NetworkToolbox: Methods and Measures for Brain, Cognitive, and Psychometric Network Analysis in R

by Alexander P. Christensen

Abstract This article introduces the **NetworkToolbox** package for R. Network analysis offers an intuitive perspective on complex phenomena via models depicted by nodes (variables) and edges (correlations). The ability of networks to model complexity has made them the standard approach for modeling the intricate interactions in the brain. Similarly, networks have become an increasingly attractive model for studying the complexity of psychological and psychopathological phenomena. **NetworkToolbox** aims to provide researchers with state-of-the-art methods and measures for estimating and analyzing brain, cognitive, and psychometric networks. In this article, I introduce **NetworkToolbox** and provide a tutorial for applying some the package's functions to personality data.

Introduction

Open science is ushering in a new era of psychology where multi-site collaborations are common and big data are readily available. Often, in these data, noise is mixed in with relevant information. Thus, researchers are faced with a challenge: deciphering information from the noise and maintaining the inherent structure of the data while reducing its complexity. Other areas of science have tackled this challenge with the use of network science and graph theory methods. Psychology is beginning to follow suit, changing the way we conceptualize psychological (Cramer et al., 2012; Schmittmann et al., 2013) and psychopathological phenomena (Borsboom and Cramer, 2013; Borsboom, 2017).

Psychological and psychopathological constructs are intrinsically complex. Over the years, researchers have focused on reducing phenomena to only the most significant variables. This reductionist approach has given us a greater understanding of what variables matter and what variables don't. Nonetheless, the more we reduce, the less we know about the phenomena as a whole (Barabási, 2011). With network science, we can begin to put psychological phenomena back together again, embracing the complexity.

Here, I introduce **NetworkToolbox**, a package for R (Team, 2018), available on the Comprehensive R Archive Network (`https://cran.r-project.org/web/packages/NetworkToolbox/index.html`). **NetworkToolbox** offers various network science methods and measures, which can be applied to brain, cognitive, psychometric and other data suitable for network analysis. Networks are made up of nodes (e.g., circles) that are connected by edges (e.g., lines) to other nodes. Nodes represent variables (e.g., brain regions, words, questionnaire items) and edges represent a relation between two nodes (e.g., correlations, partial correlations), which can be undirected (i.e., bidirectional) or directed, and weighted or unweighted (all weights are equivalent). Thus, networks are conceptually simple yet capable of considerable complexity. In addition, their graphical depictions offer representations that have intuitive interpretations (Epskamp et al., 2012).

There are several network analysis packages that are already implemented in R, such as **statnet** (Goodreau et al., 2008), **igraph** (Csardi and Nepusz, 2006), and **sna** (Butts and others, 2008). Indeed, there are R packages that specifically focus on brain (e.g., **brainGraph**; Watson, 2017) and psychometric (e.g., **qgraph**, Epskamp et al., 2012; **IsingFit**, van Borkulo et al., 2014) networks. **NetworkToolbox** differentiates itself by including state-of-the-art network methods and measures that are not currently available in these other packages. Notably, network visualization aspects are not considered in this package. Where necessary, I refer readers to visualization sources that are compatible with **NetworkToolbox**'s outputs. In general, the R package **qgraph** (Epskamp et al., 2012) is recommended for network visualizations.

Psychometric Network Analysis

In this section, I provide explanations for many methods and measures in **NetworkToolbox** and include associated code implemented in R. The main body of **NetworkToolbox** is devoted to psychometric network analysis; however, several functions, such as network construction methods and network measures, could be applied more generally. To provide examples of functions in **NetworkToolbox**, I will use psychometric data but I will provide basic interpretations, so that measures can be

generalized to other domains. All analyses conducted in this article were performed using R version 3.4.4 (2018-03-15) and **NetworkToolbox** version 1.2.1. Before exploring the methods and measures in **NetworkToolbox**, I will introduce the psychometric data that are used throughout the paper.

Data

The data included in **NetworkToolbox** are centered around the personality trait Openness to Experience scale of the NEO personality inventory (NEO-PI-3, McCrae et al., 2005; NEO-FFI-3, McCrae and Costa Jr, 2007). People high in Openness to Experience are described as *creative*, *curious*, *imaginative*, *unconventional*, and *intellectual*. The NEO Openness to Experience inventories have 6 facets: *actions*, *aesthetics*, *fantasy*, *feelings*, *ideas*, and *values*, representing distinct characterizations of the global trait. **NetworkToolbox** includes psychometric and brain data, which are openly available for researchers to use.

For the psychometric data, I will use data that includes four Openness to Experience inventories (Big Five Aspects Scale, DeYoung et al., 2007; HEXACO-100, Lee and Ashton, 2016; NEO-PI-3, McCrae et al., 2005; Woo et al.'s Openness to Experience Inventory, Woo et al., 2014), which was previously analyzed in Christensen et al. (2018a). For this article, the analyses of these data will be focused on the NEO-PI-3 (48 items) but data for all inventories can be found on the Open Science Framework (`https://osf.io/954a7/`). These data contain a relatively large sample (N = 802), which provides a foundation for some of the more nuanced analyses in **NetworkToolbox**. Below is code to load **NetworkToolbox** and the NEO-PI-3 data in the package.

```
#Load NetworkToolbox
library(NetworkToolbox)

#Load NEO-PI-3 data
data("neoOpen")
```

Network Construction

A popular network construction method in network analysis is the *least absolute selection and shrinkage operator* (LASSO; Tibshirani, 1996). In short, the LASSO approach penalizes the inverse covariance matrix to produce sparse networks whose connections reflect conditional independence (i.e., nodes that are connected are uniquely associated, given all other nodes in the network). Because there are already a number of other R packages that compute LASSO networks (see **bootnet**, Epskamp et al., 2018; **glasso**, Friedman et al., 2014; **IsingFit**, van Borkulo et al., 2014), they are not included in **NetworkToolbox**. **NetworkToolbox** does include several other options for constructing a network.

The main network construction methods within the package are from the *Information Filtering Networks* approach (IFN; Barfuss et al., 2016). This approach constructs networks based on the zero-order correlations between variables and induces parsimony via a topological (structural) constraint. Currently, this family of networks has four methods: the *Triangulated Maximally Filtered Graph* (TMFG; Massara et al., 2016), the *Planar Maximally Filtered Graph* (PMFG; Tumminello et al., 2005), the *Maximum Spanning Tree* (MaST; Chu and Liu, 1965; Edmonds, 1968; Mantegna, 1999), and most recently, the *Pólya filter* (Marcaccioli and Livan, 2018). Additionally, the TMFG method can be associated with the inverse covariance matrix via the Local/Global inversion method (LoGo; Aste and Di Matteo, 2017; Barfuss et al., 2016) for probabilistic graphical modeling. For brevity, I will only introduce the TMFG and LoGo methods.

The `TMFG()` method constructs the network by first sorting the edge weights (i.e., correlations) in descending order. Next, the TMFG algorithm finds the four nodes that have the largest sum of edge weights with all other nodes in the network and connects them, forming a tetrahedron. Then, the algorithm iteratively identifies and adds the node that maximizes the sum of its connections to three of the nodes already included in the network. This process is completed once the network reaches $3n$ - 6 edges (n = nodes). The resulting network is composed of 3-node and 4-node cliques (i.e., triangles and tetrahedrons, respectively), which imposes a nested hierarchy and automatically generates a *chordal* network (for more details see Barfuss et al., 2016 and Song et al., 2012). Therefore, the TMFG method of network construction is a suitable choice for modeling psychological constructs like personality traits (Christensen et al., 2018a; McCrae, 2015) and psychopathological disorders (Kotov et al., 2017; Markon et al., 2005).

