COMPARING COMMUNITY DETECTION ALGORITHMS IN PSYCHOLOGICAL DATA: A MONTE CARLO SIMULATION

PSYARXIV PREPRINT

Alexander P. Christensen
Department of Psychology and Human Development
Data Science Institute
Vanderbilt University
Nashville, TN, 37203
alexpaulchristensen@gmail.com

Luis Eduardo Garrido
Department of Psychology
Pontificia Universidad Católica Madre y Maestra
Santiago de Los Caballeros, Dominican Republic, 51000
luisgarrido@pucmm.edu.do

Kiero Guerra-Peña
Department of Psychology
Pontificia Universidad Católica Madre y Maestra
Santiago de Los Caballeros, Dominican Republic, 51000
kpena@pucmm.edu.do

Hudson Golino
Department of Psychology
University of Virginia
Charlottesville, VA, 22903
hfg9s@virginia.edu

ABSTRACT

Identifying the correct number of factors in multivariate data is fundamental to psychological measurement. Factor analysis has a long tradition in the field but it's been challenged recently by exploratory graph analysis (EGA), an approach based on network psychometrics. EGA first estimates a network and then applies the Walktrap community detection algorithm. Simulation studies have demonstrated that EGA has comparable or better accuracy for recovering the same number of communities as there are factors in the simulated data than factor analytic methods. Despite EGA's effectiveness, there has yet to be an investigation into whether other sparsity induction methods or community detection algorithms could achieve equivalent or better performance. Furthermore, unidimensional structures are fundamental to psychological measurement yet they have been sparsely studied in simulations using community detection algorithms. In the present study, we performed a Monte Carlo simulation using the zero-order correlation matrix, GLASSO, and two variants of a non-regularized partial correlation sparsity induction methods with several community detection algorithms. We examined the performance of these method-algorithm combinations in both continuous and polytomous data across a variety of conditions. The results indicate that the Fast-greedy, Louvain, and Walktrap algorithms paired with the GLASSO method were consistently among the most accurate and least biased overall.

***K*eywords** community detection · latent factors · exploratory graph analysis · network psychometrics

1 Introduction

Many psychological phenomena are complex systems of related variables. These variables often form distinct sets of variables where there are many relations between some variables relative to others. In network analysis, these sets of connected variables are referred to as *communities* (Fortunato, 2010). These communities are statistically consistent with latent factors in factor analysis when data are generated from a factor model (Golino & Epskamp, 2017; Golino, Shi, et al., 2020). Identifying latent factors in psychological data such as questionnaires, symptom checklists, and ability tests is important for theory, scale construction and validation, and statistical analyses in psychology (Cattell, 1978).

Over the last century, there have been several factor analytic methods developed to estimate the number of factors in psychological data. Principal component analysis (PCA) estimates the number of components in the data, which are used as a proxy for the number of factors in the data. Within the factor analytic framework, there exists a plethora of variants that aim to accurately recover the number of factors such as parallel analysis (Horn, 1965), the K1 rule (eigenvalues greater than 1; Kaiser, 1960), and the scree test (Cattell, 1966).

More recently, a method called exploratory graph analysis (EGA; Golino & Epskamp, 2017; Golino, Shi, et al., 2020) was introduced to the literature. EGA uses a network modeling framework, often referred to as *network psychometrics* (Epskamp, Maris, Waldrop, & Borsboom, 2018), to first estimate a network and then communities (Golino & Epskamp, 2017; Golino, Shi, et al., 2020). The default EGA algorithm first applies the graphical least absolute shrinkage and selection operator (GLASSO; Friedman, Hastie, & Tibshirani, 2008; Friedman, Hastie, & Tibshirani, 2014) to the inverse covariance matrix to estimate a Gaussian graphical model (GGM; Lauritzen, 1996) where edges (lines) represent (partial) correlations between nodes (circles), which represent variables in the network.

After, EGA applies the Walktrap community detection algorithm (Pons & Latapy, 2006), which uses random walks to determine the number and content of the communities in the network. Simulation studies have demonstrated that EGA recovers the number of simulated factors as accurately as or more accurate than most factor analytic approaches (e.g., parallel analysis; Golino & Epskamp, 2017; Golino, Shi, et al., 2020). A recent addition to the algorithm provides a so-called "unidimensionality adjustment" to enable the Walktrap algorithm to detect unidimensional structures. Golino and colleagues (2020) identified that the modularity statistic, which is used to select communities in many community detection algorithms (including the Walktrap), is biased against selecting a single community (Newman, 2006). This unidimensionality adjustment, which adds an orthogonal community to the network, takes advantage of this bias and allows the Walktrap algorithm to recover single community structures.

Despite the effectiveness of EGA to recover the number of simulated factors, there are many sparsity induction methods that can induce *sparsity* or statistical approaches that reduce the number of parameters in the network (e.g., edges; Epskamp, Kruis, & Marsman, 2017). Some sparsity induction methods may be as effective or more effective than the present standard of the GLASSO (e.g., Isvoranu & Epskamp, 2021). Similarly, there are several community detection algorithms that may be as effective or more effective than the present standard of the Walktrap (e.g., Brusco, Steinley, & Watts, 2022). Finally, the unidimensionality adjustment proposed by Golino, Shi, et al. (2020) warrants a more extensive investigation to evaluate limitations of the approach. Using a Monte Carlo simulation, our study evaluates each component of EGA's algorithm to determine whether the present standards are optimal.

1.1 Alternative Sparsity Induction Methods

To date, the GLASSO has been the standard sparsity induction method applied in psychological network studies (Epskamp & Fried, 2018). There are many other sparsity induction methods available. Each method will provide a different network structure which affects the number and content of communities detected by a community detection algorithm. One simulation study examined continuous and dichotomous data and compared the accuracy of the GLASSO and triangulated maximally filtered graph (TMFG; Massara, Di Matteo, & Aste, 2016) sparsity induction methods combined with the Walktrap community detection algorithm (Golino, Shi, et al., 2020). This study found that the GLASSO method had better accuracy and less bias than the TMFG but both performed comparably well to the best factor analytic techniques (i.e., parallel analysis).

Other sparsity induction methods used in psychometric networks such as non-regularized GGMs (Williams & Rast, 2018; Williams, Rhemtulla, Wysocki, & Rast, 2019) and Bayesian GGMs (Williams, 2021; Williams & Mulder, 2020) have been put forward in the literature. The non-regularized methods in particular are shown to better recover the edges in dense (highly connected) simulated networks, which are thought to be common in psychology (Williams & Rast, 2018; Wysocki & Rhemtulla, 2021). Because of their better edge recovery in simulated networks, this study investigated whether they would also perform better when recovering the number of simulated factors when data is generated by a factor model.

1.2 Alternative Community Detection Algorithms

There have been many simulation studies comparing different community detection algorithms (e.g., Brusco et al., 2022; Gates, Henry, Steinley, & Fair, 2016; Yang, Algesheimer, & Tessone, 2016) but none, to our knowledge, have generated data from latent factor models with relatively few total variables (e.g., < 50), few variables per factor (e.g., $\leq$ 12), and unidimensional structures. In general, most community detection algorithms were developed and validated on networks containing a large number of nodes (e.g., > 1,000; Lancichinetti & Fortunato, 2009; Yang et al., 2016). Some studies have demonstrated that these algorithms may only work well for one type of problem or data structure (Gates et al., 2016). Gates and colleagues (2016), for example, examined several community detection algorithms in structures that are common in brain network data. In their study, they generated network structures using factor models and manipulated several conditions that included the number of nodes (ranging from 25 to 1000) and communities (ranging from 1 to 20), size of edge weights (i.e., correlation between variables), and overlap between communities (i.e., cross-loadings and correlations between factors). Their study examined how community detection algorithms, when applied to correlation and euclidean distance matrices of continuous data, performed with respect to accuracy in recovering the proper placement of nodes in the communities. Of the six algorithms they examined, the Walktrap and Louvain (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008) algorithms performed the best across conditions. Importantly, their study investigated conditions where there were a small number of nodes (i.e., 25 and 75) and some correlations between factors (0.10).

Because most networks in the psychometric network literature consist of fewer than 100 nodes (89% between 0-30 nodes; Wysocki & Rhemtulla, 2021), there is a need to verify which of these algorithms work best in conditions that are more commonly found in questionnaires, symptom checklists, and ability tests such as dichotomous and polytomous data (Golino, Shi, et al., 2020). These data also tend to follow latent factor model conventions (i.e., simple structure with small cross-loadings) because of factor analysis's rich psychometric tradition in psychology. Although Gates and colleagues (2016) evaluated some conditions that might traditionally reflect psychological data (25 nodes and 2-5 communities), these conditions were not the focus of their study and consequently the generalizability of their results is limited. Moreover, their conditions were not presented or manipulated in psychometric terms that most psychological researchers are familiar with (e.g., overlap or degree to which communities are related in their connectivity vs. cross-loadings and correlation between factors, connectivity range vs. sample size and loading size, heterogeneity of community size vs. number of variables per factor). An evaluation of these algorithms across many common factor analytic conditions as well as the presentation of their results in factor analytic terms would provide more fruitful usage guidelines for applied researchers in psychology.

1.3 Unidimensionality and Community Detection Algorithms

Unidimensionality is fundamental to psychological measurement yet, to our knowledge, there have been only two simulation studies that have evaluated the performance of community detection algorithms with unidimensional structures (Gates et al., 2016; Golino, Shi, et al., 2020). Gates et al. (2016) evaluated unidimensional structures in networks with 1000 nodes. They found that the Infomap (Rosvall & Bergstrom, 2008) and Label Propagation (Raghavan, Albert, & Kumara, 2007) algorithms had near perfect accuracy in the unidimensional conditions, while Fast-greedy (Clauset, Newman, & Moore, 2004), Leading Eigenvalue (Newman, 2006), and Louvain (Blondel et al., 2008) had near zero accuracy. In these conditions, the Walktrap algorithm had variable accuracy when using correlation matrices and had near zero accuracy when using Euclidean distance matrices.

Golino, Shi, et al. (2020) evaluated unidimensional structures across a variety of conditions including sample size, factor loadings, number of variables, and data types (continuous and dichotomous). Their simulation only evaluated the Walktrap algorithm and applied sparsity induction methods, GLASSO and TMFG, and did not evaluate the full zero-order correlation matrix as in Gates and colleagues' (2016) simulation. Further, they explained that the Walktrap algorithm is not expected to work well with unidimensional structures because the modularity measure that is used to select the best partition penalizes unidimensional solutions (Newman, 2006). Consistent with this notion, the algorithms that used modularity, in one way or another, performed poorly in the unidimensional condition in Gates et al.'s (2016) study.

To circumvent this issue, Golino and colleagues (2020) proposed generating four variables that load strongly (.70) on a single factor that is orthogonal to the empirical data. EGA is then be applied to the combined simulated and empirical data. If two (or fewer) communities are returned (one community representing the simulated data and the other representing the empirical data), then the data is suggested to be unidimensional; otherwise, EGA is applied to the empirical data only. This so-called "unidimensional adjustment" appeared to work well and was on par with the best performing factor analytic methods (e.g., parallel analysis with PCA). Given the importance of unidimensional structures in psychology (e.g., Slocum-Gori & Zumbo, 2011), it's necessary to further investigate the effectiveness of this approach and whether it is appropriate for all community detection algorithms.

2 Simulation Aims

The present simulation had three aims: determine whether an alternative network method, such as non-regularized GGMs or full zero-order correlations, perform better than the standard EGA method of applying the GLASSO. We focus on non-regularized GGMs because of their relatively superior performance in simulations looking at the sensitivity and specificity of edge detection (Williams & Rast, 2018; Williams et al., 2019). The second aim was to examine several freely available algorithms that were used in Gates et al.'s (2016) simulation study as well as other algorithms included in the popular software package {igraph} (version 1.2.6; Csardi & Nepusz, 2006) in R (version 4.1.0; R Core Team, 2022). The third aim was to determine whether the unidimensionality adjustment applied in Golino, Shi, et al. (2020) improves the accuracy of some or all community detection algorithms.

Our simulation study differed from previous studies that have compared these sparsity induction methods and community detection algorithms in a couple ways. First, this study specifically analyzed the accurate recovery of the number of factors when data are generated from a factor model rather than whether the edges in simulated networks were recovered (Williams et al., 2019). Second, the present simulation generates data that aligns with conditions more commonly found in psychological data (i.e., questionnaires, symptom checklists, and ability tests); specifically, multivariate data that is polytomous with skew, has a relatively few variables per factor (e.g., 4, 8, and 12), and different magnitudes of correlations between factors (e.g., .00, .30, .50, and .70). To date, EGA and other comparisons of sparsity induction methods and community detection algorithms have largely focused on continuous or dichotomous data (Gates et al., 2016; Golino & Epskamp, 2017; Golino, Shi, et al., 2020; Hoffman, Steinley, Gates, Prinstein, & Brusco, 2018; Lancichinetti & Fortunato, 2009; Yang et al., 2016). Our study is the first, to our knowledge, to compare the performance of sparsity induction methods and community detection algorithms in polytomous data with skew. Finally, we provide an empirical example to demonstrate how these method-algorithm combinations perform on real data.

