

Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis:

A tutorial

Alexander P. Christensen^{a,1} & Hudson F. Golino^{b,1}

^a University of North Carolina at Greensboro

^b University of Virginia

¹ A.P.C. and H.F.G. contributed equally to this work.

Author Note

Correspondence concerning this article should be addressed to Alexander P. Christensen, Department of Psychology, P.O. Box 26170, University of North Carolina at Greensboro, Greensboro, NC, 27402-6170, USA. E-mail: apchrist@uncg.edu

Abstract

Exploratory graph analysis (EGA) is a recent addition to the growing field of network psychometrics. EGA has emerged as a popular approach for estimating the dimensionality of data in networks. The appeal of EGA is the visualization of the relations between variables and the deterministic allocation of variables into dimensions. Notably, networks tend to be sample-specific, making reproducibility and generalizability a key issue in network psychometrics. To resolve this issue, we've developed a novel bootstrap approach called, Bootstrap Exploratory Graph Analysis (bootEGA). bootEGA provides researchers with dimension and item stability statistics as well as item analyses that are akin to exploratory factor analysis loadings. We provide descriptions of bootEGA's functions accompanied by a step-by-step R tutorial for how to apply and interpret bootEGA's results. This tutorial is applied to real-world data to demonstrate its effectiveness at identifying problematic dimensions and items. In short, our results show that bootEGA is a robust approach for identifying the stability and robustness of dimensionality in data.

Keywords: exploratory graph analysis, stability, psychometric networks, dimensionality

Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis:

A tutorial

Exploratory graph analysis (**EGA**; Golino & Epskamp, 2017) has emerged as a powerful approach for identifying dimensions in multidimensional data. The appeal of using exploratory graph analysis, rather than traditional dimension reduction methods (e.g., exploratory factor analysis; EFA, cluster analysis, and parallel analysis; PA), is that the relations between items and dimensions are visualized via a network plot, making its results immediately interpretable (Bringmann & Eronen, 2018). In addition, the approach is data-driven, meaning that the dimensional structure is identified without the researcher's direction. These advantages have led researchers to reanalyze factor structures that were previously defined by traditional methods (Bell & O'Driscoll, 2018; Forkmann, Teismann, Stenzel, Glaesmer, & De Beurs, 2018; Golino & Demetriou, 2017) and redefine the taxonomy of constructs by examining multiple self-report scales (Christensen, Cotter, & Silvia, 2018a) or including behavioral and self-report variables in a single network (Costantini & Perugini, 2016). Despite these advantages, there are still areas where exploratory graph analysis, and psychometric network analysis more generally, can be improved.

The reproducibility of psychometric networks is an area that has received a lot of attention in the literature (Borsboom, Robinaugh, Group, Rhemtulla, & Cramer, 2018; Christensen, Kenett, Aste, Silvia, & Kwapil, 2018b; Forbes, Wright, Markon, & Krueger, 2017; Fried & Cramer, 2017). This issue arises from the sample-specificity of network models, which has led to a lack of stability in certain network measures (Christensen et al., 2018b; Epskamp, Borsboom, & Fried, 2018). For dimensionality, this sample-specificity may lead to dimensional structures that may not generalize to other samples, potentially leading to inconsistencies in the number and content of factors. In the present paper, we introduce a novel approach, *bootstrap exploratory graph analysis* (**bootEGA**), to estimate and evaluate the stability of dimensions identified by exploratory graph analysis. This approach also allows for more nuanced evaluations of the dimensionality of items (akin to cross-loadings in

EFA), allowing researchers to make more informed decisions about the dimensional structure of their data.

The current paper is organized as follows. In the first section, the EGA approach is briefly introduced. In the second section, we provide an R tutorial that introduces the bootEGA technique with descriptions of three broad applications: dimension stability, item stability and item analysis. After each description, we provide R code that demonstrates how bootEGA can be applied and interpreted in real-world psychological data, using Bell and O’Driscoll’s (2018) reanalysis of Bebbington et al.’s (2013) factor structure of paranoia as an example. The last section summarizes the approach and hints at future directions. A supplementary section presents a simulation study investigating the effectiveness of bootEGA in recovering the number of simulated factors, compared to EGA, and exploring the conditions in which the estimated number of factors are less stable. It is important to note that the current paper was written following a **fully reproducible strategy** and both the text and codes used are integrated into a single *Rmarkdown* file. The reader interested in replicating the findings and analysis presented below are encouraged to access the Open Science Framework page available in the methods section.

Exploratory Graph Analysis

Exploratory graph analysis is based on the estimation of a network model followed by the application of a community detection algorithm (Fortunato, 2010; Yang, Algesheimer, & Tessone, 2016). Network models are depicted by nodes (e.g., circles) which represent variables and edges (e.g., lines) which represent the relation (e.g., correlations) between two nodes. The main network estimation method is to construct a Gaussian Graphical Model (**GGM**; Lauritzen, 1996), where edges are the partial correlation between two nodes given all other nodes in the network. The GGM is estimated using the graphical least absolute shrinkage and selection operator (**GLASSO**; Epskamp, Waldorp, Möttus, & Borsboom, 2018), which penalizes and shrinks coefficients (with small coefficients going to zero),

providing parsimony and protection against overfitting. In psychometric networks, the GLASSO also applies the extended Bayesian information criterion (**EBIC**; Chen & Chen, 2008) to select the best fitting model (**GeLASSO**; Epskamp & Fried, 2018; Friedman, Hastie, & Tibshirani, 2008). In these networks, clusters (or sets of many connected nodes) often emerge, forming distinct *communities* or sub-components of uniquely related nodes. These communities are shown to be equivalent to latent factors (Golino & Epskamp, 2017; Golino et al., 2018).

More recently, the triangulated maximally filtered graph (**TMFG**; Massara, Di Matteo, & Aste, 2016) has been applied as an alternative network estimation method in exploratory graph analysis (Christensen et al., 2018a; Golino et al., 2018). The TMFG method uses a structural constraint that limits the number of zero-order correlations included in the network ($3n - 6$; where n is the number of variables). The TMFG algorithm begins by identifying four variables which have the largest sum of correlations to all other variables. Then, it iteratively adds each variable with the largest sum of three correlations to nodes already in the network until all variables have been added to the network. The end result is a network of 3- and 4-node cliques (i.e., a set of connected nodes), which form the constituent elements of an emergent hierarchy in the network (Song, Di Matteo, & Aste, 2012). Of note, this structure can be associated with the inverse covariance (or precision) matrix through a method called the Local/Global inversion (**LoGo**; Barfuss, Massara, Di Matteo, & Aste, 2016), yielding a GGM. Hereafter, EGA will refer to exploratory graph analysis with GeLASSO estimation and EGAtmfg will refer to exploratory graph analysis with TMFG estimation.

