Finding core-periphery structures with node influences

Xin Shen, Sarah Aliko, Yue Han, Jeremy I Skipper, and Chengbin Peng

Abstract—Detecting core-periphery structures is one of the outstanding issues in complex network analysis. Various algorithms can identify core nodes and periphery nodes. Recent advances found that many networks from real-world data can be better modeled with multiple pairs of core-periphery nodes. In this study, we propose to use an influence propagation process to detect multiple pairs of core-periphery nodes. In this framework, we assume each node can emit a certain amount of influence and propagate it through the network. Then we identify nodes with large influences as core nodes, and we utilize a maximum influence chain to construct a node-pairing network to determine core-periphery pairs. This approach can take node interactions into consideration and can reduce noises in finding pairs. Experiments on randomly generated networks and real-world networks confirm the efficiency and accuracy of our algorithm.

Index Terms—core-periphery structure, complex networks, node influences, computational modeling, unsupervised learning

1 Introduction

Development and analysis of algorithms for detecting local, mesoscale, and global structures in complex networks have significantly grown recently. Most of the research on the topic of investigating mesoscale network structure concerns a specific feature called community partitioning. In the study of community structure, the connection between nodes has a higher density within groups than between them [1]. Community detection has been applied to numerous areas, such as social networks [2], [3], [4], collaboration networks [2], information networks [2], biological networks [3], [5], [6], commerce systems [7]. A variety of methods exist to detect community structure including the hierarchical agglomeration algorithm [1], spin state optimization [8], spectral clustering [9], and non-negative matrix factorization [10].

The core-periphery structure is another type of mesoscale network structure, where a core is a group of densely interconnected nodes and a periphery is a group of sparsely interlinked nodes. For example, in the network shown in Fig.1, node 1, 2, and 3 can be considered as core nodes, and all the remaining nodes can be considered as periphery nodes. In Fig. 2, the adjacency matrix along with the graph shows another example, which is more close to reality because not every core-core or core-periphery pair is connected, making core-periphery detection more challenging. The main difference between core-periphery structure and community structure is that the former consists of densely connected core vertices and sparsely connected periphery vertices, whereas the latter is composed of densely connected nodes within groups and sparsely connected nodes between groups. The core-periphery structure has been mentioned in many studies [11], [12] and found in various fields. In biological networks, this concept has been applied to human brain functional networks [13], protein-protein interaction networks [14], [15] and metabolic networks [16]. There are also applications in networks related to social activities, such as voting networks [17], social networks [17], [18], [19], [20], collaboration networks [17], financial networks [19], and interbank networks [21], [22]. It can moreover be used to analyze airport networks [18] and transportation networks [17], [19], [23].

In complex networks, high degree nodes are not necessarily core nodes [24]. Therefore, researchers suggest developing specific core-periphery detection algorithms. Some researchers propose to find a single pair of core-periphery structure [25], [26], [27]. For example, Lee et al. proposed to use density-based and transport-based methods to detect core nodes [19]. Zelnio et al. proposed a detection method by utilizing power-law structures and degree centrality distributions [28].

When there are multiple pairs of core-periphery structures exist, the detection algorithms need to be extended accordingly. Recently, Kojaku et al. proposed to detect multiple non-overlapping groups of core-periphery structure [17], [18], [24]. Jeude et al. proposed to detect statistically significant core-periphery structures using multivariate hypergeometric distributions [29]. Xiang et al. proposed a method to detect both single and multiple pairs of core-periphery structure in complex networks and the overlapping nodes [30]. Liu et al. demonstrated the effectiveness of fractal analysis in revealing the properties of the core-periphery structure of complex networks [31]. Silva et al. also proposed

Sarah Aliko and Yue Han contributed equally to this study.

Chengbin Peng is the corresponding author.

Xin Shen and Yue Han are with College of Information Science and Engineering, Ningbo University, China.

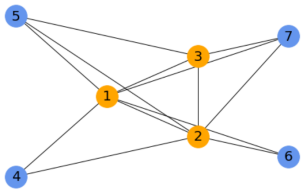
Sarah Aliko is with London Interdisciplinary Biosciences Consortium, UK, and Experimental Psychology, University College London, UK.

Jeremy I Skipper is with Experimental Psychology, University College London, UK.

Chengbin Peng is with College of Information Science and Engineering, Ningbo University, China, and Ningbo Institute of Industrial Technology, Chinese Academy of Sciences, China. (e-mail: pengchengbin@nbu.edu.cn)

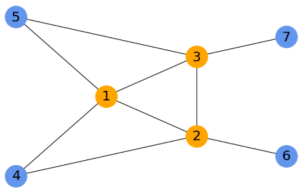
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3138436, IEEE Transactions on Network Science and Engineering

a core coefficient to evaluate the core-periphery structure of a metabolic network [16].



$$\left[\begin{array}{ccc|cccc} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

Fig. 1: The left figure shows a simple graph with a perfect core-periphery structure, where core nodes are drawn in yellow and periphery nodes are in blue, and the right matrix is the corresponding adjacency matrix.



$$\left[\begin{array}{ccc|cccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

Fig. 2: The left figure shows another graph with core-periphery structure, where core nodes are drawn in yellow and periphery nodes are in blue. The right matrix is the corresponding adjacency matrix.

In many scenarios, node connections of a network not only reflect the node similarities but also indicate node influences between connected nodes. Thus, a detection method based on influence propagation models is expected to have better performance than traditional approaches. Based on this intuition, in the present work, we propose an influence-based core-periphery detection method. The main contributions of this work are as follows.

(1) Our influence-based core-periphery detection approach can take influences from other nodes into account, and can reduce noises when looking for multiple pairs. Experimental results show that our approach achieves higher accuracy than other algorithms and exhibits better scaling to large networks. To the best of our knowledge, it is the first influence-based core-periphery detection algorithm.

(2) We prove that our approach can converge after several iterations. We also develop an error bound for our approach with appropriate assumptions. These analysis justifies the efficiency and correctness of our approach.

The rest of this manuscript is organized as follows. In Section 2, we review related work that contributes to finding core-periphery structures, including single and multiple pairs. In Section 3, a novel influence-based detection algorithm is proposed. We describe each step in detail and prove the convergence of this approach. In Section 4, a comparison with other algorithms on multiple types of networks is applied to demonstrate our method's performance and efficiency. We conclude this work in Section 5.

2 Related Work

2.1 Problem formulation

For a given graph, we can define $A \in \{0,1\}^{N\times N}$ as the adjacency matrix, where N equals the number of nodes. Therefore, we have

$$A_{ij} = \begin{cases} 1, & \text{if node } i \text{ is connected to } j, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We use an integer vector q to represent the pair identity of nodes. We use a binary vector c to indicate whether each node is a core node, where

$$c_i = \begin{cases} 1, & \text{if node } i \text{ is a core node in its pair,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Loosely speaking, core nodes are well-connected to peripheral nodes and tend to be central in a network, but peripheral nodes are not well-connected to each other. With various constraints, many traditional approaches try to maximize core-core edges and core-periphery edges [17], [24]:

$$s = \sum_{i,j} A_{ij}\Delta_q(q_i, q_j)\Delta_c(c_i, c_j) \tag{3}$$

where function $\Delta_q(\cdot,\cdot)$ returns one when the two inputs are the same and returns zero otherwise, and function $\Delta_c(\cdot,\cdot)$ returns an algorithm specific value between zero and one when either input is one and return zero when none is one. Function Δ_q makes sure that only edges connecting nodes from the same pair are considered, and it is always equal to one for single pair detection. When function Δ_c is one, at least one of the input nodes is a core node.

Constraints, such as multiplying a predefined core value with the core indicator of each node [17], or compromising with the expected value of random networks [24], are necessary for such maximization approaches, because otherwise, there is a trivial solution when all the nodes are core nodes and belong to the same pair. Rather than adopting a new constraint, in this work, we propose to use an influence-based approach to solve this issue.

2.2 Detecting core-periphery nodes

There are many methods for detecting core-periphery structures. Some can find only one pair in a network. For example, Lip et al. consider the core-periphery detection problem as a node bipartition problem, so that connections between core nodes are maximized and connections between periphery nodes are minimized [26]. Boyd et al. propose a similar notation but allow a continuous coreness score for each node, which can be computed via minimum residual singular value decomposition [27]. Some researchers utilize geodesic paths or low-rank approximation methods to develop detection algorithms [32]. Borgatti et al. propose to use the correlation between empirical networks and ideal network structures to evaluate the detection performance of such algorithms [25].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3138436, IEEE Transactions on Network Science and Engineering

2.3 Detecting multiple pairs of core-periphery nodes

Some other algorithms can find multiple pairs of core-periphery structures in a single network. For example, Kojaku et al. propose that there can be at least one block of nodes apart from the focal core-periphery structures [24]. They also propose a scalable algorithm based on the density of edges in a network to detect multiple non-overlapping core-periphery pairs in networks [18] and a method to detect core-periphery structures through computing a constant value along a core-periphery spectrum in [24]. Researchers also develop an algorithm to detect the core-periphery structure in multiple scales for a seaport network with nine hundred nodes, namely, with different parameter choices, the maximum pair size can vary from seventy nodes to all the nine hundred nodes [23]. Gu et al. develop a model-based approach to identify the core-periphery structure and applied it to functional magnetic resonance imaging data (fMRI) [13]. Battiston et al. propose a method to detect core-periphery structure based on Belief Propagation for learning through entropy maximization on both the stochastic block model and the degree-corrected stochastic block model [33]. Ma et al. propose an algorithm for detecting core-periphery structure based on a 3-tuple motif, which contains the patterns of edges as well as the property of nodes [34].

