

PAPER • OPEN ACCESS

Core-periphery structure requires something else in the network

To cite this article: Sadamori Kojaku and Naoki Masuda 2018 *New J. Phys.* **20** 043012

View the [article online](#) for updates and enhancements.

You may also like

- [Multifractal analysis for core-periphery structure of complex networks](#)
Jin-Long Liu, Zu-Guo Yu and Vo Anh
- [Deciphering the imprint of topology on nonlinear dynamical network stability](#)
J Nitzbon, P Schultz, J Heitzig et al.
- [Detecting the core of a network by the centralities of the nodes](#)
Peijie Ma, , Xuezao Ren et al.

New Journal of Physics

The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft 

IOP Institute of Physics

Published in partnership
with: Deutsche Physikalische
Gesellschaft and the Institute
of Physics



OPEN ACCESS

RECEIVED
24 October 2017

REVISED
27 February 2018

ACCEPTED FOR PUBLICATION
8 March 2018

PUBLISHED
12 April 2018

Original content from this
work may be used under
the terms of the [Creative
Commons Attribution 3.0
licence](#).

Any further distribution of
this work must maintain
attribution to the
author(s) and the title of
the work, journal citation
and DOI.



PAPER

Core-periphery structure requires something else in the network

Sadamori Kojaku  and Naoki Masuda 

Department of Engineering Mathematics, Merchant Venturers Building, University of Bristol, Woodland Road, Clifton, Bristol BS8 1UB, United Kingdom

E-mail: naoki.masuda@bristol.ac.uk

Keywords: networks, mesoscale structure, core-periphery structure

Abstract

A network with core-periphery structure consists of core nodes that are densely interconnected. In contrast to a community structure, which is a different meso-scale structure of networks, core nodes can be connected to peripheral nodes and peripheral nodes are not densely interconnected. Although core-periphery structure sounds reasonable, we argue that it is merely accounted for by heterogeneous degree distributions, if one partitions a network into a single core block and a single periphery block, which the famous Borgatti–Everett algorithm and many succeeding algorithms assume. In other words, there is a strong tendency that high-degree and low-degree nodes are judged to be core and peripheral nodes, respectively. To discuss core-periphery structure beyond the expectation of the node's degree (as described by the configuration model), we propose that one needs to assume at least one block of nodes apart from the focal core-periphery structure, such as a different core-periphery pair, community or nodes not belonging to any meso-scale structure. We propose a scalable algorithm to detect pairs of core and periphery in networks, controlling for the effect of the node's degree. We illustrate our algorithm using various empirical networks.

1. Introduction

Many complex systems, biological, physical or social, can be represented by networks [1, 2]. A network consists of a set of nodes and edges, where nodes represent objects (e.g., people, web pages) and edges represent pairwise relationships between objects (e.g., friendships, hyperlinks). A consistent observation across different types of networks is that they are often composed of communities, i.e., groups of densely interconnected nodes [3]. A community is often associated with a group of nodes sharing a role or similarity such as a circle of friends in social networks [4], a set of web pages discussing the same topic [5, 6] and a functional group of proteins [7].

Core-periphery structure is another mesoscopic structure of networks that has experienced a surge of interests in the last two decades. A core-periphery structure in its simplest form refers to a partition of a network into two groups of nodes called core and periphery, where core nodes are densely interconnected (i.e., adjacent), and peripheral nodes are adjacent to the core nodes but not to other peripheral nodes [8–10]. Core-periphery structure has been detected in a number of networks including social networks [8, 10–20], protein–protein interaction networks [15, 21, 22], neural networks [23, 24], trade networks [13, 25, 26], financial networks [14, 27–31] and transportation networks [10, 13, 14, 19, 32]. For example, in a world-trade network among countries, economically strong countries trade with other strong countries, constituting a core. Economically weak countries mainly trade with strong countries, constituting a periphery [13, 26].

Borgatti and Everett analysed core-periphery structure in quantitative terms for the first time [8]. They expressed a core-periphery structure by a core block (i.e., group of core nodes) and a periphery block (i.e., group of peripheral nodes) as shown in figure 1. The core block has many intra-block edges (the top left block in figure 1). The periphery block has relatively few intra-block edges (the bottom right block in figure 1). There may be many inter-block edges (off-diagonal blocks in figure 1) [8, 10, 12, 15, 19, 20, 24] or relatively few inter-block edges [8, 10, 16, 17, 25, 26, 31, 33–35]. The core-periphery structure expressed by blocks of nodes is classified as a discrete variant of core-periphery structure based on edge density [8–10, 12, 14, 17–20, 26, 33, 35]. There are

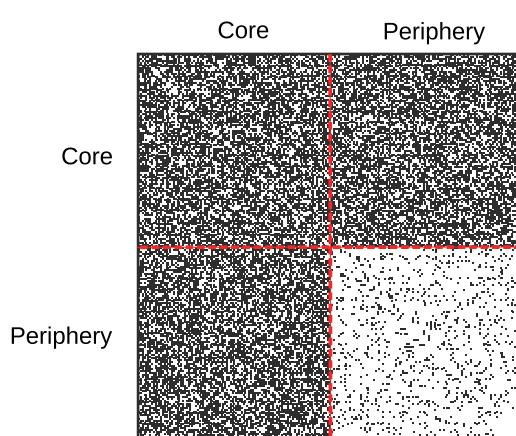


Figure 1. Adjacency matrix of a network with good-looking core-periphery structure composed of two blocks. The filled and open cells in the i th row and j th column indicate the presence and absence of the edge between nodes i and j , respectively. The dotted lines represent the boundary between the core and periphery.

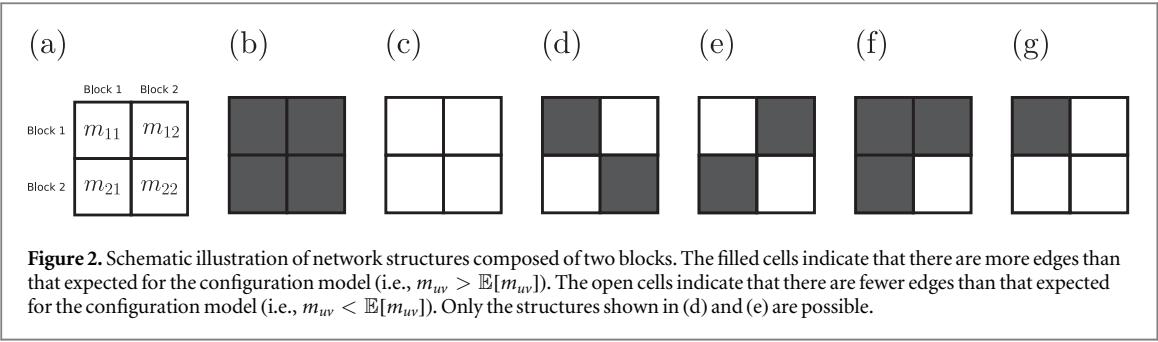
other types of core-periphery and related structure, such as continuous versions of core-periphery structure [8, 10, 14, 17, 25], transport-based core-periphery structure [13, 14, 17, 21, 36], k -core [37] and rich-clubs [38, 39].

Given that block structure of networks, or equivalently, hard partitioning of the nodes into groups, has spurred many studies such as community detection [3, 40] and the inference of stochastic block models (SBM) [6, 41], as well as its appeal to intuition, we focus on the discrete version of core-periphery structure based on edge density in the present paper. If a network has such core-periphery structure, the core block should have more intra-block edges and the periphery block should have fewer intra-block edges than a reference. We argue that the core-periphery structure that Borgatti and Everett proposed (figure 1), which many of the subsequent work is based on, is impossible if we use the configuration model [42] as the null model and there are just one core and one periphery. The configuration model is a common class of random graph models that preserve the degree or its mean value of each node. Therefore, our claim implies that there is no core-periphery structure a la mode de Borgatti and Everett beyond the expectation from the degree of each node (i.e., hubs are core nodes), which is, in fact, consistent with some previous observations [16, 28, 30].

Then, we are led to a question: what is a core-periphery structure? To answer this question, let us look at the status of the configuration model in other measurements of networks. We have a plethora of centrality measures for nodes because the degree is often not a useful measure of the importance of nodes [1]. In other words, different centrality measures provide rank orders of nodes in the given network that are not expected from the configuration model. In network motif analysis, where one looks for small subnetworks that are abundant in a given network, we discount the frequency of subnetworks that are merely explained by the degree of the nodes (i.e., configuration model) [43]. In community detection, it is conventional to use the configuration model as the null model against which one assesses the significance of community structure [3, 4, 6, 41]. To solve the conundrum that one does not discover core-periphery structure using the configuration model as the null model, we propose that one must add at least one different block apart from a core block and the corresponding periphery block for a network to have core-periphery structure that is consistent with figure 1. Such blocks may be a community, sparsely connected part, a different core-periphery pair [10, 19, 20, 32], a core that shares the periphery with the focal core-periphery pair [34] and so forth. Then, we propose a scalable algorithm to partition a network into multiple core-periphery pairs including community detection as special cases, aiming to detect core-periphery structure that is not merely explained by the degree of each node. Crucially, we use the configuration model as the null model, which is different from our previous algorithm [19].

2. Core-periphery structure needs at least three blocks

Consider an unweighted network composed of N nodes and M edges. The $N \times N$ adjacency matrix of the network is denoted by $\mathbf{A} = (A_{ij})$, where $A_{ij} = 1$ if nodes i and j ($\neq i$) are adjacent and $A_{ij} = 0$ otherwise. We assume that the network is undirected (i.e., $A_{ij} = A_{ji}$ for all $i \neq j$) and has no self-loops (i.e., $A_{ii} = 0$ for all i). Let d_i be the number of edges incident to node i (i.e., degree). As the null model of networks, we use the configuration model, i.e., a random network model preserving the degree of each node. For the configuration model, we allow multi-edges (i.e., multiple edges between nodes) and self-loops for computational ease. In fact,



multi-edges and self-loops change our quality function for finding core-periphery structure in the order of $1/N$, which is negligible if N is large. We denote by $\mathbb{E}[\cdot]$ the expectation with respect to the configuration model.

Consider a partition of the set of N nodes into B blocks (i.e., groups). Let N_u be the number of nodes in block u and m_{uv} be the number of edges between blocks u and v . Note that $m_{uv} = m_{vu}$. For notational convenience, we define m_{uu} as twice the number of self-loops in block u plus twice the number of edges between different nodes within block u . Denote by m_{uv}^{conf} the number of edges between blocks u and v in a network generated by the configuration model whose degree sequence is given by that of the original network. Suppose a network composed of $B = 2$ blocks (figure 2(a)). There are potentially six types of block structure of networks represented by two blocks. In figures 2(b)–(g), a filled block has more edges than that for the configuration model (i.e., $m_{uv} > \mathbb{E}[m_{uv}^{\text{conf}}]$), and a blank block has fewer edges than that for the configuration model (i.e., $m_{uv} < \mathbb{E}[m_{uv}^{\text{conf}}]$). The entire network would be dense if there are many intra- and inter-block edges (figure 2(b)). In contrast, the network would be sparse if there are relatively few intra- and inter-block edges (figure 2(c)). The network has community structure if there are many intra-block edges and relatively few inter-block edges (figure 2(d)). A contrasting case is a structure close to a bipartite network, where there are relatively few intra-block edges and many inter-block edges (figure 2(e)). Core-periphery structure would correspond to the case in which there are many edges within one block and few edges within the other block. With core-periphery structure, inter-block edges may be abundant (figure 2(f)) [8, 10, 12, 15, 19, 20, 24] or not (figure 2(g)) [8, 10, 16, 17, 25, 26, 31, 33–35].

Many algorithms for finding discrete versions of core-periphery structure seek a partition of nodes into one core block and one periphery block (figures 2(f) or (g)). Let us consider the karate club network [44], which has been demonstrated to have core-periphery structure [10, 13, 19, 20, 26, 32, 35]. The Borgatti–Everett (BE) algorithm partitions the $N = 34$ nodes into a core and a periphery as shown in figure 3. The detected blocks seem to suggest core-periphery structure because the core nodes are densely interconnected, whereas the peripheral nodes are sparsely interconnected. However, relative to the configuration model, the network is closer to a bipartite network than to core-periphery structure; there are fewer edges within both core and periphery blocks (i.e., $m_{11} = 10$, $\mathbb{E}[m_{11}^{\text{conf}}] = 26.25$, $m_{22} = 38$ and $\mathbb{E}[m_{22}^{\text{conf}}] = 54.25$) and more edges between the core and periphery blocks than those expected for the configuration model (i.e., $m_{12} = 54$ and $\mathbb{E}[m_{12}^{\text{conf}}] = 37.74$).

This observation is in fact universal; core-periphery structure is impossible with two blocks when the null model is the configuration model. To show this, consider a network composed of $B = 2$ blocks. The degree of each node is the same between the original network and a sample network generated by the configuration model. Therefore, the number of edges emanating from each block is also the same between the original network and the sample network. Therefore, we obtain

$$m_{11} + m_{12} = \mathbb{E}[m_{11}^{\text{conf}}] + \mathbb{E}[m_{12}^{\text{conf}}], \quad (1)$$

$$m_{21} + m_{22} = \mathbb{E}[m_{21}^{\text{conf}}] + \mathbb{E}[m_{22}^{\text{conf}}]. \quad (2)$$

Rearranging equations (1) and (2) yields

$$m_{11} - \mathbb{E}[m_{11}^{\text{conf}}] = -(m_{12} - \mathbb{E}[m_{12}^{\text{conf}}]), \quad (3)$$

$$m_{22} - \mathbb{E}[m_{22}^{\text{conf}}] = -(m_{21} - \mathbb{E}[m_{21}^{\text{conf}}]). \quad (4)$$

Equations (3) and (4) imply that if a block has more intra-block edges in the original network than in the configuration model, the same block must have fewer inter-block edges in the original network than in the configuration model. Because we assumed that the network is undirected, we obtain $m_{21} = m_{12}$ and $m_{21}^{\text{conf}} = m_{12}^{\text{conf}}$. Using these relationships, we rewrite equation (4) as

$$m_{22} - \mathbb{E}[m_{22}^{\text{conf}}] = -(m_{12} - \mathbb{E}[m_{12}^{\text{conf}}]). \quad (5)$$

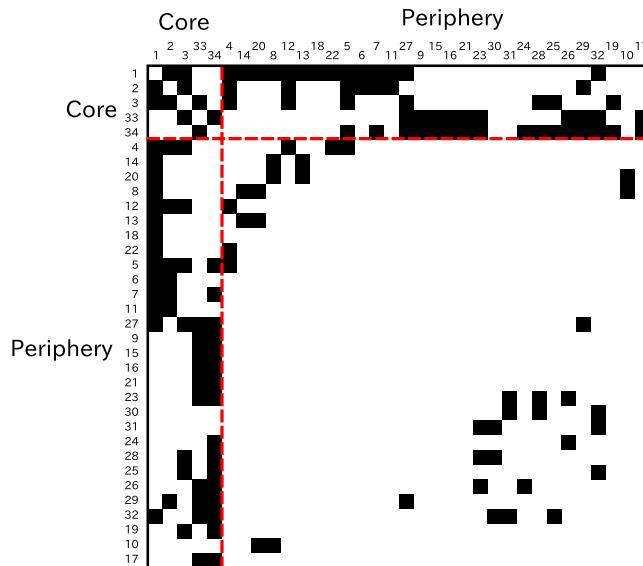


Figure 3. Core-periphery structure of the karate club network detected by the BE algorithm. The nodes are reordered. The filled and open cells represent the presence and absence of edges, respectively.

By combining equations (3) and (5), we obtain

$$m_{11} - \mathbb{E}[m_{11}^{\text{conf}}] = m_{22} - \mathbb{E}[m_{22}^{\text{conf}}]. \quad (6)$$

Equation (6) indicates that there is no network composed of two blocks such that the core block has more edges in the original network than the configuration model (i.e., $m_{11} > \mathbb{E}[m_{11}^{\text{conf}}]$) and the periphery block has fewer edges in the original network than the configuration model (i.e., $m_{22} < \mathbb{E}[m_{22}^{\text{conf}}]$). Therefore, the core-periphery structure does not exist if one partitions a network into a single core block and a single periphery block, as the BE algorithm does. It should be noted that equations (3) and (4) imply that the networks represented by figures 2(b) and (c) are also impossible. In contrast, the networks shown in figures 2(d) and (e) satisfy equations (3), (4), and (6) and therefore are possible.

Core-periphery structure is possible if the network has $B = 3$ or more blocks. To identify the block structures that are possible and those that are not, we introduce the notion of compatibility of block structure as follows. Consider a network composed of B blocks. The number of edges between blocks u and v in the original network is given by

$$m_{uv} = \sum_{i=1}^N \sum_{j=1}^N A_{ij} \delta(b_i, u) \delta(b_j, v), \quad (7)$$

where b_i is the index of the block to which node i belongs, and $\delta(\cdot, \cdot)$ is Kronecker delta. Equation (7) leads to

$$\begin{aligned} \sum_{v=1}^B m_{uv} &= \sum_{v=1}^B \sum_{i=1}^N \sum_{j=1}^N A_{ij} \delta(b_i, u) \delta(b_j, v) = \sum_{i=1}^N \delta(b_i, u) \sum_{j=1}^N A_{ij} \sum_{v=1}^B \delta(b_j, v) \\ &= \sum_{i=1}^N \delta(b_i, u) \sum_{j=1}^N A_{ij} = \sum_{i=1}^N d_i \delta(b_i, u), \quad u = 1, 2, \dots, B, \end{aligned} \quad (8)$$

where $d_i = \sum_{j=1}^N A_{ij}$ is the degree of node i . The sum $\sum_{v=1}^B m_{uv}$ is the sum of the degree of nodes in block u . Because the configuration model preserves the degree of each node, the sum $\sum_{v=1}^B m_{uv}$ is the same between the original network and the configuration model, i.e.,

$$\sum_{v=1}^B m_{uv} = \sum_{v=1}^B \mathbb{E}[m_{uv}^{\text{conf}}], \quad u = 1, 2, \dots, B. \quad (9)$$

Note that equation (9) generalises equations (1) and (2). Then, we categorise blocks into dense (i.e., $m_{uv} > \mathbb{E}[m_{uv}^{\text{conf}}]$) and sparse (i.e., $m_{uv} < \mathbb{E}[m_{uv}^{\text{conf}}]$) blocks. We say that a block structure is compatible if the designated dense and sparse blocks are realisable in the sense that equation (9) is satisfied. We describe the procedures to find compatible block structures in appendix A.