3 Methods

3.1 Simulation Design

The population models were simulated from a multivariate normal distribution with factor loadings randomly drawn from a uniform distribution. These factor loadings were manipulated to be small (0.30-0.50), moderate (0.45-0.65), and large (0.60-0.80). For brevity, the midpoints are referred to as small (0.40), moderate (0.55), and large (0.70). Cross-loadings were randomly drawn from a normal distribution with a mean of zero and a standard deviation of 0.10. The correlation between factors of orthogonal (0.00), small (0.30), moderate (0.50), and large (0.70) as well as sample size of small (250), moderate (500), large (1000), and very large (5000) were also manipulated. The number of factors—one, two, and four—were simulated to provide unidimensional and multidimensional structures that are commonly found in the psychological literature (Henson & Roberts, 2006). There were four, eight, and twelve variables per factor, which represented conditions common in scale development and validation.

The simulation design of the current study allowed for a mixed factorial design: $2 \times 4 \times 4 \times 3 \times 3 \times 3 \times 2$ (unidimensional adjustment × factor correlations × sample size × number of factors × number of variables × factor loadings × data type). Because factor correlations are not applicable to unidimensional structures, there were 1,728 total condition combinations, but only 1,296 of those were unique (when structures were unidimensional, then all conditions were replicated four times because of the correlation between factors). The oversampling of the unidimensional conditions were to evaluate the unidimensional adjustment of Golino, Shi, et al. (2020) more thoroughly. Each method-algorithm was applied to all condition combinations. There were 100 samples generated for all condition combinations.

3.2 Data Generation

We generated data from multivariate normal factor models following the same approach as Golino, Shi, et al. (2020). First, the reproduced population correlation matrix was computed:

$$\mathbf{R_R} = \mathbf{\Omega \Phi \Omega}',$$

where $\mathbf{R_R}$ is the reproduced population correlation matrix, $\mathbf{\Omega}$ is the k (variables) × r (factors) factor loading matrix,[1] and $\mathbf{\Phi}$ is the $r \times r$ correlation matrix. The population correlation matrix, $\mathbf{R_P}$, was then obtained by putting the unities on the diagonal of $\mathbf{R_R}$. Next, Cholesky decomposition was performed on the correlation matrix such that:

$$\mathbf{R_P} = \mathbf{U}'\mathbf{U}.$$

If the population correlation matrix was not positive definite (i.e., at least one eigenvalue ≤ 0) or any single item's communality was greater than 0.90, then $\mathbf{\Omega}$ was re-generated and the same procedure was followed until these criteria are met. Finally, the sample data matrix of continuous variables was computed:

$$\mathbf{X} = \mathbf{ZU},$$

[1] The traditional notation for factor loadings matrix is $\mathbf{\Lambda}$; however, to avoid confusion with the GLASSO's lambda parameter (λ), we use omega ($\mathbf{\Omega}$).

where $\mathbf{Z}$ is a matrix of random multivariate normal data with rows equal to the sample size and columns equal to the number of variables.

To generate polytomous data, each continuous variable was categorized into five categories, resembling a 5-point Likert scale, with a random skew ranging from -2 to 2 on a 0.5 interval from a random uniform distribution following the approach of Garrido, Abad, and Ponsoda (2011; 2013). A table containing the threshold values to categorize the continuous data at each skew value is provided below (Table 1).

Table 1: Skew Table for Polytomous Data

	Skew Value								
Threshold ($<$ | $\geq$)	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2
1 | 2	-1.77	-1.62	-1.44	-1.16	-0.84	-0.34	0.05	0.41	0.68
2 | 3	-1.34	-1.16	-0.94	-0.63	-0.25	0.16	0.51	0.78	1.00
3 | 4	-1.00	-0.78	-0.51	-0.16	0.25	0.63	0.94	1.16	1.34
4 | 5	-0.68	-0.41	-0.05	0.34	0.84	1.16	1.44	1.62	1.77

Note. Values are rounded to the nearest hundredths place. Thresholds on the left are less than ($<$ |) the value and thresholds on the right are greater than or equal to (| $\geq$) the value.

3.3 Modeling Methods

3.3.1 GLASSO Sparsity Induction

The GLASSO uses the least absolute shrinkage and selection operator (LASSO; Tibshirani, 1996), which is a regularization technique that reduces parameter estimates, with some estimates becoming exactly zero (for the mathematical notation, see Epskamp & Fried, 2018). The aim of this technique is to achieve a sparse network model—non-relevant edges are removed from the model, leaving only a subset of relevant (not necessarily significant) edges.

This sparsity is controlled by a parameter called *lambda* (λ). Lower values of λ remove fewer edges, increasing the possibility of including spurious associations, while larger values of λ remove more edges, increasing the possibility of removing relevant edges. When $\lambda = 0$, then the estimates are equal to the ordinary least squares solution (i.e., the partial correlation matrix). This parameter is an important part of model selection, striking a balance between sensitivity (i.e., selecting relevant edges that are truly relevant) and specificity (i.e., removing edges that are truly not relevant).

The popular approach in the network psychometrics literature is to compute models across several values of λ (usually 100) and to select the model that minimizes the extended Bayesian information criterion (EBIC; Chen & Chen, 2008; Epskamp & Fried, 2018). The EBIC model selection uses a hyperparameter (γ) to control how much it prefers simpler models (i.e., models with fewer edges; Foygel & Drton, 2010). Larger γ values lead to simpler models, while smaller γ values lead to denser models. When $\gamma = 0$, the EBIC is equal to the Bayesian information criterion. In the psychometric network literature, this approach has been termed EBICglasso and is applied via the {qgraph} package (version 1.6.9; Epskamp, Cramer, Waldorp, Schmittmann, & Borsboom, 2012) in R. For continuous data, Pearson's correlations were computed; for polytomous data, polychoric correlations were computed (but see Isvoranu & Epskamp, 2021 for criticisms).

Following the EGA approach (Golino, Shi, et al., 2020), the minimum λ value is set to 0.10, which is slightly larger than the default of 0.01 in {qgraph} (Epskamp et al., 2012). This larger value is selected to reduce the prevalence of false positive edges in the network. Next, the γ value is set to 0.50, which is the default; however, it is iteratively decreased by 0.25 when there is at least one node in the network that is disconnected. If γ reaches 0.00, then the network is used regardless of whether any nodes are disconnected.

3.3.2 Non-regularized Sparsity Induction

Two variants of a non-regularized partial correlation sparsity induction method were used. Both methods were based on a regression strategy called *neighborhood selection*, which uses node-wise multiple regression on each node in the network (Guttman, 1953; Williams et al., 2019). Multiple regression coefficients have a direct correspondence to the inverse covariance coefficients in that the negative regression coefficient ($-\beta_{ij}$) divided by the predictor variable's variance (σ_j^2) is equal to the inverse covariance between the regressed variable and the predictor variable given all other variables (θ_{ij}).

The neighborhood selection method used by Williams et al. (2019) places the multiple regression coefficients for each regressed variable across the row of each target variable with the regressed variable's variance in its respective element's position (θ_{ii}^2; i.e., variance of each variable is on the diagonal). A common method for computing partial correlations is to take the square root of the product between the absolute values of the regression coefficients (i.e., elements) in the matrix and replace their signs (i.e., $\rho_{ij} = sign(\beta_{ij})\sqrt{|\beta_{ij}| \times |\beta_{ji}|}, i \neq j$). This leads to an asymmetric covariance matrix where coefficients do not correspond to their respective transpose element (i.e., $\theta_{ij}^2 \neq \theta_{ji}^2$).

There are two approaches for determining whether an edge should be non-zero: the "and-rule" where both β_{ij} and β_{ji} must be non-zero and the "or-rule" where only one coefficient must be non-zero. Both approaches use a forward search strategy for determining non-zero coefficients, which removes predictor variables from each multiple regression that minimize some criterion until the minimum value of the criterion is achieved for the set of predictor variables. The coefficients that are not removed in the process of minimizing the criterion are retained in the network as non-zero edges, while the removed variables are set to zero. The final networks have zeros on the diagonal.

The criterion used for determining included and excluded coefficients is based on traditional model selection criteria AIC and BIC. The main difference between these criteria is that the BIC tends to penalize more complex models more severely than the AIC. For this study, we examined both the AIC and BIC approaches to edge selection with the "and-rule" because they were shown to have considerable improvements (relative to the "or-rule") in recovering the true edges in the population network structure in previous simulations (Williams et al., 2019). Both non-regularized partial correlation network models were estimated using the {GGMnonreg} package (version 1.0.0; Williams, 2019) in R.

3.3.3 Non-sparsity Comparison

For the non-sparsity comparison method, we applied the community detection algorithms to the zero-order correlation matrix, which is a common approach used in the literature (Gates et al., 2016). Because the zero-order correlation matrix does not induce sparsity and community detection algorithms can be applied directly to them, it serves as a valuable benchmark for whether (1) the unidimensionality adjustment is necessary for certain algorithms and (2) sparsity induction methods improve or inhibit the performance of community recovery in data generated from factor models.

3.3.4 Parallel Analysis Comparison

We also applied two parallel analysis algorithms—PAF and PCA. These two algorithms were chosen because they have been extensively evaluated in the literature (e.g., Garrido et al., 2013) and have shown comparable performance with EGA in a previous simulation study (Golino, Shi, et al., 2020). In short, parallel analysis generates a large number of replicate datasets by randomly resampling values from each variable from the original dataset (Horn, 1965). The number of factors (PAF) or components (PCA) whose eigenvalues in the original dataset are greater than the mean of the resampled datasets is suggested as the factor solution. The PAF models were estimated using the minimum residual estimator.

3.4 Community Detection Algorithms

This study focused on eight different community detection algorithms that are available in the {igraph} package. These algorithms are Walktrap (Pons & Latapy, 2006), Infomap (Rosvall & Bergstrom, 2008), Fast-greedy (Clauset et al., 2004), Louvain (Blondel et al., 2008), Leading Eigenvalue (Newman, 2006), Label Propagation (Raghavan et al., 2007), Spinglass (Reichardt & Bornholdt, 2006), and Edge Betweenness (Girvan & Newman, 2002).

All community detection algorithms were implemented with their default arguments in order to evaluate their baseline, "out-of-the-box" performance without researcher direction (similar to Gates et al., 2016). Moreover, all network matrices were input with absolute values to avoid the bias of some methods performing better than others because of their ability to handle negative associations. In practice, absolute values would only be used for community detection and subsequent statistical measures would use the signed values of the network. Below, we briefly describe *modularity*, a metric used to quantify the quality of community partitions, and then each algorithm (more detailed information can be found within their respective citations).

3.4.1 Modularity

A key definition for understanding many community detection algorithms is the concept of modularity (Newman, 2006). Modularity can be expressed as (Fan, Li, Zhang, Wu, & Di, 2007):

$$Q = \frac{1}{2w} \sum_{ij} (w_{ij} - \frac{w_i w_j}{2w})\ \delta(c_i, c_j),$$

where w_{ij} is the edge strength or (partial) correlation for a given node pair, and w_i and w_j are the node strength for node i and node j (respectively), w is the sum of all the edge weights in the network, c_i and c_j represents the community that node i and node j belong to (respectively), and δ is 1 if the nodes belong to the same community (i.e., $c_i = c_j$) and 0 if otherwise. Higher values of modularity reflect communities that have more connections within the community and fewer connections with other communities. Notably, modularity is zero when structures are unidimensional, making any algorithms optimized by modularity less likely to return one community (Golino, Shi, et al., 2020).

3.4.2 Walktrap

The Walktrap algorithm (Pons & Latapy, 2006) has been the most commonly applied algorithm in the psychometric network literature as the default of EGA (Golino & Epskamp, 2017; Golino, Shi, et al., 2020). The Walktrap algorithm begins by computing a transition matrix where each element represents the probability (based on node strength) of one node traversing to another. Random walks are then initiated for a certain number of steps (e.g., 4), using the transition matrix for probable destinations. Using Ward's agglomerative clustering approach (Ward, 1963), each node starts as its own community and merges with adjacent communities (based on squared distances between each community) in a way that minimizes the sum of squared distances between other communities. Modularity is then used to determine the optimal partition of communities.

3.4.3 Infomap

Similar to the Walktrap algorithm, the Infomap algorithm (Rosvall & Bergstrom, 2008) uses random walks. Different from the Walktrap algorithm, Infomap is derived from information theory with idea of "compressing" the conditional information of a random walk on the network into Huffman codes (a binary naming system; Rosvall & Bergstrom, 2008). The major difference between these two algorithms is that Infomap captures the conditional flow of information across the network in a way that maximizes the information (e.g., bits) of the random walk process. The partition function that optimizes this minimization is given by the entropy of movement between and within communities. The space of possible partitions is explored using a deterministic greedy search algorithm, which is refined using a simulated annealing approach.

3.4.4 Fast-greedy

The Fast-greedy algorithm (Clauset et al., 2004) uses modularity to identify optimal partitions in the network. Like the Walktrap algorithm, the Fast-greedy algorithm begins with each node considered as its own community and follows a hierarchical clustering algorithm. The algorithm then proceeds by iteratively combining neighboring communities in a greedy way: Each node is moved into a community that maximizes the modularity function. These aggregate communities are then merged until the modularity function can no longer be increased.