The approaches mentioned above are mostly based on edge densities between nodes and thus only take the nearest neighbors of each node or the average connections of nodes into account. However, in many real-world scenarios, nodes not directly connected with each other can also have interactions with each other. To address this gap, in this work, inspired by node ranking models [35], [36], [37], we propose to build an influence matrix to identify hidden relationships between all the node pairs and use this hidden relationship to find multiple pairs of core-periphery nodes. Theoretical analysis and experimental results justify our approach.

3 METHOD

3.1 Influence-based detection algorithm

By introducing node influences, we propose an influence-based core-periphery detection algorithm (ICPA). We assume each node can emit a certain amount of influences, and such influences can propagate to other nodes along edges. Without loss of generality, if node j exerts the most extensive influence on node i, node i belongs to the same pair as node j. Nodes with relatively large influences are considered core nodes.

In this work, we assume that each node has an influence vector, recording how it is influenced by other nodes. We use rows of a probability matrix $P \in [0,1]^{N\times N}$ to represent the influence vectors so that the row vector P_i is the influence vector of the ith node. In our approach, we also want a unique non-negative integer as the identity for each node, so we use integers from zero to $N-1$ for this purpose. We also define a max-influence indicator H of size $N \times 1$, where $H_i \in \{0,1,2,...,N-1\}$ represents the identity of the node by which node i is mostly influenced. We define $c \in \{0,1\}^N$ as an indicator for core nodes, where $c_i = 1$ if node i is a core node and $c_i = 0$ if node i is a periphery node.

In the beginning, because no influence information is available, we initialize the matrix P as a zero matrix.

Step (1): We update matrix P iteratively from $k = 0$, until P converges. In each iteration, the algorithm performs two steps.

Step (1.1): We assume each node emits a certain volume of influences which is a linear function with respect to the received self influence, such that we set the diagonal entries of P as follows and off-diagonal entries are not changed:

$$diag(P^k) = \beta + \alpha \times diag(P^{k-1}), \tag{4}$$

where $\alpha \in [0,1]$ and $\beta \in (0,+\inf]$ are both scalars.

Step (1.2): Each node sums over all the influence vectors from its neighbors using AP^k, and normalizes its own vectors so that the summation of each row vector is 1:

$$P^{k+1} = norm(AP^k), \tag{5}$$

$$k = k+1. \tag{6}$$

In short, the update rule of the ith influence vector is

$$P_i^{(k)} = \frac{1}{d_i}\sum_j \frac{A_{ij}[P_j^{(k-1)} - (1-\alpha)E_j. \times P_j^{(k-1)} + \beta E_j]}{1-(1-\alpha)P_{jj}^{(k-1)} + \beta} \tag{7}$$

where $d_i = \sum_j A_{ij}$ represents the degree of the ith node, and E_j represents the jth row of an identity matrix in which only diagonal entries are one and other entries are zeros. $.\times$ is the element-wise product for vectors. Therefore, $P_j^{(k-1)} - (1-\alpha)E_j. \times P_j^{(k-1)} + \beta E_j$ corresponds to Eq. (4). The normalization $1/[1-(1-\alpha)P_{jj}^{(k-1)} + \beta]$ corresponds to the $norm()$ function in Eq. (5) as $\sum_m P_{jm}^{(k-1)} = 1$ and $\sum_m E_{jm} = 1$.

In practice, if we choose $\alpha = 1$, this step can be considered as a typical random walk with restart model [38], [39], because Eq. (7) can be simplified as

$$P_i^{(k)} = \frac{1}{d_i}\sum_j \frac{A_{ij}[P_j^{(k-1)} + \beta E_j]}{1+\beta}. \tag{8}$$

The algorithm for obtaining the influence vectors for nodes in networks is shown in Algorithm 1.

Algorithm 1 Computing Influence Vectors (Step 1)

Input:
A: an $N \times N$ Adjacency matrix of the input network
Output:
P: Influence vectors for nodes of the input networks

```
P ← zeros(N, N)
for k = 0;k <MAX_ITER;k++ do
    for i = 0;i <N;i++ do
        Update P_i according to Eq. (7)
    end for
end for
return P
```

Intuitively, this step means that, in each iteration, each node accumulates influence vectors from its neighbors plus its own influence and then propagates the real influence to its neighbors. It can be proved that the power matrix P can converge to P^* after a few iterations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3138436, IEEE Transactions on Network Science and Engineering

Step (2): We compute vector H. For node i, we choose node H_i such that it has the largest influence to node i.

$$H_i = \arg\max_{j,j\neq i} P^*_{i,j} \tag{9}$$

We can consider H as a representation of a directed graph where each node is only connected to the node with the largest influence on it. In this directed graph, we consider each disconnected component as a core-periphery pair so that nodes and their corresponding influencers are within the same pair. Similar to non-maximum suppression in deep learning, this maximization-based approach can suppress noises when finding node pairs. This step can reduce noises in finding pairs. Although we connect each node to its top influencer by default when building the directed graph, it is also possible to connect each node with several top influencers.

Step (3): We consider the column sum of P^* as indicators of node influences to others, and thus, nodes with large influences are core nodes. We define the core score of node i as

$$cs_i = \sum_j P^*_{ji}. \tag{10}$$

Then, we choose a minimum number of nodes as core nodes so that the total influences of those nodes are above a certain ratio defined by a hyperparameter γ:

$$\begin{aligned} &\min \sum_i c_i \\ &s.t. \frac{\sum_i c_i cs_i}{\sum_i cs_i} > \gamma. \end{aligned} \tag{11}$$

This optimization can be easily solved by sorting node core scores in descending order and adding nodes to the set of core nodes one by one in this order.

The algorithm of obtaining core-periphery pairs in networks is therefore shown in Algorithm 2.

Algorithm 2 Computing core-periphery pairs (Step 2 and 3)

Input:
P: Influence vectors for nodes
Output:
q: Pair indicator, node i belongs to q_ith pair
c: Core indicator, node i is a core node if $c_i = 1$; vice versa
% Step(2)
Compute matrix H according to Eq. (9)
Build an adjacency matrix $A' \in \{0,1\}^{N\times N}$, where $A'(H_i, i) = 1$, $i \in \{0, 1, 2, ..., N-1\}$ and all the other entries are zero
For all the connected nodes in A', set a same and unique value to their corresponding q_i
% Step(3)
Compute cs_i according to Eq. (10) for all the nodes
Compute c_i according to Eq. (11) for all the nodes
return q, c

If A is a sparse matrix, the time complexity of Step 1 and 2 is $O(M)$, where M is the number of edges. If we use merge sort, the time complexity of Step 3 is $O(N \log N)$. Thus, if the average node degree is constant as N is increasing, the time complexity of the whole approach is $O(N \log N)$. On the other hand, if the average degree increases along with N, the time complexity is $O(M)$.

3.2 Theoretical analysis

In this section, we prove that our approach can converge after several iterations. For Step (1), we can have the following convergence theorem.

Theorem 1. The power matrix P can converge to P^* if it is updated according to Eq. (8).

Proof 1.
Without loss of generality, we define P^*_i as the optimal solution of P_i. We also define the total error at iteration k to be:

$$Err(k) = \sum_i |P^{(k)}_i - P^*_i|. \tag{12}$$

Since P^*_i is the optimal solution, it satisfies the following equation:

$$P^*_i = \frac{1}{d_i(1+\beta)} \sum_j A_{ij}[P^*_j + \beta E_j]. \tag{13}$$

Therefore, by Eq. (8) and the equation above, we have

$$\begin{aligned} &P^{(k)}_i - P^*_i \\ =&\frac{1}{d_i(1+\beta)} \sum_j A_{ij}[P^{(k-1)}_j + \beta E_j] \\ &- \frac{1}{d_i(1+\beta)} \sum_j A_{ij}[P^*_j + \beta E_j] \\ =&\frac{1}{d_i(1+\beta)} \sum_j A_{ij}[P^{(k-1)}_j - P^*_j] \end{aligned} \tag{14}$$

By Triangle Inequality, we can have

$$|P^{(k)}_i - P^*_i| \leqslant \frac{1}{d_i(1+\beta)} \sum_j A_{ij}|P^{(k-1)}_j - P^*_j|. \tag{15}$$

Now we can sum over all the nodes to get the total error as follows:

$$\begin{aligned} &Err(k) \\ =&\sum_i |P^{(k)}_i - P^*_i| \\ \leqslant&\sum_i \frac{1}{d_i(1+\beta)} \sum_j A_{ij}|P^{(k-1)}_j - P^*_j| \\ =&\frac{1}{1+\beta} \sum_j \sum_i \frac{A_{ij}}{d_i}|P^{(k-1)}_j - P^*_j| \\ =&\frac{1}{1+\beta} \sum_j |P^{(k-1)}_j - P^*_j| \end{aligned} \tag{16}$$

$$= \frac{1}{1+\beta} Err(k-1) \tag{17}$$

Therefore, as long as $\beta > 0$, after each iteration, the error can be decreased, so that the algorithm can converge.