To estimate the underlying dimensions in network models, exploratory graph analysis uses the **walktrap** community detection algorithm, which estimates the number and content of a network’s communities by using “random walks” over the network (Pons & Latapy, 2006). These random walks iteratively traverse over neighboring edges, with larger edge weights being more probable paths of travel. Each node is repeatedly used as a starting

point where *steps*—a jump from one node over an edge to another—are taken away from that node, forming a community boundary. A node’s community is then determined by its proportion of many, densely connected edges to few, sparsely connected edges. The algorithm is a deterministic, meaning the number and content of the communities are discovered without the researcher’s guidance. In exploratory graph analysis, these results are depicted in an easy-to-read network plot, where items in each dimension are color-coded.

Despite exploratory graph analysis’s accuracy and effectiveness in simulated (Golino & Demetriou, 2017; Golino & Epskamp, 2017; Golino et al., 2018) and real-world (Bebbington et al., 2013; Christensen et al., 2018a; Forkmann et al., 2018) data, there has yet to be an approach to examine the stability of EGA’s identification of dimensions. On the one hand, the number of dimensions identified may vary depending upon the sample being measured or the size of the sample. On the other hand, the number of dimensions might be consistent across samples but some items may be identified in one dimension in one sample and in another dimension in a different sample. These results may occur for many reasons. For example, items sorting into different dimensions may be an issue of sample size or hint that the item is multidimensional—that is, highly connected to two or more dimensions. Therefore, an approach that is able to assess the quality of EGA’s results is necessary and would allow researchers to examine how their results can be expected to generalize to other samples. Such an approach would not only lead to more accurate interpretations but also provide researchers with greater confidence in their results.

Method

Bell and O’Driscoll (2018) Dataset

To provide a tutorial of the approach, we examined a previously published dataset that has already been analyzed using EGA. Bell and O’Driscoll (2018) analyzed the factor structure of paranoia in a large sample ($n = 8576$) that had been collected in 2000 via the British National Psychiatric Morbidity Survey (Singleton, Bumpstead, O’Brien, Lee, &

Meltzer, 2003). The dataset contains 13 items related to paranoia that were assessed using the Psychosis Screening Questionnaire (PSQ; 2 items) and the Structured Clinical Interview for DSM-IV Axis II Disorders (SCID-II; 11 items; First, Benjamin, Gibbon, Spitzer, & Williams, 1997). The dataset is freely available from the UK Data Service, pending application (Office for National Statistics, 2003). The R script to preprocess the dataset has been made available from Bell and O’Driscoll through the Open Science Framework:

<https://osf.io/b9ngx>.

This dataset represents an optimal case because the sample is very large, so the results are expected to be stable. To demonstrate the full potential of the bootEGA and its functions, the original sample analyzed by Bell and O’Driscoll (2018) was randomly downsampled to 1000 observations. This strategy allows the items and dimensions to become less stable, and will help the reader better understand how the dimension stability and item analyses functions can be used.

Data Analyses

We used R (Version 3.4.4; R Core Team, 2019) and the R-packages *knitcitations* (Version 1.0.8; Boettiger, 2017), and *papaja* (Version 0.1.0.9709; Aust & Barth, 2018) for all of our analyses. The functions for all analyses used in this paper are housed in the *EGA* package (Golino, 2019). *EGA* uses the packages *ggplot2* (Wickham, 2016) and *ggraph* (Kassambara, 2018) for plots and *qgraph* for network visualization (Epskamp, Cramer, Waldorp, Schmittmann, & Borsboom, 2012). Network estimation methods are implemented via the packages *qgraph* (GeLASSO) and *NetworkToolbox* (TMFG and LoGo; Christensen, in press). Finally, the walktrap algorithm is implemented using the *igraph* package (Csardi & Nepusz, 2006). The *Rmarkdown* file containing both the text and the codes used in the current paper can be found here: <https://osf.io/wxdk7/>.

The *EGA* package can be installed in R using the *devtools* package (Wickham & Chang, 2017), since the source code is available on GitHub. To install *EGA*, the *devtools*

package needs to be loaded, and then the *EGA* package can be installed using the `install_github` function:

```
library("devtools")  
devtools::install_github('hfgolino/EGA')
```

Tutorial

After the package is installed, it can be loaded and the **EGA** function can be applied to the data. A tutorial explaining several of **EGA**'s functions and their arguments is provided elsewhere (Golino et al., 2018) and therefore will not be provide here. To begin, exploratory graph analysis with GeLASSO network estimation can be applied to Bell and O'Driscoll's (2018) dataset:

```
library("EGA")  
ega.bell <- EGA(data = paranoia_net, model = "glasso")
```

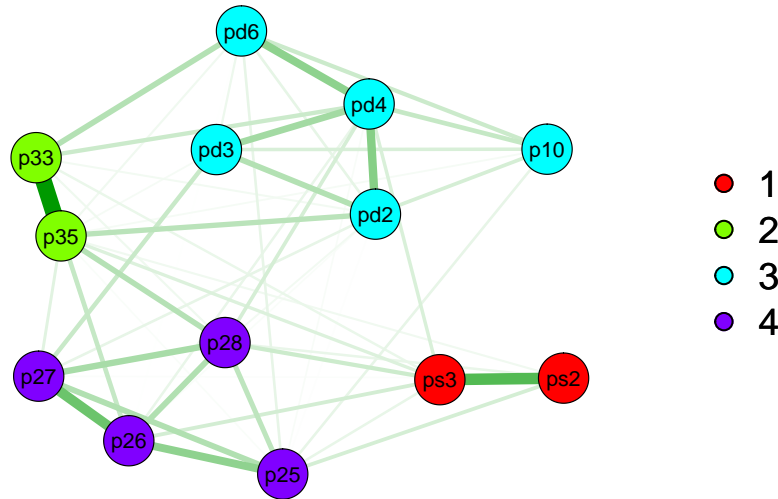



Figure 1. EGA network of the paranoia sample from Bell and O’Driscoll’s (2018) paper. Dimension 1 (red) corresponds to ideas of persecution, dimension 2 (green) corresponds to ideas of references, dimension 3 (cyan) correspond to interpersonal sensitivities, and dimension 4 (violet) corresponds to mistrust.