3.4.5 Louvain

The Louvain algorithm (also referred to as Multi-level; Blondel et al., 2008) is similar to the Fast-greedy algorithm in that it iteratively uses modularity to optimize its partitions. It differs in that its motivation is to identify hierarchical structures in large networks. Specifically it iteratively exchanges nodes between communities and evaluates the change in modularity until it no longer improves. Then, the algorithm collapses the communities into latent nodes and identifies edge weights with other observed and latent nodes, which provides the "multi-level" structure (Gates et al., 2016). In its use in this study, the algorithm was not used to identify hierarchical community structures in the network. Therefore, it's expected that this algorithm will closely align with the Fast-greedy algorithm. The final solution depends on the node order making the algorithm non-deterministic. In this study, only one run was implemented for each sample in order to evaluate its accuracy in its current form. Other strategies such as bootstrapping (Christensen & Golino, 2021) or repeated algorithm applications to the same network (De Beurs et al., 2019; Lancichinetti & Fortunato, 2012) could be used to arrive at a relatively stable organization of communities.

3.4.6 Leading Eigenvalue

The Leading Eigenvalue algorithm (Newman, 2006) is based on the spectral properties of the network using the eigenvector of the first eigenvalue of the modularity matrix to determine optimal community structures. Like Fast-greedy and Louvain algorithms, the Leading Eigenvalue algorithm uses modularity to optimize these structures; however, the first eigenvector of the modularity matrix is used, splitting the network into two communities with the aim of improving modularity. This process iteratively unfolds until there is no longer improvement in modularity.

3.4.7 Label Propagation

The Label Propagation algorithm (Raghavan et al., 2007) begins by assigning each node a unique label. Each node then adopts the same label that the majority of its neighbors have with ties being broken randomly. This process continues iteratively until each node has the same label as the majority of its neighbors. The general notion of the algorithm is that a consensus will develop among the nodes in the network. Like the Louvain algorithm, this algorithm is not deterministic and produces different results with each run. Only one run was implemented in this study.

3.4.8 Spinglass

The Spinglass algorithm comes from statistical physics and is based on the Potts model with the notion that "the problem of community detection can be mapped onto finding the ground state of an infinite ranged Potts spin glass" (Reichardt & Bornholdt, 2006, p. 1540). In essence, edges should connect nodes that are in the same spin state (i.e., community), while nodes in different states should be disconnected, which results in a "lower energy state" or ground state of the system. The model is simulated for some number of steps (e.g., 25) and the spin states in the end define the communities. This algorithm is not deterministic and only one run was implemented in this study.

3.4.9 Edge Betweenness

The Edge Betweenness algorithm (Girvan & Newman, 2002) was one of the first algorithms used to identify communities in networks. This algorithm finds edges that are frequently "between" other nodes in the network known as edge betweenness (based on the betweenness centrality; Freeman, 1977). Edge betweenness is calculated for the entire network and the edge with the highest betweenness value is removed. All edges that are affected by this removal have their edge betweenness value recalculated. This process repeats iteratively until no edges remain, making this algorithm substantially slower than the other algorithms. Modularity is used to determine the optimal cut-off.

3.5 Unidimensionality Adjustment

We used Golino et al.'s (2020) unidimensionality adjustment for all algorithms to determine if it was necessary to improve their performance on unidimensional structures. We expected that this adjustment would be necessary for algorithms that use modularity to optimize their partitions. For computational efficiency and statistical precision, we adapted this approach to "expand" the empirical correlation matrix to add four variables that are correlated 0.50 with each other (roughly equivalent to factor loadings of .70) and 0.00 with all other variables (i.e., empirical variables). This approach avoids adding noise in the form of spurious correlations between generated and empirical variables (i.e., they are exactly orthogonal) that would otherwise be present if data itself were generated.

3.6 Statistical Analyses

To evaluate the performance of the methods and algorithms, accuracy and bias were computed using the percentage of correct number of factors (PC), mean absolute error (MAE; the average absolute deviation away from the correct number of factors), and mean bias error (MBE; the average deviation away from the correct number of factors). These are defined below:

$$PC = \frac{\sum C}{N},\ for\ C = \begin{cases} 1 \text{ if } \hat{\theta} = \theta \\ 0 \text{ if } \hat{\theta} \neq \theta \end{cases},$$

$$MAE = \frac{\sum |\hat{\theta} - \theta|}{N},$$

$$MBE = \frac{\sum (\hat{\theta} - \theta)}{N},$$

where $\hat{\theta}$ is the estimated number of factors or communities, θ is the population number of factors, and N is the number of sample data matrices simulated.

A second approach was used to quantify the accuracy of the item placement of the community detection algorithms, specifically, whether the items were being identified in the correct factor. The number of communities (representing factors), for example, could be recovered correctly; however, some communities may have items that belong to a different community than the simulated factor structure. Each item in the simulated factor structure was assigned to only one factor despite the existence of cross-loadings, which were always smaller than their assigned factor's loading.

A common metric to compare communities is the Hubert-Arabie Adjusted Rand Index (Gates et al., 2016; Hoffman et al., 2018; Hubert & Arabie, 1985; Steinley, Brusco, & Hubert, 2016). The benefit of the Adjusted Rand Index (ARI) is that it ignores the labeling of item placement and is invariant to labeling permutations. The ARI therefore is a metric of item placement precision. ARI is defined below:

$$ARI = \frac{\binom{N}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{N}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]}$$

where a is the count of items placed into the same population and estimated factor and d is the count of items that are in different communities of both the population and estimated factors. Both b and c count the wrong placement of nodes between the population factor and estimated factor, respectively. ARI was not computed for unidimensional structures.

To evaluate whether method-algorithm pairs were affected by certain conditions, we computed analysis of variances (ANOVAs) for each method-algorithm combination across conditions. For the multidimensional conditions (2 and 4 number of factors), we used a fully factorial design to allow for all possible interactions between conditions. For the unidimensional conditions, we used the number of factors of 1 and 2. The purpose of this approach was to fully evaluate the trade-off of the unidimensionality adjustment. Similarly, for the unidimensional conditions, we used a fully factorial design to evaluate possible differences based on correlation between factors in bidimensional structures (two factor conditions).

Partial eta squared (η_p^2) was used for effect size. We report only large effect sizes ($\eta_p^2 = 0.14$) according to Cohen's (1992) guidelines. For full effect size results, see our Open Science Framework (OSF). All analyses were performed in R. The data, R code, and scripts can be found on the Open Science Framework: `https://osf.io/jfxad/?view_only=5ac0b62ba88543ef89c7d9f5a0679fa6`.

4 Results

4.1 Accuracy and Bias

The performance for each method-algorithm combination was separated by whether there was an adjustment for unidimensionality. For brevity, we focus first on reporting the best performing method-algorithm combinations within each method for the accuracy and bias results of both continuous and polytomous data. We then move on to discussing the ANOVAs of the best method-algorithm combinations in the multidimensional conditions. Afterward, we focus on the unidimensional conditions. To aid interpretation of the results for all possible method-algorithm and condition combinations, an interactive {Shiny} application (version 1.7.4; Chang et al., 2022) in R was created: `https://masked.shinyapps.io/community_detection_results/`.

4.1.1 Overall

The five most accurate method-algorithm combinations were: parallel analysis and PCA (89.8%), GLASSO and Louvain with unidimensionality adjustment (88.7%), GLASSO and Fast-greedy with unidimensionality adjustment (88.2%), GLASSO and Walktrap with unidimensionality adjustment (87.3%), and parallel analysis and PAF (87.1%). For the least absolute bias, the top five were: parallel analysis and PCA (0.16), GLASSO and Louvain with unidimensionality adjustment (0.16), correlation and Louvain without adjustment (0.17), GLASSO and Fast-greedy with unidimensionality adjustment, and parallel analysis and PAF (0.18). In terms of direction of bias (under factoring = negative; over factoring = positive), the best performers were GLASSO and Fast-greedy with unidimensionality adjustment (0.05), GLASSO and Louvain with unidimensionality adjustment (0.05), correlation and Spinglass with no adjustment (0.05), GLASSO and Walktrap with unidimensionality adjustment (0.06), and correlation with Leading Eigenvalue without adjustment (-0.06).

4.1.2 Continuous Data

The best performing combinations for percent correct were GLASSO with unidimensional adjustment and the Louvain (94.1%), Fast-greedy (93.7%), and Walktrap (91.7%) algorithms followed by parallel analysis with PCA (91.6%) and

PAF (89.4%; Figure 1). Breaking down results for the other three methods, correlation matrices with (81.9%) and without adjustment (88.4%) for the Louvain algorithm as well as the Spinglass with no adjustment (82.3%) were top performing. For AIC, the Walktrap (81.3%), Fast-greedy (81.1%), and Louvain (81.1%) algorithms with unidimensional adjustment were top performing. For BIC, the Louvain (68.3%), Fast-greedy (68.3%), and Walktrap (67.6%) algorithms with unidimensional adjustment were top performing.

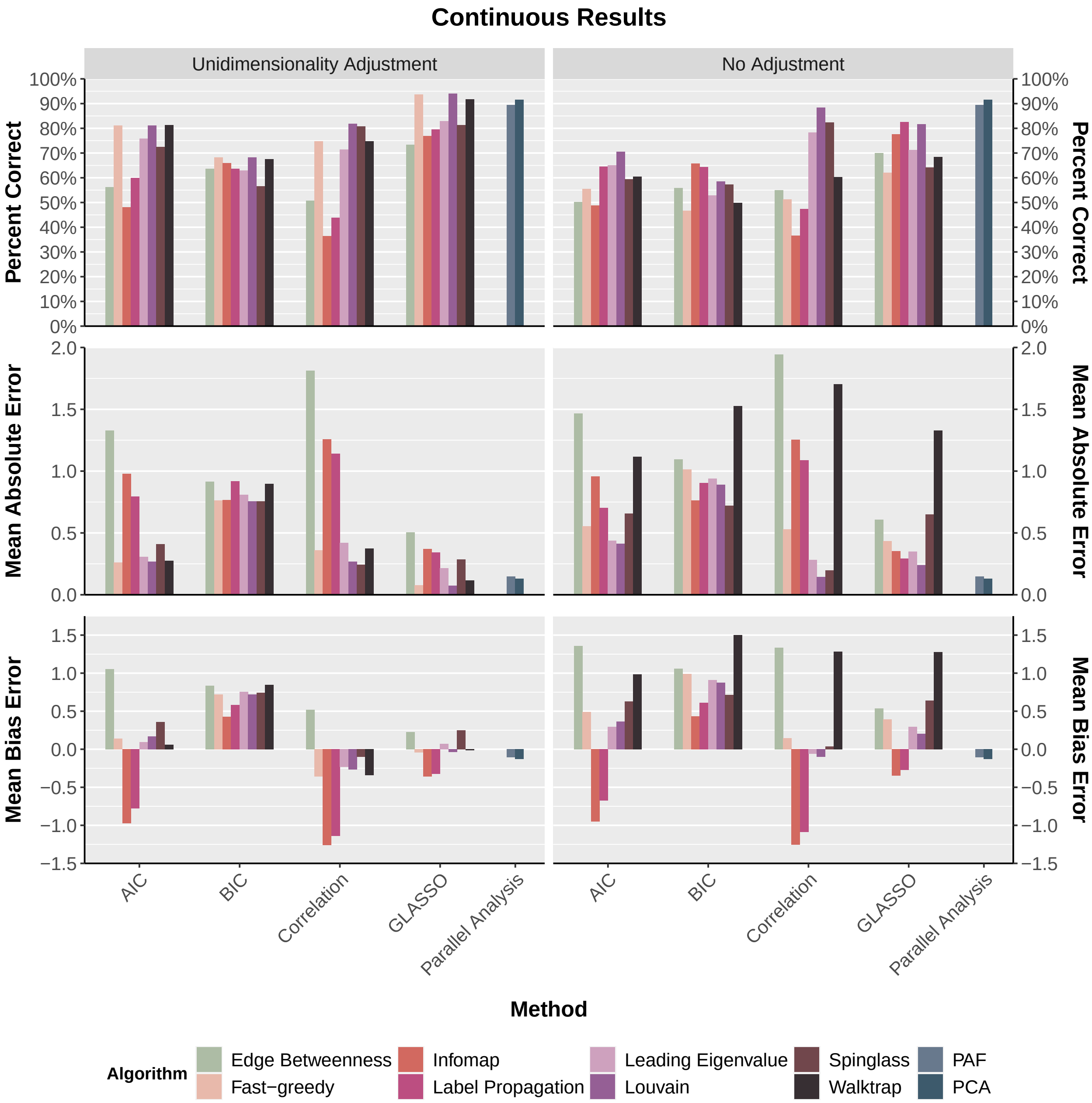


Figure 1: Continuous results for accuracy and bias measures broken down by method, algorithm, and unidimensional adjustment.

The best performing combinations for mean absolute error followed a similar pattern as the percent correct results: GLASSO with unidimensional adjustment for the Louvain (0.07), Fast-greedy (0.08), and Walktrap (0.12) algorithms were the best performing. These were followed closely by parallel analysis with PCA (0.13) and PAF (0.14). For the correlation matrices, the Spinglass algorithm with (0.24) and without (0.20) unidimensional adjustment performed the best along with the Louvain algorithm with adjustment (0.27). For AIC, the Fast-greedy (0.26), Louvain (0.27), and

Walktrap (0.28) with adjustment performed the best. For BIC, the Spinglass algorithm with (0.75) and without (0.72) unidimensional adjustment performed the best along with the Louvain algorithm with adjustment (0.76).