For Step (2) and (3), all the computations can be done within a definite number of iterations. Thus, the whole approach can converge.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3138436, IEEE Transactions on Network Science and Engineering

In the following analysis, we further justify the reliability of the computed results. For simplicity, similar to stochastic block models in community detection [40], we can define a stochastic model with core-periphery structure as follows.

Definition 1. Let A be the adjacency matrix for a graph with N nodes and K core-periphery pairs. Each pair has the same number of nodes, and the portion of core nodes in each pair is δ. All the edges are randomly connected but following constraints:
1. There are nc edges for each core node randomly connecting to other core nodes in the same pair;
2. There are np edges for each core node randomly connecting to other nodes in a different pair;
3. Connection probability between core-periphery nodes and core-core nodes in the same pair are the same;
4. Connection probability between all the other pairs of nodes are the same.

Therefore, the expected value and the variance for each connection is as follows.

$$\mathbb{E}[A_{ij}] = \begin{cases} \frac{nc\times K}{N}, & \text{if at least one of node } i \text{ and } j \\ & \text{is a core node, and both nodes} \\ & \text{are from the same pair,} \\ \frac{np\times K}{N}, & \text{otherwise.} \end{cases} \tag{18}$$

$$\mathrm{Var}[A_{ij}] = \mathbb{E}[A_{ij}](1-\mathbb{E}[A_{ij}]). \tag{19}$$

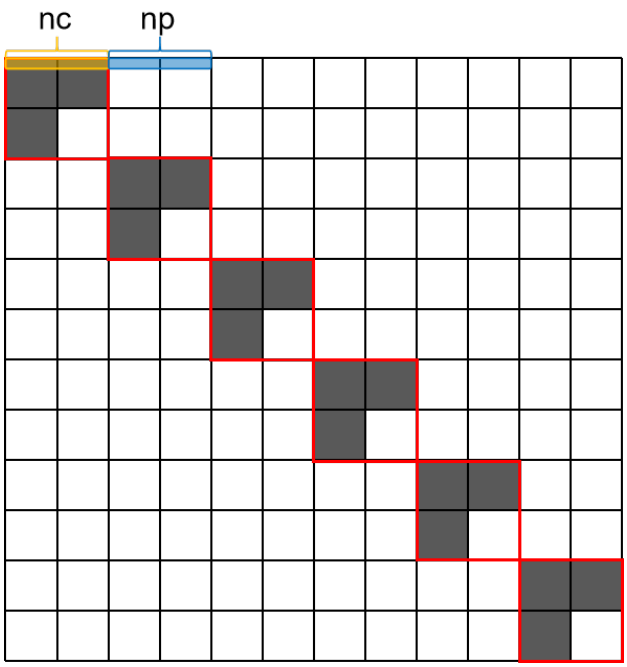


Fig. 3: Adjacency matrix of a network with multiple core-periphery pairs

Without loss of generality, Fig.3 shows an example network with multiple core-periphery pairs following Definition 1. In this figure, edges connecting nodes in the same pair are in the same red box and there are totally six pairs. The number of edges of a core node corresponding to the edges in the yellow box is nc. The number of edges of a periphery node corresponding to the edges in the blue block is np. Areas with the same color representing the same connecting probability.

Theorem 2. For a network generated by Definition 1 and core scores defined by Eq. (10), using the first order Taylor series approximation, core nodes can have larger core scores than periphery nodes with probability at least $1-\frac{(1+\delta)(N-Knp)(nc+2Knp)}{2N(1-\delta)^2(nc-np)^2}$.

Proof 2. First, we define a diagonal matrix D, in which $D_{ii} = d_i$. From Eq. (13), we have

$$(1+\beta)DP^* = A[P^* + \beta E]. \tag{20}$$

Consequently, we can have

$$[(1+\beta)D - A]P^* = \beta AE. \tag{21}$$

(22)

Therefore, as β is larger than zero, we can approximate P^* by the first-order Taylor polynomial:

$$P^* = [(1+\beta)D - A]^{-1}\beta A \tag{23}$$

$$= \beta \sum_{\kappa=0}^{\inf} [(1+\beta)D]^{-\kappa-1} A^{\kappa+1} \tag{24}$$

$$\approx \beta[(1+\beta)D]^{-1}A. \tag{25}$$

Without loss of generality, we can consider two nodes in a same pair, such that node ip is a periphery node and node ic is a core node. Then, for random networks generated from Definition 1, the expected difference of core scores of the two nodes defined by Eq. (10) can be approximated as

$$\mathbb{E}[cs_{ic} - cs_{ip}] \tag{26}$$

$$= \mathbb{E}[\sum_i P^*_{i,ic} - \sum_i P^*_{i,ip}] \tag{27}$$

$$\approx \beta(1+\beta)^{-1} \sum_i \frac{1}{d_i} \mathbb{E}[A_{i,ic} - A_{i,ip}] \tag{28}$$

$$= \beta(1+\beta)^{-1} \frac{N(1-\delta)}{K} \frac{1}{nc\delta + np(K-\delta)} \times [\frac{ncK - npK}{N}] \tag{29}$$

$$= \beta(1+\beta)^{-1} \frac{(1-\delta)(nc-np)}{nc\delta + np(K-\delta)} \tag{30}$$

where Eq. (29) the expectation of the summation of two random variables is the summation of the expectation of two random variables, and the expectations of all the other entries in $A_{:,ic}$ and $A_{:,ip}$ can be canceled except those representing connection to the periphery nodes in the same pair. By definition, if node i is a core node, $d_i = nc + np(K-1)$, and if node i is a periphery node, $d_i = nc\delta + np(K-\delta)$. We also define an auxiliary variable

$\sigma = \frac{nc+np(K-1)}{nc\delta+np(K-\delta)}$. Thus, the variance of the core score differences of node ip and ic is

$$\begin{aligned}
&\mathrm{Var}[cs_{ic} - cs_{ip}] && (31)\\
=&\mathrm{Var}[\sum_i P^*_{i,ic} - \sum_i P^*_{i,ip}] && (32)\\
\approx&\beta^2(1+\beta)^{-2}\sum_i \frac{1}{d_i^2}\mathrm{Var}[A_{i,ic} - A_{i,ip}] && (33)\\
=&\beta^2(1+\beta)^{-2}\sum_i \frac{1}{d_i^2}[\mathrm{Var}(A_{i,ic}) + \mathrm{Var}(A_{i,ip})] && (34)\\
=&\beta^2(1+\beta)^{-2}\sum_i \frac{1}{d_i^2}\{\mathbb{E}[A_{i,ic}](1-\mathbb{E}[A_{i,ic}])\\
&+\mathbb{E}[A_{i,ip}](1-\mathbb{E}[A_{i,ip}])\} && (35)\\
=&\beta^2(1+\beta)^{-2}\frac{1}{N\sigma^2[nc\delta+np(K-\delta)]^2}\\
&\times\{[Nnc-K(nc)^2][\sigma^2(1-\delta)+2\delta]+[Nnp\\
&-K(np)^2][\sigma^2(2K-1)(1-\delta)+2\delta(K-1)]\}. && (36)
\end{aligned}$$

By Chebyshev's inequality,

$$\begin{aligned}
&\Pr(cs_{ic} - cs_{ip} \geqslant 0) && (37)\\
=&1-\frac{1}{2}\Pr(|cs_{ic} - cs_{ip} - \mathbb{E}[cs_{ic} - cs_{ip}]|\\
&\geqslant \mathbb{E}[cs_{ic} - cs_{ip}]) && (38)\\
\geqslant&1-\frac{\mathrm{Var}[cs_{ic} - cs_{ip}]}{2(\mathbb{E}[cs_{ic} - cs_{ip}])^2} && (39)\\
\geqslant&1-\frac{(1+\delta)(N-Knp)(nc+2Knp)}{2N(1-\delta)^2(nc-np)^2}. && (40)
\end{aligned}$$

Theorem 3. For a network generated by Definition 1, using the first order Taylor series approximation, node i and node H_i (defined by Eq. (9)) are in the same pair with probability at least $1-\frac{(N-ncK)(2-\delta)(nc\delta+np)}{2\delta^2K(nc-np)^2(1-\delta)^2}$.

Proof 3. Without loss of generality, we can consider two nodes, such that id is a node from a different pair with respect to node i, and is is a node from the same pair as node i.

By definition, node i is a core node with probability δ. In this case, nodes in the same pair are all have dense connections with i, and so $\mathbb{E}[A_{i,is} - A_{i,id}] = \frac{ncK-npK}{N}$ by Eq. (18). Similarly, node i is a periphery node with probability $1-\delta$. In this case, if is is a core node with probability δ, $\mathbb{E}[A_{i,is} - A_{i,id}] = \frac{ncK-npK}{N}$, and if is is a periphery node, $\mathbb{E}[A_{i,is} - A_{i,id}] = 0$.

