

## Representative community divisions of networks

Alec Kirkley<sup>1,2</sup>✉ & M. E. J. Newman<sup>1,3</sup>

Methods for detecting community structure in networks typically aim to identify a single best partition of network nodes into communities, often by optimizing some objective function, but in real-world applications there may be many competitive partitions with objective scores close to the global optimum and one can obtain a more informative picture of the community structure by examining a representative set of such high-scoring partitions than by looking at just the single optimum. However, such a set can be difficult to interpret since its size can easily run to hundreds or thousands of partitions. In this paper we present a method for analyzing large partition sets by dividing them into groups of similar partitions and then identifying an archetypal partition as a representative of each group. The resulting set of archetypal partitions provides a succinct, interpretable summary of the form and variety of community structure in any network. We demonstrate the method on a range of example networks.

<sup>1</sup>Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA. <sup>2</sup>School of Data Science, City University of Hong Kong, Kowloon, Hong Kong.  
<sup>3</sup>Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109, USA. ✉email: [alec.w.kirkley@gmail.com](mailto:alec.w.kirkley@gmail.com)

Networks are widely used as a compact quantitative representation of a range of complex systems, particularly in the biological and social sciences, engineering, computer science, and physics. Many networks naturally divide into communities, densely connected groups of nodes with sparser between-group connections<sup>1</sup>. Identifying these groups, in the process known as community detection, can help us in understanding network phenomena such as the evolution of social relationships<sup>2</sup>, epidemic spreading<sup>3</sup>, and others.

There are numerous existing methods for community detection, including ones based on centrality measures<sup>4</sup>, modularity<sup>5</sup>, information theory<sup>6</sup>, and Bayesian generative models<sup>7</sup>—see Fortunato<sup>8</sup> for a review. Most methods represent the community structure in a network as a single network partition or division (an assignment of each node to a specific community), which is typically the one that attains the highest score according to some objective function. As pointed out by many previous authors, however, there may be multiple partitions of a network that achieve high scores, any of which could be a good candidate for division of the network<sup>9–14</sup>. With this in mind, some community detection methods return multiple plausible partitions rather than just one. Examples include methods based on modularity<sup>8,12,15</sup>, generative models<sup>7</sup>, and other objective criteria<sup>16,17</sup>. But while these algorithms give a more complete picture of community structure, they have their own problems. In particular, the number of partitions returned is often very large. Even for relatively small networks, the partitions may number in the hundreds or thousands, making it hard to interpret the results. How then are we supposed to make sense of the output of these calculations?

In some cases, it may happen that all of the plausible divisions of a network are quite similar to each other, in which case we can create a *consensus clustering*<sup>18</sup>, a single partition that is representative of the entire set in the same way that the mean of a set of numbers can be a useful representation of the whole. However, if the partitions vary substantially, then the consensus can fail to capture the full range of behaviors in the same way that the mean can be a poor summary statistic for broad or multimodal distributions of numbers. In cases like these, summarizing the community structure may require not just one but several representative partitions, each of which is the consensus partition for a cluster of similar network divisions<sup>14</sup>.

Finding such representative partitions thus involves clustering the full set of partitions into groups of similar ones. A few previous studies have investigated the clustering of partitions. Calatayud et al.<sup>19</sup> proposed an algorithm that starts with the single highest scoring partition (under whatever objective function is in use), then iterates through other divisions in order of decreasing score and assigns each to the closest cluster if the distance to that cluster is less than a certain threshold, or starts a new cluster otherwise. This approach is primarily applicable in situations where there is a clear definition of distance between partitions (there are many possible choices<sup>20</sup>), as the results turn out to be sensitive to this definition and to the corresponding distance threshold. Peixoto<sup>14</sup> has proposed a principled statistical method for clustering partitions using methods of Bayesian inference, which works well but differs from ours in that rather than returning a single partition as a representative of each cluster it returns a distribution over partitions. It also does not explicitly address issues of the dependence of the number of clusters on the number of input partitions.

The *minimum description length* principle posits that when selecting between possible models for a data set, the best model is the one that permits the most succinct representation of the data<sup>21</sup>. The minimum description length principle has previously

been applied to clustering of real-valued (non-network) data, including methods based on Gaussian mixture models<sup>22</sup>, hierarchical clustering<sup>23</sup>, Bernoulli mixture models for categorical data<sup>24</sup>, and probabilistic generative models<sup>25</sup>. Georgieva et al.<sup>26</sup>, for instance, have proposed a clustering framework that is similar in some respects to ours but for real-valued vector data, with the data being thought of as a message to be transmitted in multiple parts, including the cluster centers and the data within each cluster. Georgieva et al., however, only use their measure as a quality function to assess the outputs of other clustering algorithms and not as an objective to be optimized to obtain the clusters themselves. The minimum description length approach has also been applied to the task of community detection itself by Rosvall and Bergstrom<sup>27</sup>, who used it to formulate an objective function for community detection that considers the encoding of a network in terms of a partition and the node and edge counts within and between the communities in the partition.

In this paper, we use the minimum description length principle to motivate a simple and efficient method for finding representative community divisions of networks that has a number of practical advantages. In particular, it does not require the explicit choice of a partition distance function, does not depend on the number of input partitions provided the partition space is well sampled, and is adaptable to any community detection algorithm that returns multiple sample partitions. We present an efficient Monte Carlo scheme implementing our approach and test it on a range of real and synthetic networks, demonstrating that it returns substantially distinct community divisions that are a good guide to the structures present in the original sample.

