Routledge
Taylor & Francis Group

Multivariate Behavioral Research

ISSN: 0027-3171 (Print) 1532-7906 (Online) Journal homepage: https://www.tandfonline.com/loi/hmbr20

Detecting Clusters/Communities in Social Networks

Michaela Hoffman, Douglas Steinley, Kathleen M. Gates, Mitchell J. Prinstein & Michael J. Brusco

To cite this article: Michaela Hoffman, Douglas Steinley, Kathleen M. Gates, Mitchell J. Prinstein & Michael J. Brusco (2018) Detecting Clusters/Communities in Social Networks, Multivariate Behavioral Research, 53:1, 57-73, DOI: 10.1080/00273171.2017.1391682

To link to this article: https://doi.org/10.1080/00273171.2017.1391682

Published online: 08 Dec 2017.

MULTIVARIATE BEHAVIORAL RESEARCH
2018, VOL. 53, NO. 1, 57–73
https://doi.org/10.1080/00273171.2017.1391682

Routledge
Taylor & Francis Group

Detecting Clusters/Communities in Social Networks

Michaela Hoffman[a], Douglas Steinley[a], Kathleen M. Gates[b], Mitchell J. Prinstein[b], and Michael J. Brusco[c]

[a]University of Missouri, Columbia; [b]University of North Carolina, Chapel Hill; [c]Florida State University

ABSTRACT

Cohen's κ, a similarity measure for categorical data, has since been applied to problems in the data mining field such as cluster analysis and network link prediction. In this paper, a new application is examined: community detection in networks. A new algorithm is proposed that uses Cohen's κ as a similarity measure for each pair of nodes; subsequently, the κ values are then clustered to detect the communities. This paper defines and tests this method on a variety of simulated and real networks. The results are compared with those from eight other community detection algorithms. Results show this new algorithm is consistently among the top performers in classifying data points both on simulated and real networks. Additionally, this is one of the broadest comparative simulations for comparing community detection algorithms to date.

KEYWORDS

Network analysis; cluster analysis; community detection; Cohen's kappa

Introduction

Network models, while they have a long history in sociological and computer science research, are recently finding new applications in psychology. Traditionally, a network is modeled using a graph, and much network analysis is based on graph theory. A graph is composed of a set of nodes (i.e., vertices, actors) denoting the subjects being studied, and a set of edges (i.e., links, connections) reflecting the relationships between the subjects. In classical social network analysis, these nodes are people and the edges represent interpersonal relationships, such as friendships. A common example is Zachary's Karate Club, as illustrated in Figure 1, a network of collegiate karate club members where the edges indicate whether or not each pair of members is friends (Zachary, 1977). Studying social relations has its obvious applications for social scientists studying individual and group behavior, but the nodes are not limited to representing people. For example, applications of network analysis in functional magnetic resonance imaging (fMRI) research depicting the functional connectivity of the brain (e.g., Alexander-Bloch et al., 2012; Jacob et. al. 2016).

There are many potential questions of interest when analyzing network data. The focus of this paper is on one in particular: community detection. Community detection identifies clusters of nodes within networks that, in terms of their neighbors, are internally connected but externally isolated (Cormack, 1971)[1]. In the karate network example, there was a split of the club members into two factions after a disagreement. The goal of community detection here is to identify who belongs to which group, based on the members' recorded friendships. The true split in the club is depicted in Figure 1, as identified by the differentially shaded nodes.

Concomitant with the increased use of network data is a growing amount of literature dedicated to the task of correctly identifying node membership in communities. While the state of the science has advanced quickly in a short time, there are still gaps in the ability of available community detection algorithms to provide reliable results. Namely, most algorithms were designed to analyze very large networks; however, there have been very few examinations of how the methods perform in terms of actually recovering group structure similar to what has been conducted in the long history of traditional cluster analysis (Milligan, 1980; Steinley, 2003, 2006a; Steinley & Brusco, 2007, 2011). Furthermore, it is not clear whether these "scalable" algorithms will prove reliable or useful for many lines of inquiry found in psychological studies that use smaller samples of individuals. The current paper presents a novel community detection approach based on Cohen's κ for clustering social network data and provides one of the first comprehensively comparative studies of community detection algorithms.

Much like graph partitioning and blockmodeling (Wasserman & Faust, 1994; Doreian, Batagelj, & Ferligoj,

CONTACT Michaela Hoffman HoffmaMi@musc.edu Medical University of South Carolina, Charleston, SC, 29407, USA.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/hmbr.
[1]While traditionally referred to as cluster analysis in the fields of sociology and psychology, we use the increasingly popular denotation of "community detection" to tie the present work to modern literature of identifying groups (e.g., clusters) within the context of network data.

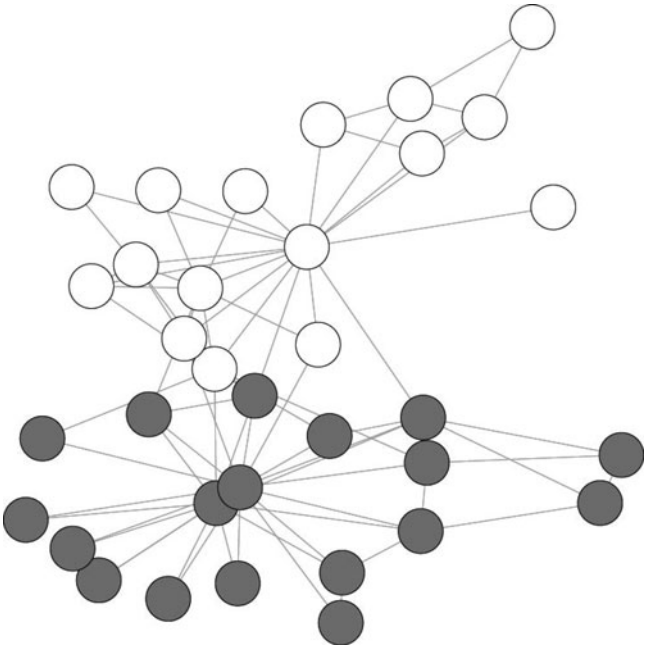

Figure 1. The Zachary's Karate Club network depicts friendships between 34 members of a collegiate karate club (Zachary, 1977). The nodes of this graph are colored to indicate group membership after the club split into two factions.

2005), community detection is dedicated to identifying subsets of individuals where members of a given subset are closer, by some measure, to each other than those in other subsets. This is analytically similar to traditional cluster analysis, but is conducted within the context of network data. Indeed, cluster analysis provides the basis for many community detection techniques. In community detection, traditional cluster analysis is often conducted not on the original network matrix but rather on one that has been recast using some sort of distance measure between individuals in the network as described in detail in the "methods" section (e.g., often referred to as a proximity matrix in cluster analysis, see Arabie, Hubert, & De Soete, 1996). Generally, these algorithms take the form of hierarchical clustering and fall into one of two classes, agglomerative or divisive. Agglomerative approaches begin with each individual in their own "community," and these communities are then iteratively merged until there is one community containing all individuals. Divisive approaches work in the opposite manner by starting with all individuals in one community and separating them into smaller and smaller communities until each individual is in their own community. In either case, the researcher must decide the stopping point or how many communities to keep.

To solve this problem, Newman and Girvan (2004) introduced a score function termed "modularity" for analytically identifying the optimal stopping point for identifying communities. This negated the need for researchers to make subjective decisions regarding where to stop the clustering procedure. Given modularity's high utility and initial successes in recovering the underlying community structures, researchers (predominantly in the fields of physics and computer science) developed algorithms in quick succession that utilize modularity. However, the increased use of "big data" formats prompted developers to focus on algorithms that were 1) computationally efficient and 2) able to reliably recover community structure in very large networks (i.e., 1,000,000 individuals).

It remains largely unclear how well these methods perform on data types that are often seen in the social sciences. For instance, researchers focusing on social networks within schools or organizations likely have far fewer individuals than the data types that motivated the currently available algorithms. Recent work suggests that Cohen's κ may provide an appropriate distance measure for use in community detection for smaller networks (Hoffman, Steinley, & Brusco, 2015). In the application of this distance measure to network data, the calculation uses both neighbors (i.e., individuals who share a connection) and nonneighbors (i.e., individuals who do not share a connection). The addition of the nonneighbors in the calculation makes it particularly suited to smaller networks as this information is meaningful, but not considered in many other binary distance measures. It also has the benefit of the nonarbitrary cutoff value, where any κ greater than zero indicates that the two individuals have more in common than would be expected by chance[2]. The present paper provides a solution in the form of an algorithm based on Cohen's κ, cluster analysis, and modularity that identifies communities of individuals even within smaller networks. We then evaluate the novel approach on simulated data and compare results with those found in a number of commonly used community detection algorithms. Next, we will apply κ on three empirical data sets. Finally, a discussion is provided for suggestions on how to choose among the available community detection algorithms.

Community detection methodology

A number of algorithms exist for finding communities in network data. This paper reviews and compares eight of the most prominent and accessible community detection algorithms. These methods are a good representation of

[2] We note that, for data that are sampled, the presence of a zero could be due to measurement or sampling error. Consequently, it is possible to observe values that are greater than zero that actually have less in common than would be expected by chance. However, we do note that, unlike Hoffman et al. (2015) where the value of zero was used for determining whether a link may be missing (or in fact, observed by chance alone), the present study does not treat the $\kappa = 0$ as a threshold for decisions to be made. Indeed, all values of κ are retained and subsumed within a distance formula; as such, we expect that any error will – in the long run – cancel itself out.

the current state of the field – both in variety and use – and are all available through the R package igraph (Csardi & Nepusz, 2006). The eight algorithms included in this study are edge betweenness (EB), fast greedy (FG), walktrap (WT), label propogation (LP), leading eigenvector (LE), multilevel community (ML), spinglass community (SG), and optimal community (OC). These algorithms are summarized below. For a detailed review of the current community detection literature, see Fortunato (2010).

