair of $(C, C^*)$ is defined by

$$F1(C, C^*) = \frac{2 \cdot \text{Precision}(C, C^*) \cdot \text{Recall}(C, C^*)}{\text{Precision}(C, C^*) + \text{Recall}(C, C^*)}$$

where the precision and recall are defined as

$$\text{Precision}(C, C^*) = \frac{|C \cap C^*|}{|C|}$$

and

$$\text{Recall}(C, C^*) = \frac{|C \cap C^*|}{|C^*|}$$

where $|C \cap C^*|$ means the number of elements that occur in both the algorithm community $C$ and the ground truth community $C^*$.

5.2.2 Motif conductance

To evaluate communities detected from networks without a ground truth community, we adopt motif conductance\cite{8} to quantify the detected community $C$. Lower motif conductance means better performance. The motif conductance for community is defined by

$$\phi_M(C) = \frac{\text{cut}_M(C, \overline{C})}{\min[\text{vol}_M(C), \text{vol}_M(\overline{C})]}$$

where $\overline{C}$ denotes the detected community, and $\overline{C}$ denotes the remainder of the nodes (the complement of $C$), $\text{cut}_M(C, \overline{C})$ is the number of instances of motif $M$ with at least one node in $\overline{C}$ and one node in $C$, and $\text{vol}_M(C)$ is the number of nodes in instances of $M$ that reside in $C$. Equation (8) is a generalization of the conductance metric in spectral graph theory, one of the most popular graph partitioning scores. We refer to $\phi_M(C)$ as the motif conductance of $C$ with respect to $M$.

5.3 Comparison with state-of-the-art algorithms

Our method has a small memory consumption, such that a machine with 8 GB RAM can afford to process a large network. We refer to the experimental results reported in some recent publications on localized community detection algorithms.

(1) **HeatKernel\cite{18}.** This is a type of graph diffusion for locally identifying a community nearby a starting seed node. The algorithm can deterministically find the community by computing the diffusion.

(2) **LEMON\cite{2}.** This method finds a local community by seeking a sparse vector in the span of local spectra derived from a short random walk from a seed set of nodes.

(3) **Personalized PageRank (PPR)\cite{17}.** Starting from a seed set, personalized PageRank can find the remaining members in the targeted community. The personalized PageRank scheme is computed using the power method and the jumpback probability $\alpha = 0.10$.

(4) **Seed Set Expansion (SSE)\cite{16}.** The seed set expansion approach starts with a phase of choosing a good seed set. The personalized PageRank scheme is then applied to expand the seeds until a community with minimal conductance is found.

(5) **LinLog+Motif\cite{10}.** This is based on force-directed embedding with motif-based weighting.

(6) **MAPPR\cite{11}.** This is a motif-based approximate personalized PageRank algorithm that finds clusters containing a seed node with minimal motif conductance. We set teleportation parameter $\rho = 0.98$ and tolerance $\epsilon = 0.0001$. 

...
(7) TECTONIC\cite{14}. This reweights the edge in the network by considering motif, and then removes all edges with weight smaller than the given threshold $\theta$ to get connected components as communities. We set $\theta = 0.06$ as suggested in Ref. [14].

(8) LCD-Motif. This is our proposed method that detects the local community by expanding the seed set from the subspace generated by running a short random walk. We set parameter dimension $l = 2$ and random walk step $k = 3$, see the detail in Section 5.7.

We obtain the source code of baseline methods from the authors, except in the cases of PPR and LinLog+Motif, and we implement them for comparison with LCD-Motif.

5.4 Results on synthetic networks

To evaluate the performance of LCD-Motif, we conduct experiments on eight synthetic networks. Figure 3 compares the average F1 score with some state-of-the-art algorithms on LFR benchmark networks. It shows that LCD-Motif performs better than other baseline methods on most LFR benchmark networks. Especially when the mixing parameter $\mu$ is between 0.2 and 0.7, as is shown in Fig. 3, LCD-Motif brings a high relative improvement compared with the best results among the baselines. Among the seven baselines, we note that SSE and HeatKernel consistently perform poorly on most synthetic networks. We further look into the communities found by SSE and HeatKernel, and find that they both tend to find communities that are much larger than the ground truth communities. This implies that both algorithms extract community structures from networks that bear little resemblance to the natural formation of the communities. The detailed F1 score is listed in Table 3.