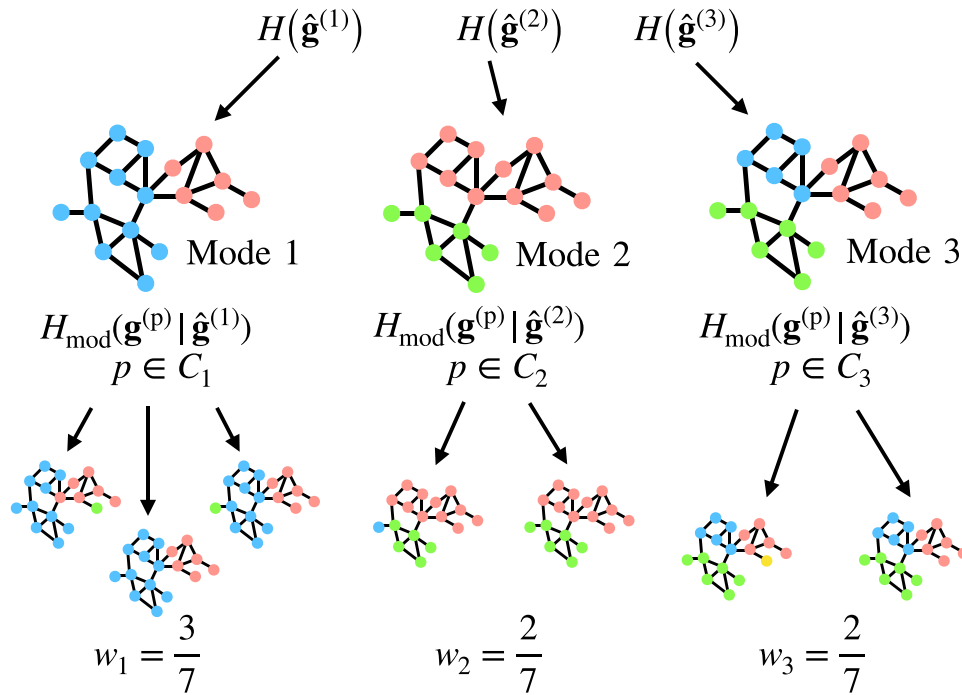
## Results and discussion

The primary goal of our proposed technique is to find representative partitions that summarize the community structure in a network. We call these representative partitions *modes*. Suppose we have an observed network consisting of  $N$  nodes and we have some method for finding community divisions of these nodes, also called partitions. We can represent a partition with a length- $N$  vector  $\mathbf{g}$  that assigns to each node  $i = 1 \dots N$  a label  $g_i$  indicating which community it belongs to.

We assume that there are a large number of plausible partitions and that our community detection method returns a subset of them. Normally we expect that many of the partitions would be similar to one another, differing only by a few nodes here or there. The goal of this paper is to develop a procedure for gathering such similar partitions into clusters and generating a mode, which is itself a partition, as an archetypal representative of each cluster. For the sake of clarity, we will in this paper use the words “partition” or “division” to describe the assignment of network nodes to communities, and the word “cluster” to describe the assignment of entire partitions to groups according to the method that we describe.

In order both to divide the partitions into clusters and to find a representative mode for each cluster, we first develop a clustering objective function based on information-theoretic arguments. The main concept behind our approach is a thought experiment in which we imagine transmitting our set of partitions to a receiver using a multi-step encoding chosen so as to minimize the amount of information required for the complete transmission.

**Partition clustering as an encoding problem.** Let us denote our set of partitions by  $D$  and suppose there are  $S$  partitions in the set, labeled  $p = 1 \dots S$ . Now imagine we wish to transmit a complete description of all elements of the set to a receiver. How should we go about this? The most obvious way is to send each of the



**Fig. 1 Transmission of a set of partitions for a network.** We first transmit a small set of “modes”  $\hat{\mathbf{g}}^{(k)}$ , archetypal partitions drawn from the larger set, which takes an amount of information equal to the sum of the entropies  $H$  of these partitions (Eq. (2)). Then each partition  $\mathbf{g}^{(p)}$  from the complete set is transmitted by describing how it differs from the most similar of the modes, which requires an amount of information equal to the modified conditional entropy  $H_{\text{mod}}$  of Eq. (4). The weight  $w_k$  is the fraction of all partitions that are part of cluster  $C_k$ , the set of partitions assigned to the representative mode  $\hat{\mathbf{g}}^{(k)}$ . The color of each node indicates its community membership within a partition.

partitions separately to the receiver using some simple encoding that uses, say, numbers or symbols to represent community labels. We could do somewhat better by using an optimal prefix code such as a Huffman code<sup>28</sup> that economizes by representing frequently used labels with shorter code words. Even this, however, would be quite inefficient in terms of information. We can do better by making use of the fact that, as we have said, we expect many of our partitions to be similar to one another. This allows us to save the information by dividing the partitions into clusters of similar ones and transmitting only a few partitions in full—one representative partition or mode for each cluster—then describing the remaining partitions by how they differ from these modes. The method is illustrated in Fig. 1.

Initially, let us assume that we want to divide the set  $D$  of partitions into  $K$  clusters, denoted  $C_k$  with  $k = 1 \dots K$ . (We will discuss how to choose  $K$  separately in a moment.) To efficiently transmit  $D$ , we first transmit  $K$  representative modes, which themselves are members of  $D$ , with group labels  $\hat{\mathbf{g}}^{(k)}$ . Then for each individual partition in  $D$  we transmit which cluster, or equivalently which mode, it belongs to and then the partition itself by describing how it differs from that mode. Since the latter information will be smaller if a partition is more similar to its assigned mode, choosing a set of modes that are accurately representative of all partitions will naturally minimize the total information, and we use this criterion to derive the best set of modes. This is the minimum description length principle, as applied to finding the optimal clusters and modes.

Following this plan, the total description length per sampled partition can be written (see Supplementary Note 1) in the form

$$\mathcal{L}_{\text{total}} = \frac{N}{S} \sum_{k=1}^K H(\hat{\mathbf{g}}^{(k)}) + H(\mathbf{c}) + \frac{N}{S} \sum_{k=1}^K \sum_{p \in C_k} H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}). \quad (1)$$

The first term represents the amount of information required to transmit the modes and is simply equal to the sum of their entropies:

$$H(\hat{\mathbf{g}}^{(k)}) = - \sum_{r=1}^{n_{m_k}} \frac{a_r^{(m_k)}}{N} \log \frac{a_r^{(m_k)}}{N}. \quad (2)$$

Here  $m_k$  is the partition label  $p$  of the  $k$ th mode,  $n_p$  is the number of communities in partition  $p$ , and  $a_r^{(p)}$  is the number of nodes in partition  $p$  that have community label  $r$ .