The best combinations for mean bias error had a different pattern. The GLASSO with unidimensional adjustment for the Walktrap (-0.01) and Louvain (-0.04) algorithms were the best followed by the correlation matrix with no adjustment for the Spinglass algorithm (0.04) and GLASSO with unidimensional adjustment for the Fast-greedy algorithm (-0.04). Continuing with the correlation matrices, Leading Eigenvalue (-0.06) and Louvain (-0.10) algorithms without adjustment were among the best performing. For parallel analysis, PAF (-0.10) and PCA(-0.13) had relatively low values. For AIC, Walktrap (0.06), Leading Eigenvalue (0.10), and Fast-greedy (0.14) algorithms with unidimensional adjustment were the best performing. For BIC, Infomap with and without (both = 0.43) unidimensional adjustment and Label Propagation with unidimensional adjustment (0.58) were the best performing.

4.1.3 Polytomous Data

The best performing combinations for percent correct were parallel analysis with PCA (87.9%) and PAF (84.6%; Figure 2). These were closely followed by correlation with no adjustment and Louvain algorithm (83.8%), GLASSO with unidimensional adjustment and Louvain (83.5%), Walktrap (83.2%), and Fast-greedy (83.0%) algorithms. Other top performing correlation algorithms were Louvain with unidimensional adjustment (79.7%) and Spinglass with (76.9%) and without (77.4%) unidimensional adjustment. For AIC, the Fast-greedy (61.1%), Walktrap (61.0%), and Louvain (60.3%) algorithms with unidimensional adjustment were top performing. For BIC, the Infomap algorithm with (46.1%) and without adjustment (46.0%) and the Fast-greedy (44.5%) algorithm with unidimensional adjustment were top performing.

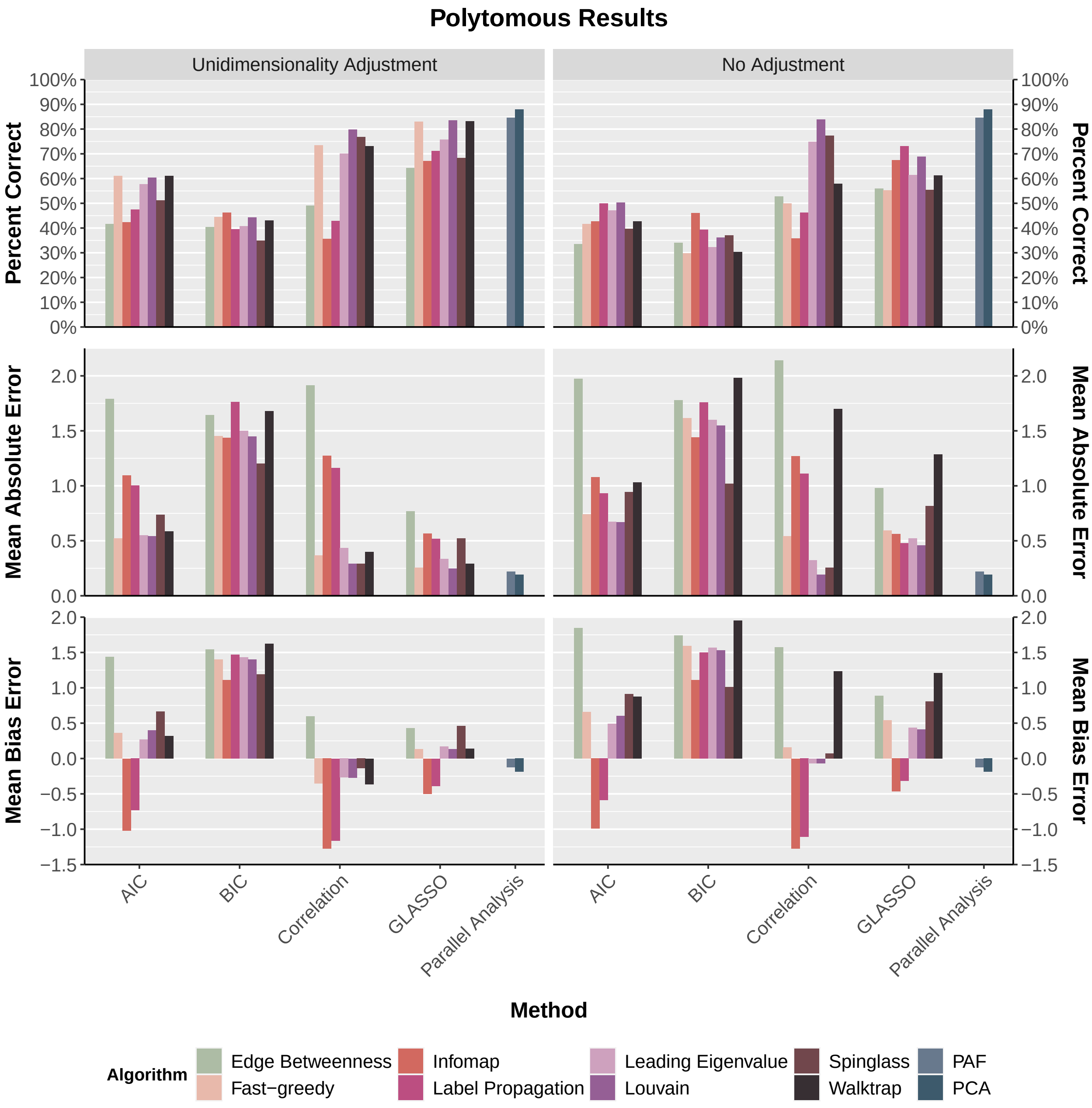


Figure 2: Polytomous results for accuracy and bias measures broken down by method, algorithm, and unidimensional adjustment.

The best performing combinations for mean absolute error for correlation matrices were the Louvain algorithm with (0.29) and without (0.19) adjustment as well as the Spinglass (0.26) algorithm with no adjustment. Parallel analysis performed as well as the best with both PCA (0.19) and PAF (0.22) algorithms. For GLASSO, the Louvain (0.25), Fast-greedy (0.26), and Walktrap (0.29) algorithms with unidimensional adjustment were top performing. For AIC, the Fast-greedy (0.52), Louvain (0.54), and Leading Eigenvalue (0.55) algorithms with unidimensional adjustment were top performing. For AIC, the Fast-greedy (0.52), Louvain (0.54), and Leading Eigenvalue (0.55) algorithms with unidimensional adjustment were top performing. For BIC, the Infomap algorithm with and without adjustment (both = 1.44) and the Spinglass (1.20) algorithm with unidimensional adjustment were top performing.

The best performing combinations for mean bias error for correlation matrices were the Louvain (-0.07), Leading Eigenvalue (-0.07), and Spinglass (0.07) algorithms without adjustment. For parallel analysis, PAF (-0.12) performed slightly better than PCA (-0.19). For GLASSO, the Louvain (0.13), Fast-greedy (0.13), and Walktrap (0.14) algorithms with unidimensional adjustment were top performing. For AIC, the Leading Eigenvalue (0.27), Walktrap (0.32), and Fast-greedy (0.36) algorithms with unidimensional adjustment were top performing. For BIC, the Infomap algorithm with (1.11) and without adjustment and the Spinglass (1.19) algorithm with unidimensional adjustment were top performing.

4.1.4 Summary of Accuracy and Bias

There were several clear patterns in the results. First, algorithms tended to perform better when there was a unidimensionality adjustment (but see some method-algorithm combinations for correlation; Figure 1 and 2). Across methods, the top performing algorithms were consistently the Fast-greedy, Louvain, and Walktrap algorithms. Across algorithms, the top performing methods were consistently GLASSO and correlation. These top performing method-algorithm combinations, especially the GLASSO method with unidimensionality adjustment, performed comparable or better than parallel analysis in continuous data but were comparable or worse in polytomous data.

4.2 Multidimensional Conditions

Shifting focus to the multidimensional conditions (two and four factors), we focus more specifically on the best performing method-algorithm combinations from the overall results. These were the correlation and GLASSO methods in combination with the Fast-greedy, Louvain, and Walktrap algorithms. Large effects for all method-algorithm combinations can be found in our Supplementary Information (SI 1).

4.2.1 Percent Correct Effects

Loading size was the only main effect that affected the GLASSO method and algorithm pairs (Fast-greedy, Louvain, and Walktrap). Correlation with Fast-greedy and Louvain was also affected. For these method-algorithm combinations, percent correct increased as loading size increased.

The rest of the main effects pertain only to the correlation method. There were separate main effects of correlation between factors and number of variables per factor for the Fast-greedy, Louvain, and Walktrap algorithms. For correlation between factors, percent correct decreased as correlation between factors increased; for number of variables per factor, percent correct increased as the number of variables per factor increased. Lastly, there was a main effect of number of factors for Fast-greedy and Walktrap such that percent correct decreased as the number of factor increased.

4.2.2 Adjusted Rand Index Effects

Overall, GLASSO and Louvain (0.96), Fast-greedy (0.95), Spinglass (0.94), and Walktrap (0.94) with no adjustment had the highest ARIs in the multidimensional conditions. This same rank-order was found when broken down by data type with minimal differences between continuous and polytomous data. Based on these results, we continued with our focus on the correlation and GLASSO methods with the Fast-greedy, Louvain, and Walktrap algorithms. All large effects for all other method-algorithm combinations are reported in the Supplemental Information (SI 2).

There were several interactions. For all three algorithms with GLASSO, there was an interaction between loading size and correlation between factors such that ARI increased as loading size increased and correlation between factors decreased. Loading size also interacted with number of variables per factor for GLASSO with Louvain and Walktrap such that ARI increased as loading size and number of variables per factor increased. A third interaction involving loading size and sample size for GLASSO and Louvain was found where ARI increased as loading size and sample size increased.

For correlation and Louvain, there were interactions between number of variables per factor and correlation between factors as well as unidimensional adjustment. In the former, ARI increased as the number of variables per factor increased and correlation between factors decreased. In the latter, ARI increased as the number of variables per factor increased and there was no adjustment. Correlation and Fast-greedy similarly had an interaction between number of variables per factor and unidimensional adjustment such that ARI increased as the number of variables per factor increased and there was no adjustment.

In terms of main effects, all algorithms for both correlation and GLASSO were affected by loading size, correlation between factors, and number of variables per factor. The effects were the same for all combinations: ARI increased as loading size or number of variables per factor increased and ARI decreased as correlation between factors increased. The correlation method had additional effects for number of factors and unidimensional adjustment. For number of factors, ARI decreased as the number of factors increased for the Fast-greedy and Walktrap algorithms. For the unidimensional adjustment, ARI was greater when there was no adjustment for the Fast-greedy and Louvain algorithms.

4.3 Unidimensional Conditions

Shifting to the unidimensional conditions, all algorithms for the correlation and GLASSO methods were considered. We did not consider AIC or BIC because of their poor overall performance. To determine the effects of the unidimensionality adjustment, we focused on the unidimensional and bidimensional (two factor) conditions. The focus on these conditions were to better evaluate the trade-off of the unidimensionality adjustment. Results were collapsed across data type because there were no substantial differences in performance.

In these conditions, the best performing method-algorithm combinations were parallel analysis and PCA (94.9%), correlation and Leading Eigenvalue with no unidimensionality adjustment (94.3%), and correlation and Louvain with no unidimensionality adjustment (93.1%). For combinations with the unidimensionality adjustment, the best performers were GLASSO and the Walktrap (90.9%), Leading Eigenvalue (90.8%), and Louvain (88.6%) algorithms.

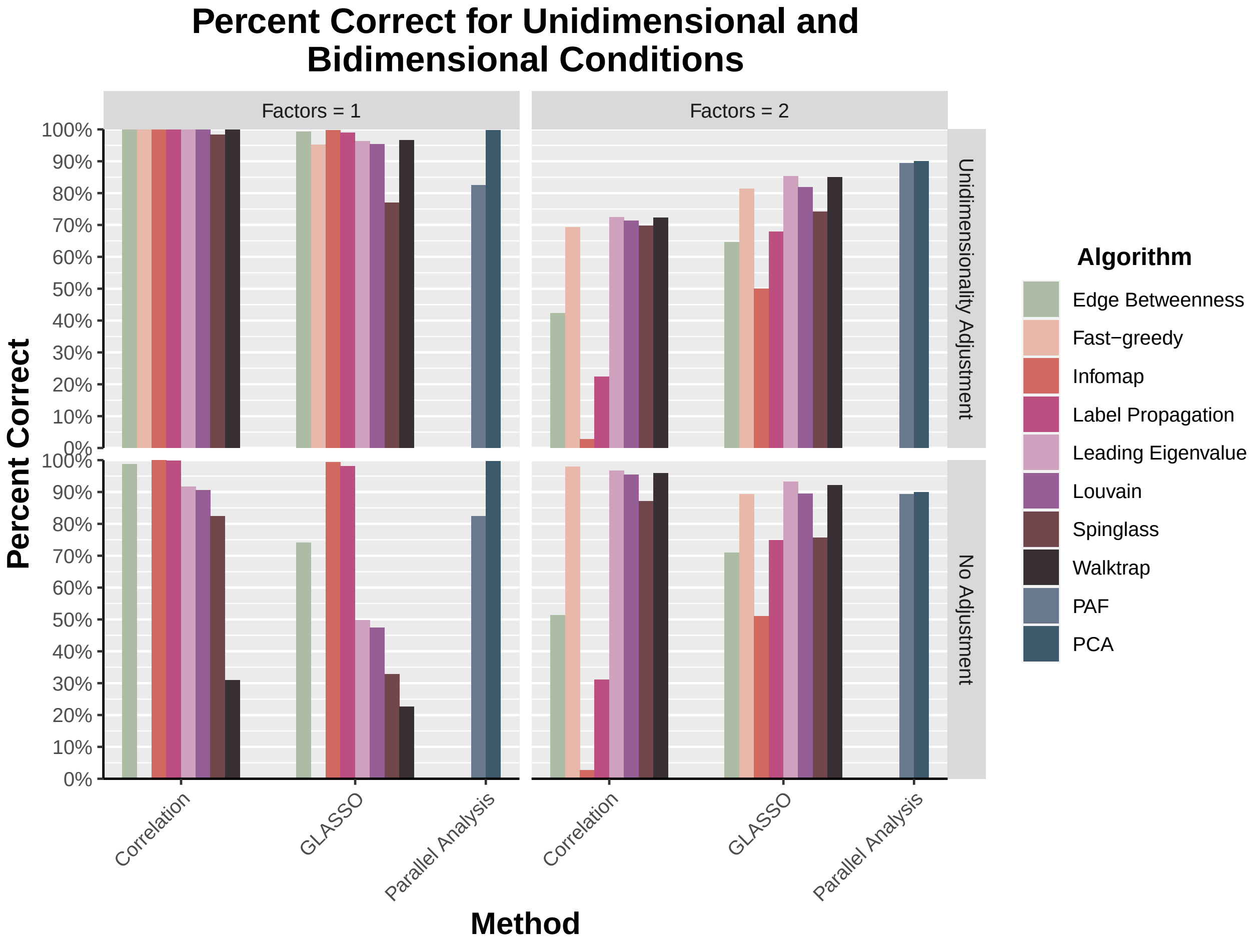


Figure 3: Combined data results for accuracy broken down by unidimensional and bidimensional conditions.