The chordal property of the TMFG network means that each 4-node clique (i.e., a diamond; a set of 4 connected nodes) in the network possesses a *chord*, which connects two of the nodes not already connected, forming two triangles. Chordal networks are thus composed of `separators`, which are the 3-node cliques that *separate* one 4-node clique from another 4-node clique. Chordal networks are a

special class of networks that can represent the independence assumptions of Bayesian (directed) and Markovian (undirected) networks (Koller and Friedman, 2009) and have a simple association with the joint probability distribution (Barfuss et al., 2016; Massara et al., 2016). This association is expressed as:

$$Q(\mathbf{X}) = \frac{\prod_{C \in Cliques} \phi_C(\mathbf{X}_C)}{\prod_{S \in Separators} \phi_S(\mathbf{X}_S)} \tag{1}$$

where Q($\mathbf{X}$) is the estimated joint probability distribution, and $\phi(_C)$ and $\phi(_S)$ are the marginal probabilities within the subsets of variables of the 4-node cliques ($\mathbf{X}_C$) and 3-node separators ($\mathbf{X}_S$), respectively. This association to the joint probability distribution can be implemented using the Local/Global inversion method (LoGo; Barfuss et al., 2016). Thus, the only difference between the TMFG and LoGo networks is the edge weights (zero-order correlations and partial correlations regressed over all others, respectively). In this way, the TMFG network embeds conditional independence within its structure (see Figure 1). An example of each function is provided below:

```
#Construct TMFG network
tmfg <- TMFG(neoOpen)$A

#Construct LoGo network
logo <- LoGo(neoOpen)
```

These networks can be visualized side-by-side (Figure 1) using **qgraph** and the following code:

```
#Load qgraph
library(qgraph)

#NEO-PI-3 defined facets
facets <- c(rep("actions", 8), rep("aesthetics", 8), rep("fantasy", 8),
rep("feelings", 8), rep("ideas", 8), rep("values", 8))

#Visualize TMFG
A <- qgraph(tmfg, groups = facets, palette = "ggplot2")
#Visualize LoGo
B <- qgraph(logo, groups = facets, palette = "ggplot2")

#Visualize TMFG and LoGo side-by-side
layout(t(1:2))
Layout <- averageLayout(A, B)
qgraph(A, layout = Layout, esize = 20, title = "TMFG")
qgraph(B, layout = Layout, esize = 20, title = "LoGo")
```

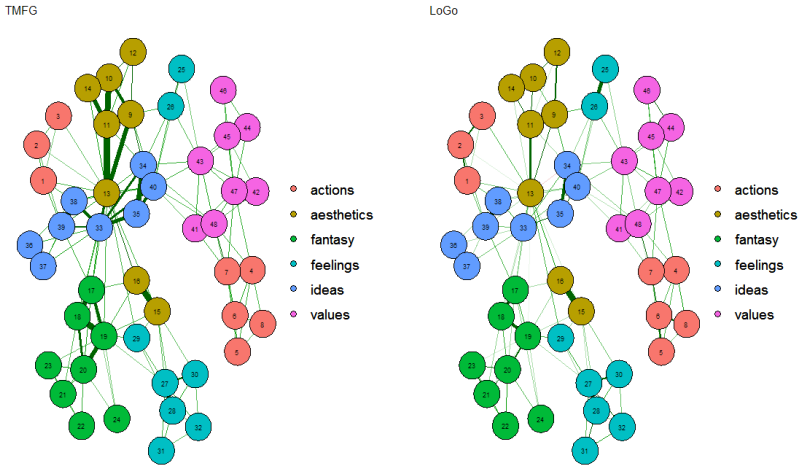

Figure 1: Plot of NEO-PI-3's items using the TMFG (left) and LoGo (right) network construction method

Network Construction Outputs Across all network construction methods, an adjacency matrix (or network) is output from the function. For most methods, this is the only output from the function.

For others, like `TMFG()`, a list containing the adjacency matrix (`A`) and other unique output from the function is supplied. Below, Table 1 provides the output of each network construction method.

Method	Output
`MaST()` (Chu and Liu, 1965; Edmonds, 1968)	adjacency matrix
`TMFG()` (Massara et al., 2016)	adjacency matrix (`A`) separators (`separators`) cliques (`cliques`)
`LoGo()` (Barfuss et al., 2016)	adjacency matrix
`ECO()` (Fallani et al., 2017)	adjacency matrix
`ECOplusMaST()` (Fallani et al., 2017)	adjacency matrix
`threshold()`	adjacency matrix (`A`) correlation critical value (`r.cv`)

Table 1: Network construction methods and outputs

The unique arguments of TMFG pertain to the `cliques` and `separators`, which are used to associate the TMFG network with the inverse covariance matrix via `LoGo()`. These outputs are a wrapper provided for `LoGo()` and are usually unimportant to the researcher. `threshold()` is the other network construction method with a unique output, `r.cv`, which provides the correlation value that was used to threshold values in the network.

Common Network Construction Arguments In general, there are several common arguments that are applied in each network construction method. Additionally, there are some arguments that are specific to a single network construction method. For all network construction methods in **NetworkToolbox**, there are a few arguments designed to handle the content and structure of the data. Below, Table 2 displays these arguments with the descriptions of their options.

Argument	Options	Description
`data`	`data`	input data or correlation matrix
`na.data`	`"listwise"`	removes any row with missing data
	`"pairwise"`	uses available data for given variable
	`"fiml"`	uses all information available
`normal`	`TRUE`	uses `cor_auto` from **qgraph**
	`FALSE`	uses Pearson's correlation

Table 2: Network construction arguments

The first argument is `data`, which can either be data or a correlation matrix. If input is data, then it should only include the variables of interest, without any identification, demographic, or descriptive variables (unless, of course, these are of interest). The second argument is `na.data`, which handles missing data. There are three options for `na.data`: `"listwise"`, `"pairwise"`, and `"fiml"`. The coarsest option is `"listwise"` deletion, which uses R's base function `na.omit()` to remove any case that has a missing observation. When using this argument, the researcher will automatically be notified exactly which rows were removed from network's construction. An example of this output is provided below:

```
Warning message: In TMFG(neoOpen, na.data = "listwise") :
10 rows were removed for missing data row(s):
234, 497, 71, 639, 99, 677, 652, 255, 150, 28
```

`"pairwise"` deletion will use the data that is available for each variable when constructing the network. Note that relations from pairwise deletion could potentially be based on different subsets of cases, which could be problematic. By far, the best and recommended option is `"fiml"` or Full Information Maximum Likelihood (FIML), which is implemented by the **psych** package (Revelle, 2017). FIML uses all information available to produce a deterministic correlation matrix (i.e., the same result every time). For this reason, FIML has been the standard missing data approach in structural equation modeling (SEM) software. Notably, network construction does take a few seconds longer when using the `"fiml"` option.

The last argument, `normal`, handles whether correlations should be used that are transformed to relations that are multivariate normal. There are multiple R packages that offer multivariate normal testing like **MVN** (Korkmaz et al., 2014), so such functions are not included in **NetworkToolbox**. Multivariate normal data are of particular importance in network and graphical modeling because there is a direct relationship between the inverse covariance matrix (also known as the *precision* matrix) and the partial correlation (conditional independence) structure. Thus, conditional independence networks fundamentally rely on a multivariate normal distribution. The necessity of this assumption and the influence of deviating from multivariate normal in network estimation are worthy of future investigation (e.g., Loh and Wainwright, 2013).

When `normal` is set to `TRUE`, then the data are correlated using **qgraph**'s `cor_auto()`. This function detects ordinal variables and uses polyserial, polychoric, and Pearson's correlations. Other correlations (e.g., tetrachoric) and association measures can also be used but they must be computed using some other function (e.g., **psych**'s `tetrachoric()`; Revelle, 2017) before they are used as input into any of the network construction functions. When `normal` is set to `FALSE`, Pearson's correlations are computed. In all network construction methods in **NetworkToolbox**, `normal` defaults to `FALSE`. It is important to note that despite the notion that Pearson's correlation assumes normality, it does not; however, the test statistic, on which significance is based, may have a different distribution when the data is non-normal. In addition, Pearson's correlation is sensitive to outliers and highly skewed variables, so data transformations or alternative correlation measures (e.g., Spearman's rho) should be considered when normality is violated.