Formally, a network is described by the graph:

(1) $\mathcal{G}$:= the graph composed of the set $\mathcal{N}$ of nodes and the set $\mathcal{L}$ of edges. All networks examined herein are undirected and binary[3];
(2) N := the number of nodes (e.g., observations or vertices) in the graph (network), where the ith node is represented as n_i and $i = 1, \ldots, N$;
(3) L := the number of edges (e.g., links or connections) in the graph (network), where the jth edge is represented as l_j and $j = 1, \ldots, L$;
(4) $\mathbf{X}_{N\times N} = \{x_{ij}\}_{N\times N}$:= the adjacency matrix corresponding to the network, where $x_{ij} = 1$ if there is an edge between n_i and n_j, and $x_{ij} = 0$ if there is not an edge between the two nodes;
(5) K := the number of communities (e.g., clusters or groups) in the network, indexed $k = 1, \ldots, K$;
(6) C_k := the set of nodes in the kth community;
(7) N_k := the number of nodes in the kth community;
(8) $L_{C_k,C_{k'}}$:= the number of edges connecting community k and community k';
(9) d_i := the degree of n_i, where the degree is the number of other nodes that n_i is connected to;
(10) $\delta(i, j)$:= an indicator function where $\delta(i, j) = 1$ if n_i and n_j are in the same community, and $\delta(i, j) = 0$ if n_i and n_j are in different communities.

Modularity

There are many specific definitions of a "community," depending on the type of network and communities being studied. Generally, a community is described as a subgroup of nodes that has many edges between its members, and a few edges connecting to nonmembers (or other communities) – once again, the classic notion of high within-community density and low between-community density that hearkens back to Cormack (1971). *Modularity* is the measure that is most commonly used: (a) to evaluate the community structure of a graph, and (b) as an objective function within many community detection methods (Newman, 2004; Newman & Girvan, 2004). Modularity is calculated, in a form described in Clauset, Newman, & Moore (2004), as:

$$Q = \frac{1}{2L}\sum_{ij}\left[x_{ij} - \frac{d_i d_j}{2L}\right]\delta(i, j). \quad (1)$$

Zero is the expected value of Q if the network's edges were formed randomly. Modularity is thus a calculation of the fraction of edges found within the same community minus the expected value for an equivalent network: one having the same partition of nodes into communities but where the edges are formed between pairs in the entire network at random. As a rule of thumb, a value greater than 0.30 implies that there is significant community structure (Clauset et al., 2004). As will be seen when the algorithms are described, the number of communities, K, is often determined based on the modularity score. In this paper, modularity is a component of all algorithms with the exception of *label propagation*.

Community detection methods

Method 1: Edge betweenness

One early community detection algorithm, sometimes referred to as Girvin and Newman's algorithm, is *edge betweenness* (Newman & Girvan, 2004). This is a divisive algorithm that works on the assumption that edges connecting nodes of separate communities will be given high centrality scores, where centrality is some measurement of the "importance" of a node within its network. *Betweenness* is an edge centrality measure calculated by counting the number of shortest paths (e.g., geodesics) between every node pair that includes the edge in question. The algorithm itself proceeds as:

1. Calculate betweenness for every edge in the graph;
2. Remove edge with the highest betweenness score;
3. Recalculate betweenness for all remaining edges;
4. Repeat from step 2.

The final partition of the nodes is selected by calculating Q at every split, and selecting the final number of communities to correspond to the maximum value of Q. This simultaneously provides the final partition and number of communities. Additionally, this is similar to a divisive hierarchical approach in that the algorithm proceeds from when the graph is connected and terminates when all nodes are in isolated clusters. The most common measure for betweenness is shortest path betweenness, which is based on the shortest geodesic paths (Wasserman & Faust, 1994, p. 190).

[3] An undirected graph implies that the edges describe a relationship that has no meaningful direction, such that the relationship is always reciprocated, creating a symmetric adjacency matrix. Binary networks are those whose relationships cannot take on values other than 1 (edge present) and 0 (no edge), meaning the edges do not have any strength information.

Method 2: Fast greedy

The *fast greedy* algorithm is an agglomerative hierarchical clustering method that optimizes Q by merging the pair of communities at each step to produce the largest increase in Q (Newman, 2004). Considering a simplified formula for Q in equation (1):

$$Q = \sum_k (e_{kk'} - e_k^2), \tag{2}$$

where $e_{kk'}$ is the fraction of edges in the network that connects C_k and $C_{k'}$, $e_{kk'} = L_{C_k,C_{k'}}/L$, and $e_k = \sum_{k'} e_{kk'}$. Then, the change in Q, ΔQ, when merging two groups is given as:

$$\Delta Q = 2(e_{kk'} - e_k e_{k'}). \tag{3}$$

The method proceeds as follows:

1. Begin with $K = N$ communities (every node is its own community).
2. Calculate change in modularity, ΔQ, for every pair of communities.
3. Merge the two communities associated with the greatest increase (or the smallest decrease) in Q.
4. Repeat from step 2 until all nodes are contained in a single community.

The final partition of nodes and number of communities is selected in the same way as the edge–betweenness algorithm. The modularity of the graph is calculated at each merge, and the level with the highest Q is selected. This particular formulation has a very similar relationship to Ward's hierarchical clustering, where the goal is to merge clusters such that the overall variance is increased the least at each step.

Method 3: Walktrap

The *walktrap* method uses short random walks on a graph to detect communities (Pons & Latapy, 2005). It assumes that random walks within a graph should get "trapped" within the communities.

A *random walk process* begins on a selected node and moves to another node chosen randomly and uniformly from its neighbors, and then proceeds to a next node in the same way, with the number of steps specified as the walk length. The length of the walk must be short enough to not be trivial, but long enough to gather community information. From the network, a transition matrix $\mathbf{P}_{N\times N} = \{p_{ij}\}$ is formed where $p_{ij} = \frac{x_{ij}}{d_i}$ is the transition probability from n_i to n_j at any step. For a random walk of length m starting at n_i, the probability of ending at n_j is P_{ij}^m. From here a distance measure, $D(i,\ j)$, between n_i and n_j is calculated:

$$D(i,\ j) = \sqrt{\sum_{n=1}^{N} \frac{(P_{in}^m - P_{jn}^m)^2}{d_n}}\quad . \tag{4}$$

This is then generalized to a distance measure between communities:

$$D(C_k, C_{k'}) = \sqrt{\sum_{n=1}^{N} \frac{(P_{C_k n}^m - P_{C_{k'} n}^m)^2}{d_n}}, \tag{5}$$

where $P_{C_k j}^m = \frac{1}{N_k}\sum_{i\in C_k} P_{ij}^m$ is the probability of going from C_k to n_j, where $n_j \notin C_k$, in m steps.

The walktrap algorithm proceeds as follows:

1. Begin with N communities (every node is its own community) and calculate the distances $D(i,\ j)$ for each pair.
2. Use Ward's criterion to merge two communities (minimize the average squared distance between each node and its community):

$$\sigma_k = \frac{1}{N}\sum_k \sum_{i\in C_k} D^2(C_k, i). \tag{6}$$

3. Update the distances between adjacent communities.
4. Repeat steps 2 and 3 until all nodes are in the same community.

Finally, the best choice of K is selected from the sequence of communities with increasing k based on the maximum modularity.

Method 4: Label propagation

Label propagation begins with each node having a unique label, and iteratively updates nodes to take on the labels of the majority of its neighboring nodes (Raghavan, Albert, & Kumara, 2007). Communities with high density will quickly adopt a common label. The process continues until all nodes belong to the community that the maximum number of its neighbors belongs to:

1. Every node begins with a unique label.
2. In a random sequential order, modify the label of each node to be that of the majority of its neighbors (in the case of no majority, a label is selected randomly).
3. Repeat step 2 until each node has the label of the majority of its neighbors: for every node i :

$$\text{If } i \text{ has label } C_k,\ \text{then } d_i^{C_k} \geq d_i^{C_k'} \quad \forall k',$$

where $C_1, \ldots, C_K$ are the community labels in the network, and $d_i^{C_k}$ is the count of node i's neighbors labeled C_k.

In the final solution, the unique labels define the communities in the network. Throughout the iterations, most

labels disappear leaving only a certain number of communities. The number of communities, K, does not need to be chosen ahead of time or with the modularity approach the previous algorithms use. Because of the ties that often occur at step 2, this algorithm does not have a unique solution. The resulting partitions, however, do tend to be similar and the authors suggest adopting an aggregation method.

Method 5: Leading eigenvector

Newman developed another algorithm based on optimizing modularity and related to spectral clustering methods called *Leading Eigenvector* (Newman, 2006). This method begins with the modularity matrix $\mathbf{B} = \{q_{ij}\}_{N\times N}$ where $q_{ij} = x_{ij} - \frac{d_i d_j}{2L}$. $\mathbf{B}$ is used in place of the Laplacian matrix of traditional spectral clustering. Considering the two cluster case, $\mathbf{p}$ is a vector partition of the graph where $\mathbf{p}_i = 1$ if v_i belongs to the first cluster and $\mathbf{p}_i = -1$ if it belongs to the second. If λ_j is an eigenvalue of $\mathbf{B}$ with associated eigenvector $\mathbf{u}_j$ then modularity can be rewritten:

$$\mathbf{B} = \frac{1}{4L}\sum_{i=1}^{N}(\mathbf{u}_i^T \cdot \mathbf{p})^2\lambda_i. \tag{7}$$

Denoting the largest eigenvalue as λ_1, its associated eigenvector, $\mathbf{u}_1$, is selected as the best partition, grouping the nodes with their signs. $\mathbf{B}$ always has a trivial eigenvector $\mathbf{u}_i = (1, 1, \ldots, 1)$ with corresponding eigenvalue $\lambda_i = 0$. In the case where there is no community structure, there will be no positive eigenvalues so the maximum is the trivial eigenvector where all nodes are part of one community.