As Figure 1 demonstrates, the results of the EGA identified 4 dimensions, reproducing Bell and O’Driscoll’s (2018) results. Although the EGA algorithm is deterministic, it’s important to investigate how stable and generalizeable these results are. To do so, bootEGA can be applied.

Bootstrap Exploratory Graph Analysis

bootEGA has two main functions: dimension stability and item analysis, which are used to evaluate the stability of a network’s dimensions and the robustness of each item’s placement within those dimensions, respectively. The bootEGA algorithm uses two data generation approaches: parametric and non-parametric. The *parametric* procedure begins by estimating a GGM using the GeLASSO or LoGo network estimation method. Using the covariance matrix given by the precision matrix, data with the same number of cases as the original sample is generated from a multivariate normal distribution. A new network is

estimated from this generated dataset and the walktrap algorithm is applied. The *non-parametric* procedure begins by resampling with replacement from the original dataset and draws the same number of cases as the original dataset. This new dataset is then used to estimate a network and the walktrap algorithm is applied. Both procedures continue iteratively until the desired number of samples is achieved (e.g., 1000 samples). The result is a large number of replica networks, which form a sampling distribution of networks that have been analyzed with the walktrap algorithm. These results are stored so dimension stability and item analysis functions can be computed.

bootEGA Arguments. The arguments for **bootEGA** are shown below:

```
bootEGA(data, n, model = c("glasso", "TMFG"), type = c("parametric",  
  "resampling"), typicalStructure = TRUE, plot.typicalStructure = TRUE,  
  ncores = 4, ...)
```

data indicates the dataset that should be used in the analysis. The dataset being used should only contain the variables of interest that were used in the **EGA** function. **n** is the number of iterations or replica networks that should be bootstrapped. **model** is the network estimation method that should be used, which defaults to "glasso". The **type** argument determines the type of bootstrap analysis that should be used. "parametric" is for the parametric bootstrap and "resampling" is for the non-parametric bootstrap.

typicalStructure is whether the median network structure should be computed (defaults to TRUE) and **plot.typicalStructure** is whether the median network should be plotted (defaults to TRUE). **ncores** is for parallel processing, speeding up the analysis. The function defaults to 4 but researchers should assess their own computer's capabilities before settling on the default. Finally, the **...** argument is for additional arguments to be passed to the network estimation methods (links to each network estimation function are included in **bootEGA**'s documentation).

Application. For the purposes of our tutorial, we'll apply bootEGA with GeLASSO using the non-parametric bootstrap (i.e., sampling with replacement) for 1,000 iterations:

```
set.seed(0)

boot.bell <- bootEGA(data = paranoia_net, model = "glasso",
                    n = 1000, type = "resampling",
                    plot.typicalStructure = FALSE)
```

bootEGA outputs a list containing several objects: `n`, `boot.ndim`, `boot.wc`, `bootGraphs`, `summary.table`, `likelihood`, `EGA`, and `typicalGraph`. `n` is the number of replica samples in the bootstrap, `boot.ndim` is the number of dimensions identified in each replica sample, `boot.wc` is each item's dimension allocation in each replica sample, and `bootGraphs` is the network of each replica sample. `EGA` is a list containing the output of the original EGA function's results. Similarly, `typicalGraph` is a list containing the median network (`graph`), an ordered matrix of each item's dimension allocation (`typical.dim.variables`), and each item's dimension allocation in the median network (`wc`). The final two objects, `summary.table` and `likelihood`, are the dimension stability statistics, which we'll examine here further.

Dimension Stability

The dimension stability function of bootEGA provides several descriptive statistics about the distribution of the number of dimensions identified. The median, standard error, and 95% confidence intervals are calculated across the replica networks, yielding a summary of the bootEGA dimension results. Moreover, the *likelihood* or proportion of times a certain number of dimensions appeared in the replica networks is also computed to provide additional information about the distribution of dimensions. An additional feature is the typical network structure, which is defined by the median edge weight for every possible node combination across the replica networks. Therefore, this network is the median network

of the distribution of replica networks. The median network is analyzed using the walktrap algorithm and represents the most likely and stable representation of the dimensional structure in the data. Indeed, our simulation demonstrates that the median network performs comparable or better than the standard EGA approach (see Supplementary Information).

Application. First, the descriptive statistics can be examined using the following R code:

```
boot.bell$summary.table
```

`summary.table` provides descriptive statistics for the number of bootstraps (`n.Boots`), median number of dimensions found across the bootstraps (`median.dim`), standard deviation (`SD.dim`), standard error (`SE.dim`), and lower (`Lower`) and upper (`Upper`) bound of the 95% confidence interval of the bootstrapped dimensions.

Table 1

bootEGA Summary Table

n.Boots	median.dim	SD.dim	SE.dim	CI.dim	Lower	Upper
1000	4	0.1332733	0.0052807	0.0103626	3.989637	4.010363

Based on the descriptive statistics (Table 1), we can see that the 4 dimensions identified by Bell and O’Driscoll (2018) are very stable (median = 4, SD = 0.13, and 95% CI [3.99,4.01]). The likelihood of dimensions provides more information into this stability by providing the distribution of the proportion of times that a certain number of dimensions was replicated. This can be examined using the following R code:

Table 2

bootEGA dimensions likelihood

# of Factors	Likelihood
2	0.001
3	0.014
4	0.985

`boot.bell$likelihood`

With `likelihood`, we can see that 4 factors was replicated the most (0.98 or 985 times of 1000 samples; Table 2). Notably, the factors ranged between 2 and 4, but the other factors were rarely identified. This analysis confirms that 4 factors are the most stable dimensional organization of the data.