The second term in Eq. (1) represents the amount of information needed to specify which cluster, or alternatively which mode, each partition in  $D$  belongs to:

$$H(\mathbf{c}) = - \sum_{k=1}^K \frac{c_k}{S} \log \frac{c_k}{S}, \quad (3)$$

where  $c_k = |C_k|$  is the number of partitions (out of  $S$  total) that belong to mode  $k$ .

The third term in Eq. (1) represents the amount of information needed to specify each of the individual partitions  $\mathbf{g}^{(p)}$  in terms of their modes  $\hat{\mathbf{g}}^{(k)}$ :

$$H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) = H(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) + \frac{1}{N} \log \Omega(p, m_k). \quad (4)$$

$H_{\text{mod}}$  is the *modified conditional entropy* of the group labels of  $\mathbf{g}^{(p)}$  given the group labels of  $\hat{\mathbf{g}}^{(k)}$ <sup>29</sup>. The normal (non-modified) conditional entropy is

$$H(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) = - \sum_{r=1}^{n_{m_k}} \sum_{s=1}^{n_p} \frac{t_{rs}^{m_k p}}{N} \log \frac{t_{rs}^{m_k p}}{a_r^{(m_k)}}, \quad (5)$$

where  $t_{rs}^{mp}$  is the number of nodes simultaneously classified into community  $r$  in partition  $\mathbf{g}^{(m)}$  and community  $s$  in partition  $\mathbf{g}^{(p)}$ . The matrix of elements  $t^{mp}$  for any pair of partitions  $m, p$  is known as a *contingency table*, and Eq. (5) measures the amount of

information needed to transmit  $\mathbf{g}^{(p)}$  given that we already know both  $\hat{\mathbf{g}}^{(k)}$  and the contingency table. To actually transmit the partitions in practice, we would also need to transmit the contingency table, and the second term in Eq. (4) represents the information needed to do this. The quantity  $\Omega(p, m)$  is equal to the number of possible contingency tables  $\mathbf{t}^{mp}$  with row and column sums  $a_r^{(m)}$  and  $a_s^{(p)}$ , respectively. This quantity can be computed exactly for smaller contingency tables and there exist good approximations to its value for larger tables<sup>29</sup>. The  $\log \Omega$  term is often omitted from calculations of conditional entropy, but it turns out to be crucial in the current application. Without it, one can minimize the conditional entropy simply by making the number of groups in the modal partition very large, with the result that the minimum description length solution is biased toward modes with many groups. The additional term avoids this bias.

In principle, before we send any of this information, we also need to transmit to the receiver information about the size of each partition and the number of modes  $K$ , which would contribute some additional terms to the description length in Eq. (1). These terms, however, are small, and moreover, they are independent of how we configure our clusters and modes, so we can safely neglect them.

A detailed derivation of Eq. (1) is given in Supplementary Note 1. By minimizing this quantity, we can now find the best set of modes to describe a given set of partitions.

**Choosing the number of clusters.** So far we have assumed that we know the number  $K$  of clusters of partitions, or equivalently the number of modes. In practice, we do not usually know  $K$  and normally there is not even one “correct” value for a given network. Different values of  $K$  can give useful answers for the same network, depending on how much granularity we wish to see in the community structure. In general, a small number of clusters—no more than a dozen or so—is most informative to human eyes, but fewer clusters also mean that each cluster will contain a wider range of structures within it. How then do we choose the value of  $K$ ? One might hope for a parameter-free method of choosing the value based, for instance, on statistical model selection techniques, in which we allow the data to dictate the natural number of clusters that should be used to describe it. For example, if the set  $D$  of partitions is drawn based on some sort of quality function—for example modularity or the posterior distribution of a generative model—then clusters of partitions will correspond to peaks in that function and one could use the number of peaks to define the number of clusters.

In practice, however, such an approach, if it existed, would not, in general, give us what we are looking for because the number of peaks in the quality function is not equivalent to the number of groups of similar-looking partitions. Peaks could be very broad, combining radically different partitions into a single cluster when they should be separated. Or they could be very narrow, producing an impractically large number of clusters whose modes differ in only the smallest of details. Or peaks could be very shallow, making them not significant at all. To obtain useful results, therefore, we prefer to allow the user to vary the number of clusters  $K$  through a tunable parameter, so as to make the members of the individual clusters as similar or diverse as desired.

A natural way to control the number of clusters is to impose a penalty on the description length objective function using a multiplier or “chemical potential” that couples linearly to the

value of  $K$  thus:

$$\mathcal{L}_{\text{total}} = \frac{N}{S} \sum_{k=1}^K H(\hat{\mathbf{g}}^{(k)}) + H(c) + \frac{N}{S} \sum_{k=1}^K \sum_{p \in C_k} H_{\text{mod}}(\mathbf{g}^{(p)} | \hat{\mathbf{g}}^{(k)}) + \lambda K. \quad (6)$$

This imposes a penalty equal to  $\lambda$  for each extra cluster added and hence larger values of  $\lambda$  will produce larger penalties. It is straightforward to show that this form makes the optimal number of clusters  $K$  independent of  $S$ —see Supplementary Note 2 for proof, and Supplementary Table 1 for a demonstration on example networks used in the paper. It is not the only choice of penalty function that achieves this goal—the central inequality in our proof is satisfied for a number of forms too—but it is perhaps the simplest and it is the one we use in this paper.

As we have said, we normally want the number of modes to be small, which means that we expect  $\lambda$  to be of order unity. In practice, we find that the choice  $\lambda = 1$  works well in many cases and this is the value we use for all the example applications presented here, although it is possible that other values might be useful in certain circumstances.