Consequently, the expected difference is

$$\begin{aligned}
&\mathbb{E}[P^*_{i,is} - P^*_{i,id}] && (41)\\
\approx&\beta(1+\beta)^{-1}\frac{1}{d_i}\mathbb{E}[A_{i,is} - A_{i,id}] && (42)\\
=&\beta(1+\beta)^{-1}[\delta\frac{1}{nc+np(K-1)}\frac{(ncK-npK)}{N} && (43)\\
&+(1-\delta)\frac{1}{nc\delta+np(K-\delta)}\frac{\delta(ncK-npK)}{N}] && (44)\\
=&\beta(1+\beta)^{-1}\frac{\delta[\sigma(1-\delta)+1]}{\sigma[nc\delta+np(K-\delta)]}\frac{ncK-npK}{N}. && (45)
\end{aligned}$$

The expected variance is

$$\begin{aligned}
&\mathrm{Var}[P^*_{i,is} - P^*_{i,id}] && (46)\\
\approx&\beta^2(1+\beta)^{-2}\frac{1}{d_i^2}\mathrm{Var}[A_{i,is} - A_{i,id}] && (47)\\
=&\beta^2(1+\beta)^{-2}\frac{1}{d_i^2}[\mathrm{Var}(A_{i,is}) + \mathrm{Var}(A_{i,id})] && (48)\\
=&\beta^2(1+\beta)^{-2}\frac{1}{d_i^2}\{\mathbb{E}[A_{i,is}](1-\mathbb{E}[A_{i,is}])\\
&+\mathbb{E}[A_{i,id}](1-\mathbb{E}[A_{i,id}])\} && (49)\\
=&\beta^2(1+\beta)^{-2}\frac{1}{N^2\sigma^2[nc\delta+np(K-\delta)]^2}\\
&\times\{ncK\delta(N-ncK)(1+\sigma^2-\sigma^2\delta)\\
&+npK(N-npK)[\delta+(\sigma^2-\sigma^2\delta)(2-\delta)]\}. && (50)
\end{aligned}$$

By Chebyshev's inequality, we have

$$\begin{aligned}
&\Pr(P^*_{i,is} - P^*_{i,id} \geqslant 0) && (51)\\
=&1-\frac{1}{2}\Pr(|P^*_{i,is} - P^*_{i,id} - \mathbb{E}[P^*_{i,is} - P^*_{i,id}]|\\
&\geqslant \mathbb{E}[P^*_{i,is} - P^*_{i,id}]) && (52)\\
\geqslant&1-\frac{\mathrm{Var}[P^*_{i,is} - P^*_{i,id}]}{2(\mathbb{E}[P^*_{i,is} - P^*_{i,id}])^2} && (53)\\
\geqslant&1-\frac{(N-ncK)(2-\delta)(nc\delta+np)}{2\delta^2K(nc-np)^2(1-\delta)^2}. && (54)
\end{aligned}$$

3.3 Extensions for Weighted Networks

For weighted networks, we can define a weight matrix W, where W_{ij} is the weight of the edge between node i and j. If we use $\hat{}$ to represent the variables in the update rule for weighted networks, the simplified version, Eq. (8) can be rewritten as follows:

$$\hat{P}_i^{(k)} = \frac{1}{\hat{d}_i}\sum_j \frac{A_{ij}W_{ij}[\hat{P}_j^{(k-1)} + \beta E_j]}{1+\beta}, \quad (55)$$

where $\hat{d}_i = \sum_j A_{ij}W_{ij}$ represents the degree of the ith node.

Similarly, in the proof of Theorem 1, if we replace A_{ij} and d_i with $A_{ij}W_{ij}$ and $\hat{d}_i$ respectively, that theorem can still work. Therefore, our approach can also converge for weighted networks.

We can also analyze the impact of edge weights by comparing the differences between Eq. (8) and (55, as follows:

$$\begin{aligned}
&|\hat{P}_i^{(k)} - P_i^{(k)}| && (56)\\
=&|\sum_j \frac{A_{ij}[\frac{W_{ij}}{\hat{d}_i}\hat{P}_j^{(k-1)} - \frac{1}{d_i}P_j^{(k-1)}]}{1+\beta}| && (57)\\
\leqslant&\frac{\sum_j A_{ij}|\frac{W_{ij}}{\hat{d}_i}\hat{P}_j^{(k-1)} - \frac{1}{d_i}P_j^{(k-1)}|}{1+\beta} && (58)\\
\leqslant&\frac{\sum_j A_{ij}(\hat{P}_j^{(k-1)} + P_j^{(k-1)})|\frac{W_{ij}}{\hat{d}_i} - \frac{1}{d_i}|}{1+\beta}. && (59)
\end{aligned}$$

If W_{ij} with $i,j \in \{0,1,2,...,N-1\}$ are generated from a random distribution of variance δ_W, the expected value of the normalized edge weight $\frac{W_{ij}}{\hat{d}_i}$ is $\frac{1}{d_i}$. If we use the second

norm for $|\cdots|$, $|\frac{W_{ij}}{\hat{d}_i} - \frac{1}{d_i}|$ is actually the variance of the normalized edge weight, namely, equal to $\frac{\delta_W}{\hat{d}_i}$. Therefore, for the same adjacency matrix, the differences between weighted networks and unweighted networks are proportional to the variance of normalized edge weights. This phenomenon is intuitively reasonable as unweighted networks can be considered a special case of weighted networks with edge weight equal to one.

4 Experiments

In this section, we compare our approach with other algorithms, where BE represents $BorgattiEverett$ algorithm [25], LC represents $LapCore$ algorithm [32], LRC represents $LowRankCore$ [32], MIN represents $MINRES$ algorithm [26], and KM represents $KMconfig$ algorithm [24]. We make comparisons using random networks and real-world networks. For our approach, we choose to set the amount of the emitted influences and that of the received influences the same, namely, $\beta = 1$. We also set γ to be 0.5 by default, meaning that core nodes take up at least half of the total influence.

For networks with ground-truth information, especially for randomly generated networks, we can use normalized mutual information (NMI) [41], [42] for evaluation. NMI allows us to measure the agreement of a given partition ω and the ground truth partition $\hat{\omega}$, and it is defined as follows

$$NMI(\omega, \hat{\omega}) = \frac{I(\omega, \hat{\omega})}{[H(\omega) + H(\hat{\omega})]/2}, \tag{60}$$

where I is mutual information, H is entropy. Based on the basic NMI definition above, we develop an NMI metric for the core-periphery structure

$$NMI_{cp} = \frac{1}{2}[NMI(c, \hat{c}) + NMI(r, \hat{r})], \tag{61}$$

where $NMI(r, \hat{r})$ indicates the accuracy for pair labels, and $NMI(c, \hat{c})$ indicates the accuracy of core labels. As the value of NMI_{cp} takes both pairs and cores information into account, the larger the value of the NMI_{cp}, the better the detection algorithm.

4.1 Case study I: Detection for random networks

We construct the random networks with several parameters: p_{cc}, p_{cp}, and p_{pp}, representing the connection probability between core nodes, that between core and periphery nodes, and that between periphery nodes. Typically, core-periphery structure arises when $p_{cc} > p_{cp} > p_{pp}$. Therefore, for simplicity, we can define $p_{cc} = \kappa^2 p$, $p_{cp} = \kappa p$, and $p_{pp} = p$, where p represents the connection probability and κ is a constant. In this experiment, we use $p = 0.25$. We use $\kappa = 1.3$, 1.5 and 1.8 respectively. Pair size varies between $N/30$ and $N/20$ where N is the network size. Half of the nodes in each pair are core nodes. We compare the performance with different network sizes as well, from 1000 nodes to 10000 nodes. In each setting, we generate three random networks and record the average performance for each algorithm.

The accuracy of different algorithms are shown in Table 1, 2, and 3. In each row, the best performance in terms of NMI_{cp} is highlighted. From these tables, we can see that ICPA performs significantly better than many others because it can find multiple pairs. When comparing with the KM algorithm, which can also identify multiple pairs, our approach is still more accurate. In Fig. 4, we analyze the sensitivity of accuracy with respect to γ. The horizontal axis representing the network size and the vertical axis representing the accuracy measured by NMI_{cp}. Lines with the same color indicate that these results are from networks generated with the same κ. From this figure, we can see that, for a given network size and generating parameter, although there are small variances in NMI_{cp} between different choices of γ, our approach always performs better than other algorithms comparing with corresponding cases in Tables 1, 2, and 3.

We also measure the run time of these algorithms using networks generated by different parameters, as shown in Figs. 5, 6, and 7. From these figures, we can see that although our approach is not fast, the growth rate is similar to most of the other approaches.