For the ANOVA, a general pattern key to unidimensionality adjustment investigation was identified: There was an interaction effect between the unidimensional adjustment and number of factors for every method-algorithm pair (including AIC and BIC) involving a modularity-related algorithm (specifically, Fast-greedy, Leading Eigenvalue, Louvain, and Walktrap). The effect, for all combinations, was that with the unidimensionality adjustment percent correct was greater for unidimensional structures than with no adjustment but percent correct was lower for bidimensional structures than with no adjustment.

For the rest of the effects, we focused on the interaction effects of the correlation and GLASSO methods with the top performing algorithms: Leading Eigenvalue, Louvain, and Walktrap. The large effects for all other method-algorithm combinations are reported in the Supplementary Information (SI 3). Other than the unidimensional adjustment and number of factors interaction, there was a crossover interaction between unidimensional adjustment and number of variables per factor for the Leading Eigenvalue and Louvain algorithms for both methods. When there were more variables per factor, percent correct was higher for the adjustment than no adjustment. When there were fewer variables per factor, percent correct was higher without the adjustment than with it. These same method-algorithm combinations also had a crossover interaction between the number of factors and number of variables per factor such that percent correct was higher with one factor and fewer variables or when there were two factors and more variables.

4.4 Alternative Unidimensionality Adjustment

The results of the unidimensional and bidimensional conditions demonstrated a strong trade-off: better performance in unidimensional conditions with an adjustment and better performance in bidimensional conditions without an adjustment. When considering options to avoid, or at least minimize, the trade-off of performance in unidimensional and bidimensional conditions with the unidimensionality adjustment, one approach might be to use an algorithm that performs well in both conditions without any adjustment.

There were two method-algorithms that were comparable to parallel analysis and PCA that did not involve any adjustment: correlation with Leading Eigenvalue and Louvain (Figure 3). A key criterion for using one of these algorithms as a unidimensionality adjustment is that they are precise—that is, they recover one community when there is one factor simulated and they recover two communities when there are two factors simulated. In both the unidimensional and bidimensional conditions, the Leading Eigenvalue algorithm (91.8% and 96.8%, respectively) had a slight edge on the Louvain algorithm (90.7% and 95.5%, respectively).

Based on this edge, we decided to investigate whether using the Leading Eigenvalue algorithm on the correlation matrix could mitigate some of the effects of the original unidimensionality adjustment. Specifically, whether using the Leading Eigenvalue on the correlation matrix to detect unidimensional structures could mitigate the performance trade-off in the two factor condition relative to the current adjustment. We used the following criterion to adjust the performance of the results with no adjustment: if correlation and Leading Eigenvalue recovered one community, then return one community; otherwise, use the unadjusted GLASSO-algorithm combination. To evaluate the performance of this criterion, we used the top performing GLASSO-algorithm combinations—Fast-greedy, Louvain, and Walktrap. We report the percent correct relative to the unidimensional adjustment results.

The GLASSO-algorithm performance overall improved by around 1% in accuracy: Louvain (89.5%), Fast-greedy (89.1%), and Walktrap (87.7%). Although this improvement seems minimal, a different story emerges when accuracy is broken down by factors. On the one hand, all algorithms' accuracies dropped by about 3% in the unidimensional conditions: Louvain (95.3% → 92.6%), Fast-greedy (95.1% → 92.6%), and Walktrap (96.7% → 92.8%). On the other hand, all algorithms' accuracies increased by about 5% in the bidimensional conditions: Louvain (81.8% → 87.1%), Fast-greedy (81.3% → 86.9%), and Walktrap (84.9% → 90.0%). This trade-off appears to strike a better balance between favoring unidimensional over bidimensional solutions relative to the previous approach. Although more investigations are needed, the Leading Eigenvalue adjustment appears to be an improvement on present standards for detecting unidimensionality in psychometric networks.

4.5 Applied Example

To demonstrate how these method-algorithm pairs might perform on real-world data, we used the Big Five Inventory dataset from the {psych} package (version 2.3.0, Revelle, 2017) in R. This dataset is comprised of 25 items with 5 items per theoretical factor (5 total factors) and follows similar characteristics to our polytomous data that were simulated in this study: variables are on a 6-point Likert scale, range in skew from -1.24 to 0.84 ($M = 0.27$ and $SD = 0.68$), and large to very large sample size ($N = 2436$ after removing cases with missing data). We applied all method-algorithm combinations evaluated in this simulation and provided ARI for the community detection method using the theoretical item placements.

The results generally paralleled our simulation findings: Louvain, Fast-greedy, and Walktrap all identified the theoretical factors and item placements within at least one method-algorithm combination (Table 2). Additionally, Spinglass all identified the theoretical factors and item placements with the GLASSO method. In terms of methods, both GLASSO and BIC had five algorithms identify the theoretical number of factors and three algorithms identify the theoretical item placements. AIC had four algorithms identify the theoretical number of factors but none identified the proper placement. Correlation did not identify the theoretical number of factors or item placements with any algorithm used. Parallel analysis with PCA identified the theoretical number of factors whereas PAF identified six factors.

To showcase the detection of unidimensional structures, we applied the method proposed by Golino and colleagues (2020), the correlation with Leading Eigenvalue combination identified in this study, and parallel analysis with PCA to each of the theoretical BFI factors. The expectation was that each of these factors should be unidimensional. Indeed, for all methods, we found that each factor was unidimensional. With no adjustment, GLASSO with the Fast-greedy, Louvain, and Walktrap algorithms produced two communities for each factor for all algorithms.

Table 2: BFI Empirical Example

Algorithm	Method	Communities/Factors	ARI
Edge Betweenness	AIC	9	0.13
	BIC	6	0.44
	Correlation	10	0.09
	GLASSO	7	0.47
Fast-greedy	AIC	**5**	0.80
	BIC	**5**	**1.00**
	Correlation	4	0.75
	GLASSO	**5**	0.89
Infomap	AIC	1	0.00
	BIC	4	0.75
	Correlation	1	0.00
	GLASSO	4	0.75
Label Propagation	AIC	1	0.00
	BIC	6	0.73
	Correlation	1	0.00
	GLASSO	6	0.73
Leading Eigenvalue	AIC	**5**	0.69
	BIC	**5**	0.89
	Correlation	3	0.33
	GLASSO	**5**	0.64
Louvain	AIC	**5**	0.80
	BIC	**5**	**1.00**
	Correlation	4	0.75
	GLASSO	**5**	**1.00**
Spinglass	AIC	**5**	0.80
	BIC	**5**	0.81
	Correlation	4	0.75
	GLASSO	**5**	**1.00**
Walktrap	AIC	4	0.48
	BIC	**5**	**1.00**
	Correlation	4	0.50
	GLASSO	**5**	**1.00**
PAF	Parallel Analysis	6	—
PCA	Parallel Analysis	**5**	—

Note. Bolded values indicate results that match the theoretical number of factors and item placement.

5 Discussion

This study examined the performance (accuracy and bias) of different network methods and several community detection algorithms for recovering the correct number of factors in data that parallel many commonly used questionnaires (e.g., personality). The aims of the study evaluated each component of the recently popularized EGA (Golino & Epskamp, 2017; Golino, Shi, et al., 2020). The standard sparsity induction approach, GLASSO, was compared against the zero-order correlation matrix and two variants of non-regularized GGMs (Williams & Rast, 2018; Williams et al., 2019). The community detection component, Walktrap, was evaluated against several other community detection algorithms that are commonly applied or open available in the {igraph} package. Third, their proposed approach to handled unidimensionality was tested more extensively. The overarching aim of the simulation was to identify what combinations of methods and algorithms would optimally perform in a wide variety of conditions common in empirical data found in psychology. As a benchmark, parallel analysis was used given its prevalence in the factor analytic literature.

Overall, we found that some sparsity induction method-algorithm combinations were comparable to the algorithms applied to the zero-order correlation matrix and parallel analysis algorithms, and that this performance was dependent on the sparsity induction method and community detection algorithms being used. Specifically, we found that the GLASSO sparsity induction method paired with the Louvain, Fast-greedy, and Walktrap algorithms all performed as well as parallel analysis or better than correlation approaches. When it came to unidimensional structures, some algorithms, particularly those optimized by modularity, performed better when there was an adjustment (i.e., adding an uncorrelated latent factor to the data; Golino, Shi, et al., 2020). We found, however, that this adjustment comes with an important trade-off: accuracy for simulated two factor structures substantially decreases.

Focusing on the sparsity induction methods, this study was the first, to our knowledge, to evaluate how different partial correlation sparsity induction methods performed when recovering the number of simulated factors in factor models. Previous work had compared the GLASSO to a correlation-based method, the TMFG, with the GLASSO showing better performance in nearly all conditions (Golino, Shi, et al., 2020). Other work, using the non-regularized partial correlation sparsity induction methods used in this study, evaluated the performance of these methods to recover the edges present in different population network structures, finding that the non-regularized approaches performed better on measures of specificity (avoidance of false positives) than the GLASSO method (Williams & Rast, 2018; Williams et al., 2019). In contrast to these studies, we found that although these methods may perform better when recovering the edges in dense simulated networks, they did not perform better with respect to recovering the correct number of simulated factors in unidimensional and multidimensional factor structures.

As for the community detection algorithms, there has been extensive evaluations of these algorithms across different literature. The closest comparison to our simulation used brain network correlation structures (Gates et al., 2016). In Gates and colleagues' (Gates et al., 2016) study, they found that for correlation matrices, the Walktrap algorithm outperformed the other algorithms on a measures of item placement. Notably, the Louvain algorithm performed the best when Euclidean Distance was used as a similarity measure. Our results generally parallel their study, finding the Walktrap and Louvain algorithms among the best performing algorithms. Our study more extensively demonstrates that these algorithms are effective on relatively small networks despite their original purpose for larger networks (>1,000 nodes; Fortunato, 2010).

It's important to emphasize that our study is the first, to our knowledge, to investigate these algorithms with respect to different methods (sparsity induction vs. zero-order correlation matrix only) but also data that was not continuous. The choice of using polytomous data with five categories was based on the ubiquitous use of 5-point Likert scales in psychological research (e.g., personality). Previous work examined how the GLASSO paired with the Walktrap algorithm performed in dichotomous data, finding that there was about a 6% difference in accuracy (better accuracy for continuous data; Golino, Shi, et al., 2020). Our results, specifically with the GLASSO, demonstrate that there is a notable drop-off in accuracy between continuous and polytomous data (around 10% for most algorithms). This larger drop-off in accuracy could be do differences in the simulated conditions between the two studies (but see Isvoranu & Epskamp, 2021). As with any simulation, the results are dependent on the characteristics of the simulated data.

The findings of this simulation warrant some consideration for future applications of EGA, which has been one of the most common approaches for community detection in network psychometrics. Our results show community detection algorithms, specifically those optimized by modularity, are ineffective for identifying unidimensional structures when the GLASSO algorithm is applied. Although several algorithms are more effective when they are applied to correlation matrices, they tended to suffer when there were one or two factors. This poses a specific problem that community detection algorithms may not be optimal for handling unidimensional structures when paired with sparsity induction methods.

We evaluated one approach, proposed by Golino, Shi, et al. (2020), that "expands" the correlation matrix with four highly correlated variables that are orthogonal to the empirical data. This approach appears to work fairly well (recovery of unidimensional structures substantially improves), however, there was a significant trade-off: recovery of two factor structures was substantially hindered. In light of this trade-off, we examined an alternative approach that uses the Leading Eigenvalue algorithm on the correlation matrix as a unidimensionality "check" before proceeding with a sparsity induction method and a subsequent community detection algorithm. We found that this alternative approach achieved a better balance between one and two factor recovery, making it an appealing alternative. Caution should be given to the results of this single study, however. Future studies should continue to evaluate these approaches to determine whether one is better than the other. A more extensive simulation that evaluates unidimensional and bidimensional structures could provide more insight into which algorithms work best and under what conditions.