With $B = 3$ blocks, eight types of block structure are compatible with equation (9) (figures 4(a)–(h)). The networks shown in figures 4(a) and (b) consist of two and three communities, respectively. The networks shown in figures 4(c) and (d) are bipartite-like and tripartite-like networks, respectively. The network shown in

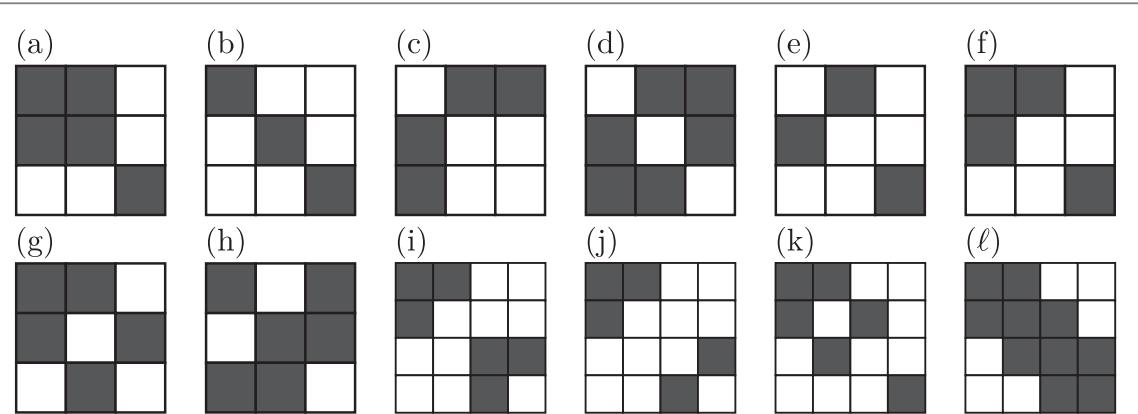


Figure 4. Schematic illustration of network structures with three or four blocks that are compatible with equation (9). We show all network structures composed of three blocks in panels (a)–(h) and four out of the 49 structures with four blocks in panels (i)–(l). The other 45 patterns with four blocks are shown in figure B1.

figure 4(e) is a union of a bipartite-like subnetwork composed of blocks 1 and 2 and a community composed of block 3. These network structures extend those viable in the case of two blocks (figures 2(d) and (e)). The networks shown in figures 4(f)–(h) contain core-periphery pairs. In figure 4(f), blocks 1 and 2 constitute a core-periphery pair, and block 3 constitutes a community. In figure 4(g), blocks 1 and 2 constitute a core-periphery pair, and blocks 2 and 3 constitute a bipartite-like subnetwork. The network shown in figure 4(h) consists of two cores (i.e., blocks 1 and 2) sharing a periphery (i.e., block 3), which is the structure studied in [34].

With $B = 4$ blocks, 49 types of block structure are compatible with equation (9). Four of them are shown in figures 4(i)–(l) for illustration (see figure B1 for the others). The network shown in figure 4(i) is composed of two non-overlapping core-periphery pairs [19, 20, 24, 32]. The network shown in figure 4(j) consists of one core-periphery pair (i.e., blocks 1 and 2) and one bipartite-like subnetwork (i.e., blocks 3 and 4). The network shown in figure 4(k) consists of one core-periphery pair (i.e., blocks 1 and 2), one bipartite-like subnetwork (i.e., blocks 2 and 3) and a community (i.e., block 4), in which the core-periphery pair and bipartite-like subnetwork overlap. The network shown in figure 4(l) has three overlapping communities, i.e., a community composed of blocks 1 and 2, one composed of blocks 2 and 3, and one composed of blocks 3 and 4.

To conclude, the core-periphery structure à la mode de Borgatti and Everett [8] relative to the configuration model can exist only when we have at least three blocks. In other words, a core-periphery pair requires a different substructure of the network that coexists in the same network, e.g., a community, bipartite-like structure, or another core-periphery pair that may overlap with the first one.

3. Methods

In this section, we first describe a new algorithm for detecting core-periphery structure, which we refer to as KM–config, based on the observations made in section 2. MATLAB and C++ codes of KM–config are available at https://github.com/skojaku/km_config/. Then, we explain other methods and data used in section 4.

3.1. Our algorithm

3.1.1. Objective function

We propose an algorithm, KM–config, to detect discrete versions of core-periphery structure in networks. In contrast to our previous algorithm that uses the Erdős–Rényi random graph as the null model [19], which we refer to as KM–ER, here we use the configuration model as the null model. This is because we are interested in the structure that is not merely explained by the node's degree.

We assume that a network consists of C non-overlapping core-periphery pairs, each of which is composed of one core block and one periphery block, e.g., figure 4(i). Each core-periphery pair should have (i) many intra-core edges, (ii) many edges between the core and the corresponding periphery (i.e., core-periphery edges), (iii) few intra-periphery edges and (iv) few edges to other core-periphery pairs (i.e., inter-pair edges). Although some previous studies do not assume property (ii) [8, 25, 27–29, 31–33, 35], we require it because otherwise one cannot relate a periphery with a particular core.

We define idealised core-periphery pairs satisfying properties (i)–(iv) [19, 24] by

$$\mathbf{A}^* = (A_{ij}^*), \quad A_{ij}^* \equiv (x_i + x_j - x_i x_j) \delta(c_i, c_j), \quad (10)$$

where $x_i = 1$ or $x_i = 0$ if node i is a core node or a peripheral node, respectively, and c_i ($1 \leq c_i \leq C$) is the index of the core-periphery pair to which node i belongs. Within each idealised core-periphery pair, every core node is adjacent to every other core node (property (i)) and also adjacent to all the corresponding peripheral nodes (property (ii)), and every peripheral node is not adjacent to any other peripheral nodes (property (iii)). Furthermore, there are no edges between different idealised core-periphery pairs (property (iv)).

We seek c_i and x_i ($1 \leq i \leq N$) that maximise similarity between \mathbf{A} and \mathbf{A}^* as defined by

$$Q_{\text{config}}^{\text{cp}} \equiv \frac{1}{2M} \sum_{i=1}^N \sum_{j=1}^N A_{ij} A_{ij}^* - \mathbb{E} \left[\frac{1}{2M} \sum_{i=1}^N \sum_{j=1}^N A_{ij}^{\text{conf}} A_{ij}^* \right], \quad (11)$$

where $\mathbf{A}^{\text{conf}} = (A_{ij}^{\text{conf}})$ is the adjacency matrix of a network generated by the configuration model. The first term on the right-hand side of equation (11) is the fraction of intra-core and core-periphery edges (i.e., $A_{ij} = A_{ij}^* = 1$), corresponding to properties (i) and (ii). The second term is the counterpart for the configuration model. The factor $1/2M$ in the first and second terms normalises $Q_{\text{config}}^{\text{cp}}$ to range in $[-1, 1]$. The remaining two properties (iii) and (iv) are also consistent with the maximisation of $Q_{\text{config}}^{\text{cp}}$. To show this, we rewrite $Q_{\text{config}}^{\text{cp}}$ as

$$Q_{\text{config}}^{\text{cp}} = -\frac{1}{2M} \sum_{i=1}^N \sum_{j=1}^N A_{ij}(1 - A_{ij}^*) + \mathbb{E} \left[\frac{1}{2M} \sum_{i=1}^N \sum_{j=1}^N A_{ij}^{\text{conf}}(1 - A_{ij}^*) \right]. \quad (12)$$

Because $\sum_{i=1}^N \sum_{j=1}^N A_{ij}(1 - A_{ij}^*)$ is the sum of the number of intra-periphery edges and that of inter-pair edges, the maximisation of $Q_{\text{config}}^{\text{cp}}$ minimises the two types of edges associated with properties (iii) and (iv).

In the configuration model, the expected number of edges between nodes i and j is given by $\mathbb{E}[A_{ij}^{\text{conf}}] = d_i d_j / 2M$ [45, 46]. Substitution of $\mathbb{E}[A_{ij}^{\text{conf}}] = d_i d_j / 2M$ and equation (10) into equation (11) yields

$$Q_{\text{config}}^{\text{cp}} = \frac{1}{2M} \sum_{i=1}^N \sum_{j=1}^N \left(A_{ij} - \frac{d_i d_j}{2M} \right) (x_i + x_j - x_i x_j) \delta(c_i, c_j). \quad (13)$$

If we restrict that all nodes are core nodes (i.e., $x_i = 1$ for $i = 1, 2, \dots, N$), $Q_{\text{config}}^{\text{cp}}$ is equivalent to the modularity [4, 46], which is used for finding communities in networks. The $Q_{\text{config}}^{\text{cp}}$ shares shortcomings with the modularity such as the resolution limit. See section 5 for further discussion.

3.1.2. Relationship to Markov stability

We can relate $Q_{\text{config}}^{\text{cp}}$ to discrete-time random walks, similar to the case of the Markov stability formalism for community detection [47–50]. Consider a random walker that moves from a node to one of the neighbouring nodes selected uniformly at random in each discrete time step. Let $T_{(c,x)(c',x')} \equiv m_{(c,x)(c',x')}/D_{(c,x)}$ be the transition probability from block (c, x) to block (c', x') , where $D_{(c,x)}$ is the sum of the degree of the nodes in block (c, x) . Let $\pi_{(c,x)} \equiv D_{(c,x)}/2M$ be the stationary probability with which the random walker visits block (c, x) . Then, one can rewrite $Q_{\text{config}}^{\text{cp}}$ as

$$\begin{aligned} Q_{\text{config}}^{\text{cp}} &= \frac{1}{2M} \sum_{c=1}^C \left(m_{(c,1)(c,1)} + 2m_{(c,0)(c,1)} - \frac{D_{(c,1)}^2}{2M} - \frac{2D_{(c,0)}D_{(c,1)}}{2M} \right) \\ &= \sum_{c=1}^C \left(\frac{D_{(c,1)}}{2M} \cdot \frac{m_{(c,1)(c,1)}}{D_{(c,1)}} + \frac{2D_{(c,0)}}{2M} \cdot \frac{m_{(c,0)(c,1)}}{D_{(c,0)}} - \frac{D_{(c,1)}^2}{4M^2} - \frac{2D_{(c,0)}}{2M} \cdot \frac{D_{(c,1)}}{2M} \right) \\ &= \sum_{c=1}^C (\pi_{(c,1)} T_{(c,1)(c,1)} + 2\pi_{(c,0)} T_{(c,0)(c,1)} - \pi_{(c,1)}^2 - 2\pi_{(c,0)}\pi_{(c,1)}). \end{aligned} \quad (14)$$

Now, imagine a random walker starting from a node i selected randomly according to the stationary density $d_i/2M$ ($1 \leq i \leq N$), at time $t = 0$. The probability that the random walker is in block (c, x) at time $t = 0$ and block (c', x') at time $t = 1$ is given by $\pi_{(c,x)} T_{(c,x)(c',x')}$, which is accounted for by the first and second terms of the right-hand side of equation (14). The corresponding probability for the configuration model is given by $\pi_{(c,x)} \pi_{(c',x')}$, which is accounted for by the third and fourth terms. Therefore, $Q_{\text{config}}^{\text{cp}}$ measures how likely a random walker moves to the core of the currently visited node in one step relative to the probability expected for the configuration model. This observation is exploited in a different algorithm to detect core-periphery structure of networks [13].

3.1.3. Maximisation of the objective function

We maximise $Q_{\text{config}}^{\text{cp}}$ using a label switching heuristic [51, 52], which we have employed in our previous algorithm, KM-ER, that uses the Erdős-Rényi random graph as the null model [19]. First, we initialise the labels by $c_i = i$ and $x_i = 1$ ($1 \leq i \leq N$). Then, we update the label of each node as follows. Suppose that node i has a neighbour in a core-periphery pair c' . We tentatively assign node i to the core (i.e., $(c_i, x_i) = (c', 1)$) and compute

the new value of $Q_{\text{config}}^{\text{cp}}$. We also tentatively assign node i to the periphery (i.e., $(c_i, x_i) = (c', 0)$) and compute $Q_{\text{config}}^{\text{cp}}$. We perform the tentative assignments for all the core-periphery pairs to which any neighbour of node i belongs. If any tentative assignments do not raise $Q_{\text{config}}^{\text{cp}}$, we do not update (c_i, x_i) . Otherwise, we update (c_i, x_i) to the tentative label (i.e., $(c', 0)$ or $(c', 1)$) giving the largest increment in $Q_{\text{config}}^{\text{cp}}$. We inspect each node in a random order. If no node has changed its label during the inspection of all the N nodes, we stop updating the labels. Otherwise, we draw a new random order and inspect each node according to the new random order, and repeat the procedure. We run this algorithm ten times starting from the same initial condition and adopt the node labelling that realises the largest value of $Q_{\text{config}}^{\text{cp}}$.

The increment in $Q_{\text{config}}^{\text{cp}}$ caused by updating node i 's label from (c, x) to (c', x') is given by

$$\begin{aligned} & \frac{1}{M} \left[\tilde{d}_{i,(c',1)} + x' \tilde{d}_{i,(c',0)} - d_i \frac{D_{(c',1)} + x' D_{(c',0)}}{2M} - \frac{d_i^2}{4M} (x' - 2(x + x' - xx') \delta(c, c')) \right] \\ & - \frac{1}{M} \left[\tilde{d}_{i,(c,1)} + x \tilde{d}_{i,(c,0)} - d_i \frac{D_{(c,1)} + x D_{(c,0)}}{2M} + \frac{d_i^2}{4M} x \right], \end{aligned} \quad (15)$$

where $\tilde{d}_{i,(c,x)} = \sum_{j=1}^N A_{ij} \delta(c_j, c) \delta(x_j, x)$ is the number of edges connecting node i and block (c, x) . When inspecting node i , we calculate equation (15) at most $2d_i$ times. Therefore, the time needed for inspecting all nodes is $\mathcal{O}(\sum_{i=1}^N d_i) = \mathcal{O}(M)$, and that of the entire algorithm is $\mathcal{O}(M \times (\text{the number of inspections over the } N \text{ nodes}))$.

3.1.4. Statistical test

We define the quality q of a core-periphery pair c by its contribution to $Q_{\text{config}}^{\text{cp}}$, i.e.,

$$q \equiv \frac{1}{2M} \sum_{i=1}^N \sum_{j=1}^N \left(A_{ij} - \frac{d_i d_j}{2M} \right) (x_i + x_j - x_i x_j) \delta(c_i, c) \delta(c_j, c). \quad (16)$$

One may deem that a core-periphery pair is significant if its q is statistically larger than the value expected for the configuration model. However, q may depend on the size (i.e., the number of nodes) n of the core-periphery pair, as is the case for the modularity [53].

Inspired by these considerations, we carry out a statistical test of the detected core-periphery pairs as follows. We generate 500 randomised networks for the given network using the configuration model. Then, we detect core-periphery pairs in each randomised network. We compute the quality \hat{q} and size \hat{n} of each core-periphery pair detected in the randomised network. On the basis of the samples of \hat{q} and \hat{n} , we infer the joint probability distribution $P(\hat{q}, \hat{n})$ using the Gaussian kernel density estimator [54, 55]. Finally, we regard the core-periphery pair detected in the original network with a quality value of q to be significant if q is statistically larger than that of the core-periphery pair of the same size n detected in the randomised networks, i.e., if $P(\hat{q} \geq q | n) \leq \alpha$, where P is the probability and α is a significance level. (See appendix C for the computation of $P(\hat{q} \geq q | \hat{n})$.) We refer to the nodes that do not belong to any significant core-periphery pair as residual nodes.

Because we carry out the test for each core-periphery pair in the original network, we have to correct the significance level to suppress false positives due to multiple comparisons. To this end, we adopt the Šidák correction [56], with which we test each core-periphery pair in the original network at a significance level of $\alpha = 1 - (1 - \alpha')^{1/C}$, where α' is the targeted significance. We set $\alpha' = 0.05$.

Empirical networks often have core-periphery pairs that are substantially larger than any of those detected in the 500 randomised networks (section 4.1). It is unlikely that one finds core-periphery pairs of the same size in randomised networks even if more samples of randomised networks are generated. The kernel density estimator enables us to infer $P(\hat{q} \geq q | n)$ for large core-periphery pairs in the original network based on the quality and size of smaller core-periphery pairs detected in randomised networks.

Quality q may be significantly large for bipartite-like pairs of blocks (figure 2(e)). Therefore, if our algorithm detects bipartite-like pairs of blocks, we manually mark them and distinguish them from the core-periphery pairs. Specifically, we regard a detected pair of blocks as bipartite-like if it has fewer intra-core edges than expected for the configuration model (i.e., if $m_{(c,1),(c,1)} < \mathbb{E}[m_{(c,1),(c,1)}]$). Otherwise we regard it as a core-periphery pair. Our algorithm did not find other types of block pairs (i.e., those shown in figures 2(b), (c) and (g)) for the networks examined in the following sections.

3.2. Other algorithms for comparison

We compare the present algorithm, KM-config, with three algorithms for finding a single core-periphery pair, i.e., the BE [8], MINRES [25, 33] and SBM [16] algorithms, and three algorithms for finding multiple core-periphery pairs, i.e., Xiang [32], Divisive [19] and KM-ER algorithms [19]. We ran the Tunç–Verma [24] algorithm but do not show the results because the Tunç–Verma algorithm did not find significant core-periphery pairs or did not terminate within 48 h on our computer (Intel 2.6 GHz Sandy Bridge processors and

4 GB of memory). It should be noted that none of these algorithms uses the configuration model as the null model.

The BE, Divisive and KM–ER algorithms intend to produce many core-periphery edges (i.e., edges connecting a core node and a peripheral node) within each core-periphery pair (figure 2(f)). With the MINRES, SBM and Xiang algorithms, core-periphery edges can be relatively sparse (figure 2(g)).

We set the parameters of these algorithms as follows. For the SBM algorithm, we set γ_k, p_{kl} ($1 \leq k, l \leq 2$) in [16] to $\gamma_1 = \gamma_2 = 0.5, p_{11} = p_{22} = \rho^2$ and $p_{12} = p_{21} = \rho^4$, where $\rho = 2M/[N(N - 1)]$. The Xiang algorithm has a parameter, denoted by $\beta \in [0, 1]$ in [32], to tune the number of core-periphery pairs. We set to $\beta = 1$. The Xiang algorithm uses a centrality measure to find core-periphery pairs. Therefore, we adopt the degree centrality measure. Note that the authors of [32] claim that the choice of the centrality measure does not considerably affect the results. With the Xiang algorithm, each node may belong to multiple core-periphery pairs. Therefore, if a node belongs to multiple core-periphery pairs, we assign the node to the core-periphery pair to which the extent of belonging is the largest. If a node belongs to multiple core-periphery pairs to the same extent, then we assign the node to one of the core-periphery pairs selected with equal probability. The other algorithms do not have parameters. As is the case of KM–config, the BE, SBM, Divisive and KM–ER algorithms are stochastic. Therefore, we run the BE, SBM, Divisive or KM–ER algorithm ten times and use the best core-periphery pairs in terms of the algorithm-specific quality function.