One can also set the value of  $\lambda$  to zero, which is equivalent to removing the penalty term altogether. In this case, there is still an optimal choice of  $K$  implied by the description length alone. Low values of  $K$ , corresponding to only a small number of modes, will give inefficient descriptions of the data because many partitions will not be similar to any of the modes, while high values of  $K$  will give inefficient descriptions because we will waste a lot of information describing all the modes. In between, at some moderate value of  $K$ , there is an optimal choice that determines the best number of clusters. An analogous method is used, for example, for choosing the optimal number of bins for histograms and often works well in that context<sup>30,31</sup>. This might appear at first sight to give a parameter-free approach for choosing the number of modes, but in fact, this is not the case because the number of modes the method returns now depends on the number of sampled partitions  $S$ , increasing as the value of  $S$  increases and diverging as  $S$  becomes arbitrarily large. When creating a histogram from a fixed set of samples this behavior is desirable—we want to use more bins when we have more data—but when clustering partitions it can result in an unwieldy number of representative modes. The linear penalty in Eq. (6) allows the user to decouple  $K$  from  $S$  and prevent the number of modes from becoming too large.

It is worth noting that one can envisage other encodings of a set of community structures that would give slightly different values for the description length. For example, when transmitting information about which cluster each sampled structure belongs to one could choose to use a single fixed-length code for the cluster labels, which would require  $\log K$  bits per sample. This would simply replace the term  $H(c)$  in Eq. (1) with  $\log K$ . One could analogously replace the terms  $H(\hat{\mathbf{g}}^{(k)})$  with their corresponding fixed-length average code sizes (per node), with values  $\log n_{m_k}$ . In general, both of these changes would result in a less efficient encoding that tends to favor a smaller number of modes. However, neither of them would affect the asymptotic scaling of the description length and the term in  $\lambda K$  would still be needed to achieve a number of modes that is independent of  $S$ . It is also possible to extend the description length formulation to a hierarchical model in which we allow the possibility of more than one “level” of modes being transmitted. However, this scheme results in a more complex output that lacks the simple

interpretation of the two-level scheme, and so we do not explore this option here.

**Minimizing the objective function.** Our goal is now to find the set of modes  $\hat{\mathbf{g}}$  that minimize Eq. (6). This could be done using any of a variety of optimization methods, but here we make use of a greedy algorithm that employs a sequence of elementary moves that merge and split clusters, inspired by a similar merge-split algorithm for sampling community structures described in Peixoto<sup>32</sup>. We start by randomly dividing our set  $D$  of partitions into some number  $K_0$  of initial clusters, then identify the mode  $\hat{\mathbf{g}}^{(k)}$  of each cluster  $C_k$  as the partition  $p \in C_k$  that minimizes  $H(\mathbf{g}^{(p)}) + \sum_{q \in C_k} H_{\text{mod}}(\mathbf{g}^{(q)}|\mathbf{g}^{(p)})$ . In other words, the initial mode for each cluster is the partition  $p$  that is closest to all other partitions  $q$  in the cluster in terms of modified conditional entropy, accounting for the entropy of  $p$  itself.

Computing the modified conditional entropy, Eq. (4), has time complexity  $O(N)$ , which means it takes  $O(NS^2/K_0^2)$  steps to compute each mode exactly if the initial clusters are the same size. This can be slow in practice, but we can obtain a good approximation substantially faster by Monte Carlo sampling. We draw a random sample  $X$  of partitions from the cluster (without replacement) and then minimize  $H(\mathbf{g}^{(p)}) + (c_k/|X|)\sum_{q \in X} H_{\text{mod}}(\mathbf{g}^{(q)}|\mathbf{g}^{(p)})$ , where as previously  $c_k$  is the size of the cluster. Good results can be obtained with relatively small samples, and in our calculations we use  $|X| = 30$ . The time complexity of this calculation is  $O(NS/K_0)$ , a significant improvement given that sample sizes  $S$  can run into the thousands or more. We also store the values of  $H(\mathbf{g}^{(p)})$  and  $H_{\text{mod}}(\mathbf{g}^{(q)}|\mathbf{g}^{(p)})$  as they are computed so that they do not need to be recomputed on subsequent steps of the algorithm.

Technically, our formulation does not require one to constrain  $\hat{\mathbf{g}}^{(k)}$  to be a member of  $C_k$ , but this restriction significantly reduces the computation time in practice by allowing stored conditional entropy values to be reused repeatedly during calculation. One could relax this restriction and choose the mode  $\hat{\mathbf{g}}^{(k)}$  of each cluster  $C_k$  to be the partition  $\mathbf{g}$  (which may or may not be in  $C_k$ ) that minimizes  $H(\mathbf{g}) + \sum_{q \in C_k} H_{\text{mod}}(\mathbf{g}^{(q)}|\mathbf{g})$ . However, we have not taken this approach in the examples presented here.

Once we have an initial set of clusters and representative modes, the algorithm proceeds by repeatedly proposing one of the following moves at random, accepting it only if it reduces the value of Eq. (6):

1. Pick a partition  $\mathbf{g}^{(p)}$  at random and assign it to the closest mode  $\hat{\mathbf{g}}^{(k)}$ , in terms of modified conditional entropy.
2. Pick two clusters  $C_{k'}$  and  $C_{k''}$  at random and merge them into a single cluster  $C_k$ , recomputing the cluster mode as before.
3. Pick a cluster  $C_k$  at random and split it into two clusters  $C_{k'}$  and  $C_{k''}$  using a  $k$ -means style algorithm: we select two modes at random from  $C_k$  and assign each partition in  $C_k$  to the closer of the two (in terms of modified conditional entropy). Then we recompute the modes for each resulting cluster and repeat until convergence is reached.

These steps together constitute a complete algorithm for minimizing Eq. (6) and optimizing the clusters, but we find that the efficiency of the algorithm can be further improved by adding a fourth move:

4. Perform step 2, then immediately perform step 3 using the merged cluster from step 2.

This extra move, inspired by a similar one in the community merge-split algorithm of Peixoto<sup>32</sup>, helps with the rapid optimization of partition assignments between pairs of clusters.