Dependency Network Analysis One argument that is specific to the `TMFG()` function is `depend`, which is used to construct a dependency network from data that has been run through the function `depend()`. The dependency network approach was introduced by Kenett et al. (2010) and produces edges in networks that show the direction of influence from one node to another (i.e., a directional network)—similar to relative importance networks (Grömping and others, 2006; Johnson and LeBreton, 2004; Robinaugh et al., 2014). This approach makes use of first-order partial correlations, which are the partial (or residual) effect of a given node j on the correlation between another pair of nodes i and k. The resulting partial correlation between nodes i and k is the correlation after the subtraction of the correlations between nodes i and j and between nodes k and j. In this way, the partial correlation provides a measure of node j's influence on the correlation between nodes i and k. Therefore, the influence of node j on node i can be determined by taking the average (or the sum) influence of node j on the correlations of node i with all other nodes. Mathematically, this can be expressed as:

$$d(i,k|j) \equiv C(i,k) - PC(i,k|j) \tag{2}$$

where $C(i,k)$ is the correlation between node i and k and $PC(i,k\,|\,j)$ is the partial correlation of node i and k given j. By subtracting the partial correlation of nodes i and k given node j from the correlation of nodes i and k, the infrequent case of where node j appears to strongly affect the correlation $C(i,k)$ when $C(i,j)$, and $C(k,j)$ have small values is avoided. In order to determine the influence of node j on node i, the average influence of node j is defined across all relative effects of the correlations where node k is not node j:

$$D(i,j) = \frac{1}{N-1} \sum_{k \neq j}^{N-1} d(i,k|j) \tag{3}$$

As defined, the dependency matrix D has (i,j) elements, which are the dependency of node i on node j. Put differently, this measure specifies the average influence of node j on node i ($j \rightarrow i$). In a dependency network, the arrow pointing from node j to node i signifies that node j influences node i. Although the correlation matrix is symmetric, the dependency matrix is asymmetric, which means the influence of node j on node i, $D(j,i)$, is not equal to the influence of node i on node j, $D(i,j)$. It's important to note that these directional influences are not interpreted as causal relations but instead as predictive relationships. These networks are, however, a step towards inferring causal relations between two nodes and may function as a "pre-test" for data and methods that can infer causal relations (Jacob et al., 2016). Therefore, the advantage of the dependency network approach is that the mutual influence between two nodes is not assumed to be equal but rather one node is assumed to have more influence in the relationship than the other.

The TMFG algorithm was adapted, in accordance with Kenett, Tumminello, Madi, Gur-Gershgoren, Mantegna, and Ben-Jacob (2010), to handle the asymmetric relations that are produced by the dependency network approach. Kenett et al.'s (2010) adaption sorts the weights of $D(i,j)$ in a descending order, and whichever link of $D(i,j)$ or $D(j,i)$ is greater is retained. This is done to keep information about the main direction of influence and to avoid multiple links (although the *actual* direction of influence could be both ways). Below is an example of how to apply the dependency network approach

and the network it produces (Figure 2):

```
#Dependency matrix
depmat <- depend(neoOpen)

#Construct dependency TMFG network
deptmfg <- TMFG(depmat, depend = TRUE)$A

#Visualize dependency TMFG network
layout(c(1, 1))
qgraph(deptmfg, layout="spring", groups = facets, palette = "ggplot2",
label.prop = 1, vsize = 4, title = "Dependency TMFG")
```

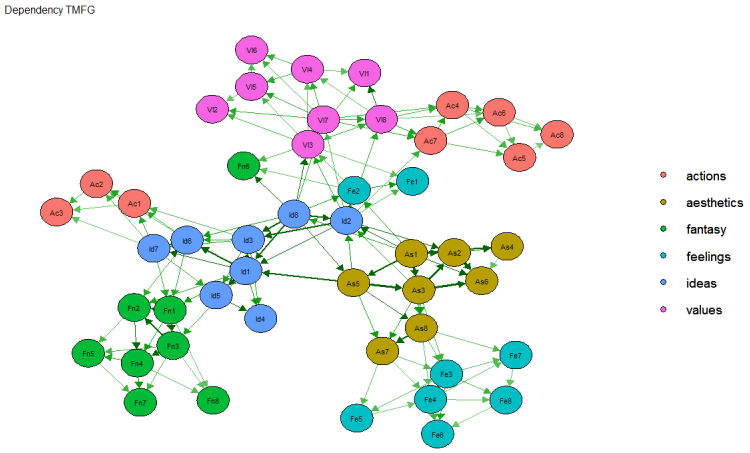

Figure 2: Plot of NEO-PI-3's items using the dependency TMFG network construction method

Network Measures

After network construction, network measures can be applied to quantify the information in the network. In **NetworkToolbox**, there are a several network measures available. First, I will start with local network characteristics, which have received the most attention in the psychometric network literature.

Local Network Characteristics The local network characteristics of a network describe how each node contributes to the overall network. Node centrality measures are the most common way to measure local network characteristics. Node centrality measures quantify each node's influence in the network. Some standard measures that are used are betweenness centrality (BC; `betweenness()`; Freeman, 1977), closeness centrality (LC; `closeness()`), degree (*k*; `degree()`), and node strength (NS; `strength()`). Because these measures are frequently used in the network science literature, I will assume the reader has encountered these measures before.

Randomized Shortest Paths Betweenness Centrality One measure that closely resembles the standard centrality measures is the randomized shortest paths betweenness centrality (RSPBC; Kivimäki et al., 2016). The difference between the standard BC and the RSPBC is the computation of the shortest paths—the smallest number of steps to get from one node to another—in the network. The standard BC measures how often a node is on the absolute shortest path (i.e., the most direct route) from one node to another. In contrast, the RSPBC measures how often a node is on the random shortest path (i.e., random "jumps") from one node to another. The randomness of the jumps are adjusted using the Bolztmann probability distribution, which can be manipulated with the argument `beta` (Kivimäki et al., 2016).

```
rspbc(A, beta = 0.01)
```

Higher `beta` values will decrease the randomness of the paths with `beta` = 10 being closer to the standard BC; lower `beta` values will increase the randomness of the paths with `beta` = .0001 being closer to degree. The `beta` argument defaults to .01, which is based on Kivimäki et al.'s (2016) recommendation. In **NetworkToolbox**, the RSPBC is adjusted so that the lowest value is one (the

node is included in its own shortest path) and the rest of the values are relative to this point (i.e., the minimum RSPBC value - 1 is subtracted from each node's RSPBC value).