This is extended to the case of more than two clusters by repeating the division into two groups on the individual communities to increase the graph modularity. The modularity matrix here is always calculated from the whole graph, not just that of the community being divided. This can proceed until the network is divided into N communities, or stop earlier when there is no longer any increase in modularity. Thus, the optimal number of communities K is defined by the point where there are no more divisions that can be made to increase modularity.

Method 6: Multilevel community (Louvain method)

Multilevel is another greedy modularity maximization-based approach originally developed for weighted networks. The goal of the "multilevel community" algorithm is to find the high modularity divisions of the network without the resolution issues associated with greedy optimization of modularity (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008):

Phase 1

1. Begin with N communities (every node is its own community).
2. Considering each node n_i sequentially, calculate ΔQ, the change in modularity made by moving n_i to the community of each of its neighbors n_j;
   (a) Make the move associated with the maximum gain (or use a tie breaking rule), if the gain is positive;
   (b) If there is no possible improvement in modularity, n_i remains in its community.
3. Repeat step 2 until no further improvement can be made.

Phase 2

4. Begin by creating a new network where the nodes are the communities' output from the first phase:
   (a) Weight the edges between these new "nodes" with the sum of the edges between the members of the corresponding communities; note edges within communities are now self-loops;
   (b) Reapply *Phase 1* to this network.
5. Repeat *Phase 2* until no more changes increase modularity.

This algorithm does have a potential problem of local optima, because the output is dependent on the order of the nodes being considered. The authors argue that this does not significantly influence the modularity obtained, but does influence computation time. However, for similar clustering algorithms (such as variations of the K-means clustering algorithm), Steinley (2003) found that the quality of the final solution can be greatly affected by the order in which the observations are considered.

Method 7: Spinglass community

Spinglass community is a method based on the Potts model, which comes out of the field of statistical mechanics. The method created by Reichardt and Bornholdt (2006) maps a network onto a zero-temperature K-Potts model with nearest-neighbor interactions. The Potts model is a system of spins that can be in K different states, optimizing an energy function. Connecting this to community detection, the spin states are the group labels of the nodes, and the energy of the spin system is the quality function of the communities. The quality function

defined here is the Hamiltonian:

$$\mathcal{H}(\{\mathcal{P}\}) = -\sum_{i\neq j} v_{ij} \underbrace{x_{ij}\delta(i,\,j)}_{\text{internal edges}} + \sum_{i\neq j} w_{ij} \underbrace{(1-xij)\delta(i,\,j)}_{\text{internal nonedges}} + \sum_{i\neq j} y_{ij} \underbrace{x_{ij}(1-\delta(i,\,j))}_{\text{external edges}} - \sum_{i\neq j} z_{ij} \underbrace{(1-x_{ij})(1-\delta(i,\,j))}_{\text{external nonedges}}, \quad (8)$$

where $\mathcal{P}$ is the partition of the data set, and recall that $\delta(i,\,j) = 1$ when v_i and v_j are in the same community and equal to zero otherwise. The values v_{ij}, w_{ij}, y_{ij}, and z_{ij} are weights, and must be selected to adjust the contribution of edges and nonedges. If p_{ij} denotes the normalized probability that an edge exists between n_i and n_j ($\sum_{i\neq j} x_{ij} = 2L$), then using $v_{ij} = 1 - \gamma p_{ij}$ and $w_{ij} = \gamma p_{ij}$ leads the Potts model to be associated with modularity.

Specifically, with undirected and unweighted networks, the Hamiltonian is simplified to

$$\mathcal{H}(\{\mathcal{P}\}) = -\sum_{i\neq j}(x_{ij} - \gamma p_{ij})\delta(i,\,j). \quad (9)$$

The edges present determine which dyads prefer to be in the same spin state/community. When $p_{ij} = d_i d_j/2L$ and $\gamma = 1$, the modularity in equation (1) can be rewritten:

$$Q = -\frac{1}{L}\mathcal{H}(\{\mathcal{P}\}). \quad (10)$$

Thus to maximize modularity on a network is equivalent to minimizing the Hamiltonian. This algorithm uses efficient update rules to directly optimize modularity. This method is slow to run, and can only be used for networks with up to a few thousand nodes.

Method 8: Optimal community

Optimal Community searches through all possible partitions of a graph and chooses the one that maximizes modularity. While this method is robust, it is the most limited in terms of applications, as it is not recommended for graphs with greater than 50 nodes. This task is $\mathcal{NP}$-complete (Brandes et al., 2008).

Proposed method: Cohen's κ

Cohen's κ

Cohen's κ was originally developed as a score calculated from the counts in a contingency table (see Table 1), determining whether or not the agreement between two categorical variables is greater than chance (Cohen, 1960).

Table 1. The contingency table from which κ is calculated for each pair of nodes n_i and n_j across all other nodes n_h.

	Node n_i	
Node n_j	Linked to Node n_h	Not Linked to Node n_h
Linked to node n_h	A	B
Not Linked to node n_h	C	D

Considering the binary case, the basic equation is as follows:

$$\kappa_{ij} = \frac{2(AD - BC)}{(A+B)(C+D) + (A+C)(B+D)}. \quad (11)$$

Agreement is counted with A (both select 1) and D (both select 0). The B and C counts indicate disagreement (where only one is 1). κ falls between -1 and 1, where a score of 0 is the level of random chance agreement.

When applied to networks, Cohen's κ can be used to assess the similarity between nodes in the network by comparing their neighborhoods (Hoffman et al., 2015). Instead of considering responses over a number of items, the variables are now two nodes, and the presence or absence of edges to the remaining nodes of the network makes up the contingency table counts. In a network, a score of $\kappa = 1$ would signify structural equivalence of two nodes (e.g., both nodes have the exact same pattern of edges to other nodes). Structural equivalence is an important concept from the blockmodeling literature. It is defined where two nodes play the exact same role in a network in terms of their edges, and would be considered interchangeable (Doreian et al., 2005).

The actual bounds of the Cohen's κ score, rather than being -1 and 1, have been demonstrated in practice to be a function of the marginal distributions of the contingency tables (Karelitz & Budescu, 2013). In the network application, this change in boundary will happen in the particular cases where the two nodes being compared have very different degrees (e.g., one has many neighbors and the other has very few). While Karelitz and Budescu (2013) provided a correction to κ to ensure that the bounds are recalibrated to -1 and 1 regardless of the marginal distributions of the induced contingency table, implementing the correction in this setting would sacrifice the strict definition of structural equivalence.

The network application of Cohen's κ was recently studied for the problem of link prediction (Hoffman et al., 2015) and demonstrated that the κ scores between pairs of nodes can be used as a proximity measure to detect edges (links) that are missing from the network (or predict where edges will form in the future). To do this, κ was calculated for each potential edge (each pair of unlinked nodes). Those with a score greater than zero were identified as missing edges. In Hoffman et al.'s simulation study,

κ was among the top performers when compared to other binary measures of similarity. Given the performance of the proposed algorithm as seen in the following section and the importance of structural equivalence as a theoretical concept in network analysis, we use the uncorrected version of κ in developing our community detection algorithm.

Cohen's k method for community detection

Here, Cohen's κ is further extended into the domain of network analysis by using it as the basis for a new method for community detection. The following section describes the procedure used, and then the analysis and comparison to other methods on both simulated and real-world data.

First, Cohen's κ is calculated as a similarity measure for each pair of nodes in the network, both linked and unlinked. This is done by considering $\mathbf{X}$. Specifically, for any pair of row vectors corresponding to n_i and n_j, $\mathbf{x}_i^{'}$ and $\mathbf{x}_j^{'}$, in the adjacency matrix the four quantities can be quickly computed from the following inner products:

$$
\begin{aligned}
A &= \mathbf{x}_i^{'}\mathbf{x}_j \\
B &= \mathbf{x}_i^{'}(\mathbf{1} - \mathbf{x}_j) \\
C &= (\mathbf{1} - \mathbf{x}_i)^{'}\mathbf{x}_j \\
D &= (\mathbf{1} - \mathbf{x}_i)^{'}(1 - \mathbf{x}_j)
\end{aligned}
$$

and, using equation (2), a value of κ for the pair of nodes can be computed.

It should be noted that in this application, the sum of the four values in the contingency table will not be the size of the total network (N) but rather $N - 2$. This is because for each pair of n_i and n_j, we do not consider the presence or absence of the edge between them x_{ij}, or self-links x_{ii}. At first, it seems natural to include the edge between n_i and n_j; however, preliminary analysis indicated that this reduced the performance of the proposed method of community detection. The primary cause of the reduction is that all of the edges between any pair of nodes, both "true positives" (e.g., edges between nodes in the same community) and "false positives" (e.g., edges between nodes in different communities) are considered to be indicative of membership in the same community. Consequently, this introduces a bias in the procedure where any potential measurement is disregarded, decreasing overall recovery of community structure.

From these calculations, a symmetric matrix $\mathbf{K}$ is formed, whose entry κ_{ij} contains the Cohen's κ score for nodes n_i and n_j. An example of this can be seen in Figure 2. The top part of the figure shows an example adjacency matrix, while the bottom part of the figure shows the associated matrix $\mathbf{K}$. The rows and columns in both of these are already permuted into blockmodel form, grouping the nodes within each of the three communities, to demonstrate how the κ scores reflect the communities. Within communities, the scores approach 1, whereas, between communities, the score approaches -1.

To determine the assignments of the nodes to communities, we conduct a K-means clustering approach (Steinley, 2006a) using the initialization method recommended by Steinley and Brusco (2007) which is by using the results of Ward's method for hierarchical clustering (Ward, 1963). Like K-means clustering, Ward's method attempts to find communities that have the smallest within-community variances. By initializing K-means clustering with Ward's method, we avoid some of the problems of locally optimal solutions known to occur with K-means clustering (Steinely, 2003, 2006b), and we are also able to provide solutions that are guaranteed to be as good as or better than conducting Ward's method on its own.