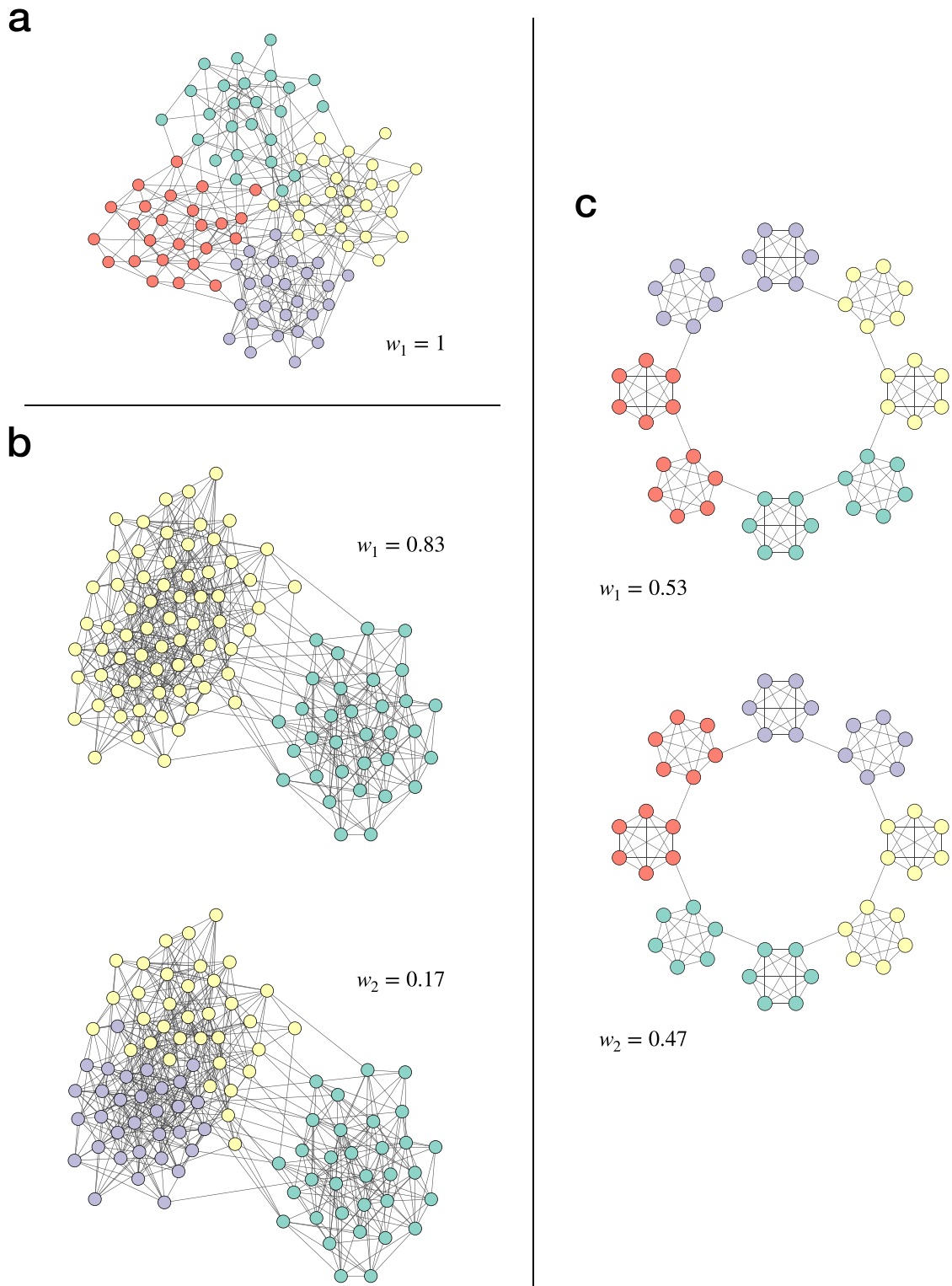
We continue performing these moves until a prescribed number of consecutive moves are rejected without improving the objective function. We find that this procedure returns very consistent results despite its random nature. If results were found to vary between runs it could be worthwhile to perform random restarts of the algorithm and adopt the results with the lowest objective score. However, this has not proved necessary for the examples presented here.

The algorithm has  $O(NS)$  time complexity per move in the worst case (which occurs when there is just a single cluster), and is fast in practice. In particular, it is typically much faster than the community detection procedure itself for current community detection algorithms, so it adds little to the overall time needed to analyze a network. We give a range of example applications in the next section.

**Example applications: synthetic networks.** In the following sections, we demonstrate the application of our method to a number of example networks, both real and computer-generated. For each example, we perform community detection by fitting to the non-parametric degree-corrected block model<sup>33</sup> and sampling 10 000 community partitions from the posterior distribution of the model using Markov chain Monte Carlo. These samples are then clustered using the method of this paper with the cluster penalty parameter set to  $\lambda = 1$ , the number of Monte Carlo samples for estimating modes to  $|X| = 30$ , and the number of initial modes to  $K_0 = 1$ . We also calculate for each mode  $k$  a weight  $w_k = c_k/S$  equal to the fraction of all partitions in  $D$  that fall in cluster  $k$ , to assess the relative sizes of the clusters.

As the first test of our method, we apply it to a set of synthetic (i.e., computer-generated) networks specifically constructed to display varying degrees of ambiguity in their community structure. Figure 2a shows results for a network generated using the planted partition model, a symmetric version of the stochastic block model<sup>34,35</sup> in which  $N$  nodes are assigned in equal numbers to  $q$  communities, and between each pair of nodes  $i, j$  an edge is placed with probability  $p_{\text{in}}$  if  $i$  and  $j$  are in the same community or  $p_{\text{out}}$  if  $i$  and  $j$  are in different communities. In our example we generated a network with  $N = 100$  nodes,  $q = 4$  communities, and  $p_{\text{in}} = 0.25$ ,  $p_{\text{out}} = 0.02$ . Though it contains four communities, by its definition, this network should exhibit only a single mode, the structure “planted” into it in the network generation process. There will be competing individual partitions, but they should be distributed evenly around the single modal structure rather than multimodally around two or more structures. And indeed our algorithm correctly infers this as shown in the figure: it returns a single representative structure in which all nodes are grouped correctly into their planted communities. Given the random nature of the community detection algorithm, it would be possible for a small number of nodes to be incorrectly assigned in the modal structure, simply by chance, but in the present case this did not happen and every node is assigned correctly.