One other noteworthy analysis is to compare the original (`EGA`) and median (`bootEGA`) network results:

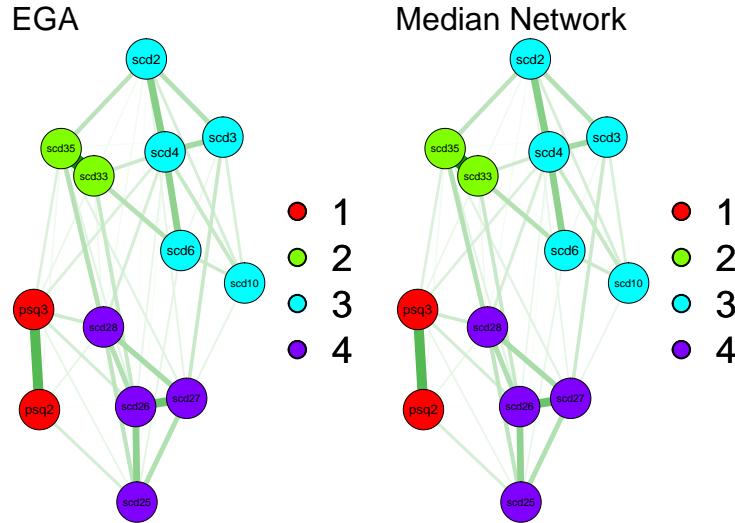


Figure 2. EGA (left) and bootEGA’s median network (right) of the paranoia sample from Bell and O’Driscoll’s (2018) paper. For both networks, dimension 1 (red) corresponds to ideas of persecution, dimension 2 (green) corresponds to ideas of references, dimension 3 (cyan) correspond to interpersonal sensitivities, and dimension 4 (violet) corresponds to mistrust.

As Figure 2 demonstrates, the results of the original EGA are identical to the median network results, providing additional support for stability of the dimensions identified by the original EGA. In short, these results demonstrate that Bell and O’Driscoll’s (2018) are sufficiently stable and are likely to generalize to other samples. Moreover, further analyses of item stability and analysis would not be warranted given these results. For demonstrative purposes, however, the original sample analyzed by Bell and O’Driscoll (2018) was randomly downsampled to 1000 observations using the following R code:

```
set.seed(9324)
ids <- sample(nrow(paranoia_net),1000)
random.sample <- paranoia_net[ids,]
```

With this random downsample, we can perform the EGA and bootEGA analyses again.

```
ega.rand <- EGA(data = random.sample, model = "glasso")
boot.rand <- bootEGA(data = random.sample, n = 1000,
                     model = "glasso", type = "resampling")
```

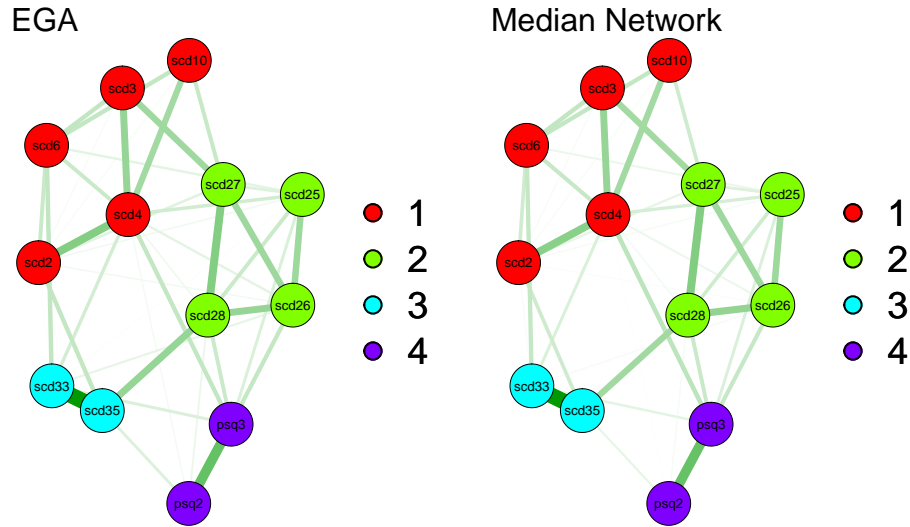


Figure 3. EGA (left) and bootEGA's median network (right) of the random downsample from Bell and O'Driscoll's (2018) paper. For both networks, dimension 1 (red) corresponds to interpersonal sensitivities, dimension 2 (green) corresponds to mistrust, dimension 3 (cyan) correspond to ideas of references, and dimension 4 (violet) corresponds to ideas of persecution.

Once again, the **EGA** results returned 4 dimensions, which was replicated by the median network provided by **bootEGA**. Looking deeper into the summary table, however, we find that although the median number of dimensions estimated remains the same (four), the confidence interval increased from 0.01 to 0.05 (Table 3).

Table 3

bootEGA summary table (random subsample, $n = 1000$)

n.Boots	median.dim	SD.dim	SE.dim	CI.dim	Lower	Upper
1000	4	0.593956	0.0235345	0.0461828	3.953817	4.046183

This is corroborated by the likelihood distribution of dimensions (Table 4). Although 4 dimensions is still the most likely, there is a drastic drop in replication percentage (from 0.98 to 0.58).

Table 4

bootEGA dimensions likelihood (random subsample, $n = 1000$)

# of Factors	Likelihood
2	0.034
3	0.366
4	0.579
5	0.021

Using the item analysis functions of `bootEGA`, we can probe further into which items might be causing this instability.

Item stability

The item analyses function of `bootEGA` provides an item stability measure and item statistics measure (akin to factor loadings in EFA). These analyses require dimensions to be specified (so-called *confirmatory dimensions*), typically using exploratory graph analysis, which are then used to “confirm” each item’s presence in a certain dimension. The item stability measure takes each item and computes how often they appear in each possible dimension across the replica network. The algorithm begins by identifying the largest portion of items that replicate in each confirmatory dimension using the Rand index (eq. 1; Rand, 1971).

$$R = \frac{a + b}{a + b + c + d} \quad (1)$$

a is the number of items from the confirmatory dimension that are the same in the replica network's dimension, b is the number of items that are not included in both, c is the number of items that are in the confirmatory dimension but not the replica network's dimension, and d is just the opposite of c .

The replica network dimension with the largest Rand index is assigned the same dimension as the corresponding confirmatory dimension, then the second largest Rand index with the corresponding confirmatory dimension, and so on. This is done to ensure that the replica network's dimensions achieve maximum similarity to confirmatory dimensions. On the one hand, some replica networks may have fewer dimensions than the confirmatory dimensions. In these instances, the confirmatory dimension that has the most items in the corresponding replica network's dimension is assigned. On the other hand, some replica networks may have more dimensions than the confirmatory dimensions. In these instances, after the confirmatory dimensions are identified in the replica network's dimensions, the additional dimensions are iteratively assigned based on the largest number of items in each of the remaining dimensions of the replica network (e.g., dimension 3 with 8 items, dimension 4 with 4 items, and dimension 5 with 2 items). Once completed, a proportion for each item in each dimension is computed, representing the stability of that item being identified in each dimension.