This procedure requires the number of communities, K, to be known a priori. In the case where the number of communities in our network is known, we would have a final solution with the data partitioned into the desired number of communities at this point. However, in most empirical applications, the number of communities in the data is not known. To address this problem, we can conduct the proposed cluster analysis from 2 to N clusters and calculate the modularity for each[4]. The final solution selected is that which maximizes the modularity. Modularity was chosen as the objective function in order to be consistent with other community detection algorithms considered, of which all but label propagation use. By allowing all of the algorithms (excepting label propagation) to choose the number of communities based on modularity, any differences in performance are due to the performance of algorithms, not the performance of the algorithms *plus* the method for choosing the number of communities.

In summary, the algorithm proceeds as follows[5]:

1. Calculate $\mathbf{K}_{N\times N} = \kappa_{ij}$.
2. Implement a K-means cluster analysis initialized with Ward's method for $k = 2, \ldots, N$. Note that the implementation of the K-means algorithm requires a squared-Euclidean distance matrix to be derived from $\mathbf{K}$ (this is similar to the creation of the distance matrix in the Walktrap algorithm).
3. Calculate Q for each partition.
4. Select the partition that maximizes Q.

[4] This step can be time consuming to implement when the network is very large, so we recommend choosing a reasonable stopping point; for the small simulations in this paper, N was used for the stopping point, but for the larger simulations $K = 20$ was the maximum K considered.

[5] This algorithm has complexity $O(N^2KT)$, where T is the number of iterations the K-means algorithm takes to converge, which generally is not too many (Steinley, 2003).

Adjacency matrix A

0	1	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	1	0	0	0	0
1	1	1	0	1	0	0	0	0	0	1	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	0	1	0	0	1	0	0	0	0
1	0	0	0	0	1	1	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	1	0	1	0	0	0	0	1
0	0	0	0	0	1	0	0	1	0	0	1	0	0	0
0	0	1	1	0	0	1	0	0	0	0	1	1	1	1
0	0	0	0	1	0	0	0	0	1	1	0	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
0	0	0	0	1	0	0	0	1	0	1	1	1	1	0

Cohen's κ matrix K

1.00	0.81	0.64	0.64	0.49	-0.18	-0.05	-0.36	-0.18	-0.41	-0.21	-0.41	-0.52	-0.52	-0.41
0.81	1.00	0.81	0.81	0.65	-0.44	-0.36	-0.08	-0.44	-0.36	-0.05	-0.27	-0.44	-0.44	-0.27
0.64	0.81	1.00	1.00	0.49	-0.52	-0.05	-0.18	-0.52	-0.41	-0.27	-0.10	-0.18	-0.18	-0.10
0.64	0.81	1.00	1.00	0.49	-0.52	-0.05	-0.18	-0.52	-0.41	-0.27	-0.10	-0.18	-0.18	-0.10
0.49	0.65	0.49	0.49	1.00	-0.59	-0.44	-0.27	-0.27	-0.12	0.24	-0.30	0.05	0.05	-0.30
-0.18	-0.44	-0.52	-0.52	-0.59	1.00	0.26	0.57	0.57	0.26	-0.34	-0.27	-0.44	-0.44	-0.27
-0.05	-0.36	-0.05	-0.05	-0.44	0.26	1.00	0.26	0.42	0.13	-0.30	-0.12	0.03	0.03	-0.12
-0.36	-0.08	-0.18	-0.18	-0.27	0.57	0.26	1.00	0.13	0.42	-0.34	-0.59	-0.44	-0.44	-0.27
-0.18	-0.44	-0.52	-0.52	-0.27	0.57	0.42	0.13	1.00	0.26	-0.34	0.05	-0.08	-0.08	-0.41
-0.41	-0.36	-0.41	-0.41	-0.12	0.26	0.13	0.42	0.26	1.00	-0.18	-0.28	0.03	0.03	0.20
-0.21	-0.05	-0.27	-0.27	0.24	-0.34	-0.30	-0.34	-0.34	-0.18	1.00	0.22	0.52	0.52	0.22
-0.41	-0.27	-0.10	-0.10	-0.30	-0.27	-0.12	-0.59	0.05	-0.28	0.22	1.00	0.65	0.65	0.68
-0.52	-0.44	-0.18	-0.18	0.05	-0.44	0.03	-0.44	-0.08	0.03	0.52	0.65	1.00	1.00	0.65
-0.52	-0.44	-0.18	-0.18	0.05	-0.44	0.03	-0.44	-0.08	0.03	0.52	0.65	1.00	1.00	0.65
-0.41	-0.27	-0.10	-0.10	-0.30	-0.27	-0.12	-0.27	-0.41	0.20	0.22	0.68	0.65	0.65	1.00

Figure 2. Example of an adjacency matrix **A** with three communities, containing five nodes each, and the matrix **K** containing the Cohen's κ scores for each pair of nodes.

Simulations

The simulation study was broken into two parts. The first part only analyzed networks smaller than 150 observations (nodes), whereas the larger simulation examined larger networks with more than 200 observations. It is worth noting that these scenarios can be seen as "small" and "large" networks in the context of social sciences; however, they are both dwarfed by some of the network sizes often employed in the physical sciences. Each simulation study has a distinct data generation approach to facilitate the investigation of network properties that may be of differential interest as network size increases.

Simulation 1: Small networks

Simulation structure

The Cohen's κ algorithm (CK) for community detection is tested by analyzing its ability to recover known communities on a variety of simulated data sets. First a set of smaller networks are considered, containing less than 150 nodes. For this simulation the communities are of equal size ($N_k = 20$), and consistent with Hoffman et al. (2015), three characteristics of the networks varied. First, three different numbers of communities were used: 2, 4, and 6. The second and third factors deal with the strength of the community structure of the network. *Density within communities* describes the percentage of edges present between nodes of the same cluster (using 90%, 75%, and 60%). Lower percentages here are associated with a less-defined group structure, and it would be expected that the algorithms have more difficulty detecting the communities. *Density between communities* is the percentage of edges present between nodes belonging to different communities, modified at three levels: 10%, 25%, and 40%. As this percentage increases, the separate communities move "closer" to each other, which is expected to make community detection more difficult.

The method used to generate network data with these different characteristics is carried out by generating a symmetric square adjacency matrix with a blockmodel structure. The K blocks on the diagonal representing the

K clusters have entries generated with the probability of the given "density within," and the of-diagonal blocks representing edges between the clusters have entries generated with a probability equal to the given "density between." Ten networks are created at each combination of the three network characteristics.

For comparison, these same networks are analyzed with other commonly used community detection algorithms described above: EB, FG, WT, LP, LE, ML, and SG. Note that, here, optimal partitioning (OP) is not feasible to be included in the full simulation study as many of the networks will have sample sizes larger than 50 (see the Appendix on restrictions on OP for community detection). The remaining seven community detection algorithms were all implemented within the R package igraph, using the default settings. The final partitions from each algorithm are compared to the true node assignments using the adjusted Rand index (ARI; see Hubert & Arabie, 1985; Steinley, 2004; Steinley, Hendrickson, & Brusco, 2015, for details on the ARI and some of its properties) which is the preferred method to evaluate cluster recovery. The formula for computing the ARI when comparing two partitions is

$$\mathrm{ARI} = \frac{\binom{N}{2}(A'+D') - [(A'+B')(A'+C') + (C'+D')(B'+D')]}{\binom{N}{2}^2 - [(A'+B')(A'+C') + (C'+D')(B'+D')]}, \tag{12}$$

where A' is the number of pairs of nodes laced in the same community in both partitions, B' is the number of pairs placed in the same community in the first partition but in different communities in the second partition, C' is the number of pairs placed in different communities in the first partition but the same community in the second partition, and D' is the number of pairs placed in different communities in both partitions. For simulation studies (see Steinley, 2003, 2006b), one of the partitions is the "true" community structure while the other is the community structure recovered by the algorithm being evaluated. Generally, an ARI score above 0.65 indicates that the method was able to find the community structure (Steinley, 2004). For those networks with no community structure (where the relative density between clusters is 1) an ARI of 0 is expected.

Results

The averages of the ARIs for the different network characteristics can be found in Table 2. Analysis of variance (ANOVA) results followed by a Tukey's test for multiple mean comparison showed no significant difference between the overall means of the top performing algorithms: Cohen's κ, walktrap, multilevel, and spinglass.

Table 2. Results for the small network simulation. These are the ARIs comparing the results of each method to the true partition averaged over the varying network characteristics.

	CK	FG	EB	WT	LP	LE	ML	SG
Number of communities								
2	0.91*	0.86	0.78	0.89	0.46	0.91	0.90	0.89
4	0.85	0.65	0.50	0.82	0.25	0.67	0.86	0.88*
6	0.79	0.44	0.36	0.76	0.17	0.50	0.78	0.83*
8	0.70	0.31	0.31	0.70	0.09	0.38	0.68	0.76*
Density within								
60%	0.60	0.41	0.37	0.57	0.15	0.44	0.60	0.64*
75%	0.85	0.59	0.49	0.82	0.24	0.64	0.86	0.90*
90%	0.99*	0.69	0.60	0.98	0.33	0.76	0.95	0.99
Density between								
10%	1.00*	0.82	0.94	1.00	0.62	0.84	1.00	0.99
25%	0.89	0.56	0.38	0.86	0.08	0.63	0.87	0.92*
40%	0.55	0.32	0.14	0.52	0.03	0.37	0.54	0.61*
Average	0.81	0.56	0.49	0.79	0.24	0.61	0.80	0.84*

Note: "*" indicates the best performing algorithm per condition. For the average performance, there was no significant difference between κ, walktrap, multilevel, and spinglass. CK, Cohen's κ; EB, Edge betweenness; FG, Fast greedy; WT, Walktrap; LP, Label propagation; LE, Leading eigenvector; MC, Multilevel; SG, Spinglass.