When considering the community detection component, we found evidence that the Louvain and Fast-greedy algorithm are worthwhile considerations for adoption into the EGA framework. Because the two algorithms are relatively similar and demonstrated similar performance, preference should be given to the Louvain algorithm because it also provides a hierarchical or "multi-level" structuring of communities. Such hierarchical structuring is important for determining different levels of taxonomies that often exist in psychological assessment instruments (e.g., personality, psychopathology; Castro, Ferreira, & Ferreira, 2020; Jiménez et al., 2022; Kotov et al., 2017). Moreover, an additional algorithm also provides another method to compare results from EGA such that the best fitting or most theoretically consistent model can be chosen based on the results (e.g., Golino, Moulder, et al., 2020).

In sum, our simulation demonstrates that there are parts of the EGA algorithm that could be improved. First, the GLASSO sparsity induction method appears to have an advantage over non-regularized approaches when it comes to accurate latent factor recovery. Second, the Louvain and Walktrap algorithms appear to work well with the GLASSO method and are competitive with parallel analysis approaches. Third, the unidimensionality adjustment proposed by Golino, Shi, et al. (2020) demonstrated a substantial trade-off between unidimensional and bidimensional structures, favoring unidimensional over bidimensional. We identified a potential alternative that uses the Leading Eigenvalue algorithm on the zero-order correlation matrix that may be a more balanced comprise between unidimensional and bidimensional detection. Future work should continue to validate and explore alternatives to the components of the EGA algorithm to combine into the most robust network psychometrics dimension reduction procedure possible.

6 Funding Information

L.E.G. and K.G-P. were supported by Grant 2018-2019-1D2-085 from the Fondo Nacional de Innovación y Desarrollo Científico y Tecnológico (FONDOCYT) of the Dominican Republic.

7 Open Science Framework

`https://osf.io/jfxad/`

8 CRediT

The authors made the following contributions.

Alexander P. Christensen: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - Original Draft Preparation, Writing - Review & Editing;

Luis Eduardo Garrido: Conceptualization, Investigation, Methodology, Supervision, Validation, Writing - Review & Editing;

Kiero Guerra-Peña: Conceptualization, Investigation, Methodology, Supervision, Validation, Writing - Review & Editing;

Hudson Golino: Conceptualization, Investigation, Methodology, Resources, Software, Supervision, Validation, Writing - Review & Editing.

9 Correspondence

Correspondence concerning this article should be addressed to Alexander P. Christensen, Department of Psychology and Human Development, Vanderbilt University, Nashville, TN, 37203. E-mail: alexpaulchristensen@gmail.com

10 References

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*, P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

Brusco, M., Steinley, D., & Watts, A. L. (2022). A comparison of spectral clustering and the walktrap algorithm for community detection in network psychometrics. *Psychological Methods*. https://doi.org/10.1037/met0000509

Castro, D., Ferreira, F., & Ferreira, T. B. (2020). Modularity of the personality network. *European Journal of Psychological Assessment*, *36*, 998–1008. https://doi.org/10.1027/1015-5759/a000613

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, *1*, 245–276. https://doi.org/10.1207/s15327906mbr0102_10

Cattell, R. B. (1978). *The scientific use of factor analysis in behavioral and life sciences*. Boston, MA: Springer. https://doi.org/10.1007/978-1-4684-2262-7

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., … Borges, B. (2022). *shiny: Web Application Framework for R*. Retrieved from https://CRAN.R-project.org/package=shiny

Chen, J., & Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, *95*, 759–771. https://doi.org/10.1093/biomet/asn034

Christensen, A. P., & Golino, H. (2021). Estimating the stability of psychological dimensions via bootstrap exploratory graph analysis: A Monte Carlo simulation and tutorial. *Psych*, *3*(3), 479–500. https://doi.org/10.3390/psych3030032

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

Cohen, J. (1992). A power primer. *Psychological Bulletin*, *112*, 155–159. https://doi.org/10.1037/0033-2909.112.1.155

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, *1695*, 1–9. Retrieved from https://www.semanticscholar.org/paper/The-igraph-software-package-for-complex-network-Cs/%C3/%A1rdi-Nepusz/1d2744b83519657f5f2610698a8ddd177ced4f5c?p2df

De Beurs, D., Fried, E. I., Wetherall, K., Cleare, S., O'Connor, D. B., Ferguson, E., … O'Connor, R. C. (2019). Exploring the psychology of suicidal ideation: A theory driven network analysis. *Behaviour Research and Therapy*, *120*, 103419. https://doi.org/10.1016/j.brat.2019.103419

Epskamp, S., Cramer, A. O. J., Waldorp, L. J., Schmittmann, V. D., & Borsboom, D. (2012). qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software*, *48*, 1–18. https://doi.org/10.18637/jss.v048.i04

Epskamp, S., & Fried, E. I. (2018). A tutorial on regularized partial correlation networks. *Psychological Methods*, *23*, 617–634. https://doi.org/10.1037/met0000167

Epskamp, S., Kruis, J., & Marsman, M. (2017). Estimating psychopathological networks: Be careful what you wish for. *PLoS ONE*, *12*(6). https://doi.org/10.1371/journal.pone.0179891

Epskamp, S., Maris, G., Waldrop, L. J., & Borsboom, D. (2018). Network psychometrics. In P. Irwing, D. Hughes, & T. Booth (Eds.), *The Wiley handbook of psychometric testing, 2 volume set: A multidisciplinary reference on survey, scale and test development*. New York, NY: Wiley. https://doi.org/10.1002/9781118489772.ch30

Fan, Y., Li, M., Zhang, P., Wu, J., & Di, Z. (2007). Accuracy and precision of methods for community identification in weighted networks. *Physica A: Statistical Mechanics and Its Applications*, *377*, 363–372. https://doi.org/10.1016/j.physa.2006.11.036

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

Foygel, R., & Drton, M. (2010). Extended Bayesian information criteria for Gaussian graphical models. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Advances in neu-*

ral information processing systems (pp. 604–612). Retrieved from https://papers.nips.cc/paper/4087-extended-bayesian-information-criteria-for-gaussian-graphical-models

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 35–41. Retrieved from https://www.jstor.org/stable/3033543

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*, 432–441. https://doi.org/10.1093/biostatistics/kxm045

Friedman, J., Hastie, T., & Tibshirani, R. (2014). *glasso: Graphical lasso – estimation of Gaussian graphical models*. Retrieved from https://CRAN.R-project.org/package=glasso

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2011). Performance of Velicer's minimum average partial factor retention method with categorical variables. *Educational and Psychological Measurement*, *71*, 551–570. https://doi.org/10.1177/0013164410389489

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2013). A new look at Horn's parallel analysis with ordinal variables. *Psychological Methods*, *18*, 454–474. https://doi.org/10.1037/a0030005

Gates, K. M., Henry, T., Steinley, D., & Fair, D. A. (2016). A Monte Carlo evaluation of weighted community detection algorithms. *Frontiers in Neuroinformatics*, *10*, 45. https://doi.org/10.3389/fninf.2016.00045

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, *99*, 7821–7826. https://doi.org/10.1073/pnas.122653799

Golino, H., & Epskamp, S. (2017). Exploratory Graph Analysis: A new approach for estimating the number of dimensions in psychological research. *PLoS ONE*, *12*, e0174035. https://doi.org/10.1371/journal.pone.0174035

Golino, H., Moulder, R., Shi, D., Christensen, A. P., Neito, M. D., Nesselroade, J. R., … Boker, S. M. (2020). Entropy Fit Index: A new fit measure for assessing the structure and dimensionality of multiple latent variables. *Multivariate Behavioral Research*. https://doi.org/10.1080/00273171.2020.1779642

Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., … Martinez-Molina, A. (2020). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, *25*, 292–320. https://doi.org/10.1037/met0000255

Guttman, L. (1953). Image theory for the structure of quantitative variates. *Psychometrika*, *18*, 277–296.

Henson, R. K., & Roberts, J. K. (2006). Use of exploratory factor analysis in published research: Common errors and some comment on improved practice. *Educational and Psychological Measurement*, *66*, 393–416. https://doi.org/10.1177/0013164405282485

Hoffman, M., Steinley, D., Gates, K. M., Prinstein, M. J., & Brusco, M. J. (2018). Detecting clusters/communities in social networks. *Multivariate Behavioral Research*, *53*(1), 57–73. https://doi.org/10.1080/00273171.2017.1391682

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, *30*, 179–185. https://doi.org/10.1007/BF02289447

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

Isvoranu, A.-M., & Epskamp, S. (2021). Which estimation method to choose in network psychometrics? Deriving guidelines for applied researchers. *Psychological Methods*. https://doi.org/10.1037/met0000439

Jiménez, M., Abad, F. J., Garcia-Garzon, E., Golino, H., Christensen, A. P., & Garrido, L. E. (2022). Dimensionality assessment in generalized bi-factor structures: A network psychometrics approach. *PsyArXiv*. https://doi.org/10.31234/osf.io/2ujdk

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, *20*, 141–151. https://doi.org/10.1177/001316446002000116

Kotov, R., Krueger, R. F., Watson, D., Achenbach, T. M., Althoff, R. R., Bagby, R. M., … Zimmerman, M. (2017). The hierarchical taxonomy of psychopathology (HiTOP): A dimensional alternative to traditional nosologies. *Journal of Abnormal Psychology*, *126*, 454–477. https://doi.org/10.1037/abn0000258

Lancichinetti, A., & Fortunato, S. (2009). Community detection algorithms: A comparative analysis. *Physical Review E*, *80*, 056117. https://doi.org/10.1103/PhysRevE.80.056117

Lancichinetti, A., & Fortunato, S. (2012). Consensus clustering in complex networks. *Scientific Reports*, *2*, 336. https://doi.org/10.1038/srep00336

Lauritzen, S. L. (1996). *Graphical models*. Oxford, UK: Clarendon Press.

Massara, G. P., Di Matteo, T., & Aste, T. (2016). Network filtering for big data: Triangulated Maximally Filtered Graph. *Journal of Complex Networks*, *5*, 161–178. https://doi.org/10.1093/comnet/cnw015

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, *103*, 8577–8582. https://doi.org/10.1073/pnas.0601602103

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, *10*, 191–218. https://doi.org/10.7155/jgaa.00185

R Core Team. (2022). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

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

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

Revelle, W. (2017). *psych: Procedures for psychological, psychometric, and personality research*. Evanston, Illinois: Northwestern University. Retrieved from https://CRAN.R-project.org/package=psych

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, *105*, 1118–1123. https://doi.org/10.1073/pnas.0706851105

Slocum-Gori, S. L., & Zumbo, B. D. (2011). Assessing the unidimensionality of psychological scales: Using multiple criteria from factor analysis. *Social Indicators Research*, *102*, 443–461. https://doi.org/10.1007/s11205-010-9682-8

Steinley, D., Brusco, M. J., & Hubert, L. (2016). The variance of the adjusted Rand index. *Psychological Methods*, *21*, 261–272. https://doi.org/10.1037/met0000049

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Ward, J. H. (1963). Hierarchical clustering to optimise an objective function. *Journal of the American Statistical Association*, *58*, 238–244.

Williams, D. R. (2019). *GGMnonreg: Estimate non-regularized Gaussian graphical models*. Retrieved from https://github.com/donaldRwilliams/GGMnonreg

Williams, D. R. (2021). Bayesian estimation for gaussian graphical models: Structure learning, predictability, and network comparisons. *Multivariate Behavioral Research*, *56*(2), 336–352. https://doi.org/10.1080/00273171.2021.1894412

Williams, D. R., & Mulder, J. (2020). BGGM: Bayesian Gaussian graphical models in R. *Journal of Open Source Software*, *5*(51), 2111. https://doi.org/10.21105/joss.02111

Williams, D. R., & Rast, P. (2018). Back to the basics: Rethinking partial correlation network methodology. *Journal of Mathematical and Statistical Psychology*. https://doi.org/10.1111/bmsp.12173

Williams, D. R., Rhemtulla, M., Wysocki, A. C., & Rast, P. (2019). On nonregularized estimation of psychological networks. *Multivariate Behavioral Research*, *54*, 719–750. https://doi.org/10.1080/00273171.2019.1575716