A plot is also produced displaying the proportion of times that an item is replicated in the confirmatory dimension. Items with high proportions (e.g., $\geq .80$) suggest that the item is stable and consistently identified in the confirmatory dimension. Items with relatively low proportions (e.g., $< .80$) suggest that the item may be multidimensional and potentially problematic. Low stability items provide inferences into why certain dimensions may not be as stable as others. Because the proportions for all dimensions are provided, items with low stability in their confirmatory dimension can be identified and the appropriate dimension can be determined. Some items, for example, may have a higher replication proportion in another confirmatory dimension, suggesting they would be a better fit in that dimension.

Other items may replicate equally well between two or more dimensions, including a dimension not included in the confirmatory dimension. These items might suggest multidimensionality and likely a poor fit within the dimensional structure of the data.

itemConfirm Arguments. `itemConfirm` has four arguments. The first, `bootega.obj`, is a `bootEGA` object, which is the variable where the `bootEGA` analyses is stored (in our example: `boot.bell`). The next argument is `confirm`, which is an input for the dimension membership of each item. For our example, we will use the results from the original `EGA` analysis (`ega$wc`). The `item.rep` argument is used to filter out values below a certain replication proportion (i.e., the number of times an item replicates in a specific dimension) in the table output by `itemConfirm`. `item.rep` defaults to 0.1 or 100 replications out of 1000 samples. This value can be lowered to 0 to see the full results. Finally, the last argument, `plot.ic`, is whether the plot of `itemConfirm`'s results should be produced. The plot illustrates the replication proportion of the `confirm` item dimension allocations only. This is to provide a quick and easy read of how well items are replicating in their defined dimension.

```
itemConfirm(bootega.obj, confirm, item.rep = 0.1, plot.ic = TRUE)
```

`itemConfirm` outputs a list containing the objects: `item.confirm`, `item.likelihood`, and `wc`. The object, `item.confirm`, is a vector representing the proportion of times each item is replicated within the defined dimension (also given in the plot). `item.likelihood` is the proportion of times each item replicated within each possible dimension. Values less than `item.rep` will appear as blank entries in this matrix and values of 0 will always appear blank in this matrix.

Application. This R code can be used to run the analysis:

```
ic.bell <- itemConfirm(boot.rand, confirm = ega.rand$wc,  
                      item.rep = 0.1, plot.ic = TRUE)
```

The first step is to examine the plot to determine whether items are replicating in their specified dimensions (Figure 4). Below is the plot and R code used to access it:

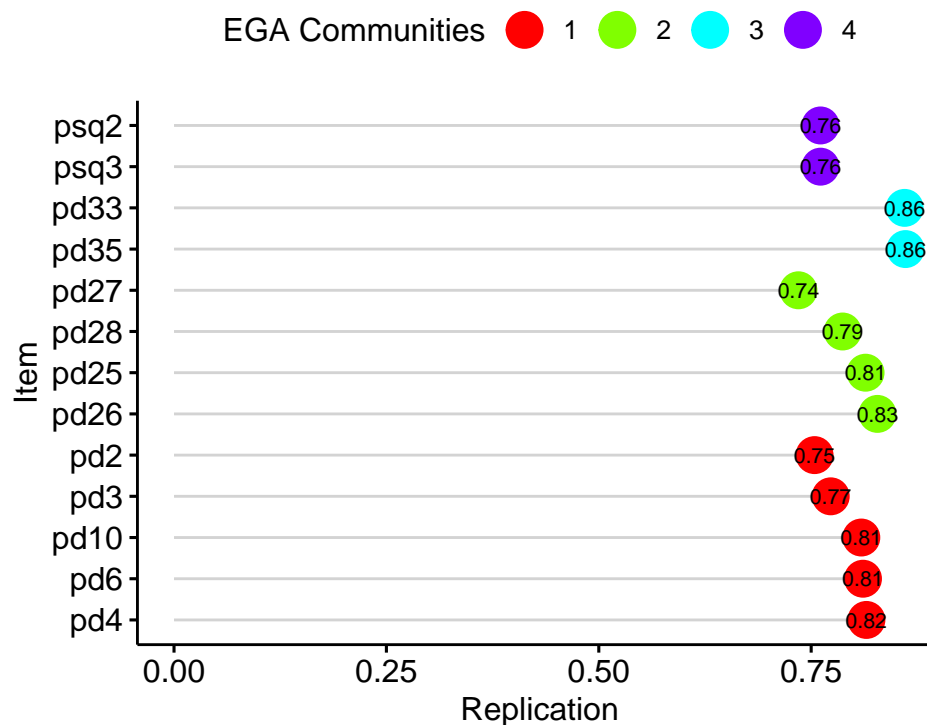


Figure 4. Item replication proportions for the specified dimensions in the itemConfirm function. Colors correspond to the original color-code and dimensions identified by the EGA function (random subsample)

```
ic.bell$ic.plot
```

The results from the plot suggest that nearly all items are replicating in their designated dimensions (i.e., `ega.rand$wc`) about 75% of the time. In general, these results are satisfactory, but it's helpful to view the replication proportions in the context of all dimensions (Table 5). This table can be generated by using the following R code:

Table 5

bootEGA item likelihood (random subsample)

	1	2	3	4
psq2		0.12	0.113	0.761
psq3		0.149		0.761
pd33			0.86	
pd35			0.861	
pd27	0.199	0.735		
pd28		0.787		0.129
pd25		0.814		
pd26		0.828		
pd2	0.754	0.106	0.126	
pd3	0.773	0.141		
pd10	0.809	0.112		
pd6	0.811		0.101	
pd4	0.815			

```
ic.bell$item.likelihood
```

Table 5 reveals a more nuanced picture of which dimensions certain items are replicating in more. The lowest within-dimension stability item, *pd27*, is replicating in a different dimension (i.e., dimension 1) about 19.90% of the time. Other notable items include: *psq2* and *psq3*. Based on their overlap in replication proportions in dimension 2, it's probable that these items are being allocated to that dimension and leading to the 3 dimension solution seen in the dimension likelihood result (Table 4).