As expected the higher density within communities and lower density between communities were associated with improved community recovery for all of the algorithms. Better results were also associated with fewer communities, though in this simulation this result is confounded with the overall network size.

Simulation 2: Larger networks

Simulation structure

For the larger sized networks, there is more flexibility in network generation due to being able to generate small communities in the presence of realistic sparseness, allowing the varying of five different characteristics for the networks. These encompass a wider range of network structures that might be observed in practice, and provide scenarios where the measure Cohen's κ might be compromised by the overall density of the network. Three sizes of networks are considered (N= 200, 400, and 600) and four number of communities (K= 2, 4, 6, and 8). The size of clusters for these networks is not held constant, instead, consistent with Milligan's (1980) seminal approach, there are three different distributions of the nodes into communities: nodes are evenly distributed into clusters; one cluster contains 10% of the nodes and the rest are evenly distributed, and one cluster contains 60% of the nodes with the rest evenly distributed. To create network conditions varying the strength of community structure, nine levels of density within were considered, in increments of 0.1 from 0.1 to 0.9. Larger values of density within correspond to more edges between nodes of the same cluster, making stronger communities. Finally, the density between communities, reflecting how well

separated the communities are, was calculated relative to the density within communities. Varied at 10 levels from 0.1 to 1, the relative density between is the ratio of the density between and density within. For example, when density within is 0.9, and the relative density between is 0.1, the density between is actually $0.1 \times 0.9 = 0.09$. As the density between communities decreases, the communities are increasingly separated from each other, making a stronger community structure overall in the network. In the extreme case, where the relative density between communities is 1, the density between is equal to the density within, making a graph without a real community structure. These data are analyzed separately, as the comparisons to the "true" community assignments are not meaningful.

All conditions were completely crossed in this experimental design, giving $3 \times 4 \times 3 \times 9 \times 10 = 3{,}240$ combinations, with 10 replications each, creating 32,400 networks. A subset of the algorithms from the smaller simulation study was compared to the Cohen's κ approach: FG, WT, LP, LE, ML. The algorithms EB, SG, and OP were not included due to the size of the simulation[6].

Results: Cluster recovery

The ARIs for each algorithm averaged over the different network characteristics can be found in Table 3. The proposed method (CK; average ARI = 0.60) and walktrap (WT; average ARI = 0.59) were the top performing algorithms overall in this simulation. The two are significantly different, likely due to the large sample size; however, the effect size related to their difference is small ($d_{\text{Cohen}} = 0.033$). Both were significantly different from the other four methods. Given that CK had the highest average ARI (both overall and within each level of every condition except when the relative density between was 0.1 or 0.2), subsequent probing of the effects will focus solely on this algorithm.

The probing of the results for CK to understand the influence of the network characteristics on this method of community detection takes the form of a traditional ANOVA and is consistent with other simulation studies of this nature (Shireman, Steinley, & Brusco, 2016). A five-way ANOVA, where the response was the ARI for CK and the factors were the different simulation characteristics that were varied, including potential two-way interactions is utilized to compare the effects of the five manipulated network variables[7]. Consistent with other simulation studies, the favored effect size here is the variance accounted for (VAF) for each of the simulation factors. The most influential factor was the relative density between communities ($\eta^2 = 0.44$). Referring to the first column of Table 3, it can be seen that the performance decreases steadily as the relative density between communities is increased, beginning with a very good recovery average of ARI = 0.90 when the relative density between is 0.1, to the very poor ARI = 0.07 when the relative density between is 0.9. The next largest effect was the density within communities ($\eta^2 = 0.17$). Here the communities with the highest density within proved the easiest to recover with an average ARI = 0.83, and the lowest density within had an average ARI = 0.26. The interaction parameter for the density within and relative density between had the next largest effect size ($\eta^2 = 0.05$). The

Table 3. The ARIs comparing the results of each method to the true partition averaged over the varying network characteristics for the larger network simulation.

	CK	WT	ML	LE	FG	LP
Relative community sizes						
Equal	0.64*	0.62	0.62	0.51	0.46	0.13
One large	0.61*	0.60	0.45	0.51	0.50	0.05
One small	0.55*	0.53	0.44	0.32	0.28	0.08
Number of communities (k)						
2	0.72*	0.68	0.52	0.56	0.54	0.14
4	0.66*	0.65	0.60	0.55	0.52	0.10
6	0.55*	0.54	0.49	0.39	0.35	0.06
8	0.48	0.47	0.40	0.29	0.26	0.05
Size of network (N)						
200	0.50*	0.50	0.41	0.37	0.34	0.08
400	0.62*	0.60	0.52	0.46	0.43	0.09
600	0.68*	0.66	0.57	0.51	0.47	0.09
Density within communities						
0.1	0.26*	0.25	0.16	0.16	0.17	0.05
0.2	0.42*	0.40	0.29	0.26	0.26	0.07
0.3	0.51*	0.50	0.39	0.34	0.33	0.08
0.4	0.58*	0.56	0.47	0.41	0.39	0.09
0.5	0.63*	0.61	0.54	0.47	0.43	0.09
0.6	0.68*	0.66	0.59	0.52	0.48	0.10
0.7	0.73*	0.71	0.64	0.57	0.51	0.10
0.8	0.77*	0.76	0.69	0.61	0.56	0.10
0.9	0.83*	0.82	0.74	0.66	0.61	0.10
Relative density between communities						
0.1	0.90	0.91*	0.79	0.71	0.74	0.54
0.2	0.87	0.87*	0.76	0.66	0.66	0.18
0.3	0.83*	0.82	0.71	0.61	0.59	0.05
0.4	0.77*	0.75	0.65	0.55	0.51	0.01
0.5	0.69*	0.67	0.57	0.48	0.44	0.00
0.6	0.58*	0.55	0.46	0.41	0.35	0.00
0.7	0.44*	0.40	0.33	0.31	0.26	0.00
0.8	0.26*	0.23	0.18	0.20	0.15	0.00
0.9	0.07*	0.07	0.05	0.06	0.04	0.00
Average	0.60*	0.59	0.50	0.45	0.42	0.09

Note: "*" indicates the best performing algorithm per condition. CK, Cohen's κ; FG, Fast greedy; WT, Walktrap; LP, Label propagation; LE, Leading eigenvector; ML, Multilevel.

[6] While they may be useful for networks in the smallest condition where $N = 200$, the number of combinations and iterations makes them unfit for the simulation study.

[7] We note that the ARI is not normally distributed; however, Steinley (2006b) showed that transforming the ARI does not change the interpretation or ordering of the results, indicating that in this setting of a completely balanced factorial analysis of variance, the results are robust to the nonnormality of ARI. Further, Steinley, Brusco, and Hubert, (2016) showed that the ARI is well approximated by the normal distribution, providing more confidence in using ANOVA for determining the relative magnitude of the effect sizes. Additionally, given that we are not relying on inference or even interpreting the p-values, rather, just the decomposition of the overall sum of squares, the formal assumptions of the ANOVA model do not pose a barrier for interpreting the magnitude of the effects

Table 4. The average ARIs comparing the results of the proposed method with the true partition for the different levels of density within (rows) and relative density between (columns).

	Relative density between									
Density within	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.9	0.95	0.94	0.94	0.94	0.93	0.90	0.84	0.67	0.33	0.00
0.8	0.95	0.94	0.94	0.93	0.91	0.87	0.75	0.52	0.16	0.00
0.7	0.94	0.93	0.93	0.92	0.88	0.80	0.65	0.41	0.09	0.00
0.6	0.93	0.92	0.92	0.89	0.83	0.73	0.55	0.30	0.04	0.00
0.5	0.93	0.90	0.90	0.86	0.79	0.64	0.45	0.21	0.01	0.00
0.4	0.92	0.91	0.86	0.81	0.70	0.53	0.34	0.12	0.00	0.00
0.3	0.91	0.86	0.83	0.72	0.57	0.41	0.22	0.06	0.00	0.00
0.2	0.87	0.80	0.70	0.57	0.42	0.26	0.11	0.01	0.00	0.00
0.1	0.73	0.59	0.43	0.31	0.18	0.09	0.02	0.00	0.00	0.00

average ARIs for this interaction can be found in Table 4. The degrading performance of the algorithm can be seen moving toward the bottom right corner of the table, as the structure of the communities becomes less clear.

After accounting for the within- and between-community densities, the most influential network characteristic was the number of communities ($\eta^2 = 0.05$), where increasing K was associated with a decreasing average ARI. This effect was followed by the interaction term for the number of communities and the relative community sizes ($\eta^2 = 0.05$). Of this interaction, the results are particularly poor for the cases with one small community for the two community case (ARI = 0.46), whereas in the case of two communities, ARI = 0.83 for both the equal cluster size and one large cluster conditions. This is likely because of the acute difference in community size here, where one contains 10% and the other 90% of the nodes. Finding small communities is a common difficulty in cluster analysis and community detection (Steinley, 2003, 2006b). In this simulation, as the number of communities increases within a given network size, the size of the individual communities decreases. This is also reflected in compared performance across network sizes, where the ARIs increase as the network size does ($\eta^2 = 0.03$). The remaining network factors had little influence on the results (with $\eta^2 \leq 0.01$).

Results: Estimating the number of clusters

Looking at the difference between the true number of communities and the number of communities recovered in Table 5, we see an overestimation in the number of communities where the values are positive and an underestimation where the values are negative. Generally, the number of communities was underestimated most often for all the algorithms except multilevel clustering (ML). CK seems to overestimate communities on average in three conditions. The first is in the lowest density within condition. This results is unsurprising, as the less connected the communities are the more likely they are to be broken apart by the algorithm. The overestimation in number of communities is also seen in the high density between condition, possibly due to between-community edges pulling apart communities. For the two community networks, the number of communities is also overestimated on average.