For a second, more demanding example we construct a network using the full (non-symmetric) stochastic block model, which is more flexible than the planted partition model. If  $\mathbf{g}$  denotes a vector of community assignments as previously, then an edge in the model is placed between each node pair  $i, j$  independently at random with probability  $\omega_{\mathbf{g}, \mathbf{g}'}$ , where the  $\omega_{\mathbf{g}, \mathbf{g}'}$  are parameters that we choose. For our example, we create a



**Fig. 2 Representative modes and their corresponding weights for three synthetic example networks. a** Planted partition model with 100 nodes, 4 communities, and connection probabilities  $p_{in} = 0.25$  and  $p_{out} = 0.02$ . **b** Network of 99 nodes generated using the stochastic block model with a mixing matrix of the form given in Eq. (7) with  $p_s = 0.27$ ,  $p_m = 0.08$ , and  $p_b = 0.01$ . **c** Ring of eight cliques of six nodes each, connected by single edges, based on the example in ref. <sup>36</sup>. Representative partitions are identified by minimizing Eq. (6) with penalty parameter  $\lambda = 1$  for 10,000 community partition samples. The color of each node indicates its community membership within a partition, and  $w_k$  is the weight of mode  $k$ .

network with three communities and with parameters of the form

$$\omega = \begin{bmatrix} p_s & p_m & p_b \\ p_m & p_s & p_b \\ p_b & p_b & p_s \end{bmatrix}, \quad (7)$$

where  $p_s$  is the within-group edge probability,  $p_m$  and  $p_b$  are between-group probabilities, and  $p_s > p_m > p_b$ . In our particular example, the network has  $N = 99$  nodes divided evenly between the three groups and  $p_s = 0.27$ ,  $p_m = 0.08$ ,  $p_b = 0.01$ . This gives the network a nested structure in which there is a clear separation between group 3 and the rest, and a weaker separation between groups 1 and 2. This sets up a deliberate ambiguity in the community structure: does the “correct” structure have three groups or just two? As shown in Fig. 2b, our method accurately pinpoints this ambiguity, finding two representative modes for the network, one with three separate communities and one where communities 1 and 2 are merged together.

A third synthetic example network is shown in Fig. 2c, the “ring of cliques” network of Fortunato and Barthelemy<sup>36</sup>, in which a set of cliques (i.e., complete subgraphs) are joined together by single edges to create a loop. In this case, we use eight cliques of six nodes each. Good et al.<sup>12</sup> found this kind of network to have an ambiguous community structure in which the cliques joined together in pairs rather than forming separate communities on their own. Since there are two symmetry-equivalent ways to divide the ring into clique pairs this also means there are two equally good divisions of the network into communities. Good et al. performed their community detection using modularity maximization, but similar behavior is seen with the method used here. Most sampled community structures show the same division into pairs of cliques, except for a clique or two that may get randomly assigned as a whole to a different community. Our algorithm readily picks out this structure as shown in Fig. 2c, finding two modes that correspond to the two rotationally equivalent configurations. Moreover, the two modes have approximately equal weight  $w_k$  in the sampling, indicating that the Monte Carlo algorithm spent a roughly equal amount of time on partitions near each mode.

**Example applications: real networks.** Turning now to real-world networks, we show that our method can also accurately summarize community structure found in a range of practical domains. (Further examples are given in Supplementary Fig. 1, under Supplementary Note 3.) The results demonstrate not only that the method works but also that real-world networks commonly do have a multimodal community structure that is best summarized by two or more modes rather than by just a single consensus partition, although our method will return a single partition when it is justified—see the section on *Example applications: synthetic networks* above.

Figure 3a shows results for one well-studied network, the co-purchasing network of books about politics compiled by Krebs (unpublished, but see<sup>37</sup>), where two books are connected by an edge if they were frequently purchased by the same buyers. It has been conjectured that this network contains two primary communities, corresponding to politically left- and right-leaning books, but the network contains more subtle divisions as well. A study by Peixoto<sup>14</sup> found 11 different types of structure—what we are here calling “modes.” Many of these modes, however, differed only slightly, by the reassignment of a few nodes from one community to another. Applying our method to the network we find, by contrast, just two modes as shown in the figure, suggesting that our algorithm is penalizing minor variations in structure more heavily than that of Ref. 14. The two modes we

find have four communities each. In the one on the left in Fig. 3a, these appear to correspond approximately to books that are politically liberal (red), center-left (purple), center-right (green), and conservative (yellow); in the one on the right, they are left-liberal (green), liberal (red), center (purple), and conservative (yellow).

Figure 3b shows a different kind of example, a social network of self-reported friendships among US high school students drawn from the National Longitudinal Study of Adolescent to Adult Health (the “Add Health” study)<sup>38,39</sup>. The particular network we examine here is network number 5 from the study with 157 students. (Two nodes with degree zero were removed from the network before running the analysis.) As the figure shows, the method in this case finds three modes, each composed of half a dozen core communities of highly connected nodes whose boundaries shift somewhat from one mode to another, as well as a set of centrally located nodes (pale pink and yellow in the figure) that seem to move between communities in different modes. The movement of nodes from one community to another may be a sign of different roles played by core and peripheral members of social circles, or of students with a broad range of friendships.