For the core-periphery pairs detected by the six previous algorithms, we carry out our previously proposed statistical test [19] that adopts the Erdős–Rényi random graph model as the null model. The statistical test runs as follows. Suppose that a network is composed of a single core-periphery pair. We generate 500 randomised networks using the Erdős–Rényi random graph with the same number of nodes and edges as the original network. Then, we detect a single core-periphery pair in each of the randomised networks using the BE algorithm and compute its quality by

$$\frac{\sum_{i=1}^N \sum_{j=1}^{i-1} (A_{ij} - \rho)(A_{ij}^* - \rho^*)}{\sqrt{\sum_{i=1}^N \sum_{j=1}^{i-1} (A_{ij} - \rho)^2} \sqrt{\sum_{i=1}^N \sum_{j=1}^{i-1} (A_{ij}^* - \rho^*)^2}}, \quad (17)$$

where A^* is given by equation (10) and $\rho^* = \sum_{i=1}^N \sum_{j=1}^{i-1} A_{ij}^*/[N(N - 1)/2]$. If the quality of the core-periphery pair detected in the original network is larger than a fraction $1 - \alpha$ of those detected in the randomised networks, then we regard the core-periphery pair in the original network as significant. It should be noted that this test is not applicable when the null model is the configuration model. If we use the configuration model as the null model, any core-periphery pair detected in the original network will be judged to be insignificant because no network is partitioned into a single core-periphery pair whose q value is larger than that for the configuration model.

If we detect multiple core-periphery pairs in the original networks, we apply the same statistical test for each of them [19]. Specifically, for each core-periphery pair, we construct a subnetwork composed of the nodes and edges within the focal core-periphery pair. Then, we apply the statistical test to the subnetwork. We correct the significance level using the Šidák correction [56]; we test each core-periphery pair in the original network at a significance level of $\alpha = 1 - (1 - \alpha')^{1/C}$, where $\alpha' = 0.05$ and C is the number of core-periphery pairs detected in the original network.

3.3. Data

We analyse the 12 empirical networks listed in table 1. We discard the direction and weight of the edge.

In the karate club network, each node represents the member of a university's karate club [44]. Two members are defined to be adjacent if they frequently interact outside the club activities. The club experienced a fissure as a result of a conflict between the instructor and the president. Based on their self-reports, each node has a label indicating either the instructor's side (15 members), president's side (16 members) or neutral (3 members).

In the dolphin social network, each node represents a dolphin living near Doubtful Sound in New Zealand [57]. An edge between two dolphins indicates that they were frequently observed in the same school during 1994 and 2001. Each dolphin has a label indicating the sex, i.e., female (25 dolphins), male (33 dolphins) and unknown (4 dolphins).

In the network of novel Les Misérables, each node is a character of the book [58]. Two characters are defined to be adjacent if they appear in the same chapter. The book consists of 365 chapters, most of which are a few pages long.

In the Enron email network, each node is an email account of the staff of Enron Inc [59]. An edge indicates that an email was sent from one account to another account during the observation period.

Table 1. Empirical networks used in the present paper: the karate club network [44], dolphin social network [57], network of characters in Les Misérables [58], Enron email network [59], network of jazz musicians [60], co-authorship network in network science [46], political blog network [5], worldwide airport network [61, 62], protein–protein interaction network [63], network of chess players [64], co-authorship network in the arXiv astro-ph section [65] and the Internet at the level of AS [64]. We exclude isolated nodes and self-loops from the networks. We count the multi-edges between a pair of nodes as a single edge.

Network	<i>N</i>	<i>M</i>	Assortativity	Degree	
				Average	Maximum
Karate [44]	34	78	-0.475	4.59	17
Dolphin [57]	62	159	-0.044	5.13	12
Les Misérables [58]	77	254	-0.165	6.60	36
Email [59]	151	1527	-0.059	20.23	74
Jazz [60]	198	2742	0.020	27.70	100
Network science (co-authorship) [46]	379	914	-0.082	4.82	34
Blog [5]	1222	16 714	-0.221	27.36	351
Airport [61, 62]	2939	15 677	0.051	10.67	242
Protein [63]	3023	6149	-0.126	4.07	129
Chess [64]	7115	55 779	0.371	15.68	181
Astro-ph (co-authorship) [65]	18 771	198 050	0.205	21.10	504
Internet [64]	34 761	107 720	-0.215	6.20	2760

In the jazz network, each node represents a jazz musician [60]. Two jazz musicians are defined to be adjacent if they have played in the same band.

In the co-authorship network, each node represents a researcher in network science [46]. An edge indicates that two researchers have a joint paper. The nodes and edges were retrieved from all the references cited by two influential review papers on network science. Then, the author of [46] manually added some nodes and edges and excluded those not belonging to the largest connected component.

In the blog network, each node represents a blog on the United States presidential election in 2004 [5]. Each edge indicates that one blog has a hyperlink to the other blog on its top page. The blogs and their labels were collected from several blog directories [5]. If a blog was unlabelled or had conflicting labels, the authors of [5] manually determined the label. There are 586 liberal blogs and 636 conservative blogs.

In the worldwide airport network, each node is an airport [61, 62]. An edge represents a direct commercial flight between two airports. We use the network provided in [62].

In the protein–protein interaction network, each node is a human protein [63]. An edge indicates the presence of physical interaction between two proteins.

In the network of chess players, each node represents a chess player [64]. Two players are adjacent if they have played before.

In the co-authorship network of the arXiv astro-ph section, each node is a researcher [65]. An edge indicates that two researchers have a joint paper in the arXiv’s astro-ph section.

In the network of the Internet, a node is an autonomous system (AS), i.e., a set of routers (or IP routing prefixes) managed by a network operator [64]. An edge indicates a logical peering relationship between two ASes.

4. Results

4.1. Quality and size of detected core-periphery pairs

The circles in figure 5 represent the quality and size (defined as the number of nodes) of core-periphery pairs detected by KM–config in the 12 empirical networks. A larger core-periphery pair tends to have a large quality, q . This is also the case for the randomised networks (crosses in figure 5). Some core-periphery pairs detected in the empirical networks have a significantly larger q value than those of the same size detected in the randomised networks. Our statistical test suggests that these core-periphery pairs are significant (circles outside the shaded regions in figure 5). We find bipartite-like pairs in the 7 out of the 12 networks (squares in figure 5), some of which are significant in 2 out of the 7 networks (figures 5(i) and (l)). In 2 out of the 12 networks, we find significant core-periphery pairs that are larger than any of those detected in the corresponding randomised networks (figures 5(g) and (l)).

4.2. Core nodes are not necessarily hub nodes

With KM–config, whether the node belongs to a core or periphery is not strongly associated with the node’s degree. To show this, we carry out a receiver operating characteristic (ROC) analysis (figure 6). Let us regard θN

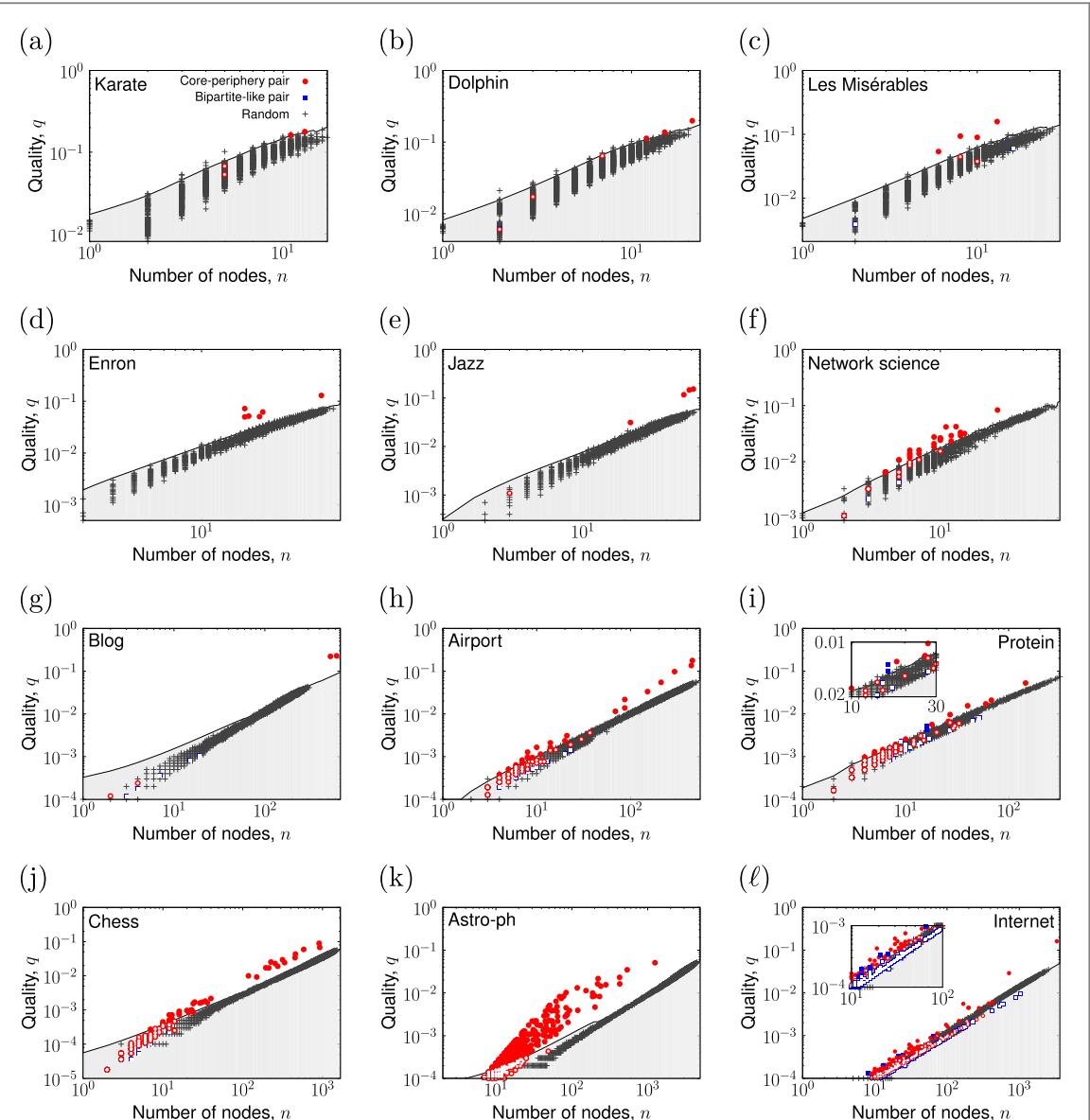


Figure 5. The quality of a core-periphery pair or bipartite-like pair, q , plotted against its number of nodes, n , in the 12 empirical networks and the corresponding randomised networks. We used the KM–config algorithm. The filled squares and filled circles represent significant core-periphery pairs and significant bipartite-like pairs, respectively. In the shaded regions, the detected core-periphery and bipartite-like pairs are insignificant, as represented by open circles and open squares, respectively. The crosses represent core-periphery or bipartite-like pairs detected in randomised networks. The insets in panels (i) and (l) magnify the region where many significant bipartite-like pairs lie.

nodes ($\theta \in \{0, 1/N, 2/N, \dots, 1\}$) with the largest degree as hub nodes and the remaining nodes as non-hub nodes. The ROC curves show the relationship between the fraction of hub nodes in the set of significant core nodes (i.e., true positive rate) and that in the set of significant peripheral nodes (i.e., false positive rate) when one varies the threshold θ . If all core nodes have a larger degree than all peripheral nodes, the ROC curve passes through $(0, 1)$ of the unit square (figure 6). If the degree of core nodes and that of peripheral nodes obey similar distributions, then the ROC curve is close to the diagonal line for the entire range of θ .

The area under the curve (AUC) of each ROC curve is shown in table 2. If the two distributions are completely separated, the AUC is equal to one. If they completely overlap, the AUC is equal to 0.5. The AUC values for the BE, MINRES, SBM, Xiang and Divisive algorithms are fairly large (mostly above 0.95) for all the networks. Therefore, these algorithms have a strong tendency to classify the nodes with a large degree as core nodes and those with a small degree as peripheral nodes. The AUC values for KM–ER are also large (above 0.81) but not as large as those for the five algorithms. Finally, KM–config determines the role (i.e., core or periphery) of each node by the degree of the node to the least extent, as suggested by the smallest AUC values across different networks among all the algorithms. These results on the AUC values are consistent with visual observations one can make in figure 6.

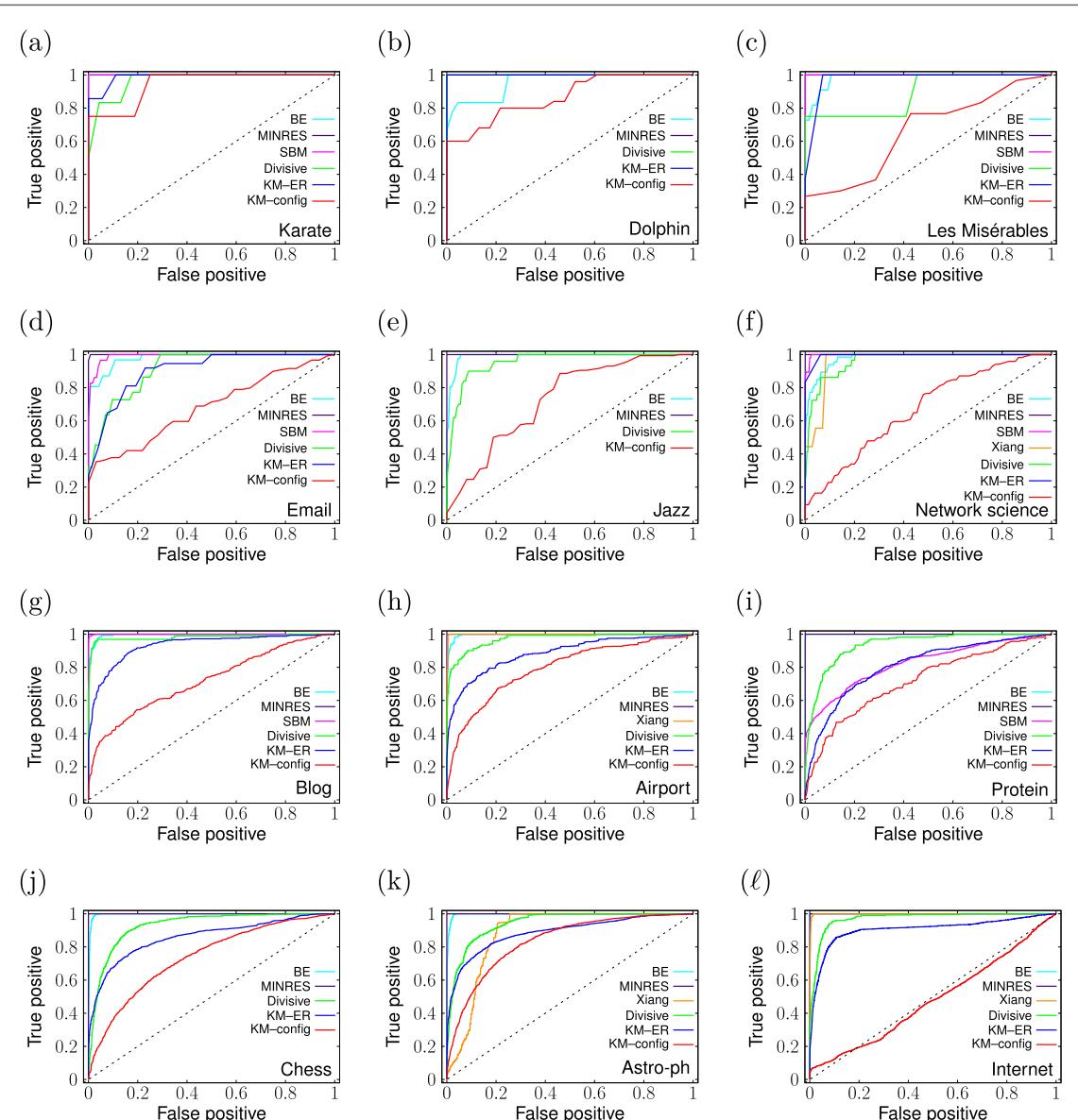


Figure 6. With KM-config, whether the node belongs to a core or periphery is not strongly associated with the node's degree. Each curve represents the relationship between the fraction of hub nodes in the set of significant core nodes (i.e., true positive rate) and that in the set of significant peripheral nodes (i.e., false positive rate). The dashed lines are the diagonal. The ROC curve is not shown if an algorithm does not detect any significant core-periphery pairs. Some ROC curves perfectly overlap on top of each other, and this occurs if and only if the ROC curves pass through $(0, 1)$. Most previous algorithms classify nodes into a core and periphery largely based on the degree of nodes.

Table 2. AUCs of the ROC curves shown in figure 6.

Network	BE	MINRES	SBM	Xiang	Divisive	KM-ER	KM-config
Karate	1.000	1.000	1.000	*	0.957	0.984	0.938
Dolphin	0.953	1.000	*	*	1.000	1.000	0.859
Les Misérables	0.982	1.000	1.000	*	0.886	0.955	0.610
Email	0.978	0.999	0.990	*	0.910	0.893	0.670
Jazz	0.989	1.000	*	*	0.953	*	0.717
Network science	0.979	1.000	0.998	0.958	0.961	0.990	0.664
Blog	0.995	1.000	0.999	*	0.981	0.932	0.718
Airport	0.996	1.000	*	0.999	0.972	0.885	0.793
Protein	1.000	1.000	0.824	*	0.936	0.810	0.717
Chess	0.997	1.000	*	*	0.928	0.860	0.737
Astro-ph	0.997	1.000	*	0.883	0.943	0.888	0.834
Internet	1.000	1.000	*	0.999	0.972	0.905	0.483

Note. The asterisk indicates that the algorithm does not detect significant core-periphery pairs.