Table 5. The average difference between the number of true communities and the number of communities recovered by each algorithm for the large simulation.

	CK	WT	ML	LE	FG	LP
Relative community sizes						
Equal	− 0.26	− 0.11	0.46	− 0.84	− 1.55	− 3.59
One large	− 2.23	− 1.40	− 0.73	− 2.31	− 2.31	− 3.94
One small	− 0.09	0.87	1.21	− 0.68	− 1.50	− 3.67
Number of communities (k)						
2	0.83	1.89	2.30	0.66	0.62	− 0.86
4	− 0.24	0.07	0.75	− 0.44	− 0.83	− 2.74
6	− 1.32	− 0.83	− 0.22	− 1.76	− 2.48	− 4.69
8	− 2.72	− 1.98	− 1.58	− 3.56	− 4.46	− 6.65
Size of network (N)						
200	− 0.76	0.19	0.37	− 1.21	− 1.63	− 3.75
400	− 0.87	− 0.33	0.35	− 1.32	− 1.85	− 3.73
600	− 0.95	− 0.50	0.23	− 1.30	− 1.89	− 3.72
Density within communities						
0.1	0.19	5.21	3.13	− 0.24	− 0.32	− 3.83
0.2	− 0.46	0.62	1.68	− 1.11	− 1.39	− 3.78
0.3	− 0.79	− 0.54	0.83	− 1.42	− 1.79	− 3.74
0.4	− 0.93	− 0.98	0.26	− 1.53	− 1.94	− 3.73
0.5	− 1.00	− 1.19	− 0.17	− 1.54	− 2.02	− 3.72
0.6	− 1.10	− 1.28	− 0.41	− 1.51	− 2.07	− 3.71
0.7	− 1.22	− 1.25	− 0.63	− 1.44	− 2.10	− 3.70
0.8	− 1.22	− 1.26	− 0.83	− 1.40	− 2.19	− 3.70
0.9	− 1.24	− 1.25	− 1.02	− 1.27	− 2.26	− 3.70
Relative density between communities						
0.1	− 0.88	0.76	− 0.18	− 0.09	− 1.19	− 2.00
0.2	− 0.99	0.40	− 0.23	− 0.59	− 1.54	− 3.68
0.3	− 1.11	0.16	− 0.16	− 0.98	− 1.75	− 3.94
0.4	− 1.20	− 0.06	− 0.06	− 1.26	− 1.88	− 3.99
0.5	− 1.22	− 0.36	0.12	− 1.51	− 1.93	− 4.00
0.6	− 1.18	− 0.63	0.36	− 1.65	− 1.95	− 4.00
0.7	− 1.13	− 0.73	0.65	− 1.75	− 1.95	− 4.00
0.8	− 0.54	− 0.82	1.00	− 1.83	− 1.94	− 4.00
0.9	0.49	− 0.65	1.34	− 1.82	− 1.96	− 4.00
Average	− 0.86	− 0.21	0.31	− 1.27	− 1.79	− 3.73

Note: The values in the table are the average of the number of communities recovered – the true number of communities in each generated network. Consequently, negative numbers denote that the method indicated too many communities. CK, Cohen's κ; FG, Fast greedy; WT, Walktrap; LP, Label propagation; LE, Leading eigenvector; ML, Multilevel.

Outside of these conditions, and overall, the number of communities is underestimated. For each of the algorithms, the number of communities is underestimated the most in the eight community condition, and the condition with one large community. This reinforces the idea that for community detection algorithms in general, the most difficult situations are when one community is dominant and when there are many different communities – a situation where there are this one central community and several small satellite communities.

Empirical network data

Finally, the performance of the algorithms is compared on some real network data sets with known community structure. These analyses are performed in the same way

as the simulated networks, using the ARI to compare the partitions given by each algorithm to the true partitions when available. All eight methods from igraph, in addition to the proposed method (CK), are compared here, though optimal (OP) is not used for the School Data due to its size.

(1) Zachary's Karate Club (Zachary, 1977) is a data set often seen in the community detection literature. This network defines $M = 78$ friendships between $N = 34$ members of a karate club at a university in the 1970s. The club splits into $K = 2$ groups after a disagreement.

(2) The American College Football network (Girvan & Newman, 2002) contains $N = 115$ Division 1 football teams divided into the $K = 11$ conferences they belonged to in the 2000 season. The edges represent games played during the season ($M = 615$). Most of the games (but not all) are between teams in the same conference, creating the community structure. Because this network includes all Division I collegiate football games in that year, there are five teams included who do not belong to any conference (independent schools), adding noise to the data. For the calculation of the ARI, these teams are labeled as their own communities here, giving a total of $K = 12$ communities.

(3) School Data (Choukas-Bradley, Giletta, Neblett, & Prinstein, 2015) is a network of high school students and their self-reported friendships. Students were asked to write down the name of their best friend, and then from a sociometric roster where they marked the name of their best friend as well as all other friends they had. For this network, we considered any friendship nomination of these as an edge and made all edges reciprocal creating an undirected, unweighted network. This leaves us with $n = 412$ students connected with $m = 2062$ friendships (an overall density of 0.0243). The main distinguishing feature of this data set versus the Karate Club and Football data set is that the number of communities *is not* known a priori.

Results: When the number of communities is known

The ARIs comparing the partitions output by each algorithm to the true community structure of the two real data can be found in Table 6 [8]. When using the Cohen's κ method, the resulting two community partitions for the Karate Club data had an ARI of unity when compared to the true partition. Furthermore, κ was the only method to have perfect recovery of the true network.

Table 6. The ARI values comparing the solutions from each algorithm to the true known partition for the real network data, followed by the number of clusters indicated in each solution.

Data Set	Measure	CK	EB	FG	WT	LP	LE	ML	SG	OP
Karate										
	ARI	1.00	0.47	0.68	0.33	0.57	0.51	0.46	0.54	0.54
	$\hat{k}$	2	5	3	5	2	4	4	4	4
Football										
	ARI	0.90	0.78	0.48	0.82	0.78	0.46	0.81	0.85	0.81
	$\hat{k}$	12	10	6	10	11	8	10	11	10

Note: CK, Cohen's κ; EB, Edge betweenness; FG, Fast greedy; WT, Walktrap; LP, Label propagation; LE, Leading eigenvector; ML, Multilevel; SG, Spinglass; OP, Optimal.

Cohen's κ was also the top performing algorithm on the American College Football data set. It divided the network into 12 communities (e.g., conferences). With an ARI of 0.90, there were only six misclassified teams. The four independent teams were placed with conferences closest to them, which, here, we do not count as misclassifications, notwithstanding that they technically belong to none of the conferences. These bring up a different issue not addressed here, that of outliers in a network.Table 6

Results: When the number of communities is unknown

Detecting community structure in the school data is somewhat more difficult because the number of communities is not known a priori. Figure 3 provides a graphical representation of the network, and the nodes are colored by community for each of the algorithms. The layout of the network was created using the Fruchterman–Reingold function in the igraph package; this force-directed method creates the graph by modeling the distances between nodes with attractive and repulsive forces, which is very similar to traditional multidimensional scaling (Borg & Groenen, 1997). At face value, it appears that there are three clusters.

Table 7 indicates the agreement (in terms of ARI) between the various algorithms, while Table 8 breaks down information about community sizes and within-community densities. Two of the methods, edge betweenness and walktrap, extract quite a large number of communities as compared to the other methods: 16 and 15, respectively. While it is possible that many communities may be present in a given data set, doubts are raised

[8] Upon inspecting , it is seen that the performance of the different algorithms is not exactly the same as what was seen in the simulation. While Cohen's κ still performed the best, other methods were "shuffled" around somewhat (e.g., label propagation performed better on these data sets than in the simulation). This should not be overly concerning as the simulation results provide the average performance over numerous, independent data sets; however, they do not imply that methods that perform better on average perform better in every instance.