TABLE 1: NMI_{cp} of different algorithms when $\kappa = 1.3$

N	NMI_{cp} of $\kappa = 1.3$					
	ICPA	BE	LC	LRC	MIN	KM
1000	**0.5089**	0.0270	0.0001	0.0050	0.0062	0.4900
2000	**0.5177**	0.0250	0.0001	0.0002	0.0028	0.4960
3000	**0.5149**	0.0250	0.0001	0.0002	0.0023	0.4910
4000	**0.5174**	0.0220	0.0002	0.0003	0.0014	0.4970
5000	**0.5118**	0.0230	0.0002	0.0001	0.0013	0.5000
6000	**0.5115**	0.0190	0.0002	0.0002	0.0011	0.4930
7000	**0.5207**	0.0230	0.0003	0.0001	0.0010	0.4970
8000	**0.5232**	0.0230	0.0009	0.0001	0.0012	0.5220
9000	**0.5173**	0.0210	0.0002	0.0002	0.1470	0.4990
10000	**0.5270**	0.0250	0.0002	0.0001	0.1070	0.5000

TABLE 2: NMI_{cp} of different algorithms when $\kappa = 1.5$

N	NMI_{cp} of $\kappa = 1.5$					
	ICPA	BE	LC	LRC	MIN	KM
1000	**0.5697**	0.0620	0.0004	0.0042	0.0096	0.4960
2000	**0.5857**	0.0710	0.0012	0.0010	0.0048	0.4940
3000	**0.5663**	0.0690	0.0007	0.0002	0.0038	0.5020
4000	**0.5608**	0.0680	0.0008	0.0003	0.0027	0.4990
5000	**0.5772**	0.0750	0.0010	0.0001	0.0024	0.5050
6000	**0.5668**	0.0640	0.0009	0.0001	0.0020	0.5010
7000	**0.5716**	0.0670	0.0013	0.0001	0.0018	0.5030
8000	**0.5866**	0.0570	0.0010	0.0001	0.0016	0.5040
9000	**0.5828**	0.0730	0.0007	0.0001	0.0013	0.5040
10000	**0.5881**	0.0620	0.0008	0.0001	0.0013	0.5040

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3138436, IEEE Transactions on Network Science and Engineering

TABLE 3: NMI_{cp} of different algorithms when $\kappa = 1.8$

N	NMI_{cp} of $\kappa = 1.8$					
	ICPA	BE	LC	LRC	MIN	KM
1000	**0.6402**	0.0690	0.0045	0.0047	0.0125	0.5230
2000	**0.6440**	0.0640	0.0055	0.0008	0.0051	0.5260
3000	**0.6371**	0.0630	0.0055	0.0004	0.0047	0.5250
4000	**0.6438**	0.0620	0.0045	0.0001	0.0035	0.5230
5000	**0.6520**	0.0680	0.0034	0.0001	0.0026	0.5260
6000	**0.6543**	0.0660	0.0050	0.0003	0.0024	0.5190
7000	**0.6480**	0.0580	0.0038	0.0003	0.0018	0.5180
8000	**0.6405**	0.0650	0.0022	0.0001	0.0017	0.5200
9000	**0.6560**	0.0610	0.0051	0.0001	0.0014	0.5250
10000	**0.6591**	0.0610	0.0031	0.0001	0.0013	0.5160

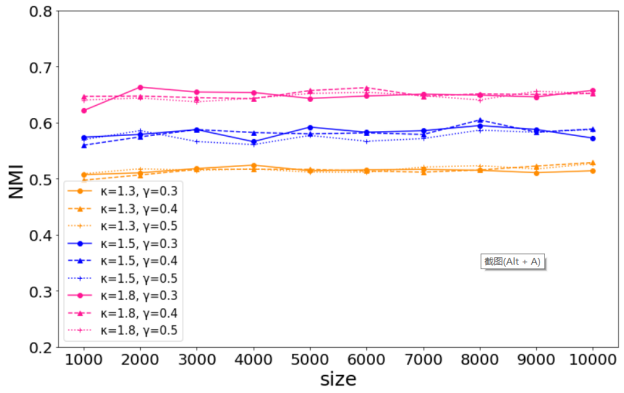


Fig. 4: NMI_{cp} of ICPA with different choices of γ under different input networks determined by κ

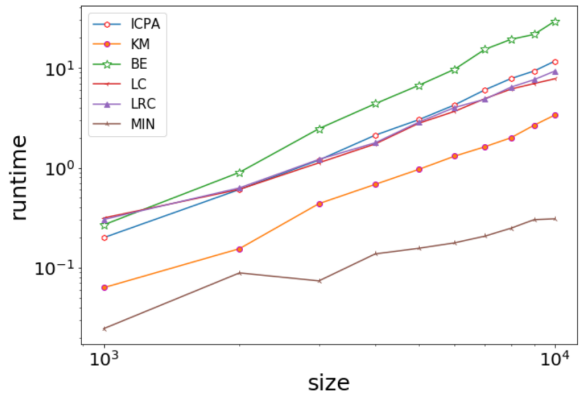


Fig. 5: Runtime of different algorithms with $\kappa = 1.3$

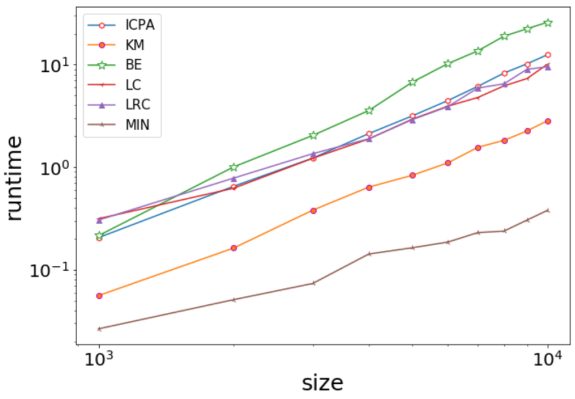


Fig. 6: Runtime of different algorithms with $\kappa = 1.5$

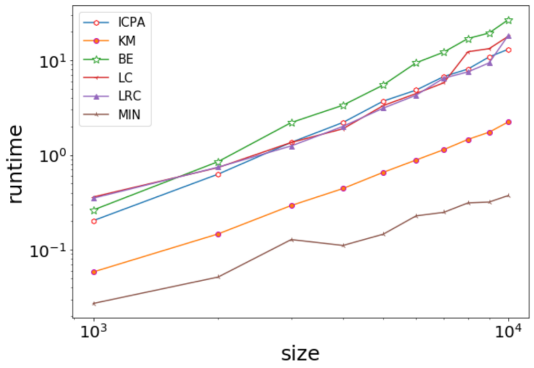


Fig. 7: Runtime of different algorithms with $\kappa = 1.8$

4.2 Case study II: Detection in scientist co-authorship network

In this section, we test our algorithm with a real-world network: the co-authorship network. We use the scientist co-authorship network from 2010 (NNS2010) [32] with 552 nodes. This network is expected to have a clear core-periphery structure: each node represents one scientist, and if two scientists have co-authorship, there is a connection between the two nodes. The core nodes represent the scientists who have large influences in the research community, surrounded by periphery nodes representing loosely connected and less influential scientists. The connection has a weight based on the degree of the cooperation, which can be considered as a factor affecting the influence propagation.

When comparing with algorithms detecting only a single core-periphery pair, we show the names, scores, and pair identities calculated by ICPA in Table4 and by other algorithms in Table5. The node rankings are generally similar. From Fig.8a, Fig.8c, and Fig.8e to Fig.8h, we can find that the core nodes in yellow color identified by our approach are more evenly distributed and better connected with periphery nodes when comparing with others, and thus our approach is better.

When comparing with multiple pair detectors as shown in Table6, we consider the pairs with two representative nodes M.E.J. Newman and A.L. Barabasi, as an example. In each pair, nodes are sorted in descending order of the core score, and they are similar. From Fig.8b to Fig.8d in which different colors represent different pairs, we can see that our approach can find better clusters as core-periphery pairs identified by our approach are more meaningful.

4.3 Case study III: Detection for Email networks

This section tests our algorithm using another real-world network: the Email network [43]. Nodes in the network represent email accounts, and when two accounts exchange messages, there is a connection between two corresponding nodes. Fig. 9 shows the comparison in detecting core nodes and core-periphery pairs. In Fig. 9b and 9d, nodes in the same pair are drawn using the same color. We can find that comparing algorithms detecting multiple core-periphery

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3138436, IEEE Transactions on Network Science and Engineering

TABLE 4: The top 30 names, scores, and pairs that nodes belong to according to the top 30 nodes calculated by ICPA

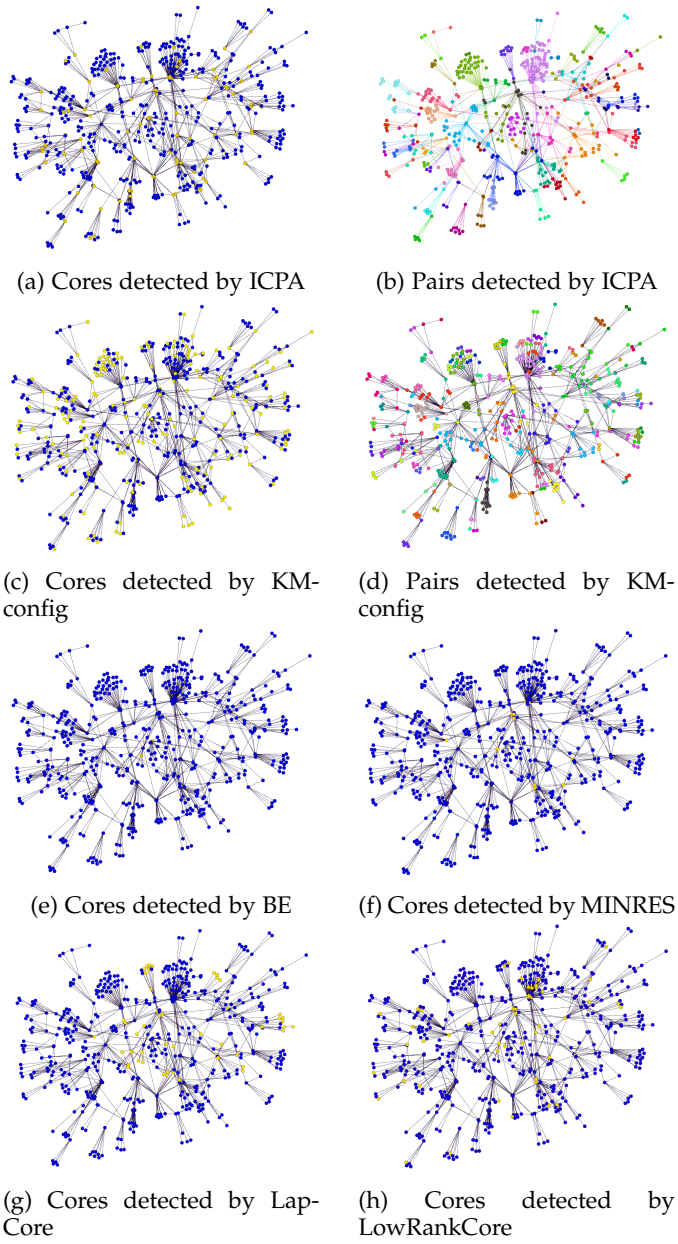
Author's name	Score	Pair-id
Newman,M.E.J.	0.0183	8
Barabasi,A.L.	0.0168	11
Jeong,H.	0.0111	11
Boccaletti,S.	0.0108	16
Sole,R.V.	0.0100	49
Vicsek,T.	0.0087	11
Kurths,J.	0.0083	36
Latora,V.	0.0083	32
Pastor-Satorras,R.	0.0082	24
Moreno,Y.	0.0081	2
Amaral,L.A.N.	0.0079	47
Arenas,A.	0.0076	21
Vespignani,A.	0.0076	24
Kahng,B.	0.0075	7
Porter,M.A.	0.0075	50
Kertesz,J.	0.0073	1
Diaz-Guilera,A.	0.0072	21
Stauffer,D.	0.0066	44
Caldarelli,G.	0.0064	31
Bornholdt,S.	0.0063	17
Guimera,R.	0.0063	47
Hu,G.	0.0060	22
Tomkins,A.S.	0.0060	13
Havlin,S	0.0059	37
Oltvai,Z.N.	0.0056	11
Koch,C.	0.0055	23
Sokolov,I.M	0.0053	54
Kleinberg,J.M	0.0053	4
Barthelemy,M	0.0053	47
Kaski,K	0.0052	1

TABLE 5: The ten names of co-authors who belong to the same pairs as Newman,M.E.J. and Barabasi,A.L. are found by single-pair algorithms.

Single-pair algorithms			
BE	**LapCore**	**MINRES**	**LowRankCore**
Newman,M.E.J.	Newman,M.E.J.	Newman,M.E.J.	Newman,M.E.J.
Barabasi,A.L.	Barabasi,A.L.	Barabasi,A.L.	Barabasi,A.L.
Boccaletti,S.	Vicsek,T.	Vicsek,T.	Pacheco,A.F.
Vicsek,T.	Kurths,J.	Kurths,J.	Albert,R.
Kurths,J.	Kertesz,J.	Boccaletti,S.	Vazquez,A.
Kertesz,J.	Albert,R.	Jeong,H.	Long,B.
Kaski,K.	Kaplan,T.D.	Latora,V.	Vicsek,T.
Tomkins,A.S.	Yoon,C.N.	Kahng,B.	Ravasz,E.
Jeong,H.	Pacheco,A.F.	Arenas,A.	Jeong,H.
Amaral,L.A.N.	Long,B.	Porter,M.A.	Boguna,M.

TABLE 6: Names of co-authors who belong to the same pairs as the top two largest pairs found by multiple-pairs algorithms.

Multiple-pair algorithm			
ICPA		**KM-config**	
pairID=11	**pairID=8**	**pairID=22**	**pairID=5**
Barabasi,A.L.	Newman,M.E.J.	Barabasi,A.L.	Newman,M.E.J.
Jeong,H.	Watts,D.J.	Jeong,H.	Watts,D.J.
Vicsek,T.	Strogatz,S.H.	Oltvai,Z.N.	Strogatz,S.H.
Oltvai,Z.N.	Moore,C.	Bianconi,G.	Moore,C.
Albert,R.	Dodds,P.S.	Palla,G.	Dodds,P.S.
Bianconi,G.	Clauset,A.	Farkas,I.J.	Callaway,D.S.
Palla,G.	Park,J.	Mongru,D.A.	Matthews,P.C.
Ben-Jacob,E.	Girvan,M.	Abel,D.	Jin,E.M.
Farkas,I.J.	Mirollo,R.E.	Ravasz,E.	Muhamad,R.
PerezVicente,C.J.	Martin,M.	Derenyi,I.	Sabel,C.F.



(a) Cores detected by ICPA (b) Pairs detected by ICPA

(c) Cores detected by KM-config (d) Pairs detected by KM-config

(e) Cores detected by BE (f) Cores detected by MINRES

(g) Cores detected by LapCore (h) Cores detected by LowRankCore

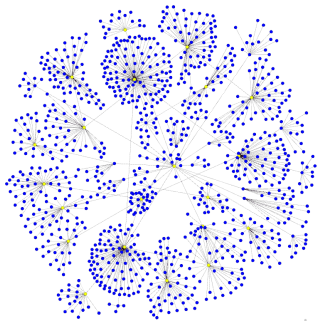
Fig. 8: Core nodes and pairs detected by ICPA and other algorithms in Author Network, which in (a),(c),(e),(f),(g), and (h), the blue nodes represent the periphery nodes and the yellow nodes represents the core nodes detected by different algorithms. In (b) and (d), nodes in the same color are identified as the same pair by ICPA and KM-config, respectively.

pairs, node pairs identified by our method are more coherent in node clusters. In other subfigures, core nodes are drawn in yellow, while periphery nodes are in blue. From these subfigures, we can see that our approach can find the core nodes that are better connected with periphery nodes than other approaches. Therefore, our approach can find multiple pairs and identify core nodes better than others.

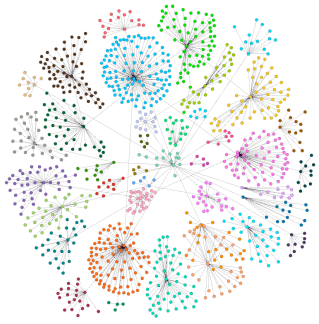
4.4 Case study IV: Detection for Brain networks

Here we test how different algorithms detect core-periphery structures in brain functional connectivity

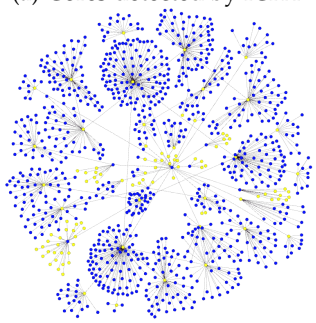
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3138436, IEEE Transactions on Network Science and Engineering



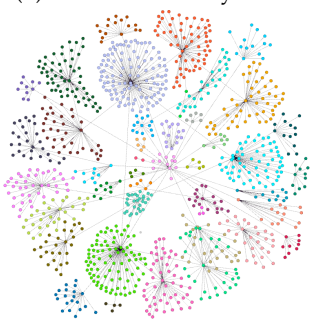
(a) Cores detected by ICPA



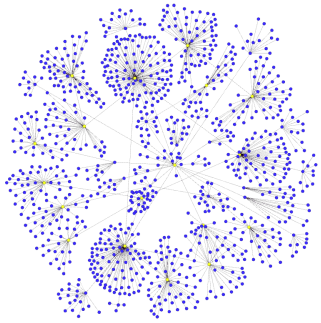
(b) Pairs detected by ICPA



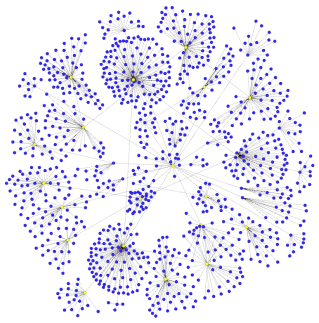
(c) Cores detected by KM-config



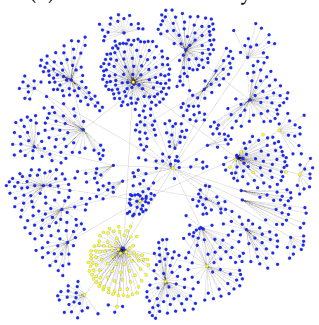
(d) Pairs detected by KM-config



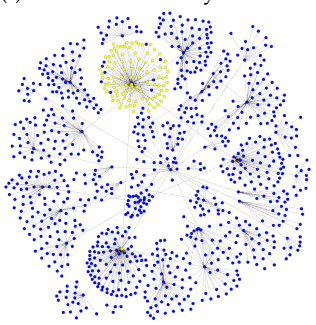
(e) Cores detected by BE



(f) Cores detected by MINRES



(g) Cores detected by LapCore



(h) Cores detected by LowRankCore

Fig. 9: Core nodes and pairs detected by ICPA and other algorithms in Email Network, which in (a),(c),(e),(f),(g), and (h), the blue nodes represent the periphery nodes and the yellow nodes represents the core nodes detected by different algorithms. In (b) and (d), nodes in the same color are identified as the same pair by ICPA and KM-config, respectively.

networks. The data was from our 'Naturalistic Neuroimaging Database' (NNDb), which consists of fMRI scans from full-length movie-watching (data available at https://openneuro.org/datasets/ds002837) [44]. We randomly selected 20 of 86 NNDb participants for testing core-periphery algorithms: 10 each who watched '500 Days of Summer' or 'Citizenfour'.