Item statistics

The item statistics function provides further inferences into which confirmatory dimension each item may belong to. Across the replica networks, node strength (i.e., the sum of the connections to a node) is computed for each item's connections within its specified dimension (i.e., replica-specific dimension) and its connections to every other dimension (e.g., Blanken et al., 2018). Each item's strength in each dimension is then averaged across all replica networks. Items with a greater average of strength within their specified dimension suggests that they are relatively stable and most strongly associate with their assigned dimension. Items with a greater proportion of strength in a dimension different from their specified dimension suggests that they may belong to another dimension. There may also be cases where items are equally split between two dimensions, reflecting that the item is likely multidimensional. In these instances, a researcher may want to investigate this item further to determine whether the item is problematic and should be removed from dimensional analyses.

itemIdent Arguments. The final function is to examine how strong an item's connections are within and between dimensions, computed using the `itemIdent` function. Node strength has been shown to be equivalent to EFA factor loadings; however, it typically represents a combination of dominant and cross-dimension loadings (Hallquist, Wright, & Molenaar, 2019). In this function, a node's strength is separated by its within- and cross-dimension connections. Thus, the result is a matrix akin to EGA factor loadings. `itemIdent` has four arguments. The first, `bootega.obj`, is a `bootEGA` object, which is the variable where the bootEGA analyses is stored (in our example: `boot.bell`). The next argument is `confirm`, which is an input for the dimension membership of each item (e.g., `ega$wc`). The `rep.val` argument represents the replication value for what the cut-off should be to identify item instability. `rep.val` should be set between 0 and 1. For example, suppose two variables in a dataset replicates only in 450 out of 1000 bootstraps. If the `rep.val` argument is set to .45, the resulting table will flag these variables with an **X** in the

last column. Thus, this argument can be used to quickly identify problematic items. Finally, the last argument, `item.rep`, is the value for lowest likelihood allowed in `itemConfirm`'s `item.likelihood` output. This argument removes noise from the output table to allow an easier interpretation.

```
itemIdent(bootega.obj, confirm, rep.val = 0.8, item.rep = 0.1)
```

`itemIdent` outputs a matrix which contains the node strength of each item within each dimension.

Application. The following R code can be used to store and produce the results:

```
itemIdent(boot.rand, confirm = ega.rand$wc,  
          rep.val = 0.80, item.rep = 0.1)
```

Table 6

Node strength for each item in each dimension (random subsample)

	Dimension	1	2	3	4	Rep<=0.8
psq2	4		0.114	0.116	0.375	X
psq3	4		0.371		0.414	X
pd33	3			0.584		
pd35	3			0.582		
pd27	2	0.379	0.587			X
pd28	2		0.611		0.222	X
pd25	2		0.416			
pd26	2		0.639			
pd2	1	0.463	0.179	0.189		X
pd3	1	0.452	0.227			X
pd10	1	0.401	0.151			
pd6	1	0.49		0.179		
pd4	1	0.778				

The output of `itemIdent` (Table 6) shows the node strength of each item in each dimension. When split up this way, these values can be interpreted similarly to EFA factor loadings (Hallquist et al., 2019). A notable difference, however, is that these values are always positive, while EFA factor loadings can be negative. This key difference should be taken into consideration when interpreting this table. When looking at some of the less stable items mentioned before, *pd27* and *psq3* have the largest strength (0.38 and 0.37, respectively) in dimensions outside of their designation (dimension 1 and 2, respectively). This suggests that these items are more likely to be pulled into other dimensions. Furthermore, it suggests at the potential for these items to potentially be multidimensional;

however, we know from the larger sample that they are stable within these dimensions.

Summary

The bootEGA approach offers researchers a tool to analyze the stability and generalizability of their networks. Within this approach, there are functions aimed at determining the stability of network dimensions as well as item-level analyses that are able to identify potential problematic or multidimensional items. This approach opens up new avenues for researchers interested in using psychometric network analysis for scale development and psychometric assessment evaluations. This tutorial provides all the knowledge necessary for researchers, including those unfamiliar with R, to feasibly apply the bootEGA approach to their own data.

References

- Aust, F., & Barth, M. (2018). *papaja: Create APA manuscripts with R Markdown*. Retrieved from <https://github.com/crsh/papaja>
- Barfuss, W., Massara, G. P., Di Matteo, T., & Aste, T. (2016). Parsimonious modeling with information filtering networks. *Physical Review E*, *94*, 062306. <https://doi.org/10.1103/PhysRevE.94.062306>
- Bebbington, P. E., McBride, O., Steel, C., Kuipers, E., Radovanovic, M., Brugha, T., . . . Freeman, D. (2013). The structure of paranoia in the general population. *The British Journal of Psychiatry*, 419–427. <https://doi.org/10.1192/bjp.bp.112.119032>
- Bell, V., & O’Driscoll, C. (2018). The network structure of paranoia in the general population. *Social Psychiatry and Psychiatric Epidemiology*, *53*, 1–8. <https://doi.org/10.1007/s00127-018-1487-0>
- Blanken, T. F., Deserno, M. K., Dalege, J., Borsboom, D., Blanken, P., Kerkhof, G. A., & Cramer, A. O. (2018). The role of stabilizing and communicating symptoms given overlapping communities in psychopathology networks. *Scientific Reports*, *8*, 5854. <https://doi.org/10.1038/s41598-018-24224-2>
- Boettiger, C. (2017). *knitcitations: Citations for ‘knitr’ markdown files*. Retrieved from <https://CRAN.R-project.org/package=knitcitations>
- Borsboom, D., Robinaugh, D. J., Group, P., Rhemtulla, M., & Cramer, A. O. (2018). Robustness and replicability of psychopathology networks. *World Psychiatry*, *17*, 143–144. <https://doi.org/10.1002/wps.20515>
- Bringmann, L. F., & Eronen, M. I. (2018). Don’t blame the model: Reconsidering the network approach to psychopathology. *Psychological Review*, *125*, 606–615. <https://doi.org/10.1037/rev0000108>
- Chen, J., & Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, *95*(3), 759–771.

<https://doi.org/10.1093/biomet/asn034>

Christensen, A. P. (in press). NetworkToolbox: Methods and measures for brain, cognitive, and psychometric network analysis in R. *R Journal*, 1–18.

<https://doi.org/10.31234/osf.io/6kmav>

Christensen, A. P., Cotter, K. N., & Silvia, P. J. (2018a). Reopening openness to experience: A network analysis of four openness to experience inventories. *Journal of Personality Assessment*, 1–15. <https://doi.org/10.1080/00223891.2018.1467428>

Christensen, A. P., Kenett, Y. N., Aste, T., Silvia, P. J., & Kwapil, T. R. (2018b). Network structure of the Wisconsin Schizotypy Scales–Short Forms: Examining psychometric network filtering approaches. *Behavior Research Methods*, 50, 2531–2550.