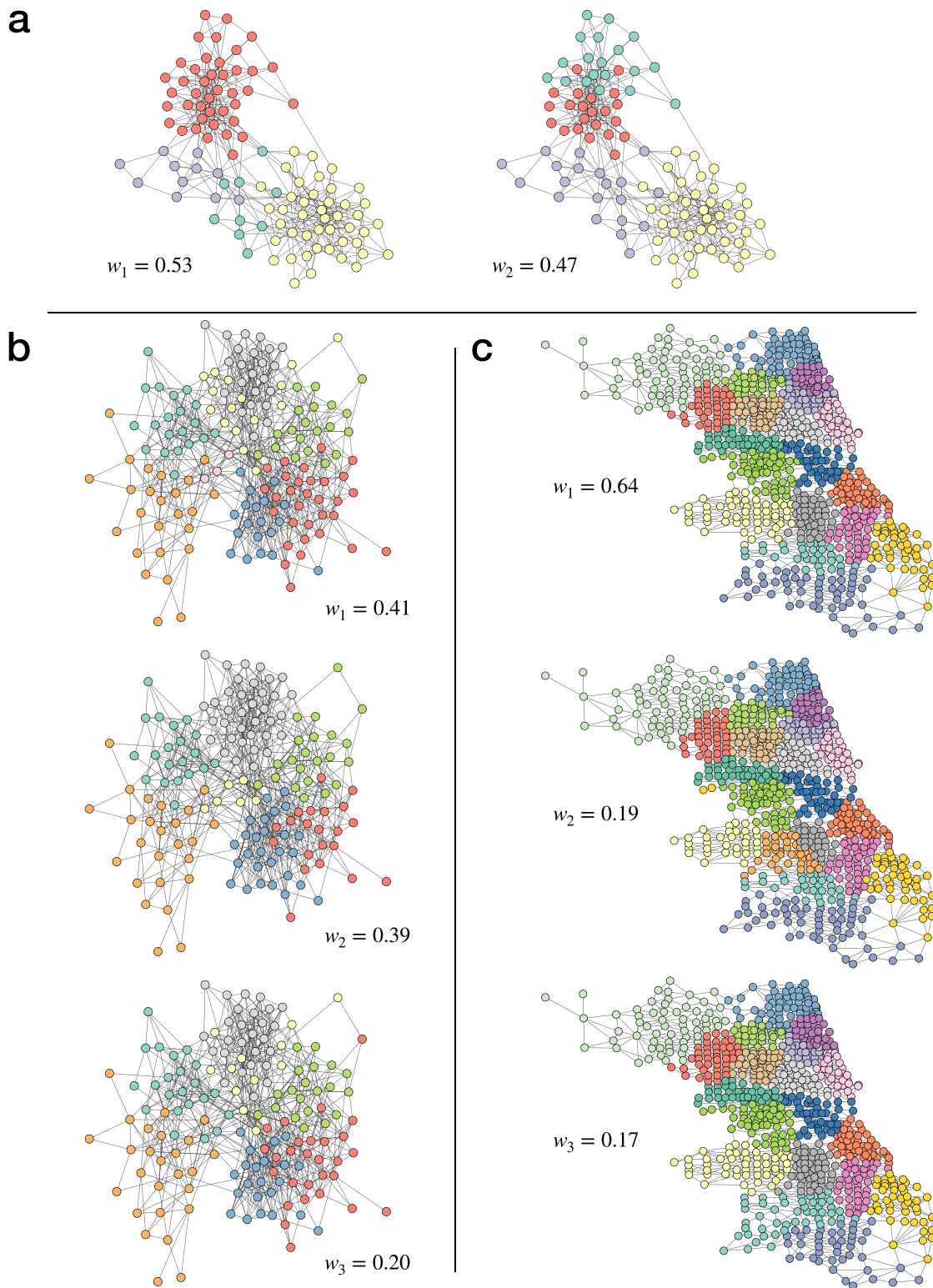
In Fig. 3c, we show a third type of network, a geographic network of census tracts in the city of Chicago (USA). In this network, the nodes represent the census tracts and two nodes are joined by an edge if the two corresponding tracts share a border<sup>40</sup>. Community detection applied to this network tends to find contiguous local neighborhoods. Our algorithm finds three modes that differ primarily in the communities on the southwest side of the city where the density of census tracts is lower (though it is unclear whether this is the driving factor in the variation of community structure).

## Conclusion

In this paper, we have presented a method for summarizing the complex output of community detection algorithms that return multiple candidate network partitions. The method identifies a small number of archetypal partitions that are broadly representative of high-scoring partitions in general. The method is based on fundamental information-theoretic principles, employing a clustering objective function equal to the length of the message required to transmit a set of partitions using a specific multi-step encoding that we describe. We have developed an efficient algorithm to minimize this objective and we give examples of applications to both synthetic and real-world networks that exhibit nontrivial multimodal community structure.

One can envisage many potential applications of this approach. As mentioned in *Example applications: real networks*, the representative community partitions for a social network could highlight distinct roles or reveal information about the diversity of a node’s social circle. In networks for which we have additional node metadata, we could investigate how individual attributes are associated with the representative partitions. Multimodal community structure may also be of interest in spatial networks, for instance for assessing competing partitions, as in mesh segmentation in engineering and computer graphics<sup>41</sup>. More generally, in the same way that any measurement can be supplemented with an error estimate, any community structure analysis could be supplemented with an analysis of competing partitions to help understand whether the optimal division is representative of the structure of the network as a whole.

The techniques presented in this paper could be extended in a number of ways. Our framework is applicable to any set of partitions—not just community divisions of a network but partitions



**Fig. 3 Representative modes and their corresponding weights for three real-world example networks. a** Network of political book co-purchases<sup>37</sup>. **b** High school friendship network<sup>38,39</sup>. **c** Network of adjacent census tracts in the city of Chicago<sup>40</sup>. Representative partitions are identified by minimizing Eq. (6) with penalty parameter  $\lambda = 1$  for 10,000 community partition samples. The color of each node indicates its community membership within a partition, and  $w_k$  is the weight of mode  $k$ .



of any set of objects or data items—so it could be applied in any situation where there are multiple competing ways to cluster objects. All that is needed is an appropriate measure of the information required to encode representative objects and their corresponding clusters. One potential application within network science could be to the identification of representative networks within a set sampled from some generative model, such as an exponential random graph model<sup>42</sup>. These extensions, however, we leave for future work.

### Data availability

All data needed to evaluate the conclusions in the paper are present in the paper and/or Supplementary Materials, except for the real (non-synthetic) network data sets, which are available from the original sources cited.

### Code availability

Code for the partition clustering algorithm presented in this paper is available at <https://github.com/aleckirkley/Community-Representatives>

Received: 27 July 2021; Accepted: 28 January 2022;

Published online: 17 February 2022

### References

- Newman, M. *Networks* 2nd edn (Oxford University Press, 2018).
- Bedi, P. & Sharma, C. Community detection in social networks. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **6**, 115–135 (2016).
- Huang, W. & Li, C. Epidemic spreading in scale-free networks with community structure. *J. Stat. Mech.* **2007**, P01014 (2007).
- Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proc. Natl Acad. Sci. USA* **99**, 7821–7826 (2002).
- Newman, M. E. J. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **69**, 066133 (2004).
- Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc. Natl Acad. Sci. USA* **105**, 1118–1123 (2008).
- Peixoto, T. P. in *Advances in Network Clustering and Blockmodeling* (eds Doreian, P., Batagelj, V. & Ferligoj, A.) 289–332 (Wiley, 2019).
- Fortunato, S. Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010).
- Guimerà, R., Sales-Pardo, M. & Amaral, L. A. N. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E* **70**, 025101 (2004).
- Massen, C. P. & Doye, J. P. K. Identifying communities within energy landscapes. *Phys. Rev. E* **71**, 046101 (2005).
- Reichardt, J. & Bornholdt, S. Statistical mechanics of community detection. *Phys. Rev. E* **74**, 016110 (2006).
- Good, B. H., de Montjoye, Y.-A. & Clauset, A. Performance of modularity maximization in practical contexts. *Phys. Rev. E* **81**, 046106 (2010).
- Riolo, M. A. & Newman, M. E. J. Consistency of community structure in complex networks. *Phys. Rev. E* **101**, 052306 (2020).
- Peixoto, T. P. Revealing consensus and dissensus between network partitions. *Phys. Rev. X* **11**, 021003 (2021).
- Zhang, P. & Moore, C. Scalable detection of statistically significant communities and hierarchies, using message passing for modularity. *Proc. Natl Acad. Sci. USA* **111**, 18144–18149 (2014).
- Guimerà, R. & Sales-Pardo, M. Missing and spurious interactions and the reconstruction of complex networks. *Proc. Natl Acad. Sci. USA* **106**, 22073–22078 (2009).
- Gong, M., Ma, L., Zhang, Q. & Jiao, L. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Phys. A Stat. Mech. Appl.* **391**, 4050–4060 (2012).
- Lancichinetti, A. & Fortunato, S. Consensus clustering in complex networks. *Sci. Rep.* **2**, 1–7 (2012).
- Calatayud, J., Bernardo-Madrid, R., Neuman, M., Rojas, A. & Rosvall, M. Exploring the solution landscape enables more reliable network community detection. *Phys. Rev. E* **100**, 052308 (2019).
- Vinh, N. X., Epps, J. & Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010).
- Grünwald, P. D. & Grünwald, A. *The Minimum Description Length Principle* (MIT Press, 2007).
- Tabor, J. & Spurek, P. Cross-entropy clustering. *Pattern Recognit.* **47**, 3046–3059 (2014).
- Wallace, R. S. & Kanade, T. Finding natural clusters having minimum description length. In *10th International Conference on Pattern Recognition* 438–442, (IEEE Press, 1990).
- Li, T., Ma, S. & Ogihara, M. Entropy-based criterion in categorical clustering. In *Proc. Twenty-first International Conference on Machine Learning* 68 (Association for Computing Machinery, 2004).
- Narasimhan, M., Jojic, N. & Bilmes, J. A. Q-clustering. *Adv. Neural Inf. Process. Syst.* **18**, 979–986 (2005).
- Georgieva, O., Tschumitschew, K. & Klawonn, F. Cluster validity measures based on the minimum description length principle. In *Proc. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* 82–89 (Springer-Verlag, 2011).
- Rosvall, M. & Bergstrom, C. T. An information-theoretic framework for resolving community structure in complex networks. *Proc. Natl Acad. Sci. USA* **104**, 7327–7331 (2007).
- Cover, T. M. & Thomas, J. A. *Elements of Information Theory* (John Wiley, 1991).
- Newman, M. E. J., Cantwell, G. T. & Young, J.-G. Improved mutual information measure for clustering, classification, and community detection. *Phys. Rev. E* **101**, 042304 (2020).
- Doane, D. P. Aesthetic frequency classifications. *Am. Stat.* **30**, 181–183 (1976).
- Hall, P. Akaike's information criterion and Kullback-Leibler loss for histogram density estimation. *Probab. Theory Relat. Fields* **85**, 449–467 (1990).
- Peixoto, T. P. Merge-split Markov chain Monte Carlo for community detection. *Phys. Rev. E* **102**, 012305 (2020).
- Peixoto, T. P. Nonparametric Bayesian inference of the microcanonical stochastic block model. *Phys. Rev. E* **95**, 012317 (2017).
- Holland, P. W., Laskey, K. B. & Leinhardt, S. Stochastic blockmodels: first steps. *Soc. Netw.* **5**, 109–137 (1983).
- Karrer, B. & Newman, M. E. J. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107 (2011).
- Fortunato, S. & Barthelemy, M. Resolution limit in community detection. *Proc. Natl Acad. Sci. USA* **104**, 36–41 (2007).
- Newman, M. E. J. Modularity and community structure in networks. *Proc. Natl Acad. Sci. USA* **103**, 8577–8582 (2006).
- Bearman, P. S., Moody, J. & Stovel, K. Chains of affection: the structure of adolescent romantic and sexual networks. *Am. J. Sociol.* **110**, 44–91 (2004).
- Udry, J. R., Bearman, P. S. & Harris, K. M. National Longitudinal Study of Adolescent Health <http://www.cpc.unc.edu/addhealth> (1997).
- Kirkley, A. Information theoretic network approach to socioeconomic correlations. *Phys. Rev. Res.* **2**, 043212 (2020).
- Shamir, A. in *Computer Graphics Forum*, Vol. 27, 1539–1556 (The Eurographics Association and John Wiley & Sons, 2008).
- Lusher, D., Koskinen, J. & Robins, G. *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications* (Cambridge University Press, 2012).

### Acknowledgements

This work was funded in part by the US Department of Defense NDSEG fellowship program (to A.K.) and by the National Science Foundation under grant number DMS-2005899 (to M.E.J.N.).

### Author contributions

A.K. and M.E.J.N. designed the study and wrote the manuscript, and A.K. performed the mathematical and computational analysis.

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s42005-022-00816-3>.

**Correspondence** and requests for materials should be addressed to Alec Kirkley.

**Peer review information** *Communications Physics* thanks Daniel Edler and the other anonymous reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.

**Reprints and permission information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022