The improvement of the RSPBC over the standard BC is its distribution of values. Often, the standard BC has a skewed distribution where few nodes have extremely large values and most nodes have low values or values of zero. The RSPBC also has many large values but there is a greater gradation of values across the nodes, and nodes that are clearly "between" other nodes are not given a value of zero (as they would with the standard BC). The node 41 in Figure 1, for example, has a standard BC log value of 0 but a RSPBC log value of 6.62. In comparison, node 6 in Figure 1 has a larger log value for standard BC (1.10) but a smaller log value for RSPBC (6.34). To demonstrate these differences, a plot is provided below (Figure 3).

```
#Randomized shortest paths betweenness centrality
bc <- betweenness(tmfg)
rbc <- rspbc(tmfg, beta = .01)

#Plot for comparison
plot(cbind(log(bc+1),log( rbc+1)), xlab = "Standard BC (log)",
ylab = "RSPBC (log)", main = "RSPBC on Standard BC", pch = 16)
text(6, 2, labels = paste("r = ", round(cor(bc, rbc, method = "kendall"), 2)))
```

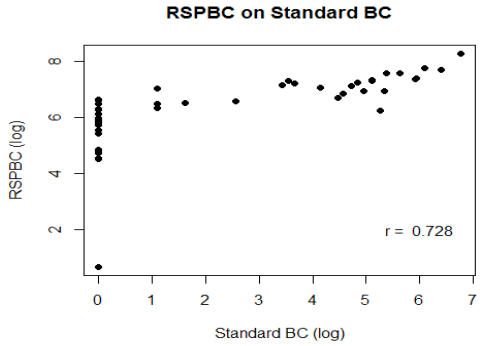

Figure 3: Plot of log-transformed RSPBC on log-transformed Standard BC

The rank-order comparison (r = .73) suggests that the measures are related but provide slightly different information. Indeed, the rank-order of the nodes is generally the same, with most of the differentiation being provided by the greater gradation of RSPBC values for nodes with small standard BC values. Notably, because RSPBC is continuous, rather than an all-or-nothing measure (like the standard BC), it's likely that it has greater reliability than the standard measure. The issue of standard BC's low stability is well known in the psychometric network literature (Epskamp et al., 2018). Therefore, the RSPBC stands as a reasonable alternative for researchers interested in measuring the betweenness centrality of their network. Moreover, the RSPBC's interpretation also provides a more reasonable explanation of a node's centrality influence in the network. The standard BC, for example, suggests that only the nodes on the absolute shortest path are influential in the spread of activation (i.e., the most central nodes' effect on other nodes). In contrast, the RSPBC suggests that this influence is mainly spread through the most central nodes but other avenues, that aren't the absolute most direct route but are still a short path to other nodes, could also potentially activate (or effect) other nodes.

Hybrid Centrality Another local network characteristic is an overall measure of centrality. This is provided by the `hybrid()` centrality (HC), which is not a specific centrality measure but rather a composite of BC, LC, NS, k, and the eigenvector centrality (EC; Christensen et al., 2018c; Pozzi et al., 2013). Pozzi et al. (2013) used this measure to assess risk associated with core and peripheral stocks in the stock market, finding more peripheral stocks were less risky to invest in. Christensen et al. (2018c) used this measure to assess the overall centrality of items in a self-report schizotypy—sub-clinical characteristics of schizophrenia—inventory. They found that more central items in the network were related, above and beyond intermediate and peripheral items, to interview-rated schizophrenia-spectrum symptoms.

The HC rank orders each centrality measure (lower values receiving a higher rank) using weighted and unweighted networks. Then, the ranks are added together and subtracted by the number of centrality measures used (e.g., eight), forming the numerator. The denominator is the number of

centrality measures times the number of nodes minus one. The mathematical expression is provided below:

$$HC = \frac{BC^w + BC^u + LC^w + BC^u + k^u + NS^w + EC^w + EC^u - 8}{8 \times (N - 1)} \tag{4}$$

where w signifies the weighted measure and u signifies the unweighted measure. Note that degree (k) and node strength (NS) are only included once because the weighted degree *is* node strength. Other than the adjacency matrix (`A`), the HC includes one argument for how `BC` should be computed. The options for `BC` include `"standard"` for standard BC and `"random"` for RSPBC, with the default being `"standard"`.

Node Impact One final local network characteristic is node `impact()`, which measures a node's influence on the structure of the network. To be more precise, node impact quantifies the difference between the network's average shortest path length (ASPL; the mean of the shortest number of edges for all nodes to get to other nodes in the network) when the node is removed from the network minus the ASPL when all nodes are included in the network (Kenett et al., 2011, 2013). Positive values suggest that removal of the node increases the ASPL of the network; negative values suggest that removal of the node decreases the ASPL of the network. Thus, a node with a positive impact *increases* the interconnectedness of the network when present in the network and a node with a negative impact *decreases* the interconnectedness of the network when present in the network. Other than centrality, node impact could be interpreted in terms of the community (i.e., clustered sets of connected nodes) structure of the network. A positive impact value means that when the node is removed, the distinctiveness (or "orthogonal-ness") of the communities in the network increases (Cotter et al., 2018). Conversely, a negative impact value means that when the node is removed, the distinctiveness of the communities in the network decreases.

Meso-scale Network Characteristics Meso-scale features of networks are becoming increasing popular in network analysis (Blanken et al., 2018; Giscard and Wilson, 2018; Golino and Epskamp, 2017; Golino and Demetriou, 2017; Golino et al., 2018). Meso-scale network characteristics evaluate the sub-network structure of the network or how the entire network can be summarized into smaller components (i.e., communities). These communities have been shown to be equivalent to factors or dimensions (Golino and Epskamp, 2017). The advantage of using networks to identify hierarchical structure in data is that they do not rely on the researcher's interpretation of scree plots, eigenvalues, or component loadings, which are used in more traditional dimension reduction methods (such as Principal Components Analysis; PCA). Instead, the hierarchical structure in networks is emergent—that is, based on the relations between variables, nodes cluster together and form distinct communities that are arise from the data.

Community detection is the main measure of meso-scale network characteristics as well as the foundation for other meso-scale characteristics. Therefore, the selection of a community detection algorithm is of the utmost importance. One of the most commonly applied algorithms is the walktrap algorithm (Golino and Epskamp, 2017; Pons and Latapy, 2006) but it is by no means the only one (for a review see Fortunato, 2010). The current standard in the psychometric network literature is to apply Exploratory Graph Analysis (EGA), which implements the walktrap algorithm using **igraph**. Currently, this approach has two implementations, which are based on different network construction methods: the graphical LASSO Golino and Epskamp (2017); Golino and Demetriou (2017) and TMFG (Golino et al., 2018). Both implementations have yielded promising results in simulation and real-world data, demonstrating comparable or better accuracy than traditional methods of dimension reduction (e.g., PCA, parallel analysis, cluster analysis; Christensen et al., 2018b; Golino and Epskamp, 2017; Golino et al., 2018). Because these methods are already supplied in **EGA** (Golino, 2018), I will not cover them here. Instead, I present a complementary measure that can be used to examine the "centralness" of communities in the overall network.

Community Closeness Centrality The concept of this measure extends from node-wise closeness centrality, with ASPL of the nodes within each community being used rather than the nodes. Each node's local ASPL ($ASPL_i$, i.e., each individual node's ASPL) is computed for all nodes. Then, the average across these nodes is taken and the reciprocal is calculated. This is done for each community. The interpretation of this measure is akin to the closeness node centrality—that is, communities closer to the overall center of the network have a higher value. This measure provides an objective classification of how central specific communities are in the network. Thus, the relative importance of a community in the network can be obtained. In psychometric networks, this is related to scale-inventory correlations. Below, an example is provided to demonstrate the close relationship between these two measurements.

```
#Unique facets
uniq <- unique(facets)

#Initialize matrix
corr <- matrix(0, nrow = length(uniq), ncol = 2)

#Name rows
row.names(corr) <- uniq

#Name columns
colnames(corr) <- c("Scale2Inv", "CommCent")

#Compute scale-to-inventory correlations
for(i in 1:length(uniq))
{
    #Identify facet members
    target <- which(facets == uniq[i])
    corr[i, 1] <- cor(rowMeans(neoOpen[, -target]), rowMeans(neoOpen[, target]))
}

#Compute community impact
corr[,2] <- comm.close(tmfg, comm = facets)

#Compute correlation
round(cor(corr, method = "kendall"), 2)
```

The rank-order correlation (r = .73) suggests that the measures are closely related but might not be taken as strong enough evidence for measuring the same thing. One reason for this difference is that the network's communities are organized differently than the theoretical definitions. Three items of the *actions* facet, for example, are not connected to the other items of the *actions* facet (Figure 1). This issue can be resolved by applying EGA to determine the network's community structure. Below is code to first apply and compare EGA dimensions to the theoretical dimensions.

```
#Load EGA
library(EGA)

#Estimate dimensions
ega <- EGA(neoOpen, model = "TMFG")
ega.var <- as.character(ega$dim.variables$items)

#Order by item
ega.ord <- ega$dim.variables[match(colnames(neoOpen), ega.var), 2]

#Visualize theoretical factors
A <- qgraph(tmfg, groups = as.factor(facets), palette = "ggplot2")

#Visualize walktrap factors
B <- qgraph(tmfg, groups = as.factor(ega.ord), palette = "ggplot2")

#Compare theoretical and walktrap factors
layout(t(1:2))
Layout <- averageLayout(A,B)
qgraph(A, layout = Layout, esize = 20, title = "Theoretical")
qgraph(B, layout = Layout, esize = 20, title = "Walktrap")
```

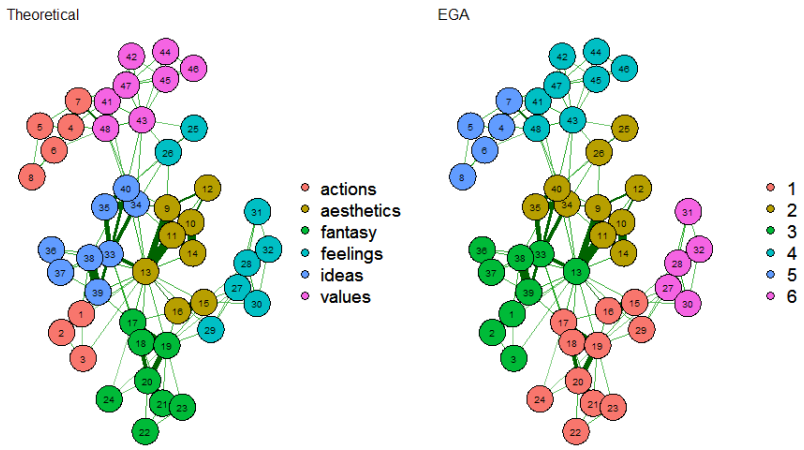

Figure 4: The theoretical facets are depicted on the left and the network-derived facets are depicted on the right

Notably, although number of the EGA-derived communities were equivalent to the number of theoretical facets, the items do not identically correspond between their designations. This does not suggest that the theoretical facets are incorrect nor does it suggest that the network-derived facets are incorrect. Network models are sample specific, which means a network with a different sample might demonstrate results more or less in line with the theoretical facets or the facets provided by EGA. In general, however, these results reproduce the theoretical facets defined by the NEO-PI-3. Next, code is provided to compute the similarity between the scale-to-inventory correlations with the EGA dimensions.

```
#Unique facets
uniq <- unique(ega.ord)

#Initialize matrix
corr <- matrix(0, nrow = length(uniq), ncol = 2)

#Name rows
row.names(corr) <- uniq

#Name columns
colnames(corr) <- c("Scale2Inv", "CommCent")

#Compute scale-to-inventory correlations
for(i in 1:length(uniq))
{
    #Identify facet members
    target <- which(ega.ord == uniq[i])
    corr[i, 1] <- cor(rowMeans(neoOpen[, -target]),rowMeans(neoOpen[, target]))
}

#Compute community closeness centrality
corr[, 2] <- comm.close(tmfg, comm = ega.ord)

#Compute correlation
round(cor(corr, method = "kendall"), 2)
```

With the EGA dimensions, there is a perfect rank-order correlation (r = 1) between the scale-to-inventory correlations and the community closeness centrality measure, suggesting there is a meaningful relationship between these two measures. In the context of networks more generally, this measure provides a quantitative approach to determining which communities are more central to the overall network.

Network Adjusted Means and Sums

Network models offer an attractive alternative to latent variable models, such as factor analysis because they provide a data-driven approach for determining the structure of a construct. Networks also

provide a way to go beyond sum scores by analyzing the smallest components of the overall construct (i.e., items and symptoms; Fried and Nesse, 2015). In addition, their emergent communities do not rely on constraints imposed by the researcher, which allows sample-specific expressions to emerge. To date, however, there has yet to be an approach developed to extract latent scores like traditional latent variable models (although network models have been combined with latent variable models; see Epskamp et al., 2017). Below, I introduce a function designed to extract network adjusted means and sums and propose a framework for establishing a latent network score. First, I briefly discuss the arguments and outputs of the function. After, I provide the conceptual framework for computing a network latent score, similar to latent scores calculated via confirmatory factor analysis (CFA). Finally, I demonstrate its effectiveness by comparing the network adjusted score to latent variable scores from a CFA model.

Arguments and Output Below are the arguments for the network adjusted means and sums function, `nams`:

```
nams(data, A, comm = c("walktrap", "louvain"),
standardize = TRUE, adjusted = c("mean", "sum"), ...)
```

`data` is an input for the data, which must be included in order to estimate the network adjusted mean or sum scores. The argument `A` is for an adjacency matrix (i.e., the network) associated with the data. Next, `comm` allows the researcher to define the communities of the network using a vector, which may be the output from a community detection algorithm or manually defined (e.g., theoretical communities). The default for `comm` is to treat the network as a single community. If a community detection algorithm (`"walktrap"` or `"louvain"`) is used, then `...` will handle additional arguments for the specified community detection algorithm. The argument `adjusted` is for whether the researcher would like the network adjusted mean or sum as the output (defaults to `"mean"`). Finally, the `standardize` argument is for whether the network adjusted scores should be standardized or not (defaults to `TRUE`). Note that when `standardize = TRUE`, the `adjusted` argument is ignored.

`nams()` outputs a list containing three objects: `NetAdjScore`, `FactorItems`, and `FactorCor`. `NetAdjScore` contains the network adjusted score, which is specified by the `adjusted` and `standardize` arguments. `FactorItems` displays a table of nodes corresponding to their respective factors (this is particularly useful when using a community detection algorithm). Finally, `FactorCor` provides the correlations between each community's network adjusted score including the overall network adjusted score.

Conceptual Framework The implementation of the this method requires a comprehensive measure of a node's influence in the network. As discussed above, the hybrid centrality is optimal for this task because it captures multiple centrality measures in a single overall centrality measure. The RSPBC is used for the betweenness centrality measure when computing the hybrid centrality in `nams()` because of its ability to differentiate lower values of betweenness (Figure 3).

Network as One Community Using the NEO-PI-3 data as an example, the hybrid centrality of each node is used as a weight for each participant's response—that is, their response on 5-point Likert scale ranging from 1 (*Strongly Disagree*) to 5 (*Strongly Agree*). This is done by multiplying each participant's response to each variable by the hybrid centrality value for each variable. This process is repeated for each participant and variable in the network. The participant's network adjusted mean or sum can be computed by taking the average or sum across their weighted responses. Then, the participants average or total can be normalized by dividing by the mean of the hybrid centrality values. The formula for this metric is described below:

$$\theta_j|u_j = \frac{\sum_{i=1}^{n} h_i u_{ij}}{\frac{1}{n}\sum_{i=1}^{n} h_i} \tag{5}$$

where i is a variable (i.e., a node), j is a case (e.g., a participant), u_{ij} is the response value for variable i and case j, n is the total number of nodes, h_i is the hybrid centrality value of node i, and $\theta_j \mid u_j$ is the latent score of participant j given the response pattern u_j. The stated equation provides the latent score estimate for the general construct that is being measured by the network. If the researcher is only interested in the construct associated with the entire network, then no additional computation is necessary.

Example Below I've provided code to compare the network's latent variable scores to the scores from a CFA latent variable model. For comparison, The network's latent variable scores are plotted on the CFA latent variable scores as well as the participant's mean scores on the CFA latent variable

scores. First, the theoretical CFA model must be built and the latent variable scores extracted. To do so, the **lavaan** (Rosseel, 2012) package was used.

```
#Load lavaan
library(lavaan)

#Build NEO model
neo.model <- 'actions =~ Act1 + Act2 + Act3 + Act4 + Act5 + Act6 + Act7 + Act8
aesthetics =~ Aes1 + Aes2 + Aes3 + Aes4 + Aes5 + Aes6 + Aes7 + Aes8
fantasy =~ Fan1 + Fan2 + Fan3 + Fan4 + Fan5 + Fan6 + Fan7 + Fan8
feelings =~ Fee1 + Fee2 + Fee3 + Fee4 + Fee5 + Fee6 + Fee7 + Fee8
ideas =~ Ide1 + Ide2 + Ide3 + Ide4 + Ide5 + Ide6 + Ide7 + Ide8
values =~ Val1 + Val2 + Val3 + Val4 + Val5 + Val6 + Val7 + Val8
open =~ actions + aesthetics + fantasy + feelings + ideas + values'

#Fit CFA model
fit <- cfa(neo.model, data = neoOpen, estimator = "WLSMV")

#Compute latent variable scores
cfaScores <- lavPredict(fit)
cfaLV <- scale(cfaScores[, 7])
```

Then, the network's overall latent variable scores must be extracted using `nams()`:

```
#Compute network latent variable scores
netScores <- nams(neoOpen, tmfg, comm = facets)
netLV <- netScores$Standardized$overall
```

Finally, the plots can be generated:

```
#Plot network latent variable on CFA latent variable
layout(t(c(1, 2)))
plot(cfaLV, netLV,
     main = "Network Latent Variable\non CFA Latent Variable",
     ylab = "Network Latent Variable",
     xlab = "CFA Latent Variable")

#Compute correlation
netCor <- cor(cfaLV, netLV)

#Compute root mean square error
netRoot <- rmse(cfaLV, netLV)

#Add text to plot
text(-2, 2, labels = paste("r = ", round(netCor, 2),
                           "\nrmse = ", round(netRoot, 3)))

#Compute standardized participant means
pmeans <- scale(rowMeans(neoOpen))

#Plot participant means on CFA latent variable
plot(cfaLV, pmeans,
     main = "Participant Means on\nCFA Latent Variable",
     ylab = "Participant Means",
     xlab = "CFA Latent Variable")

#Compute correlation
meansCor <- cor(cfaLV, pmeans)

#Compute root mean square error
meansRoot <- rmse(cfaLV, pmeans)

#Add text to plot
text(-2, 2, labels = paste("r = ", round(meansCor, 2),
                           "\nrmse = ", round(meansRoot, 3)))
```

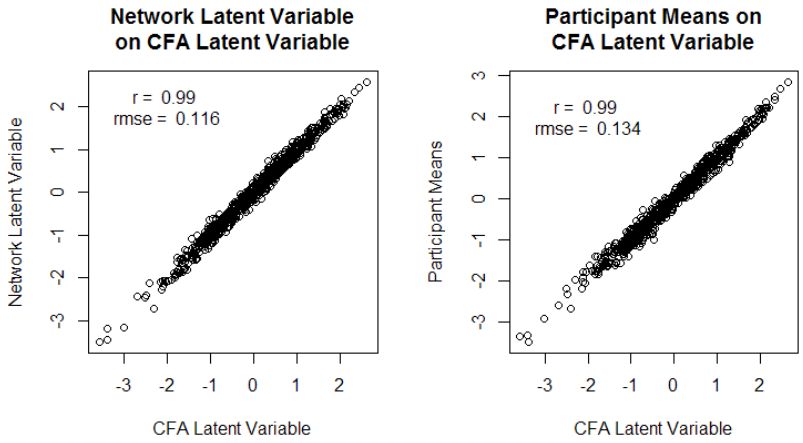

Figure 5: Plot of the standardized network latent scores on the standardized CFA latent scores (left) and the standardized participant's means on the standardized CFA latent scores (right)

In general, the network's latent scores are closer to the CFA's latent scores than the participant mean scores. The correlations, however, were high for both the network latent scores and the participant means. Figure 5 and the root mean square error show that the network's latent scores (rmse = .116) are more closely related to the CFA latent scores than the participant means (rmse = .134). Notably, the relationship between the network latent scores and the CFA latent scores is not perfect, suggesting there are some differences between the two measures.

Network as Many Communities The network's latent community scores differ slightly from the global network scores (i.e., the network as one community). For the community scores, it's important to consider the relative position of each community in the network. In order to factor in the positions of communities in the network, the community closeness centrality is computed for each community. The community closeness centrality provides a weight for nodes that are in communities that are more central in the network. Thus, nodes with low hybrid centrality that are in more central communities still receive additional weight because of their overall positioning in the network. The sum of the community closeness and global hybrid centrality is then used as the weight that is multiplied across the participant's response to each item within that community. The sum or mean of these facet scores are computed and normalized using the respective average of the weights in the community. This process repeats until all communities are computed. Finally, the mean or sum may not average or total to the overall network latent variable mean or sum. To adjust for this, the difference between the facets mean or sum total and the overall network latent mean or sum is taken and proportionally distributed across the facets.

Example Because the CFA latent scores and network latent scores were already computed, they can be immediately examined with the following code:

```
#Network latent facet scores
netFacet <- matrix(0, nrow = 6, ncol = 2)

for(i in 1:6)
{
    netFacet[i, 1] <- cor(cfaScores[, i], netScores$Standardized[, i])
    netFacet[i, 2] <- rmse(scale(cfaScores[, i]),netScores$Standardized[, i])
}

#Identify unique facets
uniq <- unique(facets)

#Participant facet means
meanFacet <- matrix(0, nrow = 6, ncol = 2)
for(i in 1:6)
{
    meanFacet[i, 1] <- cor(cfaScores[, i], rowMeans(neoOpen[, which(facets == uniq[i])]))
    meanFacet[i, 2] <- rmse(scale(cfaScores[, i]), scale(rowMeans(neoOpen[, which(facets == uniq[i])])))
}
```

```
#Compare network latent facet scores and participant facet means
comp <- cbind(netFacet, meanFacet)
row.names(comp) <- c("actions", "aesthetics",
                     "fantasy", "feelings",
                     "ideas", "values")
colnames(comp) <- c("netCor", "netRMSE", "meanCor", "meanRMSE")
```

Below is the table that is displaying the results from the above code.

	Network Latent Facet Scores		**Participant Facet Means**	
	r	rmse	*r*	rmse
actions	**.96**	**.278**	.92	.394
aesthetics	.98	**.176**	.98	.220
fantasy	**.98**	**.174**	.97	.255
feelings	.97	.238	.97	**.204**
ideas	**.99**	**.171**	.98	.211
values	.97	.225	**.98**	**.217**

Table 3: Pearson's correlation (*r*) and root mean square error (rmse) are shown in relation to the CFA latent facet scores. Bolded values signify better values.

The results shown in Table 3 suggest that the network latent community scores are equal to or more related to the CFA latent facet scores than the participant facet means. Once again, this suggests that the network latent scores are effective for estimating latent scores more generally. Similar to the global latent scores, the network latent community scores were not perfectly related to the CFA latent facet scores. Taken together, however, this provides evidence that latent network scores are plausible. Moreover, the advantage of the proposed method is that network adjusted sums and means can be computed, meaning sum scores can be altered based on each item's position in the network, providing differentiation between identical sum scores—that is, unless two cases with identical sum scores have identical data patterns, they will have different scores based on the structure of the network.

Summary

Network science is a rapidly developing statistical approach in psychology. The appeal of the approach is the intuitive modeling of complexity that is intrinsic in many psychological phenomena. **NetworkToolbox** is designed to equip researchers with different network tools that have been developed over decades of work in graph theory and the complex systems domains. In this article, many functions of **NetworkToolbox** were introduced with relevant details for appropriate use. The examples of code used throughout this article can be used as a blueprint for any researcher interested in running their own psychometric network analysis. Notably, not all of the functions available in **NetworkToolbox** were discussed; however, documentation with examples are provided within the package. Many of the arguments discussed in this article are also used across the package. Notably, **NetworkToolbox** is not exhaustive—there are many other methods and measures for constructing, analyzing, and quantifying networks. Because of this, **NetworkToolbox** will continue to expand to equip researchers with the newest advances in the domain.

Computational Details

The results of this paper were obtained using R 3.4.4 with the **Networktoolbox** 1.2.1 package and the networks were visualized using the **qgraph** 1.4.4 package. EGA was provided by the **EGA** 0.4 package (available from GitHub: https://github.com/hfgolino/EGA). The CFA model was generated using the **lavaan** 0.5-23.1097 (BETA) package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Bibliography

T. Aste and T. Di Matteo. Sparse causality network retrieval from short time series. *Complexity*, 2017, 2017. URL https://doi.org/10.1155/2017/4518429. [p2]

A.-L. Barabási. The network takeover. *Nature Physics*, 8(1):14–16, 2011. URL https://doi.org/10.1038/nphys2188. [p1]

W. Barfuss, G. P. Massara, T. Di Matteo, and T. Aste. Parsimonious modeling with information filtering networks. *Physical Review E*, 94(6):062306, 2016. URL https://doi.org/10.1103/PhysRevE.94.062306. [p2, 3, 4]

T. F. Blanken, M. K. Deserno, J. Dalege, D. Borsboom, P. Blanken, G. A. Kerkhof, and A. O. Cramer. The role of stabilizing and communicating symptoms given overlapping communities in psychopathology networks. *Scientific Reports*, 8(1), 2018. URL https://doi.org/10.1038/s41598-018-24224-2. [p8]

D. Borsboom. A network theory of mental disorders. *World Psychiatry*, 16(1):5–13, 2017. URL https://doi.org/10.1002/wps.20375. [p1]

D. Borsboom and A. O. Cramer. Network analysis: An integrative approach to the structure of psychopathology. *Annual Review of Clinical Psychology*, 9:91–121, 2013. URL https://doi.org/10.1146/annurev-clinpsy-050212-185608. [p1]

C. T. Butts and others. Social network analysis with **sna**. *Journal of Statistical Software*, 24(6):1–51, 2008. URL http://doi.org/10.18637/jss.v014.i06. [p1]

A. P. Christensen, K. N. Cotter, and P. J. Silvia. Reopening openness to experience: A network analysis of four openness to experience inventories. *Journal of Personality Assessment*, 2018a. URL https://doi.org/10.1080/00223891.2018.1467428. [p2]

A. P. Christensen, G. M. Gross, H. F. Golino, P. J. Silvia, and T. R. Kwapil. Exploratory graph analysis of the multidimensional schizotypy scales. *PsyArXiv*, 2018b. URL https://doi.org/10.31234/osf.io/vqgsx. [p8]

A. P. Christensen, Y. N. Kenett, T. Aste, P. J. Silvia, and T. R. Kwapil. Network structure of the wisconsin schizotypy scales-short forms: Examining psychometric network filtering approaches. *Behavior Research Methods*, pages 1–20, 2018c. URL https://doi.org/10.3758/s13428-018-1032-9. [p7]

Y. J. Chu and T. H. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400, 1965. URL https://ci.nii.ac.jp/naid/10030090917/en/. [p2, 4]

K. N. Cotter, A. P. Christensen, and P. J. Silvia. Understanding inner music: A dimensional approach to musical imagery. *Psychology of Aesthetics, Creativity, and the Arts*, 2018. [p8]

A. O. Cramer, S. Sluis, A. Noordhof, M. Wichers, N. Geschwind, S. H. Aggen, K. S. Kendler, and D. Borsboom. Dimensions of normal personality as networks in search of equilibrium: You can't like parties if you don't like people. *European Journal of Personality*, 26(4):414–431, 2012. URL https://doi.org/10.1002/per.1866. [p1]

G. Csardi and T. Nepusz. The **igraph** software package for complex network research. *InterJournal, Complex Systems*, 1695(5):1–9, 2006. URL https://pdfs.semanticscholar.org/1d27/44b83519657f5f2610698a8ddd177ced4f5c.pdf. [p1]

C. G. DeYoung, L. C. Quilty, and J. B. Peterson. Between facets and domains: 10 aspects of the big five. *Journal of Personality and Social Psychology*, 93(5):880–896, 2007. URL https://doi.org/10.1037/0022-3514.93.5.880. [p2]

J. Edmonds. Optimum branchings. *Mathematics and the Decision Sciences*, 1(335-345), 1968. [p2, 4]

S. Epskamp, A. O. Cramer, L. J. Waldorp, V. D. Schmittmann, D. Borsboom, and others. **qgraph**: Network visualizations of relationships in psychometric data. *Journal of Statistical Software*, 48(4):1–18, 2012. URL https://doi.org/10.18637/jss.v048.i04. [p1]

S. Epskamp, M. Rhemtulla, and D. Borsboom. Generalized network psychometrics: Combining network and latent variable models. *Psychometrika*, 82(4):904–927, 2017. URL https://doi.org/10.1007/s11336-017-9557-x. [p11]

S. Epskamp, D. Borsboom, and E. I. Fried. Estimating psychological networks and their accuracy: A tutorial paper. *Behavior Research Methods*, 50(1):195–212, 2018. URL https://doi.org/10.3758/s13428-017-0862-1. [p2, 7]

F. D. V. Fallani, V. Latora, and M. Chavez. A topological criterion for filtering information in complex brain networks. *PLoS Computational Biology*, 13(1):e1005305, 2017. URL https://doi.org/10.1371/journal.pcbi.1005305. [p4]

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010. URL https://doi.org/10.1016/j.physrep.2009.11.002. [p8]

L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977. URL https://www.jstor.org/stable/3033543. [p6]

E. I. Fried and R. M. Nesse. Depression sum-scores don't add up: Why analyzing specific depression symptoms is essential. *BMC Medicine*, 13(1):72, 2015. URL https://doi.org/10.1186/s12916-015-0325-4. [p11]

J. Friedman, T. Hastie, and R. Tibshirani. *Glasso: Graphical Lasso- Estimation of Gaussian Graphical Models*, 2014. URL https://CRAN.R-project.org/package=glasso. R package version 1.8. [p2]

P.-L. Giscard and R. C. Wilson. A centrality measure for cycles and subgraphs ii. *Applied Network Science*, 3(1):9, 2018. URL https://doi.org/10.1007/s41109-018-0064-5. [p8]

H. F. Golino. **EGA***: Exploratory Graph Analysis - Estimating the Number of Dimensions in Psychological Data*, 2018. URL http://github.com/hfgolino/EGA. R package version 0.4. [p8]

H. F. Golino and A. Demetriou. Estimating the dimensionality of intelligence like data using exploratory graph analysis. *Intelligence*, 62:54–70, 2017. URL https://doi.org/10.1016/j.intell.2017.02.007. [p8]

H. F. Golino and S. Epskamp. Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PloS one*, 12(6):e0174035, 2017. URL https://doi.org/10.1371/journal.pone.0174035. [p8]

H. F. Golino, D. Shi, L. Nieto, L. E. Garrido, and A. P. Christensen. Investigating the performance of exploratory graph analysis and traditional techniques to identify the number of latent factors. *under review*, 2018. [p8]

S. M. Goodreau, M. S. Handcock, D. R. Hunter, C. T. Butts, and M. Morris. A **statnet** tutorial. *Journal of Statistical Software*, 24(9):1–27, 2008. URL https://doi.org/10.18637/jss.v024.i09. [p1]