<https://doi.org/10.3758/s13428-018-1032-9>

Costantini, G., & Perugini, M. (2016). The network of conscientiousness. *Journal of Research in Personality*, 65, 68–88. <https://doi.org/10.1016/j.jrp.2016.10.003>

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695, 1–9. Retrieved from

<https://pdfs.semanticscholar.org/1d27/44b83519657f5f2610698a8ddd177ced4f5c.pdf>

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., Borsboom, D., & Fried, E. I. (2018). Estimating psychological networks and their accuracy: A tutorial paper. *Behavior Research Methods*, 50, 195–212.

<https://doi.org/10.3758/s13428-017-0862-1>

Epskamp, S., Cramer, A. O., 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., Waldorp, L. J., Möttus, R., & Borsboom, D. (2018). The gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, 1–28.

<https://doi.org/10.1080/00273171.2018.1454823>

First, M. B., Benjamin, L. S., Gibbon, M., Spitzer, R. L., & Williams, J. B. (1997).

Structured clinical interview for dsm-iv axis ii personality disorders.

Forbes, M. K., Wright, A. G., Markon, K. E., & Krueger, R. F. (2017). Evidence that psychopathology symptom networks have limited replicability. *Journal of Abnormal Psychology*, 126, 969–988. <https://doi.org/10.1037/abn0000276>

Forkmann, T., Teismann, T., Stenzel, J.-S., Glaesmer, H., & De Beurs, D. (2018). Defeat and entrapment: More than meets the eye? Applying network analysis to estimate dimensions of highly correlated constructs. *BMC Medical Research Methodology*, 18(1), 16. <https://doi.org/10.1186/s12874-018-0470-5>

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

Fried, E. I., & Cramer, A. O. (2017). Moving forward: Challenges and directions for psychopathological network theory and methodology. *Perspectives on Psychological Science*, 12, 999–1020. <https://doi.org/10.1177/1745691617705892>

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>

Golino, H. F. (2019). *EGA: Exploratory Graph Analysis – estimating the number of dimensions in psychological data*. Retrieved from <https://github.com/hfgolino/EGA>

Golino, H. F., & Demetriou, A. (2017). Estimating the dimensionality of intelligence like data using Exploratory Graph Analysis. *Intelligence*, 62, 54–70. <https://doi.org/10.1016/j.intell.2017.02.007>

Golino, H. F., & 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., Shi, D., Garrido, L. E., Christensen, A. P., Nieto, M. D., Sadana, R., &

- Thiyagarajan, J. A. (2018). Investigating the performance of exploratory graph analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial.
- Hallquist, M., Wright, A. C. G., & Molenaar, P. C. M. (2019). Problems with centrality measures in psychopathology symptom networks: Why network psychometrics cannot escape psychometric theory. *PsyArXiv*, 1–73. <https://doi.org/10.31234/osf.io/pg4mf>
- Kassambara, A. (2018). *ggpubr: 'ggplot2' based publication ready plots*. Retrieved from <https://CRAN.R-project.org/package=ggpubr>
- 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>
- Office for National Statistics. (2003). *Psychiatric morbidity among adults living in private households, 2000*. [data collection]. UK Data Service. SN:4653, <http://doi.org/10.5255/ukda-sn-4653-1>.
- Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, 10(2), 191–218.
- R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336), 846–850.
- Singleton, N., Bumpstead, R., O'Brien, M., Lee, A., & Meltzer, H. (2003). Psychiatric morbidity among adults living in private households, 2000. *International Review of Psychiatry*, 15(1-2), 65–73. <https://doi.org/10.1080/0954026021000045967>
- Song, W.-M., Di Matteo, T., & Aste, T. (2012). Hierarchical information clustering by means of topologically embedded graphs. *PLoS One*, 7, e31929.

<https://doi.org/10.1371/journal.pone.0031929>

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer.

Wickham, H., & Chang, W. (2017). *devtools: Tools to make developing r packages easier*.

Retrieved from <https://CRAN.R-project.org/package=devtools>

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>

Supplementary Information

Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis: A tutorial

SI Method

A Monte-Carlo simulation was implemented to investigate the accuracy of bootEGA in estimating the number of simulated factors, and the conditions where the stability of the estimate is less pronounced. The data generation strategy followed the factor model approach implemented by Golino et al. (2018). In short, the data generation starts by computing the (full-rank) population correlation matrix, then proceeds to perform a Cholesky decomposition of the population matrix. Next, the sample data matrix of continuous variables is computed. Differently from Golino et al. (2018), in the current paper the continuous variables were not dichotomized, since the goal is to check how well does bootEGA recover the number of simulated factors in more ideal scenarios. Since the present paper is the first to introduce the bootEGA technique, it is important to understand which conditions leads to poor estimates not related to the type of data generated. The sample size used in the current simulation was large (5000) for the same reason. Previous simulation studies showed that with large sample sizes the accuracy of the dimensionality assessment techniques are higher (Garrido, Abad, & Ponsoda, 2011, 2013; Golino et al., 2018).

Four variables were systematically controlled: factor loadings (.40, .55 and .70), variables per factor (4 and 6), number of factors (2 and 4), and factor correlations (0.00, 0.30, 0.50 and 0.70). In total, a $3 \times 2 \times 2 \times 4$ (factor loadings, variables per factor, number of factors, factor correlations, method and bootstrap) between-subject design was implemented, resulting in 48 conditions. For each condition, 500 datasets were generated, resulting in 24,000 datasets. The within-subject part of the simulation regarded the bootEGA approach, with two variables: the method (GeLASSO vs. TMFG) and the bootstrap (parametric vs. resampling). The *bootEGA* technique was implemented using 500 bootstraps for each condition. So, in total, 48 million dimensionality estimates were obtained, $48 \times 500 \times 500 \times 4$

(between subject conditions, number of replicates, number of bootstraps).

In order to evaluate the performance of the *bootEGA* algorithm, two complementary criteria was be used: the accuracy or percentage of correct estimates (PC) and the normalized mutual information (NMI; Danon, Diaz-Guilera, Duch, & Arenas (2005)). The accuracy criterion varies between zero (0% correct estimates) and one (100% correct estimates). The normalized mutual information measures how well the estimated assignment of items per factor reflects the simulated structure. A value of one indicates that the estimated factors perfectly reflects the distribution of items per factor in the population, and a value of zero indicates that the items departs completely from the distribution found in the population.