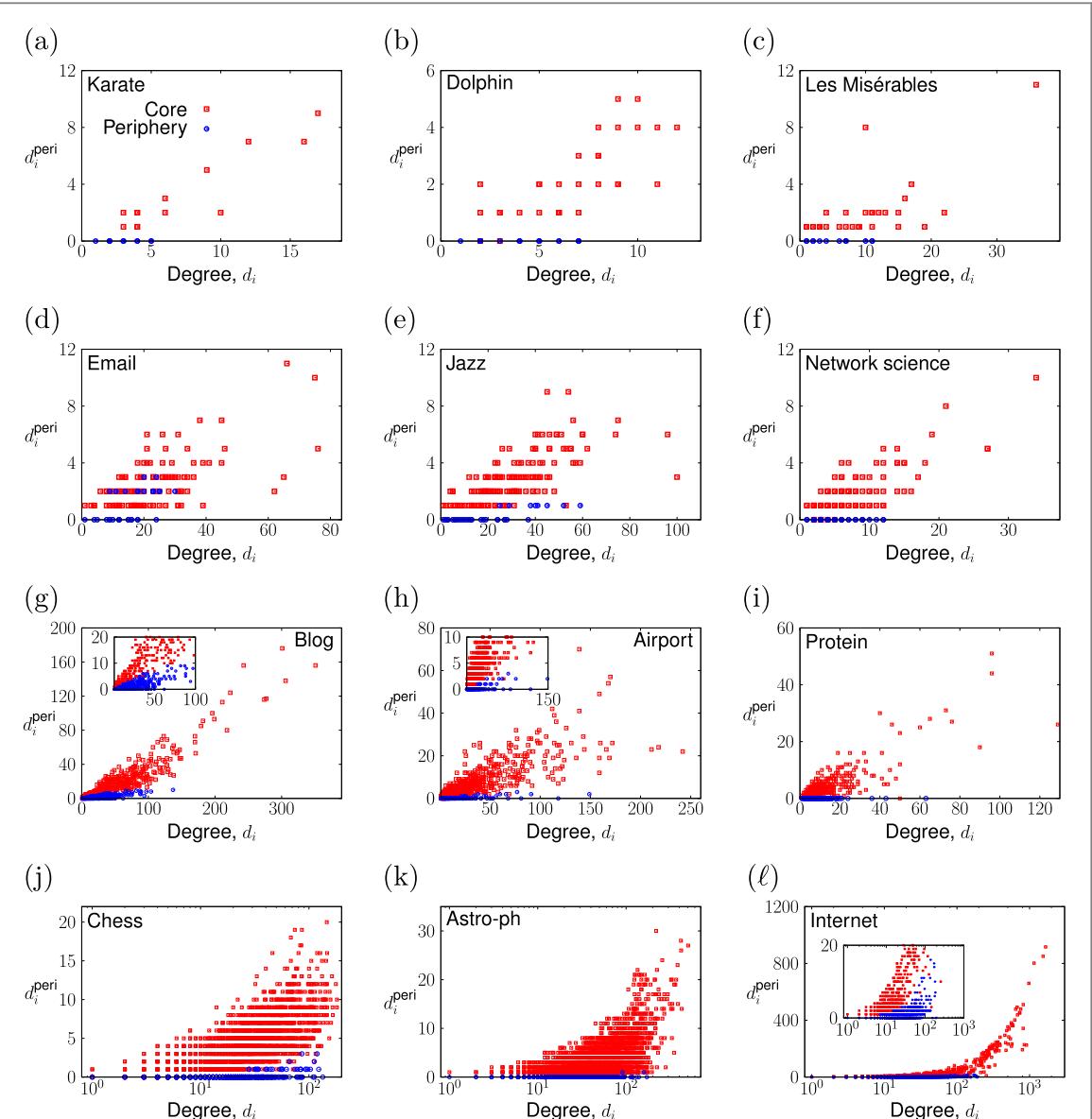


Figure 7. Relationships between d_i and d_i^{peri} for the empirical networks. The squares and circles indicate core nodes and peripheral nodes identified by KM–config, respectively. The insets of the panels (g), (h) and (l) magnify the regions with small d_i^{peri} values.

To illuminate on the meaning of the core and peripheral nodes detected by KM–config, let us denote by d_i^{core} and d_i^{peri} the number of neighbouring core and peripheral nodes of node i , respectively, within the core-periphery pair to which node i belongs. For each node i , we plot d_i^{peri} against degree d_i in figure 7. We find that peripheral nodes are adjacent to a smaller number of peripheral nodes within the same core-periphery pair (i.e., a small d_i^{peri}) than core nodes with a similar d_i value would do. This result is consistent with the concept of core-periphery structure based on edge density. However, this property is not necessarily respected if one classifies nodes according to the degree of each node. In fact, with the other six algorithms, core nodes and peripheral nodes are less distinct from each other in terms of d_i^{peri} (figures D1–D6 in appendix D). As a corollary, with KM–config, the peripheral nodes tend to be more frequently connected to the core nodes within the same core-periphery pair than the core nodes do (figure 8). For some core nodes with a large degree, d_i^{core} is equal to zero, which happens when a core-periphery pair has only one core node and forms a star.

4.3. A core-periphery pair is a community?

We compare the core-periphery pairs identified by KM–config and communities in networks. Here we determine communities by modularity maximisation using the Louvain algorithm [52]. We run it ten times and adopt the node partition that realises the largest modularity value. Table 3 reports the modularity values for the node partition identified by the Louvain algorithm and that determined by KM–config, with the insignificant

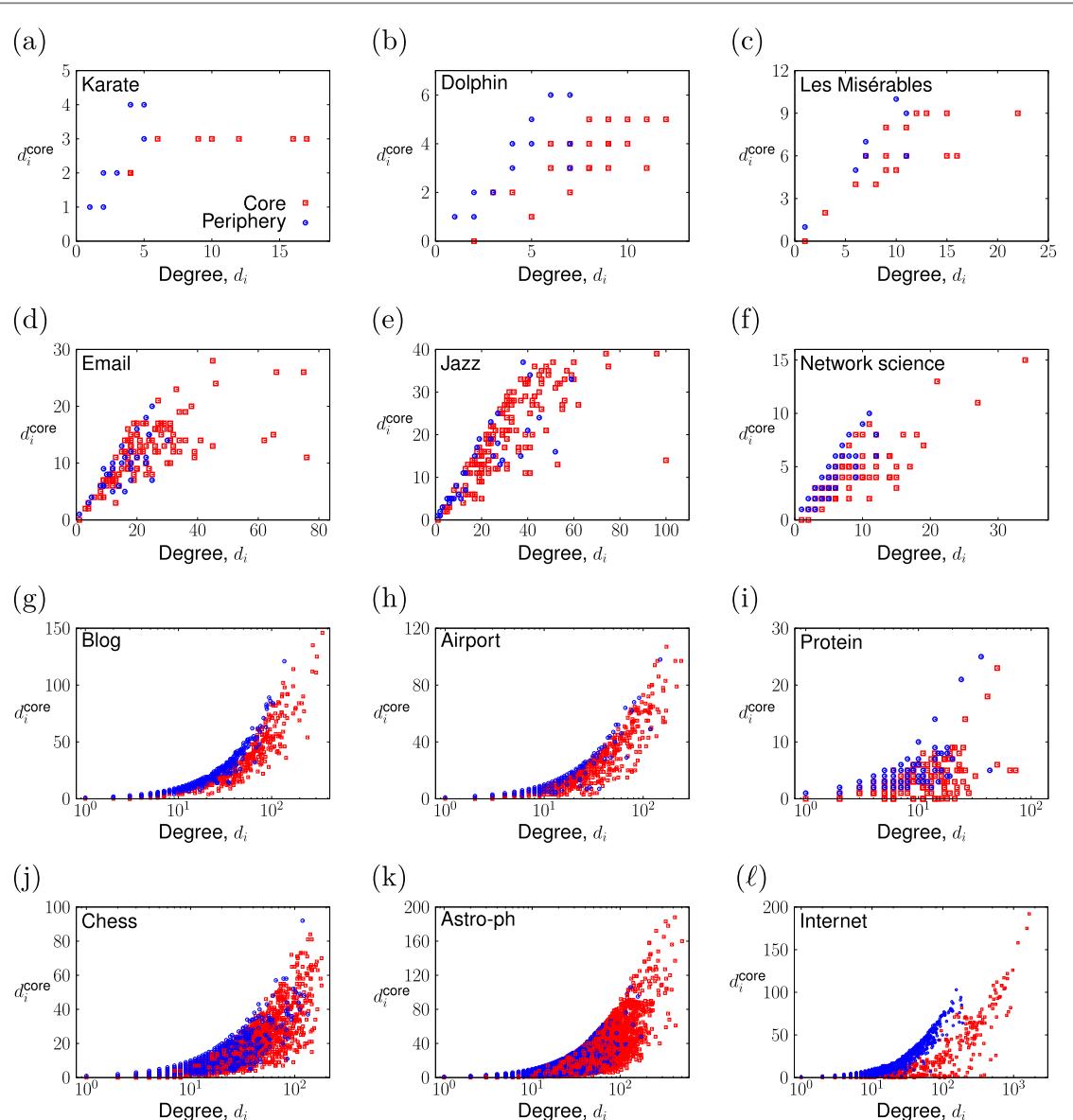


Figure 8. Relationships between d_i and d_i^{core} for the empirical networks. The squares and circles indicate core nodes and peripheral nodes identified by KM-config, respectively.

Table 3. Modularity for communities determined by the Louvain algorithm and that for the core-periphery pairs determined by KM-config.

Network	Louvain	KM-config
Karate	0.416	0.417
Dolphin	0.520	0.518
Les Misérables	0.535	0.542
Email	0.420	0.419
Jazz	0.445	0.445
Network science	0.815	0.741
Blog	0.426	0.426
Airport	0.642	0.615
Protein	0.626	0.483
Chess	0.505	0.508
Astro-ph	0.574	0.555
Internet	0.521	0.459

core-periphery pairs being included. The modularity value for the node partitioning into core-periphery pairs is close to that obtained by the modularity maximisation for most of the empirical networks. Therefore, the detected core-periphery pairs may be similar to communities.

This result poses a question whether a core-periphery pair is a community in the traditional sense, and if so whether the KM–config algorithm effectively classifies the nodes in each community into a core and a periphery according to the composition of intra- and inter-community edges that each node owns. To examine this point, we analyse the role of each node using a cartographic representation of networks [66, 67]. With the cartographic representation, the role of each node i in a network is characterised by the standardised within-module degree $z_i \in [-\infty, \infty]$ and the participation coefficient $p_i \in [0, 1]$ [66, 67]. They are defined by

$$z_i \equiv \frac{\tilde{d}_{i,c_i} - \langle \tilde{d}_{c_i} \rangle}{\sigma_{c_i}}, \quad (18)$$

$$p_i \equiv 1 - \sum_{c=1}^C \left(\frac{\tilde{d}_{i,c}}{d_i} \right)^2, \quad (19)$$

where $\tilde{d}_{i,c}$ is the number of neighbours of node i in the c th core-periphery pair ($1 \leq c \leq C$), c_i is the core-periphery pair to which node i belongs, $\langle \tilde{d}_{c_i} \rangle$ is the average of \tilde{d}_{j,c_i} over the nodes j in the c_i th core-periphery pair including the case $j = i$, and σ_{c_i} is the unbiased estimation of the standard deviation of \tilde{d}_{j,c_i} over the nodes j in the c_i th core-periphery pair. A large z_i value indicates that node i has relatively many neighbours within the same core-periphery pair. The p_i value is the smallest if node i is adjacent only to the nodes in a single core-periphery pair and largest if node i is adjacent to an equal number of nodes across all core-periphery pairs. In the cartographic representation of networks, each node i is classified according to the position (z_i, p_i) of the node in the z - p space. The nodes are categorised into seven roles [66, 67]. Here we do not use this categorisation rule but examine the distributions of the core and peripheral nodes in the z - p space.

Figure 9 shows z_i and p_i of each node for the 12 empirical networks. KM–config classifies the nodes having very large z_i as core nodes. However, for the other nodes, the values of z_i and p_i are not predictive of whether a node is in the core or periphery. Therefore, the core and periphery that we propose are distinct from the roles of nodes identified by the cartographic analysis.

We find different results for the Divisive algorithm, which first divides the network into communities and then estimates the role of each node (i.e., core or periphery). With Divisive, the core nodes have larger z_i than most of the peripheral nodes (figure D7 in appendix D), indicating that the core nodes detected by Divisive largely correspond to the hub nodes as identified by the cartographic analysis. This is because Divisive uses the BE algorithm to partition each community into a core and a periphery. As shown in figure 6, the BE algorithm classifies nodes into a core and a periphery by the degree of each node to a large extent. Therefore, Divisive regards the nodes with a large z_i as core nodes.

4.4. Case studies

In this section, we present case studies of some of the empirical networks analysed in the previous sections.

The core-periphery pairs in the karate club network are shown in figure 10. KM–config detected two significant core-periphery pairs and ten residual nodes. A majority of the members on the president side (12 members; 75%), including the president (node 34), belong to core-periphery pair 1. A majority of the members on the instructor side (11 members; 73%), including the instructor (node 1), belong to core-periphery pair 2. These results are consistent with the social conflict of the club. The residual consists of four members on the instructor side, five members on the president side and one neutral member. The significant core-periphery pairs are similar to those detected by our previous algorithm, KM–ER [19].

The core-periphery pairs in the dolphin social network are shown in figure 11. KM–config detected three significant core-periphery pairs and 14 residual nodes. Each core-periphery pair mostly consists of the dolphins of the same sex; there are two male-dominant core-periphery pairs (pairs 1 and 3) and one female-dominant core-periphery pair (pair 2). A previous study identified five communities in the dolphin network by modularity maximisation [4], three of which are similar to the present core-periphery pairs 1, 2 and 3.

For the network of Les Misérables, KM–config identified four significant core-periphery pairs and 40 residual nodes (figure 12). A majority of nodes belonging to the significant core-periphery pairs are core nodes, suggesting that each core-periphery pair resembles a community. In fact, a previous study used modularity maximisation to identify 11 communities in the same network [4], four of which are similar to our core-periphery pairs 1–4. The significant core-periphery pairs are consistent with the plot of the story; the characters in core-periphery pairs 1, 2, 3 and 4 are the members of a revolutionary student club, Thénardier family and a street gang, Fantine's relatives and her friends, and characters involved in the Champmathieu's trial, respectively. The main characters, e.g., Valjean, Javert and Cosette, are classified as residual nodes (arrows in

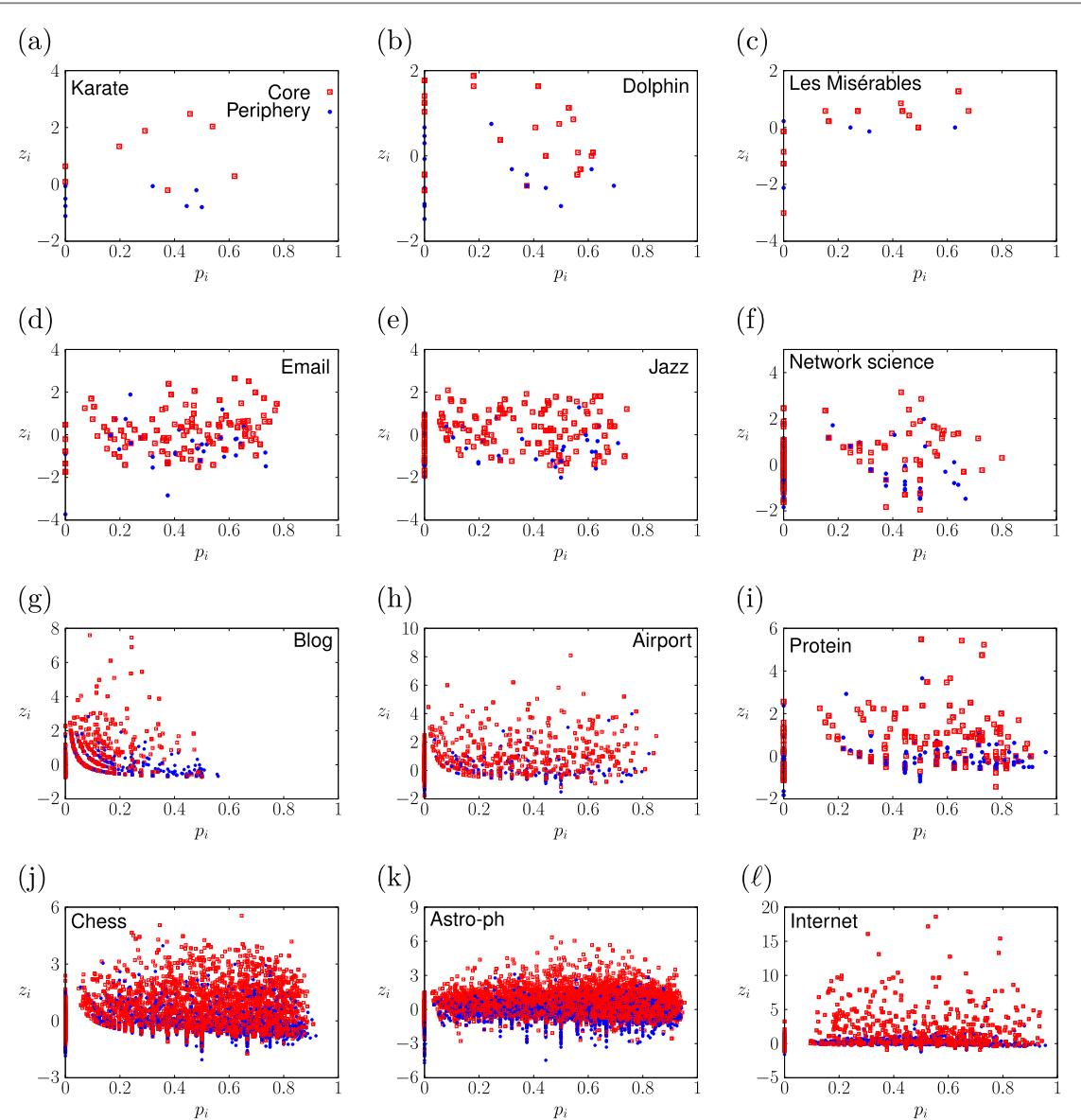


Figure 9. Cartographic analysis of the empirical networks. The squares and circles indicate core nodes and peripheral nodes identified by KM–config, respectively.

figure 12). Although they have a large degree, they are regarded as residual nodes because they belong to insignificant core-periphery pairs.

For the co-authorship network, KM–config detected 28 significant core-periphery pairs and 133 residual nodes (figure 13(a)). Detailed structure of core-periphery pairs 1–10 is shown in figure 13(b). Five core-periphery pairs (pairs 1, 5, 8, 9 and 10) have relatively many intra-core edges, many core-periphery edges and no intra-periphery edges, indicating a strong core-periphery structure. Core-periphery pair 8 contains only one peripheral node, implying a structure close to a community. Some core researchers collaborate with most of the researchers in the same core-periphery pair, e.g., Barabási, Jeong and Oltvai in core-periphery pair 1, Vázquez and Vespignani in core-periphery pair 2 and Boccaletti in core-periphery pair 4.

For the blog network, KM–config identified two core-periphery pairs and 79 residual nodes (figure 14). A majority of the blogs leaning to the conservative and to the liberal belong to core-periphery pairs 1 and 2, respectively. These core-periphery pairs are similar to those identified by KM–ER [19].

For the airport network, KM–config identified 23 significant core-periphery pairs and 983 residual nodes (figure 15). Each core-periphery pair mainly consists of the airports in the same geographical region, which agrees with the previous results [19, 67, 68]. Our previous algorithm, KM–ER, detected ten core-periphery pairs, of which the three largest core-periphery pairs based in Europe, East Asia and the USA are similar to core-periphery pairs 1, 3 and 2 detected by KM–config, respectively [19]. Properties of core-periphery pairs 1–8 are shown in table 4. Among the representative airports (i.e., the airports having the largest degree in each core-periphery pair), some peripheral airports have a larger degree than core airports, e.g., MUC (Munich) in



Figure 10. Significant core-periphery pairs in the karate club network. The rows and columns correspond to nodes. The filled and open cells represent the presence and absence of an edge, respectively. The square box bordered by the solid lines inside the adjacency matrix represents a significant core-periphery pair. The dotted lines represent the boundary between the core and periphery in the core-periphery pair. The colour of the node indicates the label of the member.

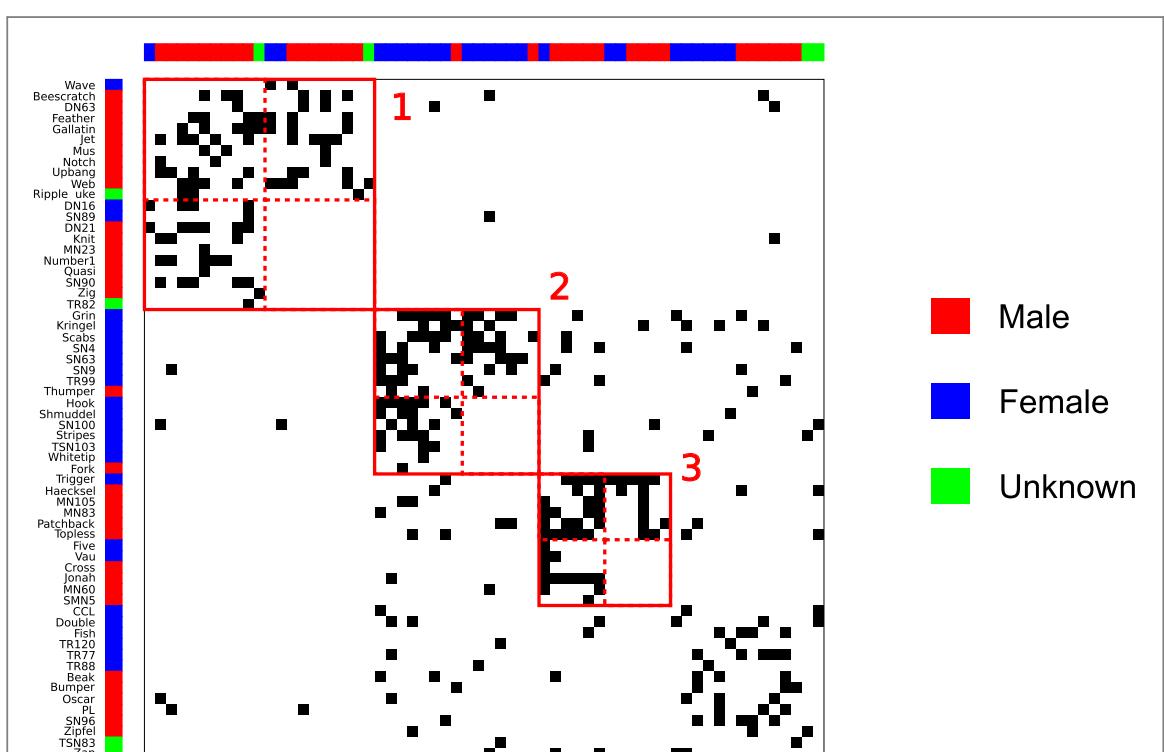


Figure 11. Significant core-periphery pairs in the dolphin social network. The colour indicates the sex of the individual.

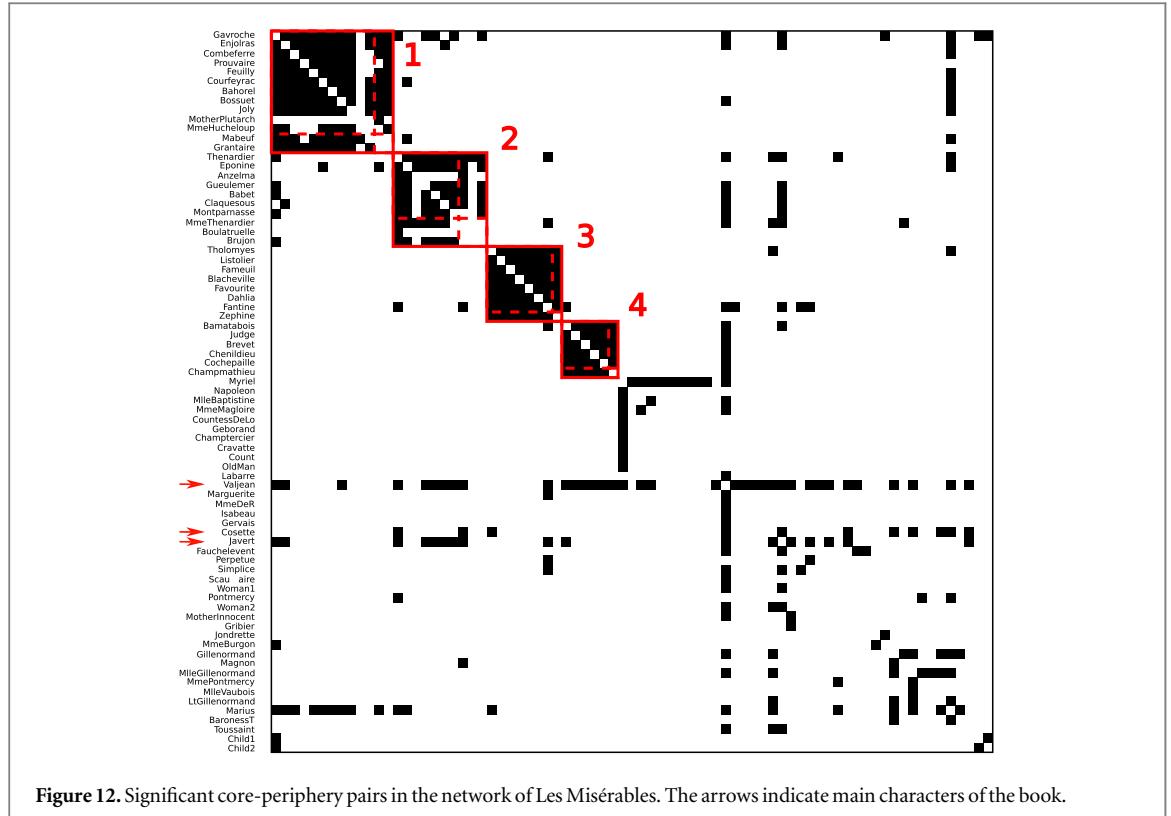


Figure 12. Significant core-periphery pairs in the network of *Les Misérables*. The arrows indicate main characters of the book.

core-periphery pair 1, SVO (Moscow) in core-periphery pair 5 and NBO (Nairobi) in core-periphery pair 8, showing that hub nodes are not always classified as core nodes.

4.5. Synthetic networks

The results in the previous sections suggest that KM–config tends to detect core-periphery pairs without using the node's degree as a main criterion but produces node partitioning consistent with the concept of core-periphery structure based on edge density. To confirm this point further, in this section we test the algorithms on model networks with a planted core-periphery structure composed of two core-periphery pairs (figure 16).

The discrepancy between the degree distribution of core nodes and that of peripheral nodes is controlled by a parameter $\mu \in \{0, 0.05, 0.1, \dots, 0.5\}$. The ‘strength’ of the core-periphery structure is controlled by a parameter $\lambda \in \{0, 0.025, 0.05, \dots, 1\}$. The model assumes four blocks. Each block consists of 200 nodes and represents a core or periphery. To generate networks, we use the degree-corrected SBM (dcSBM) [6]; it places edges such that each node i has a prescribed expected degree \bar{d}_i , and each pair of blocks u and v has an expected number \bar{m}_{uv} of edges. We set \bar{d}_i ($1 \leq i \leq N$) as follows. For the core (i.e., blocks (1, 1) and (2, 1)), we set $\bar{d}_i = 50$ for a fraction μ of nodes and $\bar{d}_i = 200$ for the remaining fraction $1 - \mu$ of nodes. For the periphery (i.e., blocks (1, 0) and (2, 0)), we set $\bar{d}_i = 50$ for a fraction $1 - \mu$ of nodes and $\bar{d}_i = 200$ for the remaining fraction μ of nodes. The fraction μ tunes the amount of overlap between the degree distribution of core nodes and that of peripheral nodes. The two distributions have no overlap if $\mu = 0$ and perfectly overlap if $\mu = 0.5$. Then, we set $\bar{m}_{(c,x)(c',x')} (1 \leq c, c' \leq 2 \text{ and } 0 \leq x, x' \leq 1)$ by

$$\bar{m}_{(c,x)(c',x')} = \lambda \bar{m}_{(c,x)(c',x')}^{\text{rand}} + (1 - \lambda) \bar{m}_{(c,x)(c',x')}^{\text{plant}}, \quad (20)$$

where λ is a mixing parameter, $\bar{m}_{(c,x)(c',x')}^{\text{rand}} = D_{(c,x)}D_{(c',x')}/2M$ is the expected number of edges between blocks (c, x) and (c', x') for the configuration model given $\bar{d}_1, \dots, \bar{d}_N$. The parameters $\{\bar{m}_{(c,x),(c',x')}^{\text{plant}}\}$ represent the number of intra-block and inter-block edges in an idealised core-periphery structure with nodes' degrees $\bar{d}_1, \dots, \bar{d}_N$. In other words, there are no edges between the different core-periphery pairs ($\bar{m}_{(c,x),(c',x')}^{\text{plant}} = 0$ for $c \neq c'$), the peripheral nodes are adjacent only to core nodes in the same core-periphery pair (i.e., $\bar{m}_{(c,0),(c,1)}^{\text{plant}} = D_{(c,0)}$ and $\bar{m}_{(c,0),(c,0)}^{\text{plant}} = 0$ for $c = 1, 2$), and the number of edges within each core is given by $\bar{m}_{(c,1),(c,1)}^{\text{plant}} = D_{(c,1)} - D_{(c,0)}$ for $c = 1, 2$. Although the dcSBM with $\{\bar{m}_{(c,x),(c',x')}^{\text{plant}}\}$ specifies a disconnected network, the dcSBM with $\{\bar{m}_{(c,x),(c',x')}^{\text{plant}}\}$ given by equation (20) produces connected networks in general unless $\lambda = 0$. We note that, if $\mu = 0.5$, the dcSBM generates bipartite networks because the number of intra-core edges is zero, i.e., $\bar{m}_{(c,1)(c,1)}^{\text{plant}} = 0$ ($c = 1, 2$).

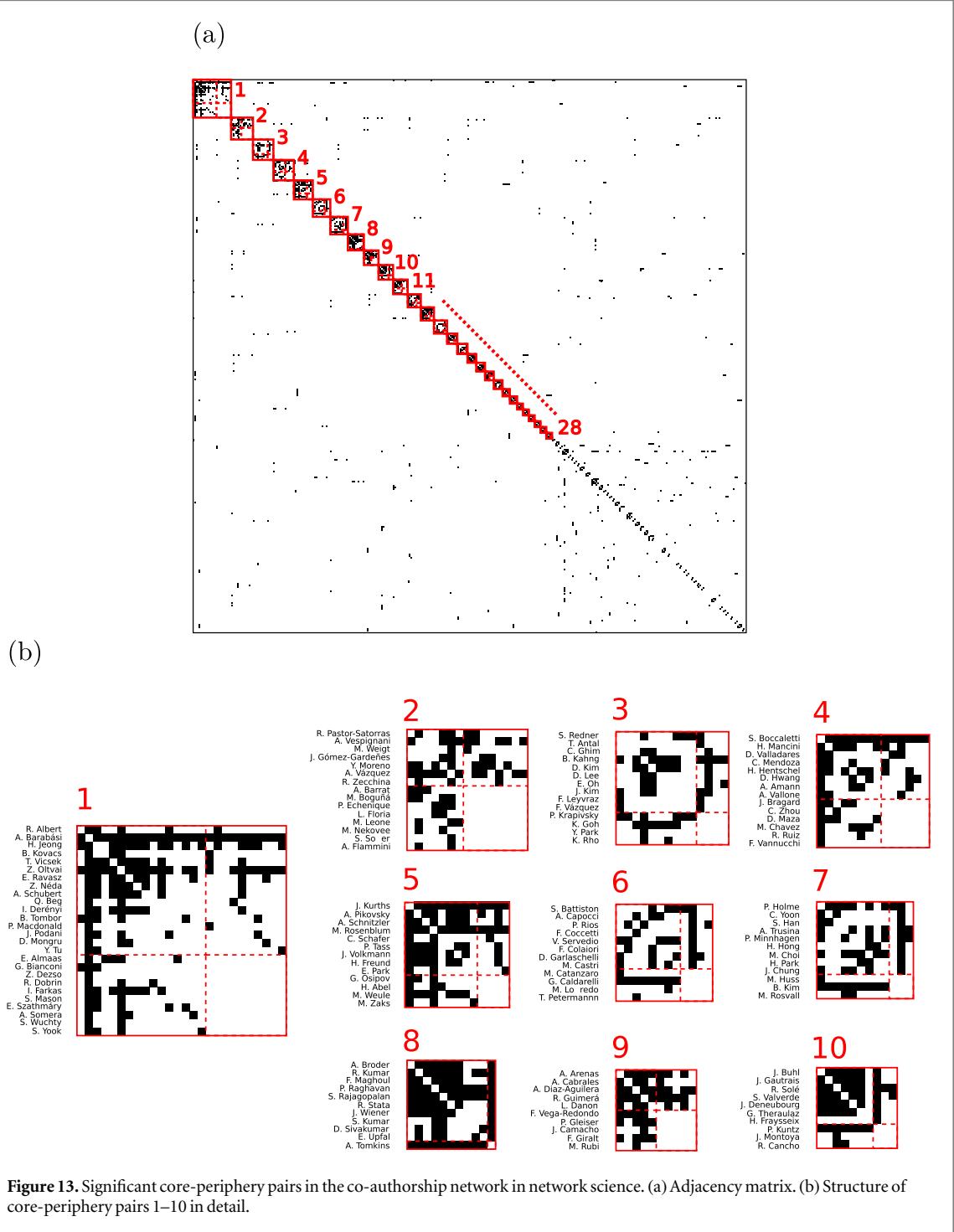


Figure 13. Significant core-periphery pairs in the co-authorship network in network science. (a) Adjacency matrix. (b) Structure of core-periphery pairs 1–10 in detail.

We evaluate the performance of algorithms by the difference between the planted and detected core-periphery structures. To quantify the difference, we use the variation of information (VI) [69] given by

$$\text{VI} = - \sum_{(c,x)} \sum_{(\hat{c},\hat{x})} R(c, x; \hat{c}, \hat{x}) \log \frac{[R(c, x; \hat{c}, \hat{x})]^2}{\left[\sum_{(\hat{c}',\hat{x}')} R(c, x; \hat{c}', \hat{x}') \right] \times \left[\sum_{(c',x')} R(c', x'; \hat{c}, \hat{x}) \right]}, \quad (21)$$

where $R(c, x; \hat{c}, \hat{x})$ is the fraction of nodes having true label (c, x) and inferred label (\hat{c}, \hat{x}) . The VI value is the smallest (i.e., zero) if and only if the partitioning of nodes by the true labels and that by the inferred labels are identical. In the computation of the VI values, we regard the set of residual nodes as a block; technically, we set $(\hat{c}_i, \hat{x}_i) = (C + 1, 0)$ for the residual nodes. We generate 30 synthetic networks and average the VI values over the 30 generated networks.

The VI values for the Xiang, Divisive, KM-ER and KM-config algorithms are shown in figure 17. We do not show the results for the other three algorithms because they do not find multiple core-periphery pairs by definition.

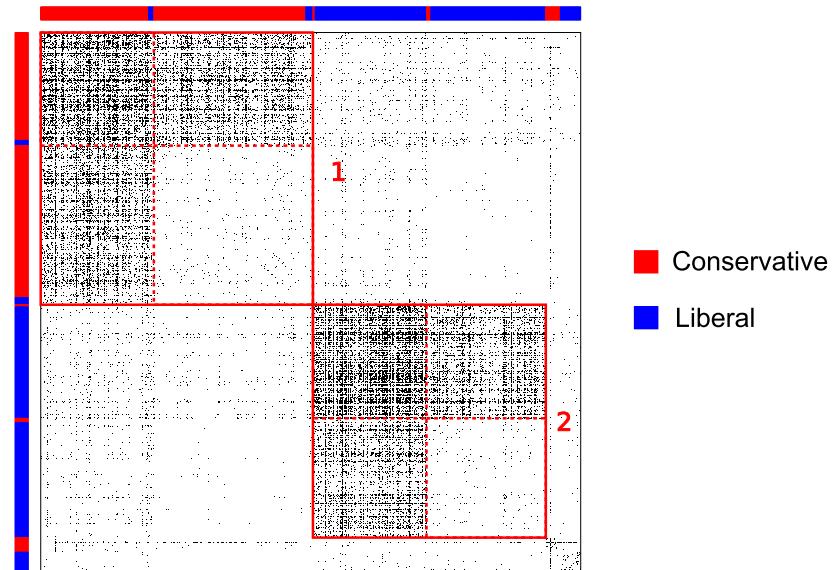


Figure 14. Significant core-periphery pairs in the political blog network. The colour of the node indicates the political leaning of individual blogs.

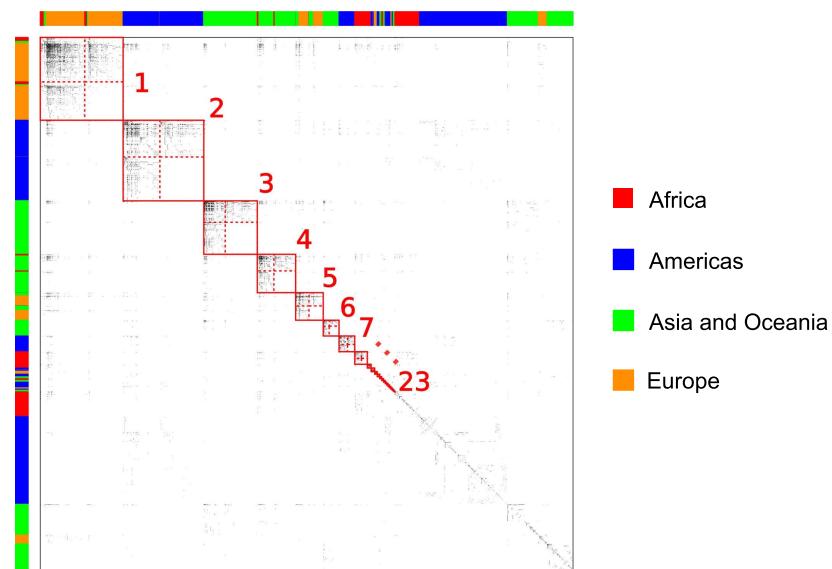


Figure 15. Significant core-periphery pairs in the airport network. The colour indicates the geographical region of the airports. Americas is the union of North, Central and South America.

When λ is large, the VI values are large because the network is close to the configuration model and has weak core-periphery structure. The VI values for the Xiang algorithm are large in the entire $\lambda-\mu$ parameter space (figure 17(a)). This is because the Xiang algorithm did not find significant core-periphery pairs in all the generated networks (i.e., all nodes are residual nodes). The VI values for the Divisive and KM-ER algorithms are relatively large for most λ values if some planted core nodes are non-hub nodes, i.e., $\mu \geq 0.15$ (figures 17(b) and (c)). The VI values for the KM-config algorithm are the smallest in most of the $\lambda-\mu$ parameter space (figure 17(d)), including the case for bipartite-like structure ($\mu \geq 0.4$). Therefore, KM-config but not the other algorithms is capable of detecting core-periphery structure even when a substantial fraction of core nodes are non-hubs and peripheral nodes are hubs.

5. Discussion

We have studied core-periphery structure using the configuration model as the null model. We have shown that discrete versions of a single core-periphery pair determined based on edge density, which many studies assume,

Table 4. Property of the eight largest significant core-periphery pairs in the airport network. The representative airports of each core-periphery pair are defined as the four core and four peripheral airports having the largest degree. The territory is defined as the country where the airport is located. If the airport is located in a sovereign state, we instead show the name of the state. IATA is a three-letter code of an airport assigned by the International Air Transport Association.

Pair	Number of airports		Representative core airport				Representative peripheral airport			
	Core	Periphery	IATA	City	Territory	Degree	IATA	City	Territory	Degree
1	245	212	FRA	Frankfurt	Germany	242	MUC	Munich	Germany	149
			CDG	Paris	France	218	OSL	Oslo	Norway	91
			AMS	Amsterdam	Netherlands	211	BUD	Budapest	Hungary	77
			LGW	London	UK	172	LPL	Liverpool	UK	66
2	203	242	ATL	Atlanta	USA	168	MEX	Mexico City	Mexico	81
			JFK	New York	USA	144	LGA	New York	USA	44
			LAS	Las Vegas	USA	139	CCS	Caracas	Venezuela	41
			YYZ	Toronto	Canada	119	SXM	Philipsburg	Netherlands Antilles	35
3	118	177	PEK	Beijing	China	170	HGH	Hangzhou	China	53
			BKK	Bangkok	Thailand	136	KIX	Osaka	Japan	47
			PVG	Shanghai	China	126	NGO	Nagoya	Japan	32
			ICN	Seoul	South Korea	121	WNZ	Wenzhou	China	32
4	92	120	DXB	Dubai	UAE	166	BAH	Bahrain	Bahrain	52
			JED	Jeddah	Saudi Arabia	99	MRU	Port Louis	Mauritius	29
			DEL	Delhi	India	93	MLE	Malé	Maldives	27
			DOH	Doha	Qatar	92	KBL	Kabul	Afghanistan	23
5	73	79	DME	Moscow	Russia	159	SVO	Moscow	Russia	118
			LED	St. Petersburg	Russia	92	DYU	Dushanbe	Tajikistan	26
			KBP	Kiev	Ukraine	76	ODS	Odessa	Ukraine	15
			TLV	Tel-aviv	Israel	69	YKS	Yakutsk	Russia	14
6	33	54	SYD	Sydney	Australia	77	DRW	Darwin	Australia	20
			MEL	Melbourne	Australia	56	TSV	Townsville	Australia	11
			BNE	Brisbane	Australia	51	NOU	Nouméa	New Caledonia	10
			AKL	Auckland	New Zealand	38	WLG	Wellington	New Zealand	6

Table 4.(Continued.)

Pair	Number of airports		Representative core airport				Representative peripheral airport			
	Core	Periphery	IATA	City	Territory	Degree	IATA	City	Territory	Degree
7	47	40	GRU	São Paulo	Brazil	83	BEL	Belém	Brazil	17
			GIG	Rio De Janeiro	Brazil	46	CGR	Campo Grande	Brazil	16
			BSB	Brasília	Brazil	46	MVD	Montevideo	Uruguay	13
			CNF	Belo Horizonte	Brazil	33	NAT	Natal	Brazil	12
8	37	32	JNB	Johannesburg	South Africa	70	NBO	Nairobi	Kenya	68
			ADD	Addis Ababa	Ethiopia	58	WDH	Windhoek	Namibia	9
			LAD	Luanda	Angola	30	PNR	Pointe-noire	Congo (Republic)	8
			DAR	Dar Es Salaam	Tanzania	23	NDJ	N'djamena	Chad	8

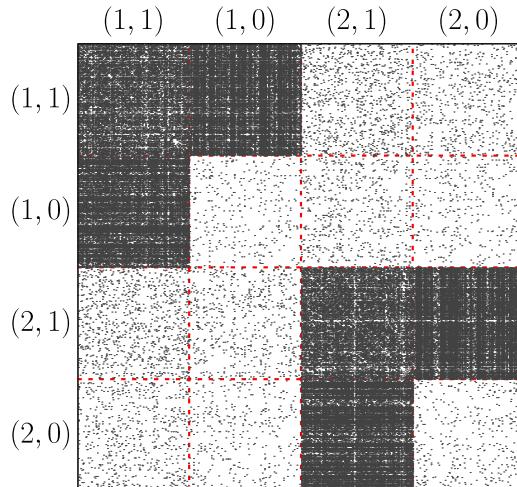


Figure 16. Adjacency matrix of a synthetic network generated with $\lambda = 0.1$ and $\mu = 0.7$. The dashed lines represent the boundaries between blocks. The label (c, x) of each block is shown at the top and left of the adjacency matrix.

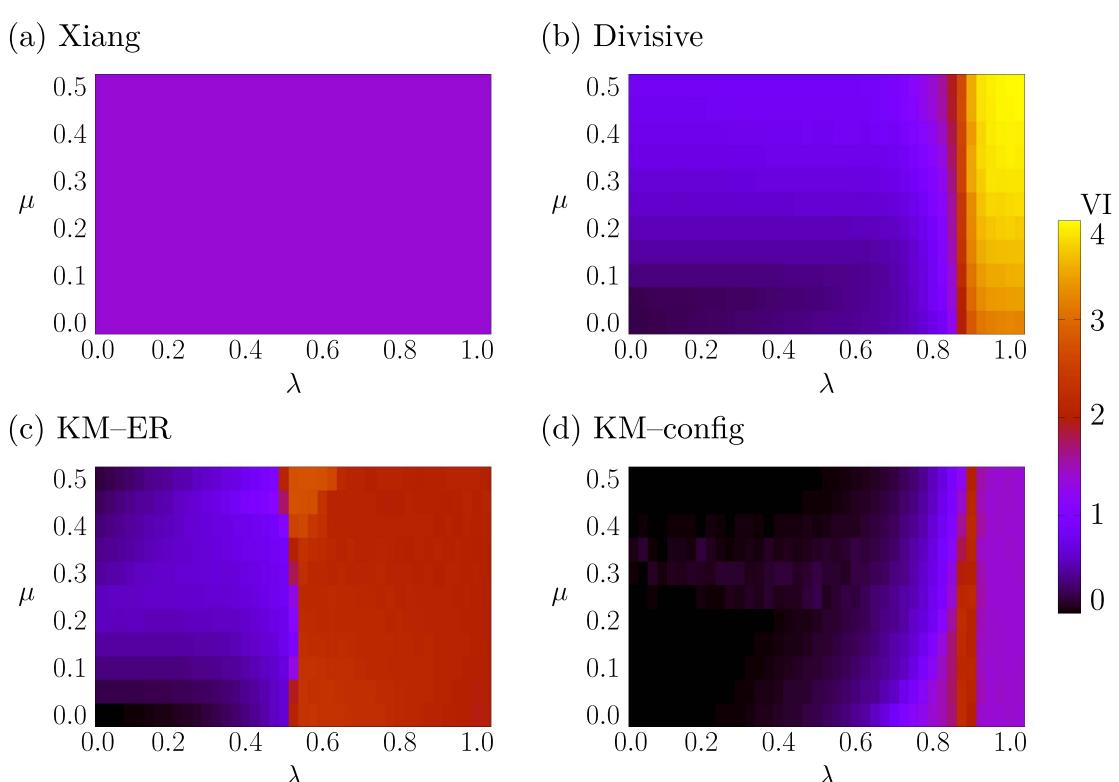


Figure 17. VI values between the planted and inferred core-periphery structure. (a) Xiang. (b) Divisive. (c) KM-ER. (d) KM-config.

can never be significant relative to the configuration model. The core-periphery structure beyond what one expects for the configuration model must accompany other meso-scale network structure such as another core-periphery pair, communities and bipartite-like subnetworks coexisting in the given network. This claim is in resonance with the studies [19, 28, 30] reporting the absence of core-periphery structure when the configuration model is used as the null model. Then, we have presented a scalable algorithm to find core-periphery structure in networks and applied it to various networks.

Our argument does not apply to continuous versions of core-periphery structure [8, 10, 14, 17, 25], in which each node belongs to the core to a different extent. A possible extension of our present algorithm (i.e.,

KM–config) to the case of continuous core-periphery structure is to replace the idealised core-periphery structure defined in equation (10) with a continuous version of idealised core-periphery structure, such as those proposed in [8, 10]. This line of investigation may reveal relationships between continuous versions of core-periphery structure, multiple core-periphery pairs and the configuration model.

Null models for networks do not have to be limited to the Erdős–Rényi random graph or the configuration model. Other null models incorporate different properties of networks such as the weight of edges [70], the sign of edge weights [71], correlations [72], bipartiteness [73] and spacial properties [74]. It is probably possible to incorporate such null models into our algorithm by modifying the null-model term of $Q_{\text{config}}^{\text{cp}}$ (i.e., the second term on the right-hand side of equation (13)).

Akin to the modularity, our quality function $Q_{\text{config}}^{\text{cp}}$ allows an interpretation in terms of random walks on networks. With a core-periphery structure, random walkers are likely to move from any node to a core node within the same core-periphery pair in a discrete time step. In a core-periphery structure on a small scale, the random walkers would reach the core in a small number of steps. In contrast, they would need a large number of steps to reach the core on a large scale. By regarding the number of steps as a resolution parameter, we may be able to identify core-periphery structure across different scales, as in the case of the Markov stability, where modularity maximisation with different values of the time resolution parameter provides information about hierarchical organisation of communities in networks [47–50].

Our quality function $Q_{\text{config}}^{\text{cp}}$ shares shortcomings with the modularity, such as the inability of finding small communities [75] and of distinguishing random from non-random structure [76]. Remedies for these problems include multi-resolution approaches [77] and statistical tests [78, 79]. Another approach is the statistical inference based on SBMs, which has been used for finding communities [6, 16, 30, 41] and core-periphery structure [16, 24, 30]. Investigation of core-periphery structure with SBMs may be a topic for future study.

Finally, we have restricted ourselves to undirected and unweighted networks. It is straightforward to incorporate the weight of edges by replacing A_{ij} on the right-hand side of equation (13) by the weight of the edge between nodes i and j . In contrast, it is non-trivial to incorporate the direction of edges. It seems that the direction of edges can be incorporated into $Q_{\text{config}}^{\text{cp}}$ by allowing an adjacency matrix to be asymmetric, as in the case of modularity [80, 81]. However, for modularity, this extension elicits a problem [82], which may also hold true for $Q_{\text{config}}^{\text{cp}}$.

Appendix A. Finding compatible block structures

We represent a block structure composed of B blocks by a symmetric $B \times B$ matrix $\mathbf{Y} = (Y_{uv})$, where

$$Y_{uv} = \begin{cases} 1 & (m_{uv} > \mathbb{E}[m_{uv}^{\text{conf}}]), \\ -1 & (m_{uv} < \mathbb{E}[m_{uv}^{\text{conf}}]). \end{cases} \quad (\text{A1})$$

Variable $Y_{uv} = 1$ or $Y_{uv} = -1$ indicates that blocks u and v are either densely (i.e., $m_{uv} > \mathbb{E}[m_{uv}^{\text{conf}}]$; filled blocks in figure 2) or sparsely (i.e., $m_{uv} < \mathbb{E}[m_{uv}^{\text{conf}}]$; opened blocks in figure 2) interconnected. Recall that we do not consider the case $m_{uv} = m_{uv}^{\text{conf}}$ because it is unlikely in general. To find compatible block structures, we generate all $2^{B(B+1)/2}$ symmetric binary matrices. Then, for each binary matrix \mathbf{Y} , we inspect the compatibility of the block structure as follows. Equation (9) can be rewritten as

$$\sum_{v=1}^B \Delta_{uv} = 0, \quad 1 \leq u \leq B, \quad (\text{A2})$$

where

$$\Delta_{uv} = m_{uv} - \mathbb{E}[m_{uv}^{\text{conf}}]. \quad (\text{A3})$$

Equations (A1) and (A3) imply

$$Y_{uv} \Delta_{uv} > 0, \quad 1 \leq u, v \leq B. \quad (\text{A4})$$

Because $Y_{uv} \neq 0$, we rewrite equation (A4) as

$$Y_{uv} \Delta_{uv} \geq 0, \quad 1 \leq u, v \leq B, \quad (\text{A5})$$

$$\Delta_{uv}^2 \geq \eta > 0, \quad 1 \leq u, v \leq B. \quad (\text{A6})$$

We set $\eta = 1$ without loss of generality. In fact, if $\eta \neq 1$, consider a rescaled variable $\tilde{\Delta}_{uv} = \Delta_{uv}/\sqrt{\eta}$. Dividing both sides of equations (A2) and (A5) by $\sqrt{\eta}$ and of equation (A6) by η yields

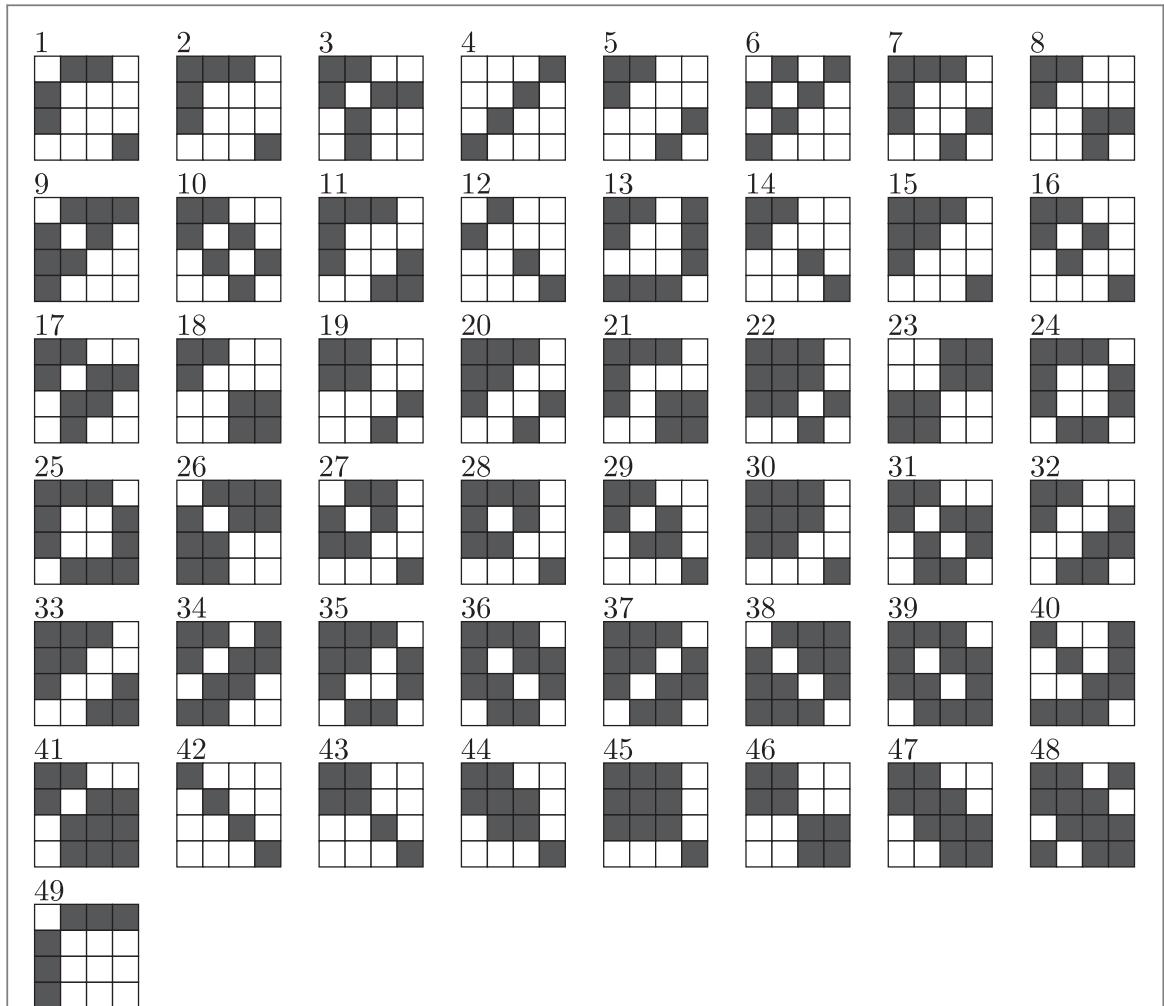


Figure B1. Schematic illustration of all network structures with four blocks that are compatible with equation (9). Patterns 5, 8, 16 and 47 are the same as those shown in figure 4.

$$\sum_{v=1}^B \tilde{\Delta}_{uv} = 0, \quad 1 \leq u \leq B, \quad (\text{A7})$$

$$Y_{uv} \tilde{\Delta}_{uv} \geq 0, \quad 1 \leq u, v \leq B, \quad (\text{A8})$$

$$\tilde{\Delta}_{uv}^2 \geq 1, \quad 1 \leq u, v \leq B, \quad (\text{A9})$$

which are equivalent to equations (A2), (A5) and (A6) with $\eta = 1$, respectively.

We seek Δ_{uv} values ($1 \leq u, v \leq B$) that simultaneously satisfy equations (A2), (A5) and (A6). To this end, we solve the following quadratic programming (QP) problem:

$$\min_{\substack{\Delta_{uv}; \\ 1 \leq u, v \leq B}} \sum_{u=1}^B \sum_{v=1}^B \Delta_{uv}^2, \quad (\text{A10})$$

subject to equations (A2), (A5) and (A6). The constraints for the QP problem are equivalent to the conditions for compatible block structure. Therefore, the block structure is compatible if and only if the QP problem is feasible. Note that we are not interested in the objective function's value, i.e., equation (A10). To solve the QP problem and hence to check whether it has a feasible solution, we use a numerical solver [83]. If we find Δ_{uv} values satisfying all the constraints by solving the QP problem, then the block structure represented by $Y = (Y_{uv})$ is compatible. Otherwise, the block structure is incompatible.

Appendix B. Network structure with four blocks

All possible network structures with four blocks that are compatible with equation (9) are shown in figure B1.

Appendix C. Estimating statistical significance of core-periphery structure

Let S be the sum of the number of core-periphery pairs detected in the 500 randomised networks. Let $\hat{q}^{(s)}$ and $\hat{n}^{(s)}$ ($1 \leq s \leq S$) be the quality and size of the s th core-periphery pair in the randomised networks, respectively. We use the Gaussian kernel density estimator [54, 55] to infer the joint probability distribution $P(\hat{q}, \hat{n})$, which gives

$$P(\hat{q}, \hat{n}) = \frac{1}{S} \sum_{s=1}^S f\left(\frac{\hat{q} - \hat{q}^{(s)}}{\sigma_{\hat{q}} h}, \frac{\hat{n} - \hat{n}^{(s)}}{\sigma_{\hat{n}} h}\right), \quad (\text{C1})$$

where $\sigma_{\hat{q}}$ and $\sigma_{\hat{n}}$ are the standard deviations of $\{\hat{q}^{(s)}\}$ and $\{\hat{n}^{(s)}\}$ ($1 \leq s \leq S$), respectively. Standard deviations $\sigma_{\hat{q}}$ and $\sigma_{\hat{n}}$ are defined as

$$\sigma_{\hat{q}} \equiv \sqrt{\frac{1}{S-1} \sum_{s=1}^S (\hat{q}^{(s)} - \langle \hat{q} \rangle)^2}, \quad (\text{C2})$$

$$\sigma_{\hat{n}} \equiv \sqrt{\frac{1}{S-1} \sum_{s=1}^S (\hat{n}^{(s)} - \langle \hat{n} \rangle)^2}, \quad (\text{C3})$$

where $\langle \hat{q} \rangle$ and $\langle \hat{n} \rangle$ are the average values of $\{\hat{q}^{(s)}\}$ and $\{\hat{n}^{(s)}\}$ ($1 \leq s \leq S$), respectively. In equation (C1), f is the probability density function of the standard bivariate normal distribution given by

$$f(y_1, y_2) \equiv \frac{1}{2\pi\sqrt{1-\gamma^2}} \exp\left(-\frac{y_1^2 - 2\gamma y_1 y_2 + y_2^2}{2(1-\gamma^2)}\right), \quad (\text{C4})$$

where γ is the Pearson correlation coefficient between $\{\hat{q}^{(s)}\}$ and $\{\hat{n}^{(s)}\}$ ($1 \leq s \leq S$), i.e.,

$$\gamma \equiv \frac{\frac{1}{S} \sum_{s=1}^S (\hat{q}^{(s)} - \langle \hat{q} \rangle)(\hat{n}^{(s)} - \langle \hat{n} \rangle)}{\sigma_{\hat{q}} \sigma_{\hat{n}}}. \quad (\text{C5})$$

In equation (C1), h is a parameter specifying the width of the Gaussian kernel density estimator. We use the Scott's rule of thumb [55], which gives $h = S^{-1/6}$.

The probability that the core-periphery pair of size n has a quality value greater than or equal to q in randomised networks is computed as

$$P(\hat{q} \geq q | n) = \frac{\int_q^\infty P(z, n) dz}{\int_{-\infty}^\infty P(z, n) dz} = \frac{\sum_{s=1}^S \int_q^\infty f\left(\frac{z - \hat{q}^{(s)}}{\sigma_{\hat{q}} h}, \frac{n - \hat{n}^{(s)}}{\sigma_{\hat{n}} h}\right) dz}{\sum_{s=1}^S \int_{-\infty}^\infty f\left(\frac{z - \hat{q}^{(s)}}{\sigma_{\hat{q}} h}, \frac{n - \hat{n}^{(s)}}{\sigma_{\hat{n}} h}\right) dz}. \quad (\text{C6})$$

Equation (C4) leads to

$$\int_{-\infty}^{y_1} f(z, y_2) dz = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y_2^2}{2}\right) \Phi\left(\frac{y_1 - \gamma y_2}{\sqrt{1-\gamma^2}}\right), \quad (\text{C7})$$

where $\Phi(y) = (2\pi)^{-1/2} \int_{-\infty}^y \exp(-z^2/2) dz$ is the cumulative distribution function of the standard normal distribution. Substitution of equation (C7) into equation (C6) yields

$$P(\hat{q} \geq q | n) = 1 - \frac{\sum_{s=1}^S \exp\left(-\frac{(n - \hat{n}^{(s)})^2}{2\sigma_{\hat{n}}^2 h^2}\right) \Phi\left(\frac{\sigma_{\hat{n}}(q - \hat{q}^{(s)}) - \gamma \sigma_{\hat{q}}(n - \hat{n}^{(s)})}{\sigma_{\hat{n}} \sigma_{\hat{q}} h \sqrt{1-\gamma^2}}\right)}{\sum_{s=1}^S \exp\left(-\frac{(n - \hat{n}^{(s)})^2}{2\sigma_{\hat{n}}^2 h^2}\right)}. \quad (\text{C8})$$

Appendix D. Results of other algorithms

Figures D1–D6 plot d_i^{peri} against degree d_i when the core-periphery structure is determined by the BE, MINRES, SBM, Xiang, Divisive and KM–ER algorithms. Figure D7 shows the cartographic representation of the 12 empirical networks when the core-periphery structure is determined by the Divisive algorithm.

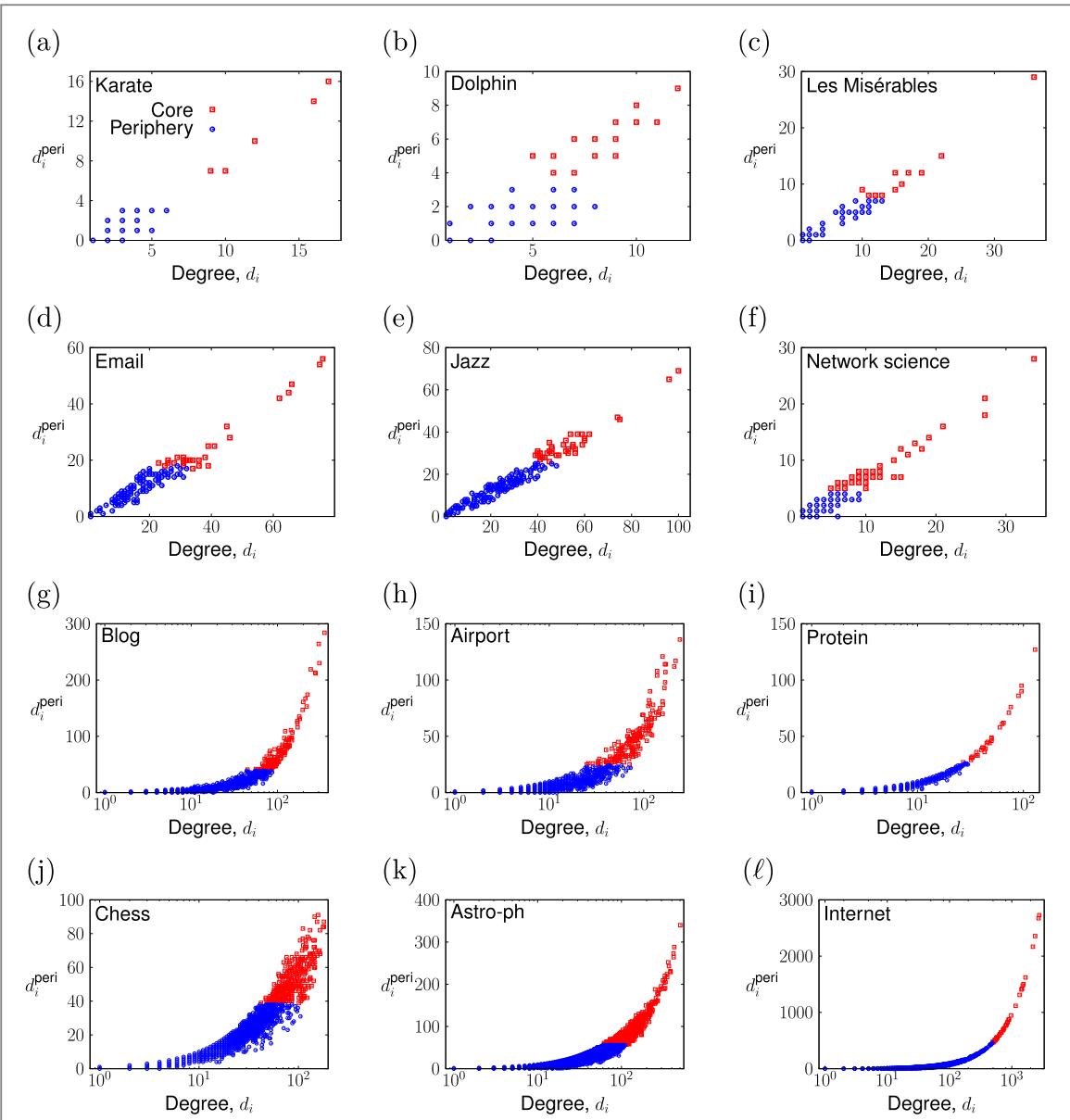


Figure D1. Relationships between d_i and d_i^{peri} for the empirical networks when the core-periphery structure is determined by the BE algorithm. The squares and circles indicate core nodes and peripheral nodes, respectively.

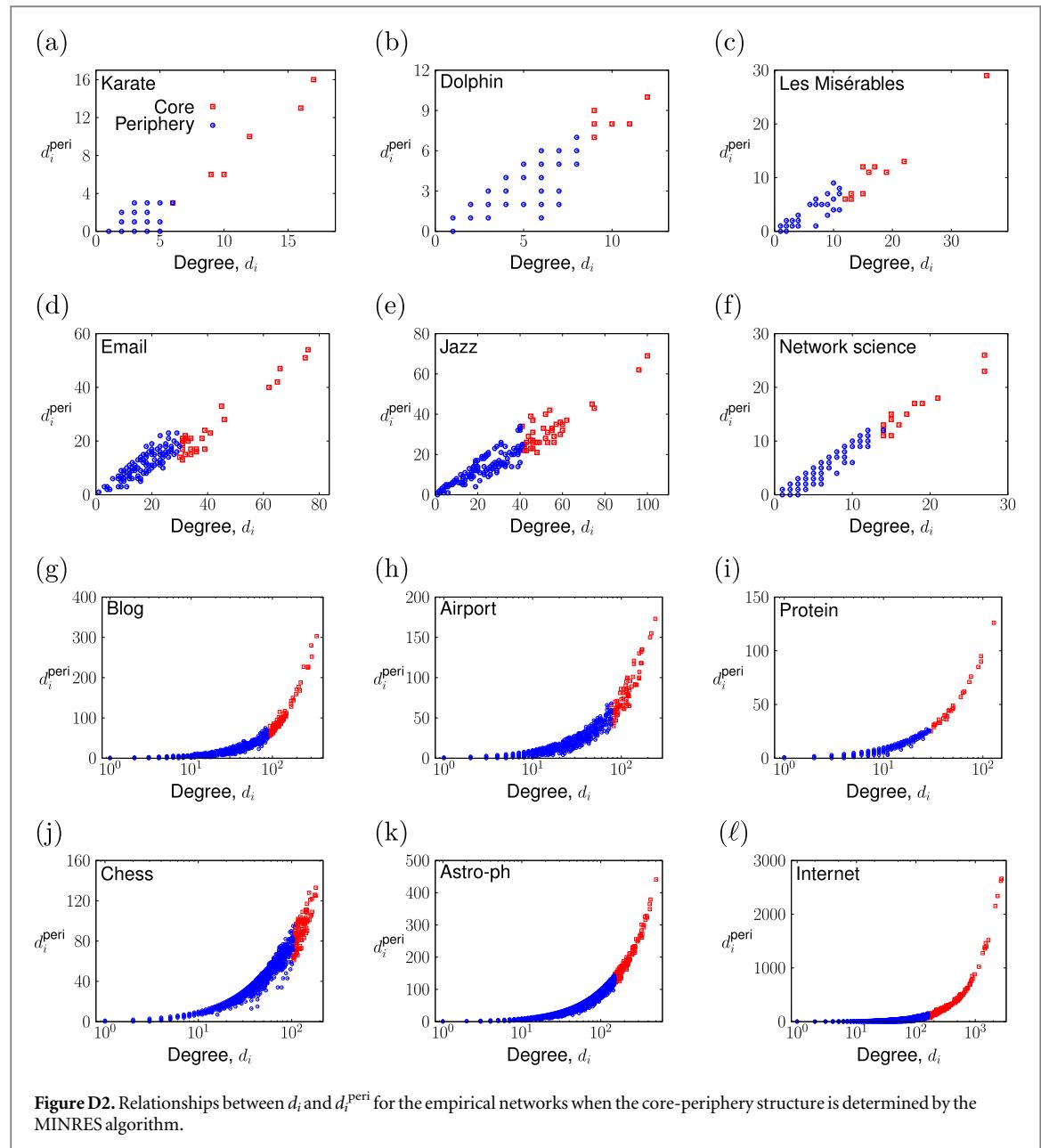


Figure D2. Relationships between d_i and d_i^{peri} for the empirical networks when the core-periphery structure is determined by the MINRES algorithm.

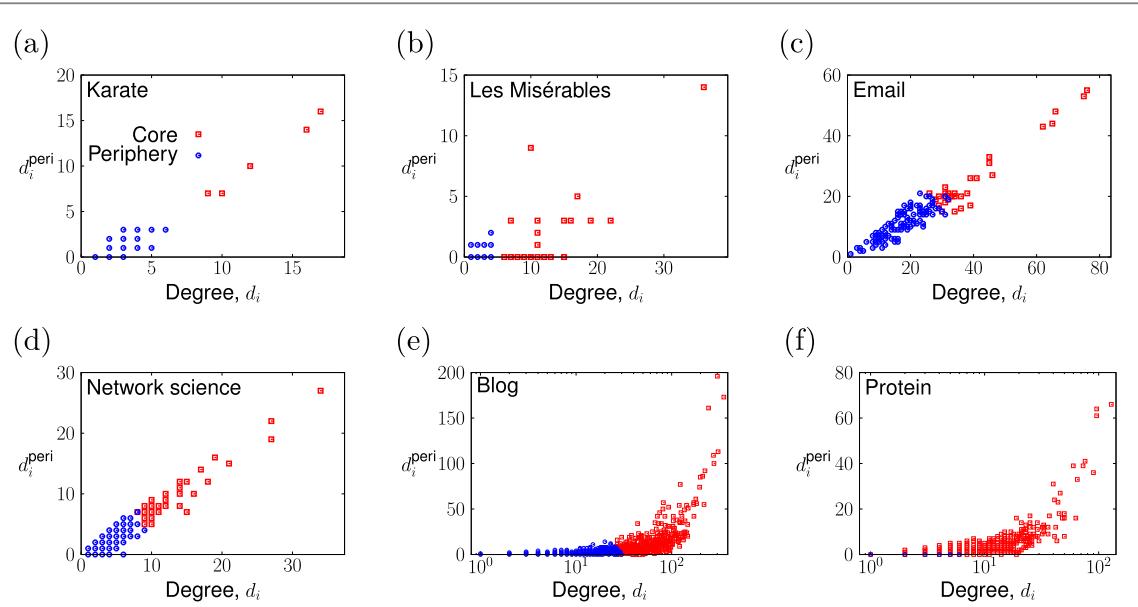


Figure D3. Relationships between d_i and d_i^{peri} for the empirical networks when the core-periphery structure is determined by the SBM algorithm. We do not show the results for the other empirical networks, for which the SBM algorithm does not find significant core-periphery pairs.

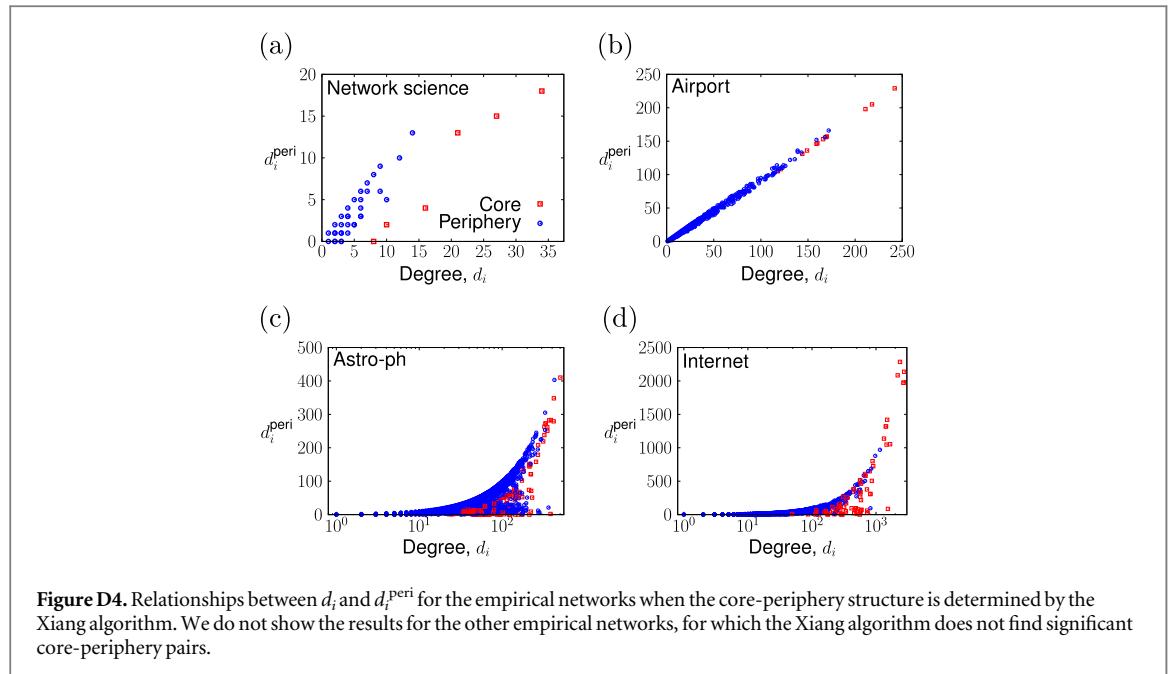
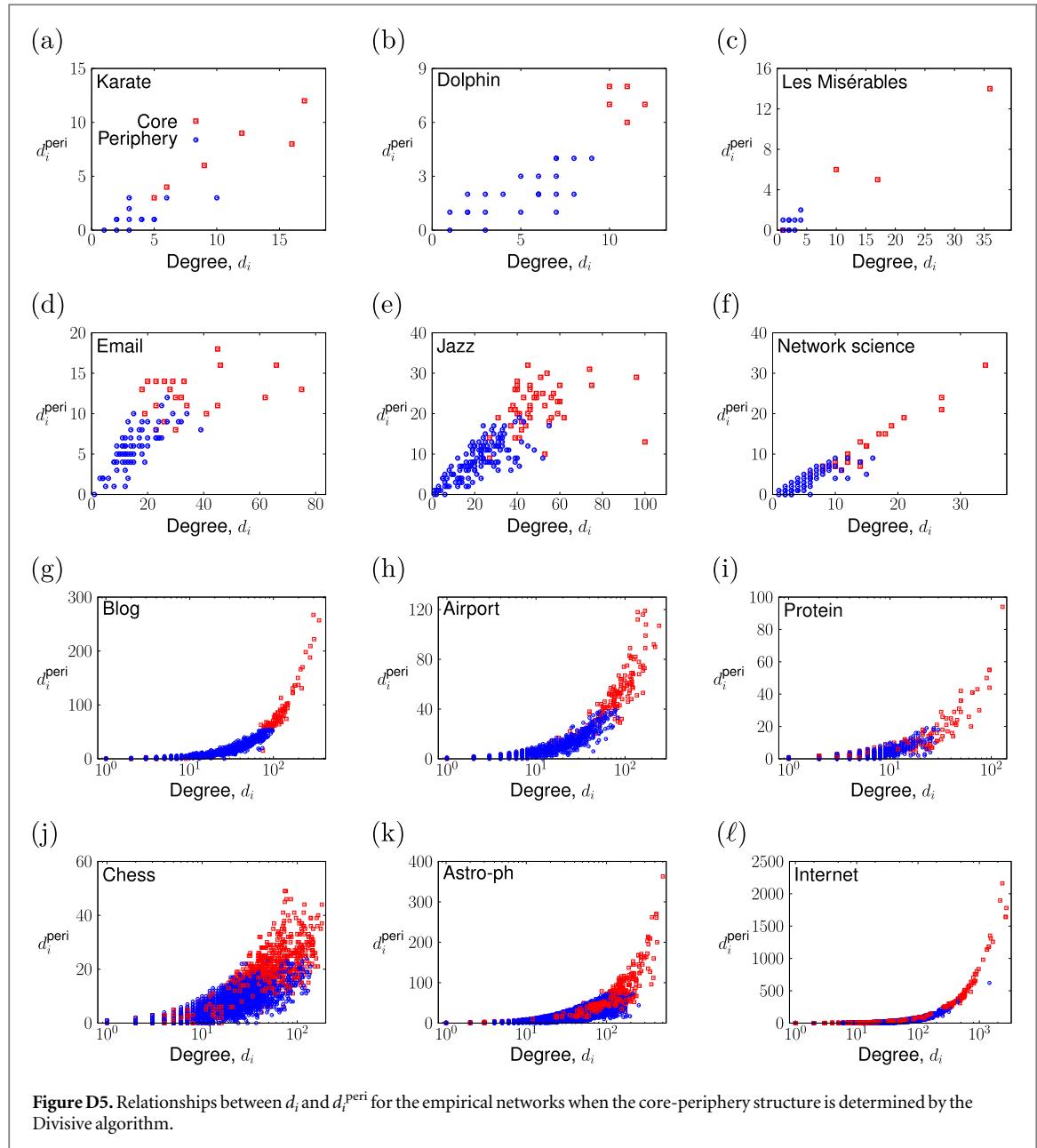


Figure D4. Relationships between d_i and d_i^{peri} for the empirical networks when the core-periphery structure is determined by the Xiang algorithm. We do not show the results for the other empirical networks, for which the Xiang algorithm does not find significant core-periphery pairs.



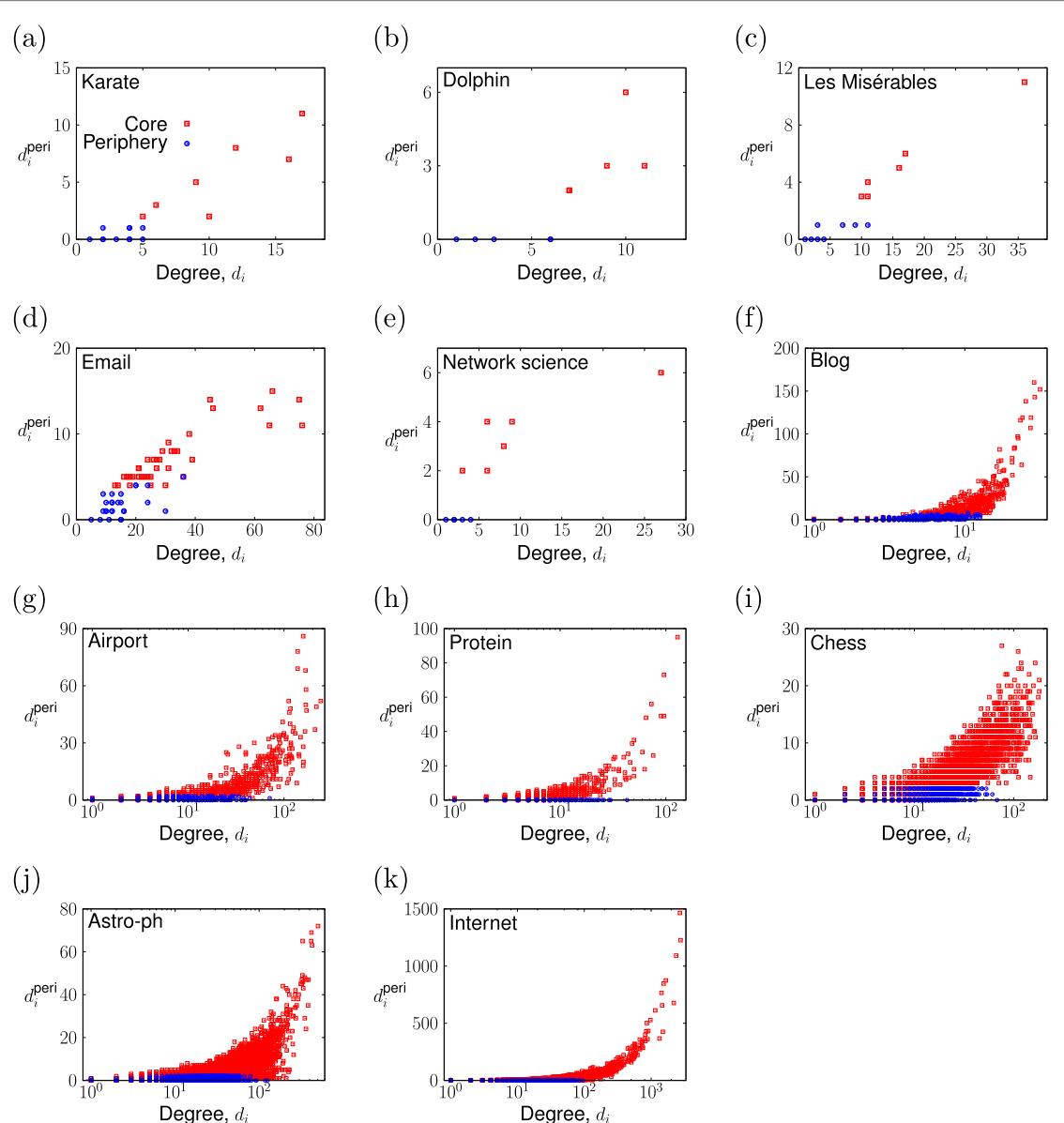


Figure D6. Relationships between d_i and d_i^{peri} for the empirical networks when the core-periphery structure is determined by KM–ER. We do not show the result for the jazz network, for which KM–ER does not find significant core-periphery pairs.

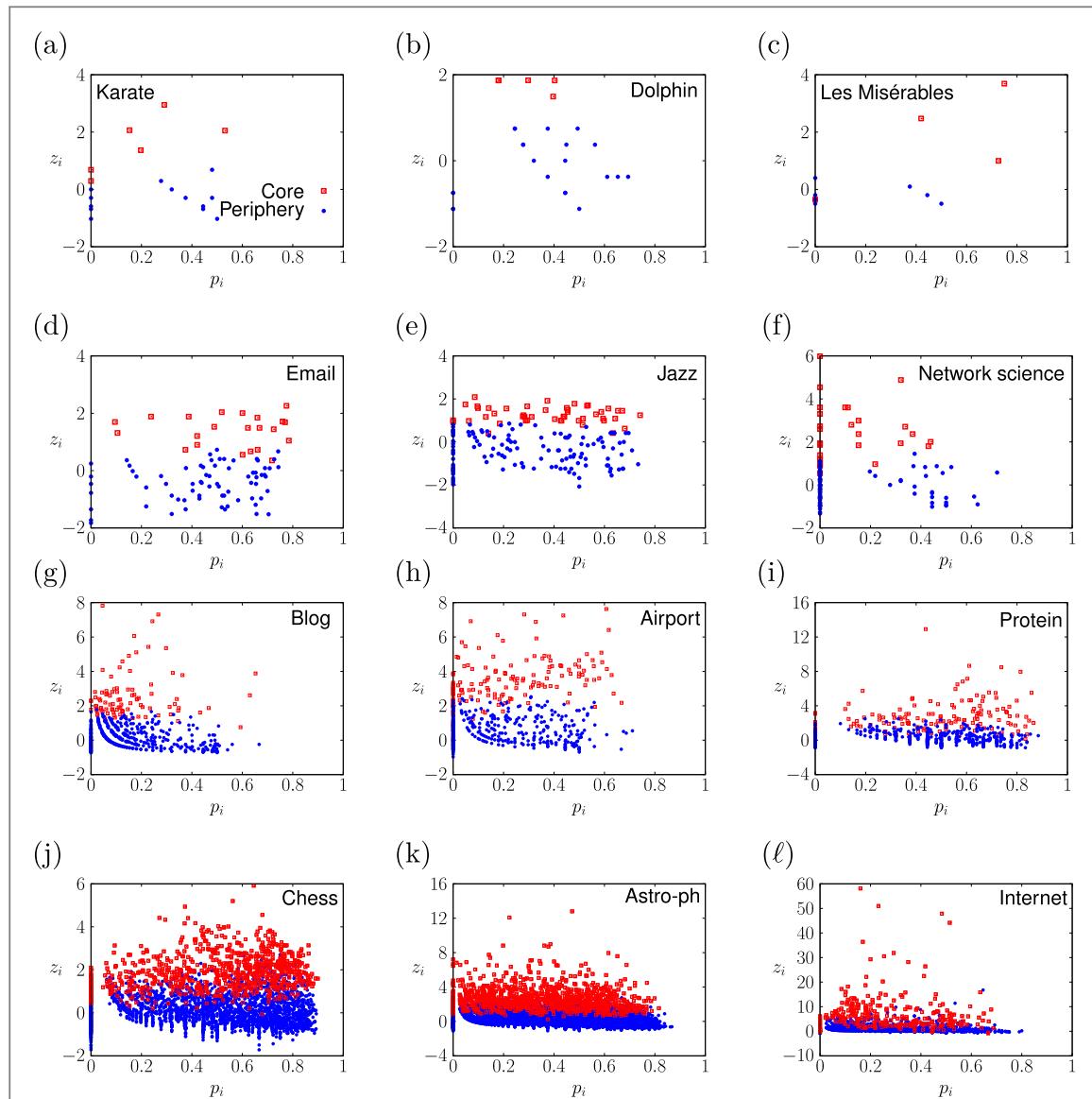


Figure D7. Cartographic analysis of the empirical networks. The core and peripheral nodes are detected by the Divisive algorithm.

ORCID iDs

Sadamori Kojaku <https://orcid.org/0000-0002-9414-6814>

Naoki Masuda <https://orcid.org/0000-0003-1567-801X>

References

- [1] Newman M E J 2010 *Networks: An Introduction* (Oxford: Oxford University Press)
- [2] Barabási A L 2016 *Network Science* (Cambridge: Cambridge University Press)
- [3] Fortunato S 2010 *Phys. Rep.* **486** 75
- [4] Newman M E J and Girvan M 2004 *Phys. Rev. E* **69** 026113
- [5] Adamic L A and Glance N 2005 *Proc. 3rd Int. Workshop on Link Discovery, LinkKDD '05* (New York: ACM) pp 36–43
- [6] Karrer B and Newman M E J 2011 *Phys. Rev. E* **83** 016107
- [7] Jonsson P F, Cavanna T, Zicha D and Bates P A 2006 *BMC Bioinform.* **7** 2
- [8] Borgatti S P and Everett M G 2000 *Soc. Netw.* **21** 375
- [9] Csermely P, London A, Wu L-Y and Uzzi B 2013 *J. Comput. Netw.* **1** 93
- [10] Rombach M P, Porter M A, Fowler J H and Mucha P J 2017 *SIAM Rev.* **59** 619
- [11] Holme P 2005 *Phys. Rev. E* **72** 046111
- [12] Boyd J P, Fitzgerald W J and Beck R J 2006 *Soc. Netw.* **28** 165
- [13] Rossa F D, Dercole F and Piccardi C 2013 *Sci. Rep.* **3** 1467
- [14] Lee S H, Cucuringu M and Porter M A 2014 *Phys. Rev. E* **89** 032810
- [15] Yang J and Leskovec J 2014 *Proc. IEEE* **102** 1892
- [16] Zhang X, Martin T and Newman M E J 2015 *Phys. Rev. E* **91** 032803

- [17] Cucuringu M, Rombach P, Lee S H and Porter M A 2016 *Eur. J. Appl. Math.* **27** 846
- [18] Gamble J, Chintakunta H, Wilkerson A and Krim H 2016 *IEEE Trans. Signal Inf. Process. Netw.* **2** 186
- [19] Kojaku S, Masuda N and Sadamori Kojaku N M 2017 *Phys. Rev. E* **96** 052313
- [20] Ma C, Xiang B, Zhang H, Chen H and Small M 2017 arXiv: 1705.04062
- [21] da Silva M R, Ma H and Zeng A-P 2008 *Proc. IEEE* **96** 1411
- [22] Bruckner S, Hüffner F and Komusiewicz C 2015 *Algorithms Mol. Biol.* **10** 16
- [23] Bassett D S, Wymbs N F, Rombach M P, Porter M A, Mucha P J and Grafton S T 2013 *PLoS Comput. Biol.* **9** e1003171
- [24] Tunç B and Verma R 2015 *PLoS One* **10** e0143133
- [25] Boyd J P, Fitzgerald W J, Mahutga M C and Smith D A 2010 *Soc. Netw.* **32** 125
- [26] Ma A and Mondragón R J 2015 *PLoS One* **10** e0119678
- [27] Craig B and von Peter G 2014 *J. Financ. Intermed.* **23** 322
- [28] in 't Veld D and van Lelyveld I 2014 *J. Banking Financ.* **49** 27
- [29] Fricke D and Lux T 2014 *Comput. Econ.* **45** 359
- [30] Barucca P and Lillo F 2016 *Chaos Solitons Fractals* **88** 244
- [31] Sardana D and Bhatnagar R 2016 *Proc. IEEE/WIC/ACM Int. Conf. on Web Intelligence, WI'16* (New York: ACM) pp 1–8
- [32] Xiang B-B, Bao Z-K, Ma C, Zhang X-Y, Chen H-S and Zhang H-F 2018 *J. Nonlinear Sci.* **28** 13122
- [33] Lip S Z W 2011 arXiv: 1102.5511
- [34] Yan B and Luo J 2018 *Network Sci.* **1**–28
- [35] Fu J, Li J, Niu Y, Wang G and Wu J 2018 *Future Gener. Comput. Syst.* **83** 454–60
- [36] Marc T and Šubelj L 2016 arXiv: 1608.03402
- [37] Alvarez-Hamelin J I, Dall'Asta L, Barrat A and Vespignani A 2006 *Advances in Neural Information Processing Systems 18* ed Y Weiss et al (Cambridge, MA: MIT Press) pp 41–50
- [38] Zhou S and Mondragón R J 2004 *IEEE Commun. Lett.* **8** 180
- [39] Colizza V, Flammini A, Serrano M A and Vespignani A 2006 *Nat. Phys.* **2** 110
- [40] von Luxburg U 2007 *Stat. Comput.* **17** 395
- [41] Peixoto T P 2017 *Phys. Rev. E* **95** 012317
- [42] Fosdick B K, Larremore D B, Nishimura J and Ugander J 2017 *SIAM Rev.* accepted
- [43] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D and Alon U 2002 *Science* **298** 824
- [44] Zachary W W 1977 *J. Anthropol. Res.* **33** 452
- [45] Newman M E J 2003 *SIAM Rev.* **45** 167
- [46] Newman M E J 2006 *Phys. Rev. E* **74** 036104
- [47] Delvenne J C, Yaliraki S N and Barahona M 2010 *Proc. Natl Acad. Sci. USA* **107** 12755
- [48] Mucha P J, Richardson T, Macon K, Porter M A and Onnela J-P 2010 *Science* **328** 876
- [49] Lambiotte R, Delvenne J C and Barahona M 2014 *IEEE Trans. Netw. Sci. Eng.* **1** 76
- [50] Masuda N, Porter M A and Lambiotte R 2017 *Phys. Rep.* **716–717** 1–58
- [51] Raghavan U N, Albert R and Kumara S 2007 *Phys. Rev. E* **76** 036106
- [52] Blondel V D, Guillaume J-L, Lambiotte R and Lefebvre E 2008 *J. Stat. Mech.* P10008
- [53] Leskovec J, Lang K J and Mahoney M W 2010 *Proc. 19th Int. Conf. World Wide Web, WWW'10* (New York: ACM) pp 631–40
- [54] Wand M P and Jones M C 1993 *J. Am. Stat. Assoc.* **88** 520
- [55] Scott D W 2012 *Handbook of Computational Statistics* (Berlin: Springer) pp 549–69
- [56] Šidák Z 1967 *J. Am. Stat. Assoc.* **62** 626
- [57] Lusseau D, Schneider K, Boisseau O J, Haase P, Slooten E and Dawson S M 2003 *Behav. Ecol. Sociobiol.* **54** 396
- [58] Knuth D E 1993 *The Stanford GraphBase: A Platform for Combinatorial Computing* (New York: ACM)
- [59] Klimt B and Yang Y 2004 *Machine Learning: ECML 2004*. ECML 2004 (*Lecture Notes in Computer Science* vol 3201) ed J F Boulicaut et al (Berlin: Springer) pp 217–26
- [60] Gleiser P M and Danon L 2003 *Adv. Comput. Syst.* **6** 565
- [61] Patokallio J OpenFlights Data
- [62] Opsahl T 2011 Why Anchorage is not (that) important: Binary ties and sample selection
- [63] Rual J, Venkatesan K, Hao T, Hirozane-Kishikawa T, Dricot A, Li N, Berriz G F, Gibbons F D, Dreze M and Ayivi-Guedehoussou N 2005 *Nature* **437** 1173
- [64] Kunegis J 2013 KONECT: The Koblenz Network Collection *Proc. of the 22nd Int. Conf. on World Wide Web* (New York: ACM) 1343–50
- [65] Leskovec J, Kleinberg J and Faloutsos C 2007 *ACM Trans. Knowl. Discovery Data* **1** 1
- [66] Guimerà R and Amaral L A N 2005 *Nature* **433** 895
- [67] Guimerà R, Mossa S, Turtschi A and Amaral L A N 2005 *Proc. Natl Acad. Sci. USA* **102** 7794
- [68] Sales-Pardo M, Guimerà R, Moreira A A and Amaral L A N 2007 *Proc. Natl Acad. Sci. USA* **104** 15224
- [69] Meilă M 2007 *J. Multivariate Anal.* **98** 873
- [70] Mastrandrea R, Squartini T, Fagiolo G and Garlaschelli D 2014 *New J. Phys.* **16** 043022
- [71] Traag V A and Bruggeman J 2009 *Phys. Rev. E* **80** 036115
- [72] MacMahon M and Garlaschelli D 2015 *Phys. Rev. X* **5** 021006
- [73] Barber M J 2007 *Phys. Rev. E* **76** 066102
- [74] Expert P, Evans T S, Blondel V D and Lambiotte R 2011 *Proc. Natl Acad. Sci. USA* **108** 7663
- [75] Fortunato S and Barthélémy M 2006 *Proc. Natl Acad. Sci. USA* **104** 36
- [76] Guimerà R, Sales-Pardo M and Amaral L A N 2004 *Phys. Rev. E* **70** 025101
- [77] Arenas A, Fernández A and Gómez S 2008 *New J. Phys.* **10** 053039
- [78] Lancichinetti A, Radicchi F and Ramasco JJ 2010 *Phys. Rev. E* **81** 046110
- [79] Zhang P and Moore C 2014 *Proc. Natl Acad. Sci. USA* **111** 18144
- [80] Arenas A, Duch J, Fernández A and Gómez S 2007 *New J. Phys.* **9** 176
- [81] Leicht E A and Newman M E J 2008 *Phys. Rev. Lett.* **100** 118703
- [82] Kim Y, Son S-W and Jeong H 2010 *Phys. Rev. E* **81** 016103
- [83] Gurobi Optimization, Inc. 2016 Gurobi optimizer