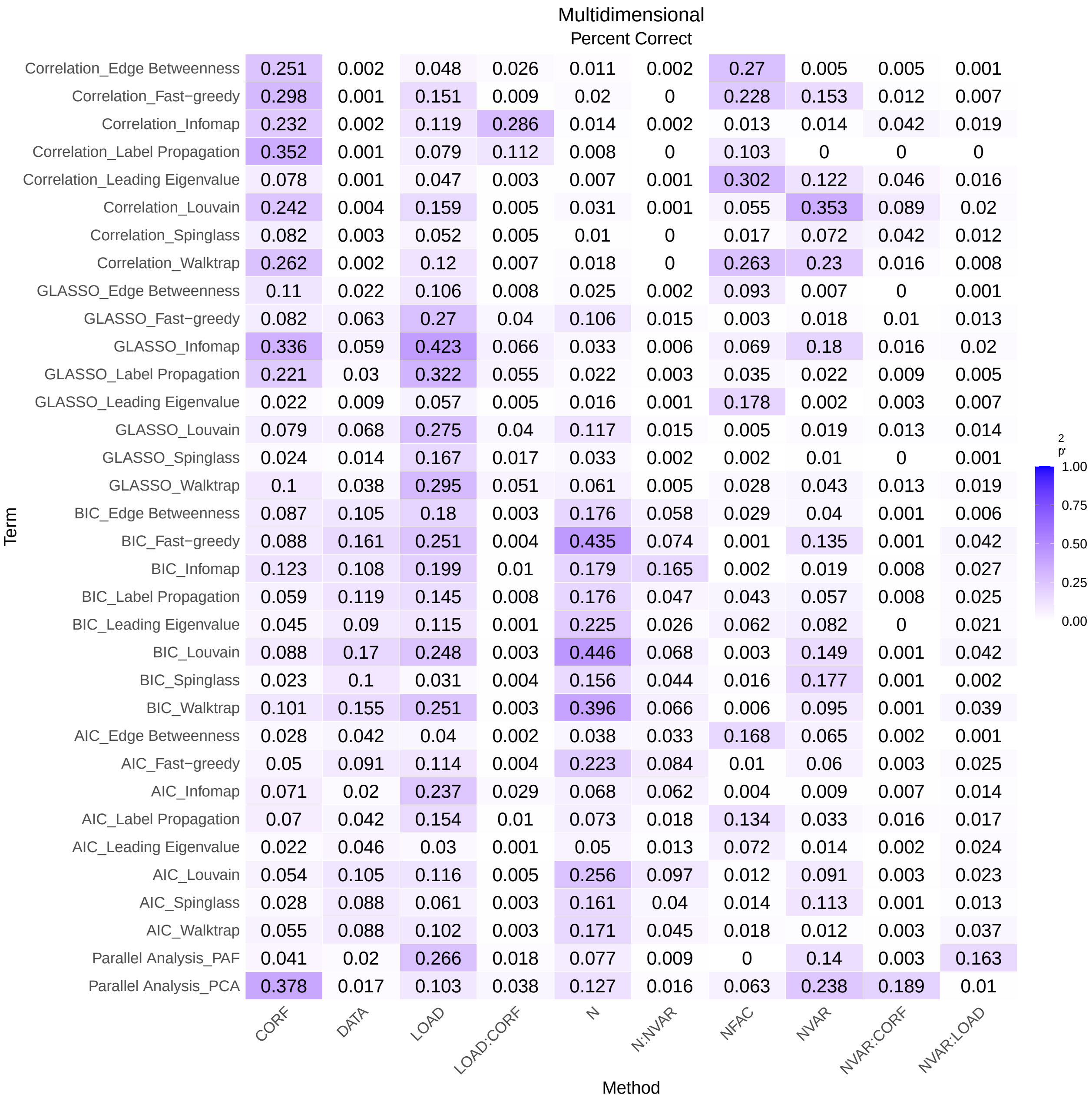
Wysocki, A. C., & Rhemtulla, M. (2021). On penalty parameter selection for estimating network models. *Multivariate Behavioral Research*, *56*(2), 288–302. https://doi.org/10.1080/00273171.2019.1672516

Yang, Z., Algesheimer, R., & Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, *6*, 30750. https://doi.org/10.1038/srep30750

11 Supplementary Information

Comparing Community Detection Algorithms in Psychological Data: A Monte Carlo Simulation

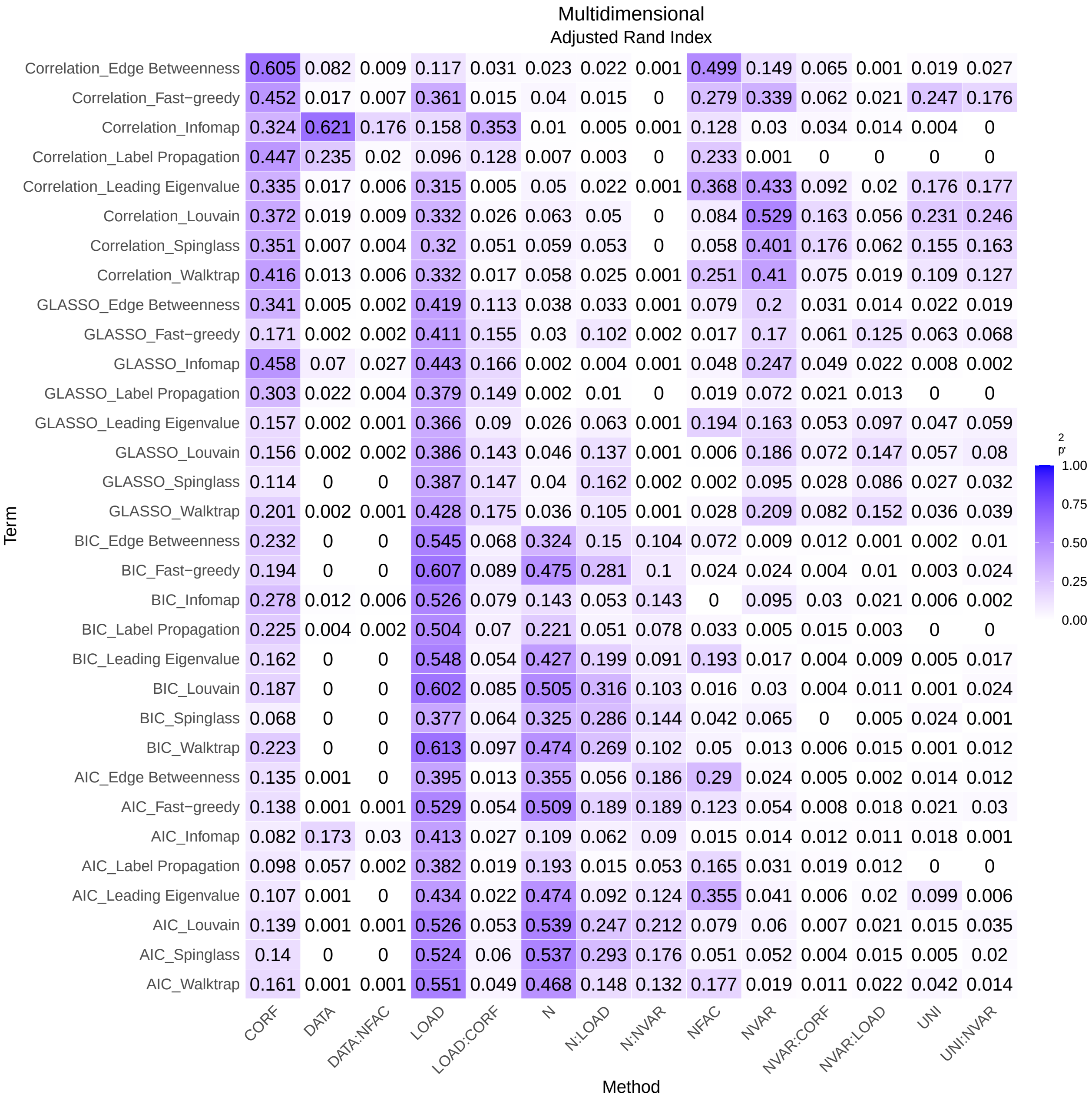
11.1 SI 1. Multidimensional Percent Correct ANOVA Effects



Term	CORF	DATA	LOAD	LOAD:CORF	N	N:NVAR	NFAC	NVAR	NVAR:CORF	NVAR:LOAD
Correlation_Edge Betweenness	0.251	0.002	0.048	0.026	0.011	0.002	0.27	0.005	0.005	0.001
Correlation_Fast−greedy	0.298	0.001	0.151	0.009	0.02	0	0.228	0.153	0.012	0.007
Correlation_Infomap	0.232	0.002	0.119	0.286	0.014	0.002	0.013	0.014	0.042	0.019
Correlation_Label Propagation	0.352	0.001	0.079	0.112	0.008	0	0.103	0	0	0
Correlation_Leading Eigenvalue	0.078	0.001	0.047	0.003	0.007	0.001	0.302	0.122	0.046	0.016
Correlation_Louvain	0.242	0.004	0.159	0.005	0.031	0.001	0.055	0.353	0.089	0.02
Correlation_Spinglass	0.082	0.003	0.052	0.005	0.01	0	0.017	0.072	0.042	0.012
Correlation_Walktrap	0.262	0.002	0.12	0.007	0.018	0	0.263	0.23	0.016	0.008
GLASSO_Edge Betweenness	0.11	0.022	0.106	0.008	0.025	0.002	0.093	0.007	0	0.001
GLASSO_Fast−greedy	0.082	0.063	0.27	0.04	0.106	0.015	0.003	0.018	0.01	0.013
GLASSO_Infomap	0.336	0.059	0.423	0.066	0.033	0.006	0.069	0.18	0.016	0.02
GLASSO_Label Propagation	0.221	0.03	0.322	0.055	0.022	0.003	0.035	0.022	0.009	0.005
GLASSO_Leading Eigenvalue	0.022	0.009	0.057	0.005	0.016	0.001	0.178	0.002	0.003	0.007
GLASSO_Louvain	0.079	0.068	0.275	0.04	0.117	0.015	0.005	0.019	0.013	0.014
GLASSO_Spinglass	0.024	0.014	0.167	0.017	0.033	0.002	0.002	0.01	0	0.001
GLASSO_Walktrap	0.1	0.038	0.295	0.051	0.061	0.005	0.028	0.043	0.013	0.019
BIC_Edge Betweenness	0.087	0.105	0.18	0.003	0.176	0.058	0.029	0.04	0.001	0.006
BIC_Fast−greedy	0.088	0.161	0.251	0.004	0.435	0.074	0.001	0.135	0.001	0.042
BIC_Infomap	0.123	0.108	0.199	0.01	0.179	0.165	0.002	0.019	0.008	0.027
BIC_Label Propagation	0.059	0.119	0.145	0.008	0.176	0.047	0.043	0.057	0.008	0.025
BIC_Leading Eigenvalue	0.045	0.09	0.115	0.001	0.225	0.026	0.062	0.082	0	0.021
BIC_Louvain	0.088	0.17	0.248	0.003	0.446	0.068	0.003	0.149	0.001	0.042
BIC_Spinglass	0.023	0.1	0.031	0.004	0.156	0.044	0.016	0.177	0.001	0.002
BIC_Walktrap	0.101	0.155	0.251	0.003	0.396	0.066	0.006	0.095	0.001	0.039
AIC_Edge Betweenness	0.028	0.042	0.04	0.002	0.038	0.033	0.168	0.065	0.002	0.001
AIC_Fast−greedy	0.05	0.091	0.114	0.004	0.223	0.084	0.01	0.06	0.003	0.025
AIC_Infomap	0.071	0.02	0.237	0.029	0.068	0.062	0.004	0.009	0.007	0.014
AIC_Label Propagation	0.07	0.042	0.154	0.01	0.073	0.018	0.134	0.033	0.016	0.017
AIC_Leading Eigenvalue	0.022	0.046	0.03	0.001	0.05	0.013	0.072	0.014	0.002	0.024
AIC_Louvain	0.054	0.105	0.116	0.005	0.256	0.097	0.012	0.091	0.003	0.023
AIC_Spinglass	0.028	0.088	0.061	0.003	0.161	0.04	0.014	0.113	0.001	0.013
AIC_Walktrap	0.055	0.088	0.102	0.003	0.171	0.045	0.018	0.012	0.003	0.037
Parallel Analysis_PAF	0.041	0.02	0.266	0.018	0.077	0.009	0	0.14	0.003	0.163
Parallel Analysis_PCA	0.378	0.017	0.103	0.038	0.127	0.016	0.063	0.238	0.189	0.01

Figure 4: All large ANOVA effects for percent across all method-algorithm combinations in multidimensional conditons.

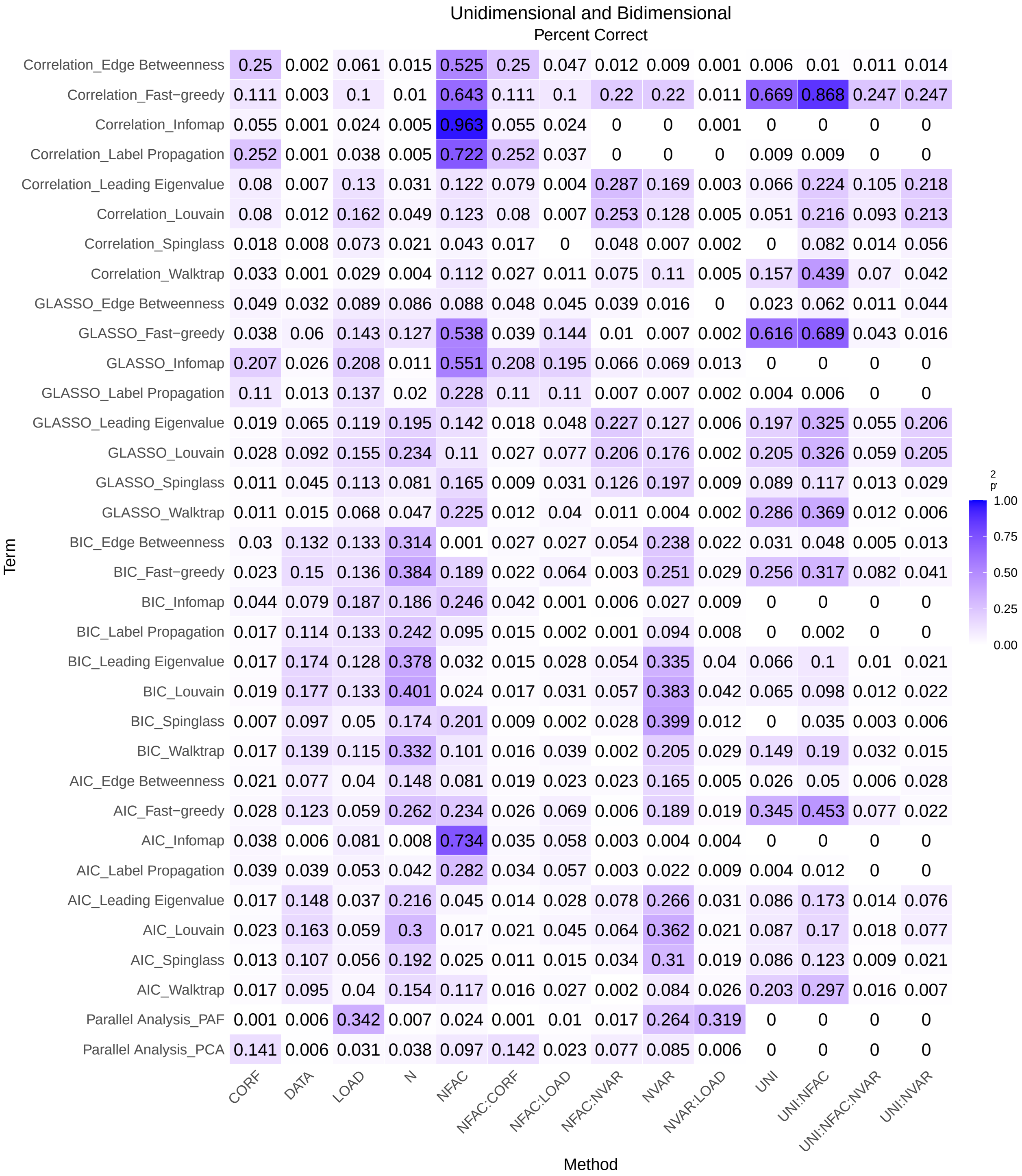
11.2 SI 2. Multidimensional ARI ANOVA Effects



Term	CORF	DATA	DATA:NFAC	LOAD	LOAD:CORF	N	N:LOAD	N:NVAR	NFAC	NVAR	NVAR:CORF	NVAR:LOAD	UNI	UNI:NVAR
Correlation_Edge Betweenness	0.605	0.082	0.009	0.117	0.031	0.023	0.022	0.001	0.499	0.149	0.065	0.001	0.019	0.027
Correlation_Fast−greedy	0.452	0.017	0.007	0.361	0.015	0.04	0.015	0	0.279	0.339	0.062	0.021	0.247	0.176
Correlation_Infomap	0.324	0.621	0.176	0.158	0.353	0.01	0.005	0.001	0.128	0.03	0.034	0.014	0.004	0
Correlation_Label Propagation	0.447	0.235	0.02	0.096	0.128	0.007	0.003	0	0.233	0.001	0	0	0	0
Correlation_Leading Eigenvalue	0.335	0.017	0.006	0.315	0.005	0.05	0.022	0.001	0.368	0.433	0.092	0.02	0.176	0.177
Correlation_Louvain	0.372	0.019	0.009	0.332	0.026	0.063	0.05	0	0.084	0.529	0.163	0.056	0.231	0.246
Correlation_Spinglass	0.351	0.007	0.004	0.32	0.051	0.059	0.053	0	0.058	0.401	0.176	0.062	0.155	0.163
Correlation_Walktrap	0.416	0.013	0.006	0.332	0.017	0.058	0.025	0.001	0.251	0.41	0.075	0.019	0.109	0.127
GLASSO_Edge Betweenness	0.341	0.005	0.002	0.419	0.113	0.038	0.033	0.001	0.079	0.2	0.031	0.014	0.022	0.019
GLASSO_Fast−greedy	0.171	0.002	0.002	0.411	0.155	0.03	0.102	0.002	0.017	0.17	0.061	0.125	0.063	0.068
GLASSO_Infomap	0.458	0.07	0.027	0.443	0.166	0.002	0.004	0.001	0.048	0.247	0.049	0.022	0.008	0.002
GLASSO_Label Propagation	0.303	0.022	0.004	0.379	0.149	0.002	0.01	0	0.019	0.072	0.021	0.013	0	0
GLASSO_Leading Eigenvalue	0.157	0.002	0.001	0.366	0.09	0.026	0.063	0.001	0.194	0.163	0.053	0.097	0.047	0.059
GLASSO_Louvain	0.156	0.002	0.002	0.386	0.143	0.046	0.137	0.001	0.006	0.186	0.072	0.147	0.057	0.08
GLASSO_Spinglass	0.114	0	0	0.387	0.147	0.04	0.162	0.002	0.002	0.095	0.028	0.086	0.027	0.032
GLASSO_Walktrap	0.201	0.002	0.001	0.428	0.175	0.036	0.105	0.001	0.028	0.209	0.082	0.152	0.036	0.039
BIC_Edge Betweenness	0.232	0	0	0.545	0.068	0.324	0.15	0.104	0.072	0.009	0.012	0.001	0.002	0.01
BIC_Fast−greedy	0.194	0	0	0.607	0.089	0.475	0.281	0.1	0.024	0.024	0.004	0.01	0.003	0.024
BIC_Infomap	0.278	0.012	0.006	0.526	0.079	0.143	0.053	0.143	0	0.095	0.03	0.021	0.006	0.002
BIC_Label Propagation	0.225	0.004	0.002	0.504	0.07	0.221	0.051	0.078	0.033	0.005	0.015	0.003	0	0
BIC_Leading Eigenvalue	0.162	0	0	0.548	0.054	0.427	0.199	0.091	0.193	0.017	0.004	0.009	0.005	0.017
BIC_Louvain	0.187	0	0	0.602	0.085	0.505	0.316	0.103	0.016	0.03	0.004	0.011	0.001	0.024
BIC_Spinglass	0.068	0	0	0.377	0.064	0.325	0.286	0.144	0.042	0.065	0	0.005	0.024	0.001
BIC_Walktrap	0.223	0	0	0.613	0.097	0.474	0.269	0.102	0.05	0.013	0.006	0.015	0.001	0.012
AIC_Edge Betweenness	0.135	0.001	0	0.395	0.013	0.355	0.056	0.186	0.29	0.024	0.005	0.002	0.014	0.012
AIC_Fast−greedy	0.138	0.001	0.001	0.529	0.054	0.509	0.189	0.189	0.123	0.054	0.008	0.018	0.021	0.03
AIC_Infomap	0.082	0.173	0.03	0.413	0.027	0.109	0.062	0.09	0.015	0.014	0.012	0.011	0.018	0.001
AIC_Label Propagation	0.098	0.057	0.002	0.382	0.019	0.193	0.015	0.053	0.165	0.031	0.019	0.012	0	0
AIC_Leading Eigenvalue	0.107	0.001	0	0.434	0.022	0.474	0.092	0.124	0.355	0.041	0.006	0.02	0.099	0.006
AIC_Louvain	0.139	0.001	0.001	0.526	0.053	0.539	0.247	0.212	0.079	0.06	0.007	0.021	0.015	0.035
AIC_Spinglass	0.14	0	0	0.524	0.06	0.537	0.293	0.176	0.051	0.052	0.004	0.015	0.005	0.02
AIC_Walktrap	0.161	0.001	0.001	0.551	0.049	0.468	0.148	0.132	0.177	0.019	0.011	0.022	0.042	0.014

Figure 5: All large ANOVA effects for Adjusted Rand Index across all method-algorithm combinations in multidimensional conditons.

11.3 SI 3. Unidimensional and Bidimensional Percent Correct ANOVA Effects



Term	CORF	DATA	LOAD	N	NFAC	NFAC:CORF	NFAC:LOAD	NFAC:NVAR	NVAR	NVAR:LOAD	UNI	UNI:NFAC	UNI:NFAC:NVAR	UNI:NVAR
Correlation_Edge Betweenness	0.25	0.002	0.061	0.015	0.525	0.25	0.047	0.012	0.009	0.001	0.006	0.01	0.011	0.014
Correlation_Fast−greedy	0.111	0.003	0.1	0.01	0.643	0.111	0.1	0.22	0.22	0.011	0.669	0.868	0.247	0.247
Correlation_Infomap	0.055	0.001	0.024	0.005	0.963	0.055	0.024	0	0	0.001	0	0	0	0
Correlation_Label Propagation	0.252	0.001	0.038	0.005	0.722	0.252	0.037	0	0	0	0.009	0.009	0	0
Correlation_Leading Eigenvalue	0.08	0.007	0.13	0.031	0.122	0.079	0.004	0.287	0.169	0.003	0.066	0.224	0.105	0.218
Correlation_Louvain	0.08	0.012	0.162	0.049	0.123	0.08	0.007	0.253	0.128	0.005	0.051	0.216	0.093	0.213
Correlation_Spinglass	0.018	0.008	0.073	0.021	0.043	0.017	0	0.048	0.007	0.002	0	0.082	0.014	0.056
Correlation_Walktrap	0.033	0.001	0.029	0.004	0.112	0.027	0.011	0.075	0.11	0.005	0.157	0.439	0.07	0.042
GLASSO_Edge Betweenness	0.049	0.032	0.089	0.086	0.088	0.048	0.045	0.039	0.016	0	0.023	0.062	0.011	0.044
GLASSO_Fast−greedy	0.038	0.06	0.143	0.127	0.538	0.039	0.144	0.01	0.007	0.002	0.616	0.689	0.043	0.016
GLASSO_Infomap	0.207	0.026	0.208	0.011	0.551	0.208	0.195	0.066	0.069	0.013	0	0	0	0
GLASSO_Label Propagation	0.11	0.013	0.137	0.02	0.228	0.11	0.11	0.007	0.007	0.002	0.004	0.006	0	0
GLASSO_Leading Eigenvalue	0.019	0.065	0.119	0.195	0.142	0.018	0.048	0.227	0.127	0.006	0.197	0.325	0.055	0.206
GLASSO_Louvain	0.028	0.092	0.155	0.234	0.11	0.027	0.077	0.206	0.176	0.002	0.205	0.326	0.059	0.205
GLASSO_Spinglass	0.011	0.045	0.113	0.081	0.165	0.009	0.031	0.126	0.197	0.009	0.089	0.117	0.013	0.029
GLASSO_Walktrap	0.011	0.015	0.068	0.047	0.225	0.012	0.04	0.011	0.004	0.002	0.286	0.369	0.012	0.006
BIC_Edge Betweenness	0.03	0.132	0.133	0.314	0.001	0.027	0.027	0.054	0.238	0.022	0.031	0.048	0.005	0.013
BIC_Fast−greedy	0.023	0.15	0.136	0.384	0.189	0.022	0.064	0.003	0.251	0.029	0.256	0.317	0.082	0.041
BIC_Infomap	0.044	0.079	0.187	0.186	0.246	0.042	0.001	0.006	0.027	0.009	0	0	0	0
BIC_Label Propagation	0.017	0.114	0.133	0.242	0.095	0.015	0.002	0.001	0.094	0.008	0	0.002	0	0
BIC_Leading Eigenvalue	0.017	0.174	0.128	0.378	0.032	0.015	0.028	0.054	0.335	0.04	0.066	0.1	0.01	0.021
BIC_Louvain	0.019	0.177	0.133	0.401	0.024	0.017	0.031	0.057	0.383	0.042	0.065	0.098	0.012	0.022
BIC_Spinglass	0.007	0.097	0.05	0.174	0.201	0.009	0.002	0.028	0.399	0.012	0	0.035	0.003	0.006
BIC_Walktrap	0.017	0.139	0.115	0.332	0.101	0.016	0.039	0.002	0.205	0.029	0.149	0.19	0.032	0.015
AIC_Edge Betweenness	0.021	0.077	0.04	0.148	0.081	0.019	0.023	0.023	0.165	0.005	0.026	0.05	0.006	0.028
AIC_Fast−greedy	0.028	0.123	0.059	0.262	0.234	0.026	0.069	0.006	0.189	0.019	0.345	0.453	0.077	0.022
AIC_Infomap	0.038	0.006	0.081	0.008	0.734	0.035	0.058	0.003	0.004	0.004	0	0	0	0
AIC_Label Propagation	0.039	0.039	0.053	0.042	0.282	0.034	0.057	0.003	0.022	0.009	0.004	0.012	0	0
AIC_Leading Eigenvalue	0.017	0.148	0.037	0.216	0.045	0.014	0.028	0.078	0.266	0.031	0.086	0.173	0.014	0.076
AIC_Louvain	0.023	0.163	0.059	0.3	0.017	0.021	0.045	0.064	0.362	0.021	0.087	0.17	0.018	0.077
AIC_Spinglass	0.013	0.107	0.056	0.192	0.025	0.011	0.015	0.034	0.31	0.019	0.086	0.123	0.009	0.021
AIC_Walktrap	0.017	0.095	0.04	0.154	0.117	0.016	0.027	0.002	0.084	0.026	0.203	0.297	0.016	0.007
Parallel Analysis_PAF	0.001	0.006	0.342	0.007	0.024	0.001	0.01	0.017	0.264	0.319	0	0	0	0
Parallel Analysis_PCA	0.141	0.006	0.031	0.038	0.097	0.142	0.023	0.077	0.085	0.006	0	0	0	0

Figure 6: All large ANOVA effects for percent across all method-algorithm combinations in unidimensional and bidimensional conditons.