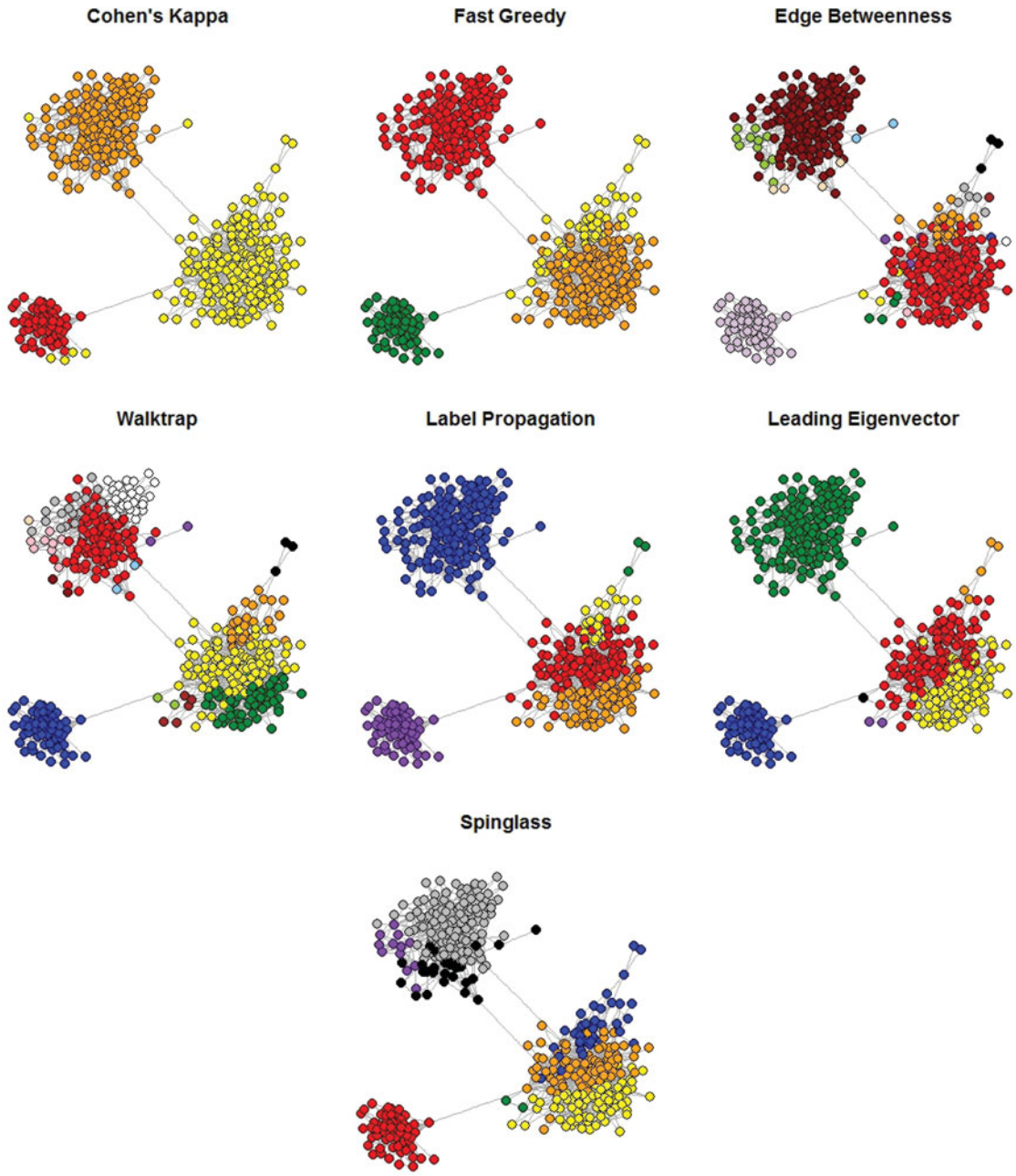

Figure 3. Plots of the communities identified in the school network indicated by different colors. The layout was produced using the Fruchterman–Reingold function in the igraph package.

Table 7. The ARI measuring agreement between the community assignments (partitions) for the school network given by each algorithm.

	CK	FG	EB	WT	LP	LE	ML	SG
CK	1	0.78	0.70	0.43	0.65	0.68	0.56	0.49
FG		1	0.82	0.51	0.77	0.75	0.55	0.64
EB			1	0.57	0.68	0.66	0.48	0.66
WT				1	0.62	0.51	0.68	0.64
LP					1	0.81	0.69	0.78
LE						1	0.65	0.63
ML							1	0.57
SG								1

Note: CK, Cohen's κ; EB, Edge betweenness; FG, Fast greedy; WT, Walktrap; LP, Label propagation; LE, Leading eigenvector; ML, Multilevel; SG, Spinglass.

here because (as seen in Table 8) several of the clusters have sizes that are quite small, with some of them being isolates (e.g., only containing one observation). Steinley (2008) introduced the following measure (in the original notation) for comparing the quality of solutions with different numbers of clusters when using square matrices:

$$\Psi = \frac{\Omega}{\Lambda K}, \tag{13}$$

where Ω is the sum of within-cluster edges (e.g., the within cluster density), Λ is the sum of between cluster edges (e.g., between cluster densities), and K is the

Table 8. Comparison of the size of communities (n_k = size of the kth community) and densities within communities for the results of each algorithm on the school network.

	CK		FG		EB		WT		LP		LE		ML		SG	
	n_k	DW	n_k	DW	n_k	DW	n_k	DW	n_k	DW	n_k	DW	n_k	DW	n_k	DW
1	204	5%	155	6%	154	7%	101	8%	155	5%	155	6%	122	6%	117	8%
2	153	6%	153	7%	136	7%	86	9%	96	8%	101	6%	118	7%	82	9%
3	55	19%	59	17%	59	17%	61	16%	79	12%	89	11%	59	17%	78	13%
4			45	11%	22	29%	59	17%	59	17%	59	17%	44	25%	59	17%
5					11	43%	34	28%	20	31%	5	40%	37	24%	36	15%
6					6	28%	26	22%	3	67%	2	50%	32	18%	26	14%
7					5	40%	20	31%			1	0%			12	40%
8					4	63%	8	63%							2	50%
9					3	44%	5	48%								
10					3	67%	3	67%								
11					3	67%	2	50%								
12					2	50%	2	50%								
13					1	0%	2	50%								
14					1	0%	2	50%								
15					1	0%	1	0%								
16					1	0%										

Note: CK, Cohen's κ; EB, Edge betweenness; FG, Fast greedy; WT, Walktrap; LP, Label propagation; LE, Leading eigenvector; MC, Multilevel; SG, Spinglass; DW, Within-community density; n_k, Size of the kth community.

number of clusters. Obviously, there is a penalty for fitting more clusters, and the best solution is the one which maximizes Ψ. For the school data, the values for Ψ, for the number of clusters chosen as indicated in Table 7, are Cohen's κ ($\Psi = 0.6707$), edge betweenness ($\Psi = 0.0249$), walktrap ($\Psi = 0.0161$), label propagation ($\Psi = 0.0363$), fast greedy ($\Psi = 0.1023$), multilevel ($\Psi = 0.0274$), leading eigenvector ($\Psi = 0.0235$), and spinglass ($\Psi = 0.0215$). As can been seen, the procedure based on κ is finding communities that are much more homogeneous and isolated without overfitting the data.

Conclusion

In this paper, we introduced a new algorithm, based on Cohen's κ, and provided one of the first broad-based investigations of community detection algorithms. The proposed algorithm provided better recovery of community structure than many other methods, under a wide range of conditions. Generally, we recommend the use of Cohen's κ as a proximity measure for input into a standard K-means clustering program for detecting community structure. While Cohen's κ performed the best on the larger networks (under a broader range of conditions), we also note that the walktrap algorithm, while significantly different, was very similar in performance. However, none of the other selected algorithms were able to match κ's recovery on the two empirical networks examined.

The difficulties in finding smaller sized communities are a common problem in community detection, particularly linked to modularity. Since most algorithms, including the newly proposed algorithm, choose a final solution based on the modularity metric, this may be the reason that too few communities are found (particularly when there were several small communities). This is sometimes called the resolution limit problem in community detection and is discussed in Fortunato & Barthelemy (2007) and Lancichinetti & Fortunato (2011). Current work is focused on examining the distribution of κ for a network as a method for determining both the presence and the number of communities.

This paper focuses on one limited area of network analysis: small- to medium-sized graphs that are undirected and unweighted. In the social sciences, however, while an undirected and unweighted structure can be imposed on the data, this type of network may not always be the best to reflect the true structure of a system or study the specific question of interest. Kindermann & Gest (2009) discuss some the methods and implications of collecting network data to identify social groups. For example, using self-report methods to build a network, data might be collected asking individuals to rate how much they "like" other people in their network, which would results in a graph with nonreciprocal and weighted edges. Alternatively, asking subjects to identify pairs who "hang out a lot" would create an undirected network, as considered in the analyses here. The methodology proposed in this paper could be easily extended to incorporate the information provided by different types of networks. The use of Cohen's κ for directed network analysis, for example, is a potential application requiring a slight modification of the similarity calculation. More work is needed in this area, particularly in the interpretation for that type of data. The framework of this algorithm could also be used for other networks where Cohen's κ is not well suited. Alternative distance measures can be used, for example, for very large networks where the quantity d, the nonneighbors, may no longer be informative.

An additional limitation to this methodology that is common to the algorithms discussed is that in the final solution, a node's membership is limited to one group, not allowing over overlapping communities. This limitation applies particularly to the *K*-means clustering algorithm used in the procedure. Replacing this with more flexible methodology is a possible avenue for further research, as it is possible that the Cohen's κ matrix could indicate overlap in the distance measures. As such, it seems particularly profitable to pursue the identification of overlapping network communities by adapting the MAthematicalProgrammingCLUStering (MAPCLUS) algorithm (Arabie & Carroll, 1980) and various cluster visualization schemes (Carroll & Corter, 1995) to the context of network analysis.

Finally, the network generation method is based on the idea/community definition that nodes within a community will be more likely to connect to each other and less likely to connect to those of other communities. In other words, within-community sections of the network will be more dense than between-community sections of the network. In terms of network analysis, the focus of the current investigation has been on the detection of cohesive subgroups that is often the goal of community detection methods. Perhaps not surprisingly, the search for groups has a rich history where the definition of a group or what makes two actors/nodes similar can take on several different meanings. For instance, as mentioned in the paper, the type of distance we focus on results in a parallel to what is usually referred to as structural equivalence (e.g., two nodes have an identical set of connections in the network). However, there are other types of equivalences that can be handled within a generalized blockmodeling framework, with the most popular being "regular equivalence." Regular equivalence itself has been formally defined as two actors are regularly equivalent if they are equally related to equivalent others; thus, they do not have to directly share neighbors, but have neighbors who are themselves similar. Blockmodeling itself has seen several recent developments (Brusco & Steinley, 2006, 2007, 2009, 2011), while Doreian et al. (2005) provide an indepth treatment of blockmodeling and its theoretical foundation.

Additional procedures for finding groups in network data include the method by Steinley, Brusco, and Wasserman's (2011) to clustering exponential random graph models. Further, it is possible to imagine communities to arise from processes different than cohesive subgroups or various types of equivalencies, such as small worlds or even small, contained sets of miniscale free networks. Overall the method based on Cohen's κ proved a viable option for the networks considered, with good community recovery across a variety of network structures, in addition to the benefits of its simplicity and interpretability. Future work is underway to examine the comparative performance of different classes of modeling (e.g., community detection, blockmodeling, and cluster-wise p^* models) when the generative truth varies. We expect the effects to be large, and we further expect that there will be a need to consider the theoretical motivations for searching for groups within a network to help prevent arriving at solutions that are mere by products of algorithms but are otherwise completely ungrounded from reality/possibility.