U. Grömping and others. Relative importance for linear regression in r: The package **relaimpo**. *Journal of Statistical Software*, 17(1):1–27, 2006. URL https://doi.org/10.18637/jss.v017.i01. [p5]

Y. Jacob, Y. Winetraub, G. Raz, E. Ben-Simon, H. Okon-Singer, K. Rosenberg-Katz, T. Hendler, and E. Ben-Jacob. Dependency network analysis (depna) reveals context related influence of brain network nodes. *Scientific Reports*, 6:27444, 2016. URL https://doi.org/10.1038/srep27444. [p5]

J. W. Johnson and J. M. LeBreton. History and use of relative importance indices in organizational research. *Organizational Research Methods*, 7(3):238–257, 2004. URL https://doi.org/10.1177/1094428104266510. [p5]

D. Y. Kenett, M. Tumminello, A. Madi, G. Gur-Gershgoren, R. N. Mantegna, and E. Ben-Jacob. Dominating clasp of the financial sector revealed by partial correlation analysis of the stock market. *PloS one*, 5(12):e15032, 2010. URL https://doi.org/10.1371/journal.pone.0015032. [p5]

Y. N. Kenett, D. Y. Kenett, E. Ben-Jacob, and M. Faust. Global and local features of semantic networks: Evidence from the hebrew mental lexicon. *PloS one*, 6(8):e23912, 2011. URL https://doi.org/10.1371/journal.pone.0023912. [p8]

Y. N. Kenett, D. Wechsler-Kashi, D. Y. Kenett, R. G. Schwartz, E. Ben Jacob, and M. Faust. Semantic organization in children with cochlear implants: Computational analysis of verbal fluency. *Frontiers in Psychology*, 4:543, 2013. URL https://doi.org/10.3389/fpsyg.2013.00543. [p8]

I. Kivimäki, B. Lebichot, J. Saramäki, and M. Saerens. Two betweenness centrality measures based on randomized shortest paths. *Scientific Reports*, 6:19668, 2016. URL https://doi.org/10.1038/srep19668. [p6]

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009. [p3]

S. Korkmaz, D. Goksuluk, and G. Zararsiz. **MVN**: An r package for assessing multivariate normality. *The R Journal*, 6(2):151–162, 2014. URL https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. [p5]

R. Kotov, R. F. Krueger, D. Watson, T. M. Achenbach, R. R. Althoff, R. M. Bagby, T. A. Brown, W. T. Carpenter, A. Caspi, L. A. Clark, and others. The hierarchical taxonomy of psychopathology (hitop): A dimensional alternative to traditional nosologies. *Journal of Abnormal Psychology*, 126(4):454–477, 2017. URL https://doi.org/10.1037/abn0000258. [p2]

K. Lee and M. C. Ashton. Psychometric properties of the hexaco-100. *Assessment*, 1073191116659134: 1–15, 2016. URL https://doi.org/10.1177/1073191116659134. [p2]

P.-L. Loh and M. J. Wainwright. Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *The Annals of Statistics*, pages 3022–3049, 2013. URL http://dx.doi.org/10.1214/13-AOS1162. [p5]

R. N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B Condensed Matter and Complex Systems*, 11(1):193–197, 1999. URL https://doi.org/10.1007/s100510050929. [p2]

R. Marcaccioli and G. Livan. A parametric approach to information filtering in complex networks: The pólya filter. *arXiv*, 2018. URL https://arxiv.org/abs/1806.09893. [p2]

K. E. Markon, R. F. Krueger, and D. Watson. Delineating the structure of normal and abnormal personality: An integrative hierarchical approach. *Journal of Personality and Social Psychology*, 88(1): 139–157, 2005. URL https://doi.org/10.1037/0022-3514.88.1.139. [p2]

G. P. Massara, T. Di Matteo, and T. Aste. Network filtering for big data: Triangulated maximally filtered graph. *Journal of Complex Networks*, 5(2):161–178, 2016. URL https://doi.org/10.1093/comnet/cnw015. [p2, 3, 4]

R. R. McCrae. A more nuanced view of reliability: Specificity in the trait hierarchy. *Personality and Social Psychology Review*, 19(2):97–112, 2015. URL https://doi.org/10.1177/1088868314541857. [p2]

R. R. McCrae and P. T. Costa Jr. Brief versions of the neo-pi-3. *Journal of Individual Differences*, 28(3): 116–128, 2007. URL https://doi.org/10.1027/1614-0001.28.3.116. [p2]

R. R. McCrae, P. T. Costa Jr, and T. A. Martin. The neo-pi-3: A more readable revised neo personality inventory. *Journal of Personality Assessment*, 84(3):261–270, 2005. URL https://doi.org/10.1207/s15327752jpa8403_05. [p2]

P. Pons and M. Latapy. Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, 10(2):191–218, 2006. URL https://doi.org/10.1007/11569596_31. [p8]

F. Pozzi, T. Di Matteo, and T. Aste. Spread of risk across financial markets: Better to invest in the peripheries. *Scientific Reports*, 3:1665, 2013. URL https://doi.org/10.1038/srep01665. [p7]

W. Revelle. **psych***: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois, 2017. URL https://CRAN.R-project.org/package=psych. R package version 1.7.8. [p4, 5]

D. J. Robinaugh, N. J. LeBlanc, H. A. Vuletich, and R. J. McNally. Network analysis of persistent complex bereavement disorder in conjugally bereaved adults. *Journal of Abnormal Psychology*, 123(3): 510–522, 2014. URL http://dx.doi.org/10.1037/abn0000002. [p5]

Y. Rosseel. **lavaan**: An r package for structural equation modeling and more. version 0.5–12 (beta). *Journal of Statistical Software*, 48(2):1–36, 2012. URL https://doi.org/10.18637/jss.v048.i02. [p12]

V. D. Schmittmann, A. O. Cramer, L. J. Waldorp, S. Epskamp, R. A. Kievit, and D. Borsboom. Deconstructing the construct: A network perspective on psychological phenomena. *New Ideas in Psychology*, 31(1):43–53, 2013. URL https://doi.org/10.1016/j.newideapsych.2011.02.007. [p1]

W.-M. Song, T. Di Matteo, and T. Aste. Hierarchical information clustering by means of topologically embedded graphs. *PLoS One*, 7(3):e31929, 2012. URL https://doi.org/10.1371/journal.pone.0031929. [p2]

R. C. Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL https://www.R-project.org/. [p1]

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, pages 267–288, 1996. URL https://www.jstor.org/stable/2346178. [p2]

M. Tumminello, T. Aste, T. Di Matteo, and R. N. Mantegna. A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences*, 102(30):10421–10426, 2005. URL https://doi.org/10.1073/pnas.0500298102. [p2]

C. D. van Borkulo, D. Borsboom, S. Epskamp, T. F. Blanken, L. Boschloo, R. A. Schoevers, and L. J. Waldorp. A new method for constructing networks from binary data. *Scientific Reports*, 4:5918, 2014. URL https://doi.org/10.1038/srep05918. [p1, 2]

C. G. Watson. **brainGraph***: Graph Theory Analysis of Brain MRI Data*, 2017. URL https://CRAN.R-project.org/package=brainGraph. R package version 1.0.0. [p1]

S. E. Woo, O. S. Chernyshenko, A. Longley, Z.-X. Zhang, C.-Y. Chiu, and S. E. Stark. Openness to experience: Its lower level structure, measurement, and cross-cultural equivalence. *Journal of Personality Assessment*, 96(1):29–45, 2014. URL https://doi.org/10.1080/00223891.2013.806328. [p2]

Alexander Christensen
Department of Psychology
P.O. Box 26170
University of North Carolina at Greensboro
Greensboro, NC, 27402-6170, USA
ORCiD: 0000-0002-9798-7037
apchrist@uncg.edu