5.5 Results on networks with ground truth community

To evaluate the performance of LCD-Motif, we conduct experiments on two realworld networks with a ground truth community — the FloridaBay food chain network and the E.coli transcription network. Figure 4a gives the experimental results on the two networks with ground truth community, and LCD-Motif achieves the highest F1 score among state-of-the-art methods. We detect local motif communities in the FloridaBay food chain network and E.coli transcription network based on $M_6$ and $M_5$, respectively. Actually, in the FloridaBay food chain network, $M_6$ models two species that prey on each other and then compete to feed on a common third species. The FloridaBay food chain network contains four layers of the ecosystem which can be modeled by $M_6$, and $M_6$ reveals the higher-order interaction patterns in the network\cite{8}. In the E.coli transcription network, a function model corresponding to a community consists of many micro-clusters with a specific link pattern, and motif $M_5$, which is a feed-forward loop, gives fundamental elements of the E.coli transcription network\cite{4}. Since these two networks contain rich network motifs, LCD-Motif generates the subspace from the motif adjacency matrix, which considers the motif information in a community. LCD-Motif can reveal the real-world network motif community from networks containing motif structures. LCD-Motif achieves the highest F1 score among state-

![Fig. 3 Average F1 score on LFR datasets with varying mixing parameter from 0.1 to 0.8.](image)

| Method          | LFR1 | LFR2 | LFR3 | LFR4 | LFR5 | LFR6 | LFR7 | LFR8 |
|-----------------|------|------|------|------|------|------|------|------|
| HeatKernel      | 0.040| 0.044| 0.060| 0.056| 0.131| 0.157| 0.167| 0.183|
| LEMON           | 0.950| 0.873| 0.768| 0.632| 0.466| 0.345| 0.203| 0.093|
| PPR             | 0.914| 0.391| 0.222| 0.126| 0.144| 0.161| 0.172| 0.194|
| SSE             | 0.314| 0.196| 0.120| 0.121| 0.132| 0.143| 0.137| 0.154|
| LinLog+Motif    | 1.0  | 0.425| 0.230| 0.172| 0.114| 0.037| 0.052| 0.037|
| MAPPR           | 1.0  | 0.472| 0.308| 0.139| 0.123| 0.059| 0.074| 0.013|
| TECTONIC        | 0.991| 0.474| 0.303| 0.133| 0.121| 0.043| 0.118| 0.008|
| LCD-Motif       | 0.970| 0.954| 0.937| 0.887| 0.829| 0.832| 0.478| 0.188|
of-the-art methods. The details of the F1 scores are listed in Table 4.

### 5.6 Results on networks without ground truth community

We also conduct experiments on three real-world networks without a ground truth community. Figure 4b gives the experimental results on three such networks: HepPh, StanfordWeb, and Slashdot. Because LinLog+Motif takes too long to produce results on StanfordWeb and Slashdot, we use empty bars to indicate these. We detect local communities in the HepPh, StanfordWeb, and Slashdot networks based on Table 4 Performance of methods on 5 public realworld networks (MC denotes motif conductance).

| Method            | FloridaBay | E.Coli | HepPh | StanfordWeb | Slashdot |
|-------------------|------------|--------|-------|-------------|----------|
| HeatKernel        | 0.1        | 0.29   | 0.981 | 0.086       | 0.993    |
| LEMON             | 0.667      | 0.273  | 0.983 | 0.960       | 0.567    |
| PPR               | 0.07       | 0.42   | 0.984 | 0.133       | 0.999    |
| SSE               | 0.3        | 0.14   | 0.507 | 0.031       | 0.561    |
| LinLog+Motif      | 0.143      | 0.194  | 0.172 | –           | –        |
| MAPPR             | 0.578      | 0.9    | 0.226 | 0.916       | 0.606    |
| TECTONIC          | 0.286      | 0.364  | 0.834 | 0.778       | 0.407    |
| LCD-Motif         | **0.735**  | **0.941** | 0.259 | 0.337       | **0.391** |

**Fig. 4** Performance of methods on realworld networks.

These networks contain many instances of network motifs. LCD-Motif finds the local community with the lowest motif conductance value, which means the highest performance value. Because baseline methods find communities at the level of individual nodes and edges, they cannot find the community correctly. It is obvious that expanding seed nodes with three-node network motifs will form more tight-knit communities. For example, dangling nodes connected by one edge to a community are ignored by the motif-based method, whereas such a node would increase the motif conductance of the community detected by edge-based methods. And LCD-Motif can find a community with a lower motif conductance compared to other methods.

### 5.7 Parameter selection

The random step $k$ and subspace dimension $l$ are the key parameters in the local motif spectral clustering algorithm. We conduct parameter sensitivity studies for these two parameters on the real-world datasets.

**Subspace dimension.** To study the parameter subspace dimension $l$, we fix the random walk step to be 3, and vary the number of dimension $l$ from 1 to 15. Figure 5a shows that changing the dimension $l$ causes some performance fluctuations. Choosing a larger dimension is undesirable because it would increase the computational cost of generating local motif spectra. In this paper, we fix $l = 2$ because the experiment suggests that setting $l = 2$ can statistically achieve low motif conductances.

**Random walk step.** To investigate how the step of the random walk affects the algorithm performance, we fix dimension $l$ to be 2, and vary random walk step $k$ from 1 to 15. Figure 5b shows that changing the random walk step $k$ causes some performance fluctuations, and a 3-step random walk yields the algorithm’s full potential. Throughout the paper, we therefore fix the random walk step $k = 3$ for the datasets.

### 5.8 Which motif should be selected?

In this section, we will introduce which motif is to be
selected to reveal higher-order organization of the network. Since it is not clear whether the network is organized based on a given motif, we use every motif in Fig. 1 to detect communities in the network, in order to look for most useful motif. For the HepPh citation network, we only analyze motifs $M_5$, $M_8$, $M_9$, and $M_{10}$, which have no bidirectional or loopy interaction patterns (other motifs have either bidirectional or loopy interaction patterns), because it is impossible in practice that two papers cite each other mutually or that three papers are cited cyclically. In citation networks, motif $M_5$ implies that a paper $B$ cites another paper $C$, which is cited by a paper $A$, which cites both papers $B$ and $C$. Motif $M_8$ implies that a paper usually cites many other papers at the same time, which is consistent with the real situation. Motif $M_9$ implies that a paper which cites some other papers is in turn cited by many other papers. Motif $M_{10}$ implies that a paper is cited by many papers at the same time, which reflects the real situation of a highly cited paper. Figure 6 shows the sweep profile plot of motif conductance of different motifs in the HepPh citation network. All four motifs reveal certain information of higher-order structures, but motif $M_5$ reveals much less higher-order structures than other motifs. Communities based on motif $M_5$ have higher motif conductance. In other words, in detected communities there are less instances of motif $M_5$ than of the other three motifs. The downward spikes of the black, green, and red curves show that motifs $M_5$, $M_9$, and $M_{10}$ reveal richer higher-order modular structures in the HepPh citation network.

6 Conclusion

The problem of identifying a small community within large networks is an important one. In this paper, we propose a method for finding a small community by seeking a sparse vector in the span of local motif spectra where seeds are in its support vector. To overcome the drawbacks of the spectral motif clustering method, we propose a novel method to construct the local motif spectra based on the singular vector approximations drawn from short random walks. We demonstrate the effectiveness of our method on realworld datasets. As the experimental results show, our algorithm achieves the best performance among state-of-the-art methods. However, many other research questions are still to be addressed. First, one direction that could be taken is to parallelize LCD-Motif for handling larger networks, because it is highly parallelizable. Second, during the process of seed set expansion, we adopt the first low motif conductance community as the target community, which usually yields a high resemblance to the ground truth community. It would be interesting to look further into some larger low motif conductance communities and see if a hierarchical structure exists. Third, it is necessary to look further into how communities based on motifs change in temporal networks.

Acknowledgment

This work was supported by the National Social Science Foundation of China (No. 16ZDA055).