The fully preprocessed fMRI data was first resampled to a five mm3 voxel size to reduce computation time. Whole-brain functional connectivity matrices were constructed using the AFNI program 3dDegreeCentrality, which performs pairwise Pearson's correlations between all voxel time series [45], [46]. We used a sliding window approach with a window size of 60 seconds and step size of 10 seconds, as suggested by [47], [48]. This resulted in 542 and 675 matrices for participants in '500 Days of Summer' and 'Citizenfour', respectively. The correlation values were then thresholded at r = 0.1 as suggested by [49], resulting in an unweighted binary (1,0) adjacency matrix.

To demonstrate that the algorithm detects core-periphery structures in brain networks that are neurobiologically plausible, we performed affinity propagation clustering on the resulting coreness scores to identify spatially similar core clusters across the duration of the movies [50]. Fig.10 shows the cut half (i.e., top half) dendrogram of core clusters from a sample participant watching '500 Days of Summer'. We display four exemplars from branches with core clusters that are prototypically associated with visual (blue), sensorimotor (green), and auditory and language processing (red). The latter, e.g., have previously been identified as forming cores [51]. Furthermore, these and other networks qualitatively display characteristics of typical patterns of brain activity in that they form larger clusters, are often bilaterally distributed, and follow gyral and sulcal boundaries and other anatomical landmarks.

To further test the performance of our algorithm against existing algorithms for detecting core-periphery pairs in brain networks, we computed the average spatial variance of nodes in the same pair. The spatial information is not used when computing the core-periphery pairs, but we can use it to infer the algorithms' performance. The assumption was that nodes in the same pair should be, on average, spatially closer than nodes in different pairs. Moreover, in order to obtain comparable results, we merged small into larger pairs so that the final pair numbers were similar to those output by the KM algorithm, which we used as a reference. Formally, the average spatial variance of nodes in the same pair var is defined as follows,

$$var = \frac{1}{m}\frac{1}{n-1}\sum_{j=1}^{m}\sum_{i=1}^{n}[(x_{ji} - \frac{\sum_i^n x_{ji}}{n})^2 \tag{62}$$

$$+ (y_{ji} - \frac{\sum_i^n y_{ji}}{n})^2 + (z_{ji} - \frac{\sum_i^n z_{ji}}{n})^2], \tag{63}$$

where x_{ji}, y_{ji}, and z_{ji} is the spatial location of the ith node in jth pair in three dimensions, m is the number of total pairs, and n is the number of nodes in each pair. The results are shown in Table 7 and Table 8 for '500 Days of Summer' and 'Citizenfour' respectively. Pairs identified by our algorithm had a much smaller spatial variance than

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3138436, IEEE Transactions on Network Science and Engineering

others, indicating that nodes in each pair were spatially closer.

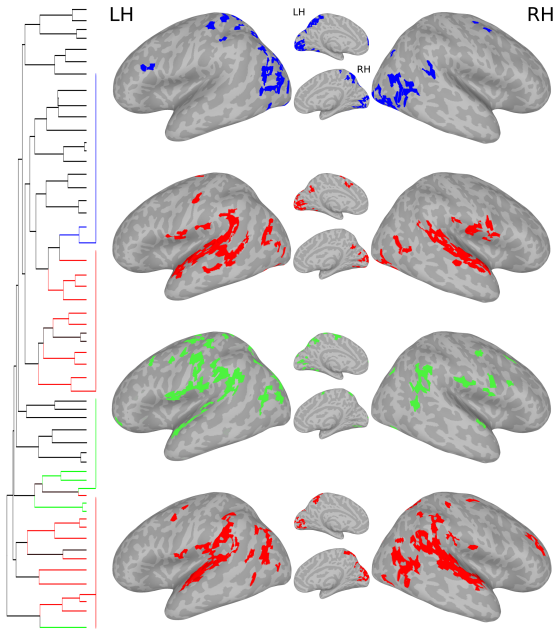


Fig. 10: Core-periphery clusters in the brain during movie watching detected by ICPA. The left is the cut half of the dendrogram showing 56 core clusters after APC clustering for one participant watching '500 Days of Summer'. Example (multi)cores are presented on the inflated left (LH) and right hemisphere (RH) lateral and medial cortical surfaces. Blue is occipital-parietal areas involved in vision; Green is pre/central sulcus and parietal cortices involved in sensorimotor processing, and red is temporal regions involved in auditory and language processing. Regions not colored constitute the periphery.

TABLE 7: Number of pairs detected by ICPA and other algorithms and the average spatial variance between detected pairs in brain networks of participants watching the movie '500 Days of Summer'.

Number of pairs detected by each algorithm					
ICPA	**KM-config**	**BE**	**LapCore**	**MINRES**	**LowRankCore**
4	3	1	1	1	1
Average spatial covariances between detected pairs in brain networks					
ICPA	**KM-config**	**BE**	**LapCore**	**MINRES**	**LowRankCore**
232.635	348.794	362.397	362.397	362.397	362.397

5 Conclusion

Core-periphery detection can be widely used in many applications. In this work, we propose a novel influence-based approach to identify multiple pairs of core-periphery structures in networks in terms of both efficiency and accuracy. It is also applicable to weighted networks, in which case we can compute the influence vectors by dispensing the influences of each node to its neighbors proportional to the corresponding edge weights. Theoretically, our method can be proved to converge and the error rate is bounded. In the experiment with random and real-world networks, our method outperforms other algorithms. Moreover, our approach has computational efficiency in analyzing large dynamic brain networks.

TABLE 8: Number of pairs detected by ICPA and other algorithms and the average spatial variance between detected pairs in brain networks of participants watching the movie 'Citizenfour'.

Number of pairs detected by each algorithm					
ICPA	**KM-config**	**BE**	**LapCore**	**MINRES**	**LowRankCore**
6	3	1	1	1	1
Average spatial covariances between detected pairs in brain networks					
ICPA	**KM-config**	**BE**	**LapCore**	**MINRES**	**LowRankCore**
229.626	348.582	362.397	362.397	362.397	362.397

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NO. 61802372), the Natural Science Foundation of Zhejiang Province (NO. LGG20F020011), the Ningbo Science and Technology Innovation Project (NO. 2018B10080), and the Open Fund of Chinese Academy of Sciences.

References

[1] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.
[2] J. Yang and J. Leskovec, "Structure and overlaps of ground-truth communities in networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 2, pp. 1–35, 2014.
[3] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
[4] M. E. Newman, "Detecting community structure in networks," *The European physical journal B*, vol. 38, no. 2, pp. 321–330, 2004.
[5] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
[6] J. Friedman, L. M. Higgins, and J. Gore, "Community structure follows simple assembly rules in microbial microcosms," *Nature ecology & evolution*, vol. 1, no. 5, pp. 1–7, 2017.
[7] H.-J. Li, Z. Bu, Z. Wang, and J. Cao, "Dynamical clustering in electronic commerce systems via optimization and leadership expansion," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5327–5334, 2019.
[8] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical review E*, vol. 74, no. 1, p. 016110, 2006.
[9] Y. van Gennip, B. Hunter, R. Ahn, P. Elliott, K. Luh, M. Halvorson, S. Reid, M. Valasik, J. Wo, G. E. Tita *et al.*, "Community detection using spectral clustering on sparse geosocial data," *SIAM Journal on Applied Mathematics*, vol. 73, no. 1, pp. 67–83, 2013.
[10] H.-J. Li, L. Wang, Y. Zhang, and M. Perc, "Optimization of identifiability for efficient community detection," *New Journal of Physics*, vol. 22, no. 6, p. 063035, 2020.
[11] H. C. White, S. A. Boorman, and R. L. Breiger, "Social structure from multiple networks. i. blockmodels of roles and positions," *American journal of sociology*, vol. 81, no. 4, pp. 730–780, 1976.
[12] D. Knoke and D. L. Rogers, "A blockmodel analysis of interorganizational networks," *Sociology & Social Research*, vol. 64, no. 1, pp. 28–52, 1979.
[13] S. Gu, C. H. Xia, R. Ciric, T. M. Moore, R. C. Gur, R. E. Gur, T. D. Satterthwaite, and D. S. Bassett, "Unifying the notions of modularity and core–periphery structure in functional brain networks during youth," *Cerebral Cortex*, vol. 30, no. 3, pp. 1087–1102, 2020.

[14] P. Csermely, A. London, L.-Y. Wu, and B. Uzzi, "Structure and dynamics of core/periphery networks," *Journal of Complex Networks*, vol. 1, no. 2, pp. 93–123, 2013.

[15] J. Yang and J. Leskovec, "Overlapping communities explain core–periphery organization of networks," *Proceedings of the IEEE*, vol. 102, no. 12, pp. 1892–1902, 2014.

[16] M. R. Da Silva, H. Ma, and A.-P. Zeng, "Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks," *Proceedings of the IEEE*, vol. 96, no. 8, pp. 1411–1420, 2008.

[17] P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha, "Core-periphery structure in networks (revisited)," *SIAM review*, vol. 59, no. 3, pp. 619–646, 2017.

[18] S. Kojaku and N. Masuda, "Finding multiple core-periphery pairs in networks," *Physical Review E*, vol. 96, no. 5, p. 052313, 2017.

[19] S. H. Lee, M. Cucuringu, and M. A. Porter, "Density-based and transport-based core-periphery structures in networks," *Physical Review E*, vol. 89, no. 3, p. 032810, 2014.

[20] J. Gamble, H. Chintakunta, A. Wilkerson, and H. Krim, "Node dominance: Revealing community and core-periphery structure in social networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 186–199, 2016.

[21] P. Barucca and F. Lillo, "Disentangling bipartite and core-periphery structure in financial networks," *Chaos, Solitons & Fractals*, vol. 88, pp. 244–253, 2016.

[22] T. Lux, "Emergence of a core-periphery structure in a simple dynamic model of the interbank market," *Journal of Economic Dynamics and Control*, vol. 52, pp. A11–A23, 2015.

[23] S. Kojaku, M. Xu, H. Xia, and N. Masuda, "Multiscale core-periphery structure in a global liner shipping network," *Scientific reports*, vol. 9, no. 1, pp. 1–15, 2019.

[24] S. Kojaku and N. Masuda, "Core-periphery structure requires something else in the network," *New Journal of Physics*, vol. 20, no. 4, p. 043012, 2018.

[25] S. P. Borgatti and M. G. Everett, "Models of core/periphery structures," *Social networks*, vol. 21, no. 4, pp. 375–395, 2000.

[26] S. Z. Lip, "A fast algorithm for the discrete core/periphery bipartitioning problem," *arXiv preprint arXiv:1102.5511*, 2011.

[27] J. P. Boyd, W. J. Fitzgerald, M. C. Mahutga, and D. A. Smith, "Computing continuous core/periphery structures for social relations data with minres/svd," *Social Networks*, vol. 32, no. 2, pp. 125–137, 2010.

[28] R. Zelnio, "Identifying the global core-periphery structure of science," *Scientometrics*, vol. 91, no. 2, pp. 601–615, 2012.

[29] J. v. L. de Jeude, G. Caldarelli, and T. Squartini, "Detecting core-periphery structures by surprise," *EPL (Europhysics Letters)*, vol. 125, no. 6, p. 68001, 2019.

[30] B.-B. Xiang, Z.-K. Bao, C. Ma, X. Zhang, H.-S. Chen, and H.-F. Zhang, "A unified method of detecting core-periphery structure and community structure in networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 1, p. 013122, 2018.

[31] J.-L. Liu, Z.-G. Yu, and V. Anh, "Multifractal analysis for core-periphery structure of complex networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 7, p. 073405, 2019.

[32] Cucuringu, Mihai, Lee, Sang, Hoon, Rombach, Puck, Porter, Mason, and A., "Detection of core-periphery structure in networks using spectral methods and geodesic paths," *European Journal of Applied Mathematics*, 2016.

[33] F. Battiston, J. Guillon, M. Chavez, V. Latora, and F. de Vico Fallani, "Multiplex core–periphery organization of the human connectome," *Journal of the Royal Society Interface*, vol. 15, no. 146, p. 20180514, 2018.

[34] C. Ma, B.-B. Xiang, H.-S. Chen, M. Small, and H.-F. Zhang, "Detection of core-periphery structure in networks based on 3-tuple motifs," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 5, p. 053121, 2018.

[35] W. Xing and A. Ghorbani, "Weighted pagerank algorithm," in *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.* IEEE, 2004, pp. 305–314.

[36] I. M. Kloumann, J. Ugander, and J. Kleinberg, "Block models and personalized pagerank," *Proceedings of the National Academy of Sciences*, vol. 114, no. 1, pp. 33–38, 2017. [Online]. Available: https://www.pnas.org/content/114/1/33

[37] Q. Liu, B. Xiang, N. J. Yuan, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, "An influence propagation view of pagerank," *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 3, Mar. 2017. [Online]. Available: https://doi.org/10.1145/3046941

[38] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: fast solutions and applications," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, 2008.

[39] J. Jung, W. Jin, L. Sael, and U. Kang, "Personalized ranking in signed networks using signed random walk with restart," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 973–978.

[40] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 83, no. 2, p. 016107, 2011.

[41] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Transactions on neural networks*, vol. 20, no. 2, pp. 189–201, 2009.

[42] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[43] Jure, Leskovec, Jon, Kleinberg, Christos, and Faloutsos, "Graph evolution: Densification and shrinking diameters," *Acm Transactions on Knowledge Discovery from Data*, 2007.

[44] S. Aliko, J. Huang, F. Gheorghiu, S. Meliss, and J. I. Skipper, "A'naturalistic neuroimaging database'for understanding the brain using ecological stimuli," *BioRxiv*, 2020.

[45] R. W. Cox, "Afni: software for analysis and visualization of functional magnetic resonance neuroimages," *Computers and Biomedical research*, vol. 29, no. 3, pp. 162–173, 1996.

[46] R. C. Craddock and D. J. Clark, "Optimized implementations of voxel-wise degree centrality and local functional connectivity density mapping in afni," *BioRxiv*, p. 067702, 2016.

[47] M. G. Preti, T. A. Bolton, and D. Van De Ville, "The dynamic functional connectome: State-of-the-art and perspectives," *Neuroimage*, vol. 160, pp. 41–54, 2017.

[48] R. Liégeois, E. Ziegler, C. Phillips, P. Geurts, F. Gómez, M. A. Bahri, B. T. Yeo, A. Soddu, A. Vanhaudenhuyse, S. Laureys *et al.*, "Cerebral functional connectivity periodically (de) synchronizes with anatomical constraints," *Brain structure and function*, vol. 221, no. 6, pp. 2985–2997, 2016.

[49] K. A. Garrison, D. Scheinost, E. S. Finn, X. Shen, and R. T. Constable, "The (in) stability of functional brain network measures across thresholds," *Neuroimage*, vol. 118, pp. 651–661, 2015.

[50] U. Bodenhofer, A. Kothmeier, and S. Hochreiter, "Apcluster: an r package for affinity propagation clustering," *Bioinformatics*, vol. 27, no. 17, pp. 2463–2464, 2011.

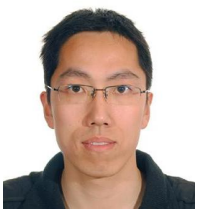
[51] L. R. Chai, M. G. Mattar, I. A. Blank, E. Fedorenko, and D. S. Bassett, "Functional network dynamics of the language system," *Cerebral Cortex*, vol. 26, no. 11, pp. 4148–4159, 2016.

Xin Shen Xin Shen is a College student at the College of Information Science and Engineering, Ningbo University, working with Dr. Chengbin Peng. She interests in complex network analysis, especially algorithms for detecting core-periphery structures in networks.

Sarah Aliko Sarah Aliko is a final year Ph.D. student at University College London, working with Dr. Jeremy I Skipper. She previously obtained her BSc in Molecular Biology from the University of Edinburgh. Her work focuses on the functional connectivity of the brain under naturalistic stimuli, in particular the neurobiology of language comprehension. She uses movies and fMRI to investigate the network organization of the brain and how language varies with context.

Yue Han Yue Han is a senior student at the College of Information Science and Engineering, Ningbo University, working with Dr. Chengbin Peng. His strong interests mainly include complex network analysis, particularly algorithms for detecting core-periphery structures in networks.

Jeremy I Skipper Jeremy I Skipper has B.A. degrees in philosophy and religion, and psychology. He has a Ph.D. from the Cognitive and Cognitive Neuroscience Program from the Department of Psychology at The University of Chicago. He was a Postdoctoral Fellow of Psychology in Psychiatry at the Sackler Institute for Developmental Psychobiology at the Weill Medical College of Cornell University. He was previously an Assistant Professor at Hamilton College and has held visiting scholar positions at Vanderbilt University and the Max Planck Institute for Psycholinguistics, Nijmegen. He is currently an Associate Professor in Experimental Psychology at University College London and the Director of the Language, Action and Brain Lab (LAB Lab). His research centers around understanding how the human brain supports language and other functions during real-world conditions and how it changes with experience or pharmacological intervention (e.g., with psychedelic drugs).



Chengbin Peng Chengbin Peng received the B.E. and M.S. degrees in computer science from Zhejiang University, China. He received a Ph.D. degree in computer science from King Abdullah University of Science and Technology, KSA, in 2015. He is currently an Associate Professor with the College of Information Science and Engineering, Ningbo University. His research interests include complex network analysis, trajectory data analysis, and machine vision.