Article information

Conflict of Interest Disclosures: Each author signed a form for disclosure of potential conflicts of interest. No authors reported any financial or other conflicts of interest in relation to the work described.

Ethical Principles: The authors affirm having followed professional ethical guidelines in preparing this work. These guidelines include obtaining informed consent from human participants, maintaining ethical treatment and respect for the rights of human or animal participants, and ensuring the privacy of participants and their data, such as ensuring that individual participants cannot be identified in reported results or from publicly available original or archival data.

Funding: This work was supported by Grant 1R21AA022074-01 from the National Institute on Alcohol Abuse and Alcoholism.

Role of the Funders/Sponsors: None of the funders or sponsors of this research had any role in the design and conduct of the study; collection, management, analysis, and interpretation of data; preparation, review, or approval of the manuscript; or decision to submit the manuscript for publication.

Acknowledgements: The ideas and opinions expressed herein are those of the authors alone, and endorsement by the authors institutions is not intended and should not be inferred.

References

Alexander-Bloch, A., Lambiotte, R., Roberts, B., Giedd, J., Gogtay, N., & Bullmore, E. (2012). The discovery of population differences in network community structure: new methods and applications to brain functional networks in schizophrenia. *Neuroimage*, 59(4), 3889–3900. DOI: https://doi.org/10.1016/j.neuroimage.2011.11.035.

Arabie, P., & Carroll, J. D. (1980). MAPCLUS: A mathematical programming approach to fitting the ADCLUS model. *Psychometrika*, 45, 211–235. DOI: 10.1007/BF02294077.

Arabie, P., Hubert, L. J., & De Soete, G. (1996). *Clustering and classification*. Singapore: World Scientific. DOI: 10.1142/9789812832153.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. DOI: http://stacks.iop.org/1742-5468/2008/i=10/a=P10008.

Borg, I., & Groenen, P. (1997). Modern multidimensional scaling. *Series in Statistics*. New York, NY: Springer. DOI: 10.1007/0-387-28981-X.

Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., & Wagner, D. (2008). On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2), 172–188. DOI: 10.1109/TKDE.2007.190689.

Brusco, M. J., & Steinley, D. (2006). Inducing a blockmodel structure on two-mode data using seriation procedures. *Journal of Mathematical Psychology*, 50, 468–477. DOI: https://doi.org/10.1016/j.jmp.2006.05.005.

Brusco, M. J., & Steinley, D. (2007). An evaluation of a variable-neighborhood search method for blockmodeling of two-mode binary matrices based on structural equivalence. *Journal of Mathematical Psychology*, 51, 325–338. DOI: https://doi.org/10.1017/nws.2013.5.

Brusco, M. J., & Steinley, D. (2009). Integer programs for one- and two-mode blockmodeling based on prespecified image matrices for structural and regular equivalence. *Journal of Mathematical Psychology*, 53, 577–585. DOI: tps://doi.org/10.1016/j.jmp.2009.08.003.

Brusco, M. J., & Steinley, D. (2011). A tabu search heuristic for deterministic two-mode blockmodeling of binary network matrices. *Psychometrika*, 76, 612–633. DOI: 10.1007/s11336-011-9221-9.

Carroll, J. D., & Corter, J. E. (1995). A graph-theoretic method for organizing overlapping clusters into trees, multiple trees, or extended trees. *Journal of Classification*, 12, 283–313. DOI: 10.1007/BF03040859.

Choukas-Bradley, S., Giletta, M., Neblett, E. W., & Prinstein, M. J. (2015). Ethnic differences in associations among popularity, likability, and trajectories of adolescents' alcohol use and frequency. *Child Development*, 86(2), 519–535. DOI: 10.1111/cdev.12333.

Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, 70(6), 066111. DOI: https://doi.org/10.1103/PhysRevE.70.066111.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46. https://doi.org/10.1177/001316446002000104.

Cormack, R. M. (1971). A review of classification. *Journal of the Royal Statistical Society. Series A (General)*, 134(3), 321–367. DOI: 10.2307/2344237.

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, *1695*(5). Cambridge, United Kingdom.

Doreian, P., Batagelj, V., & Ferligoj, A. (2005). *Generalized blockmodeling:*Vol. 25. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9780511584176.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3), 75–174. DOI: https://doi.org/10.1016/j.physrep.2009.11.002.

Fortunato, S., & Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, 104(1), 36–41. DOI: 10.1073/pnas.0605965104.

Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12), 7821–7826. DOI: 10.1073/pnas.122653799.

Hoffman, M., Steinley, D., & Brusco, M. J. (2015). A note on using the adjusted rand index for link prediction in networks. *Social Networks*, 42, 72–79. DOI: https://doi.org/10.1016/j.socnet.2015.03.002.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1), 193–218. DOI: 10.1007/BF01908075.

Jacob, Y., Winetraub, Y., Raz, G., Ben-Simon, E., Okon-Singer, H., Rosenberg-Katz, K., .. & Ben-Jacob, E. (2016). Dependency Network Analysis (DEPNA) reveals context related influence of brain network nodes. *Scientific Reports*, 6, 27444. DOI: 10.1038/srep27444.

Karelitz, T. M., & Budescu, D. V. (2013). The effect of the raters' marginal distributions on their matched agreement: a rescaling framework for interpreting kappa. *Multivariate Behavioral Research*, 48(6), 923–952. DOI: https://doi.org/10.1080/00273171.2013.830064.

Kindermann, T. A., & Gest, S. D. (2009). Assessment of the peer group: Identifying social networks in natural settings and measuring their influences. In: Rubin, K. H., Bukowski, W., & Laursen, B. *Handbook of peer interactions, relationships, and groups* (Chapter 6). New York: Guilford.

Lancichinetti, A., & Fortunato, S. (2011). Limits of modularity maximization in community detection. *Physical Review E*, 84(6), 066122. DOI: https://doi.org/10.1103/PhysRevE.84.066122.

Milligan, G. W. (1980). An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, 45(3), 325–342. DOI: 10.1007/BF02293907.

Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6), 066133. DOI: https://doi.org/10.1103/PhysRevE.69.066133.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), 036104. DOI: https://doi.org/10.1103/PhysRevE.74.036104.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 026113. DOI: https://doi.org/10.1103/PhysRevE.69.026113.

Pons, P., & Latapy, M. (2005). Computing communities in large networks using random walks. In: Yolum, P., Güngör, T., Gürgen, F., Özturan, C (Eds.), *Computer and Information Sciences-ISCIS 2005* (pp. 284–293). Berlin, Germany: Springer. DOI: 10.1007/11569596_31.

Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), 036106. DOI: https://doi.org/10.1103/PhysRevE.76.036106.

Reichardt, J., & Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74(1), 016110. DOI: https://doi.org/10.1103/PhysRevE.74.016110.

Shireman, E. M., Steinley, D.& Brusco, M. J. (2016). Local optima in mixture modeling. *Multivariate Behavioral Research*, 51, 466–481. DOI: https://doi.org/10.1080/00273171.2016.1160359.

Srivastava, A. (2010). Motif analysis in the amazon product co-purchasing network. *arXiv*:1012.4050.

Steinley, D. (2003). Local optima in *K*-means clustering: What you don't know may hurt you. *Psychological Methods*, 8(3), 294. DOI: https://doi.org/10.1037/1082-989X.8.3.294.

Steinley, D., Brusco, M. J., & Hubert, L. (2016). The variance of the adjusted Rand index. *Psychological methods*, 21(2), 261.

Steinley, D. (2004). Properties of the Hubert–Arable adjusted Rand index. *Psychological Methods*, 9(3), 386. DOI: https://doi.org/10.1037/1082-989X.9.3.386.

Steinley, D. (2006a). K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1), 1–34. DOI: 10.1348/000711005X48266.

Steinley, D. (2006b). Profiling local optima in *K*-means clustering: Developing a diagnostic technique. *Psychological Methods*, 11(2), 178. DOI: https://doi.org/10.1037/1082-989X.11.2.178.

Steinley, D. (2008). Stability analysis in *K*-means clustering. *British Journal of Mathematical and Statistical Psychology*, 61(2), 255–273. DOI: 10.1348/000711007X184849.

Steinley, D., & Brusco, M. J. (2007). Initializing *K*-means batch clustering: A critical evaluation of several techniques. *Journal of Classification*, 24(1), 99–121. DOI: 10.1007/s00357-007-0003-0.

Steinley, D., & Brusco, M. J. (2011). Evaluating mixture modeling for clustering: Recommendations and cautions. *Psychological Methods*, 16, 63–79. DOI: https://doi.org/10.1037/a0022673.

Steinley, D., Brusco, M. J., & Wasserman, S. (2011). Clusterwise p^* models for network data. *Statistical Analysis and Data Mining*, 4, 487–496. DOI: 10.1002/sam.10139.

Steinley, D., Hendrickson, G., & Brusco, M. J. (2015). A note on maximizing the agreement between partitions: A stepwise optimal algorithm and some properties. *Journal of Classification*, 32(1), 114–126. DOI: 10.1007/s00357-015-9169-z.

van Borkulo, C. D., Borsboom, D., Epskamp, S., Blanken, T. F., Boschloo, L., Schoevers, R. A., & Waldorp, L. J. (2014). A new method for constructing networks from binary data. *Scientific Reports*, 4, 5918. DOI: 10.1038/srep05918.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301), 236–244. DOI: 10.1080/01621459.1963.10500845.

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications:*Vol. 8. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9780511815478.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4), 452–473. DOI: 10.1086/jar.33.4.3629752.