SI Results

In terms of dimensionality assessment, the accuracy of EGA, bootEGA using the dimensionality estimate via the typical network or using the median of the estimated number of dimensions over n bootstraps are almost identical (.968, .968, .967, respectively). However, bootEGA with the GeLASSO method (GGM) using parametric and resampling bootstraps (.98) presents higher percentage correct estimates than and bootEGA with the TMFG method using parametric (.947) and resampling (.952). Small differences were also detected comparing the techniques by size of factor loading and size of the correlation between factors. Although the dimensionality estimates of EGA, bootEGA using the typical structure or using the median number of factors being almost identical for the Gaussian graphical model with GeLASSO regularization (both for the parametric and resampling bootstraps), the bootEGA algorithm (using resampling) improves the dimensionality estimation of the TMFG method, especially when the correlation between factors is high and factor loadings are low or moderate. In these conditions, the bootEGA algorithm is more accurate than EGA, when the TMFG method is used (see Figure SI1).

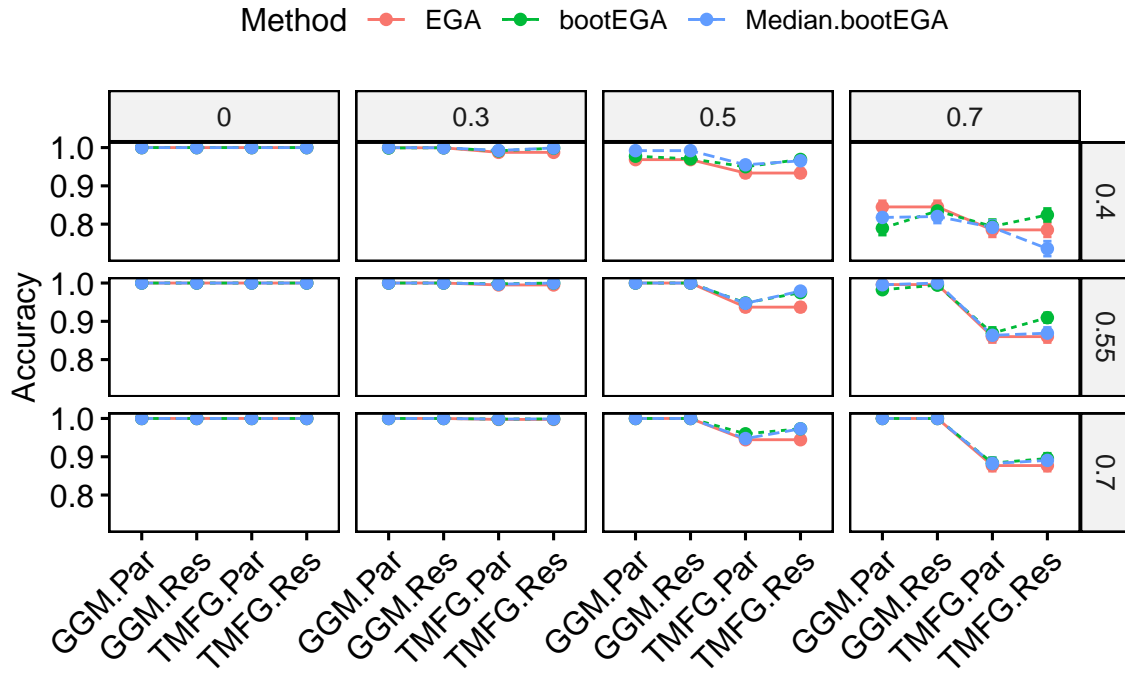


Figure SI1. Accuracy of dimension identification

In terms of the NMI, the pattern is similar. The assignment of items per factor is almost perfect ($N\bar{MI}$ for bootEGA = .99, $N\bar{MI}$ for EGA = .98), except when the correlation between factors is .70 and the factor loadings are .40 (see SI2). In this case, EGA presents a $N\bar{MI}$ of .91 and bootEGA .93.

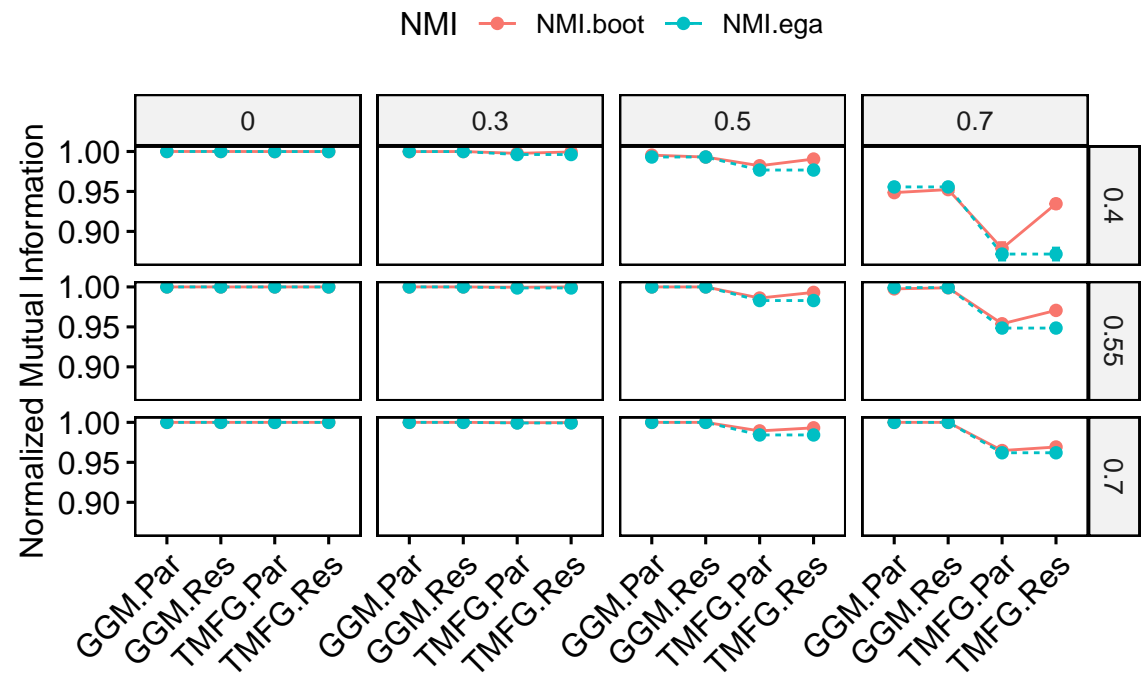


Figure SI2. Normalized Mutual Information of dimension identification

References

Danon, L., Diaz-Guilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005, P09008. <https://doi.org/10.1088/1742-5468/2005/09/P09008>

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>

Golino, H., Shi, D., Garrido, L. E., Christensen, A. P., Nieto, M. D., Sadana, R., & Thiagarajan, J. A. (2018). Investigating the performance of exploratory graph analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial.