

User Manual

Table of content/function reference list:

| | |
|---|----------|
| 1. General Introduction: | 4 |
| • <i>How to use the manual</i> | 4 |
| • <i>Quick start</i> | 4 |
| 1. Getting started: | 5 |
| 2. Workspaces , reference files, and input data | 5 |
| 2.2.1. <i>The “BrainRegions” file</i> | 6 |
| 2.2.2. <i>The “Variables” file (mandatory for statistical analyses)</i> | 6 |
| 2.2.3. <i>The “data” folder</i> | 6 |
| 2.2.4. <i>The “results” folder</i> | 6 |
| 2.3. <i>Mandatory settings (it is very important to make sure that)</i> | 6 |
| 2. Analysis setup interface | 7 |
| 1. The General Settings panel | 7 |
| 1.1. <i>Brain regions files (xls)</i> | 7 |
| 1.2. <i>File with Variables (xls)</i> | 7 |
| 1.3. <i>“Select Subjects (Corr Matrix)”</i> | 8 |
| 1.4. <i>Subject name in Filename</i> | 8 |
| 1.5. <i>Corr Matrix Array</i> | 8 |
| 1.6. <i>Create Connectivity Matrix</i> | 8 |
| 1.7. <i>Connectivity Matrices</i> | 8 |
| 1.8. <i>Create random time series</i> | 9 |
| 1.8.1.-3. <i>Randomize/Shuffle/FFT</i> | 9 |
| 1.8.4. <i>Number of random series</i> | 9 |
| 2. The Network Construction panel | 10 |
| 2.2. <i>Network nodes/Brain areas</i> | 10 |
| 2.3. <i>Network thresholds</i> | 10 |
| 2.3.1. <i>Significant</i> | 11 |

| | | |
|-----------|---|-----------|
| 2.3.2. | <i>Relative</i> | 11 |
| 2.3.3. | <i>Absolute</i> | 11 |
| 2.3.4. | <i>None</i> | 11 |
| 2.3.5. | <i>SICE</i> | 11 |
| 2.4. | <i>Generate null-model networks</i> | 11 |
| 2.4.1. | <i>Binary/weighted</i> | 11 |
| 3. | The Network Calculations panel | 12 |
| 3.1. | <i>Calculate graph metrics</i> | 12 |
| 3.2. | <i>Normalize graph metrics with random networks</i> | 12 |
| 3.3. | <i>Use random network to calc smallworldness</i> | 12 |
| 3.4. | <i>Calculate variables and export</i> | 12 |
| 4. | The Raw Matrix (link wise) panel | 13 |
| 4.1. | <i>Introduction:</i> | 13 |
| 4.2. | <i>Raw matrix</i> | 13 |
| 4.3. | <i>r to z</i> | 14 |
| 4.4. | <i>Connectivity Thr.</i> | 14 |
| 5. | The Statistics panel | 15 |
| 5.1. | <i>Do correlation/do partial correlation</i> | 15 |
| 5.2. | <i>Test against random networks (graph metrics)</i> | 16 |
| 5.3. | <i>Test against shuffled data</i> | 16 |
| 5.4. | <i>Test against random Networks (raw matrix)</i> | 16 |
| 5.5. | <i>Do Group Comparison</i> | 16 |
| 5.6. | <i>Test against random groups</i> | 17 |
| 5.8. | <i>Raw Matrix – Group</i> | 17 |
| 3. | Results viewer | 18 |
| 1. | The results selection box: | 18 |
| 1.2. | <i>Variable</i> | 18 |
| 1.3. | <i>Graphvar</i> | 18 |
| • | <i>“one-dimensional”</i> | 19 |
| • | <i>“two-dimensional”</i> | 19 |
| • | <i>“three-dimensional”</i> | 19 |
| 1.4. | <i>Threshold</i> | 19 |
| 1.5. | <i>Brain Areas</i> | 19 |

| | |
|--|----|
| 2. The results display: | 19 |
| 2.1. <i>X- and Y-axes</i> | 21 |
| 2.2. <i>Group-comparison</i> | 21 |
| 2.2.1. <i>Show/Hide group value</i> | 21 |
| 2.2.2. <i>Show All Groups</i> | 21 |
| 2.2.3. <i>Selection field</i> | 21 |
| 3. The general functions panel | 22 |
| 3.1. <i>Alpha Level</i> | 22 |
| 3.2. <i>Hide non-significant</i> | 22 |
| 3.3. <i>Show p-values</i> | 22 |
| 3.4. <i>Correction:</i> | 23 |
| 3.4.1. <i>None</i> | 23 |
| 3.4.2. <i>Bonferroni correction</i> | 23 |
| 3.4.3. <i>FDR correction</i> | 23 |
| 3.4.4. <i>Random Networks/Groups</i> | 23 |
| 3.5. <i>Show randomize...</i> | 23 |
| • <i>Correlation</i> | 23 |
| • <i>Group analysis</i> | 24 |
| 3.6. <i>Show Vars/Areas with more than __ Significant Filter</i> | 24 |
| 3.7. <i>Visualization function- "No vis function"</i> | 24 |
| 3.8. <i>Export Data</i> | 24 |
| 3.9. <i>Save</i> | 24 |
| 4. Network Inspector | 25 |
| 2. The results panel | 25 |
| 3. The options bar | 27 |
| 3..1. <i>Alpha input-box</i> | 27 |
| 3..2. <i>Show all labels</i> | 27 |
| 3..3. <i>Enable mouse-over</i> | 27 |
| 3..4. <i>Previous/Next</i> | 27 |
| 4. The network display | 27 |
| 5. Appendix | 27 |
| 1. Appendix 1: list of graph functions + description | 27 |
| 2. Appendix 2: include new functions and data handling | 31 |

1. General Introduction:

- **How to use the manual:** The manual is divided into a general introduction section, three sections covering the GraphVar GUI (*Analysis setup interface; Results viewer, Network Inspector*), and into an appendix comprising more detailed information on the underlying processes and data handling of GraphVar. The general introduction covers basic concepts and points to important data requirements (recommended to read), whereas the GUI section entails detailed description of each implemented option that can be chosen by the user. When planning advanced and large scale analyses it is suggested to carefully study the descriptions (this will moreover enable you to describe the analyses in the methods section of your paper).
- **Quick start:** For a quick start you can load the “sample workspace”, where sample data is already loaded and set up. However, please refer to the “GraphVar_QuickGuide.pdf”/read the general introduction and the section about “The General Settings panel”. Very important remarks are highlighted in red font color.
- **TIP 1:** to generate your data input (e.g. time series signals of brain regions/nodes or correlation matrices) you could use the software DPARSF (<http://rfmri.org/DPARSF>). However, some GraphVar features will only be available when generating correlation matrices based on time series signals with the GraphVar function “Create Connectivity Matrix”.
- **TIP 2:** GraphVar requires the **Matlab statistics toolbox for statistical analyses. However, calculation and export of network topological measures will also work without the statistics toolbox.** If you have installed SPM in your Matlab path it is important to know that SPM entails functions with the same name as the statistics toolbox. If errors occur during GraphVar’s statistical analyses, you may have to move the statistics toolbox to the top of your Matlab path to prevent conflicts (most likely with the function “*nanvar_base*”).
- **TIP 3:** to make your workspaces portable (i.e., to allow loading them on another computer), you should store your data in the recommended folders (“data” folder) which are automatically created in a new workspace.
- **TIP 4:** you can replace the default files (“The “BrainRegions” file and “Variables””) in the “SampleData” folder with your personalized files to load these automatically whenever a new workspace is created. When a new workspace is created GraphVar will load the .csv files but you also have the option to load .xls files from within GraphVar. However, keep the names as the original files and do not just delete without replacement.

- **TIP 5:** by default GraphVar saves all interim results to provide these for further statistical analyses. If your data storage space is limited, it is advisable to delete these interim results manually from time to time (see Appendix 2.5 for more info).

1. Getting started:

- 1.1. Unzip the GraphVar folder and put it in your preferred directory. Add the GraphVAR folder to your Matlab path, and then simply type “start_GraphVar” into the Matlab command line and the program will launch its welcome screen.

2. Workspaces , reference files, and input data:

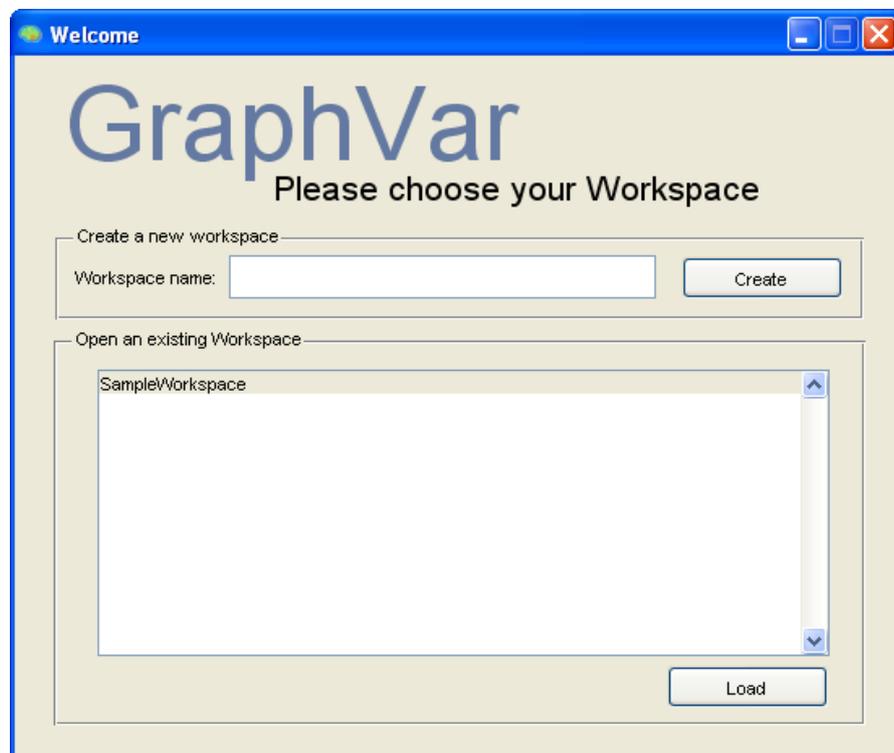


Figure 1. Choose workspace

- 2.1. You can create a new or load an existing workspace (by default workspaces are stored as subfolders within the GraphVar directory under “workspaces”). For each workspace GraphVar will automatically save the last settings that were selected before terminating the program and will respectively store the computed results if opted (see the “Save” option in the “Results viewer”). Therefore, it is advisable to create a new workspace for new projects (e.g. different subjects or data to be analyzed).
- 2.2. For each newly created workspace GraphVar will automatically generate four important files: two folders (“data” and “results”) and two .xlsx/csv files (“Variables” and

"BrainRegions"). GraphVar provides sample data in a sample workspace to make the setup process easier and to provide an idea of the data input/structure (to personalize your workspace refer to "[The General Settings panel](#)").

2.2.1. **The "BrainRegions" file** contains the regions of the AAL atlas, which are commonly used for node definition in graph analyses. However, you can substitute these regions with the nodes you need for your analysis (e.g. task-fMRI derived regions of interest or any other brain atlas). As GraphVar is not limited to a specific number of nodes you can define your own brain parcellation scheme. When doing so by creating your own BrainRegions file, it is important to keep in line with the structure of the original sheets as GraphVar will refer to it (also refer to the "ROI_template" folder).

2.2.2. **The "Variables" file (mandatory for statistical analyses):** contains subject specific data (including identifiers, ID) for statistical analyses. This file shows the way in which the data has to be organized so that GraphVar can make use of it. When you load a workspace the first time, GraphVar will show you an interface for selection of the (sample) subject ID and the subject specific data. When GraphVar is launched you will have the possibility to select your own variables sheet (please refer to "[The General Settings panel](#)"). If no excel is installed GraphVar uses .csv files.

2.2.3. **The "data" folder** contains two subfolders "CorrMatrix" and "Signals". This is where GraphVar suggests putting the data you want to analyze (this allows to load workspaces on different computers once they have been copied to the "workspace" folder). "CorrMatrix" contains subject specific $n \times n$ correlation matrices (the association matrices of the network). "Signals" contains the regions time courses that can be used to create the association matrices by computing (partial) correlations across the signals. Data format is respectively a subject specific .mat file that contains the ID of the subject in its filename. You can work with either data (CorrMatrix or Signals) but some GraphVar features will only be available when generating correlation matrices based on the time courses by using the GraphVar "[Create Connectivity Matrix](#)".

2.2.4. **The "results" folder** contains initially empty subfolders ("CorrectedAlpha; CorrMatrix; CorrResults; RandomizedTimeSeries; Saved") that will serve for storing the results of your computations (please refer to "[Appendix 2:](#)" for a detailed description of the data handling).

2.3. **Mandatory settings (it is very important to make sure that):**

2.3.1. **For statistical analyses:** subject IDs in the "Variables" sheet must match subject IDs in the subject specific .mat filename that contains the data (if you only want to compute and export the graph measures you don't need any variables sheet).

2.3.2. **The number and order of your brain parcellation scheme in [The "BrainRegions" file](#) (e.g. 90 AAL regions) must match the number and order of the subject specific $n \times n$ correlation matrices (e.g. 90 x 90) that you input to be analyzed OR must match the**

number and order of the regions time courses (this allows labeling of the nodes in the interactive results viewer and moreover allows for an easy redefinition of your network once you have loaded in all the data by simply selecting specific nodes (e.g. for analyses of sub-networks)).

2. Analysis setup interface

1. The General Settings panel:

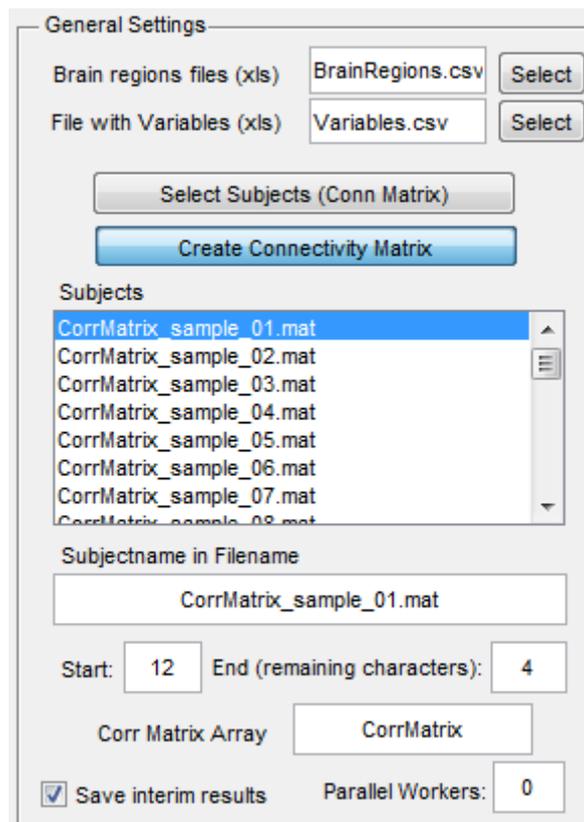


Figure 2. General Settings panel

- 1.1. **Brain regions files (xls)**: here you can select your own brain parcellation scheme (e.g. AAL atlas). When starting a new workspace this refers to the “BrainRegions.csv/xlsx” file. The labels of the node definition will respectively be loaded and displayed in the “Network Construction” panel under “Network nodes/Brain areas”.
- 1.2. **File with Variables (xls)**: here you can select the spread sheet including subject IDs and their demographic/genetic/etc. data used for statistical analyses (a selection prompt will appear for selection of the variables). When planning to do group comparisons it is also possible to enter strings to define the groups (e.g. male vs. female instead of 0 vs. 1). When starting a new workspace the variables list refers to the “Variables” file.

- 1.3. **“Select Subjects (Conn Matrix)”**: browse to the location of your correlation matrices and select all or specific subjects (each subject’s correlation matrix should be stored in a .mat file array). For how to load time courses refer to **“Create Connectivity Matrix”**.
- 1.4. **Subject name in Filename**: this field will display the first subject specific .mat data filename. As shown in Figure 2 the subject name must be specified through highlighting with the mouse cursor. Based on your selection the numbers next to **“Start”** and **“End – remaining characters”** will change. Alternatively you can specify the name directly using the number fields. This input is mandatory as it enables GraphVar to make a reference between the subject ID in the **“Variables”** file and the .mat data file for each subject.
- 1.5. **Corr Matrix Array**: refers to the name of the array (within your subject specific .mat file) in which the correlation matrix is stored. When loading subjects via **“select subjects”** a selection prompt gives you the possibility to select the respective array and the input is done automatically.
- 1.6. **Create Connectivity Matrix**: by hitting this button GraphVar will prompt you to a new specification field superimposed on the GraphVar logo. Subsequently you will be pointed to select the time series signals for creation of the subject specific association matrices. The generated connectivity matrices are stored in **“your workspace/results/CorrMatrix”**. Along with the CorrMatrix array, GraphVar creates arrays for the p-values **“PValMatrix”** (parametric p-values)/ **“RandPValMatrix”** (non-parametric p-values) and saves them in the subjects .mat file. The computed connectivity matrices are automatically loaded into the subjects list.

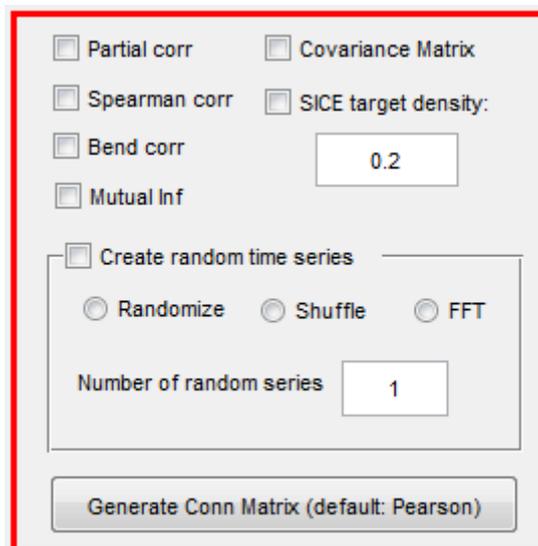


Figure 3. Create connectivity matrix

- 1.7. **Connectivity Matrices**: lets you create association matrices using the subject’s time series signals. You have the option to generate connectivity matrices based on pearson correlation, partial correlation, spearman correlation, percentage bend correlation or mutual information. When selecting a checkbox you will be given a definition by a help dialog. When constructing a network based on partial correlations you will have to be very

careful with a post-hoc redefinition of your network (which will still be possible but may be methodologically invalid). GraphVar also allows computing covariance matrices from input time series that may be used in combination with the “Network Construction -Threshold” option of estimating binary graphs with predefined densities using sparse inverse covariance estimation SICE. In addition to this, you can also produce binary graphs with a specified target density by using SICE in the “Create Connectivity Matrix” option for further usage of your choice.

1.8. **Create random time series:** by non-parametric testing of the correlation strength (r) of any two nodes in the $n \times n$ matrix against a multiple iteration derived distribution of correlation strengths between random time series (pairwise null-model-distribution), this option allows to derive p-values (retrieved by placing the “original” correlation value in the corresponding null-model distribution and determining its percentile position) for each connection in the correlation matrix that can be used for thresholding of the network (i.e. to create the adjacency matrix) or/and for determining a threshold for subsequent association-matrix based statistics (please refer to the *“The Raw Matrix (link wise) panel”*). The creation of random time series can also be used in combination with the *“Partial correlation”* option (in this case - per iteration - a “random” correlation value entering the respective pairwise null-model distribution is controlled for the random time series of the selected nodes in the network). Thresholding based on the non-parametric p-values is performed in the *“Network Construction”* panel under *“Significant/Significance level”*.

1.8.1. **Randomize:** with this option a time course consisting of random normal distributed numbers with the same mean and standard deviation as the reference time course is generated.

1.8.2. **Shuffle:** generates random time series by picking a random time point from a randomly selected other nodal time course (i.e., time series) of the subject until the new random time course consists of the equal number of elements as the original time courses. In this method previously used random time points are acceptable (i.e., sampling with replacement).

1.8.3. **FFT:** this option will cause randomizing the observed time series by taking its Fourier transform, scrambling its phase and then inverting the transform (Prichard and Theiler, 1994; Zalesky et al., 2012).

1.8.4. **Number of random series:** allows setting the number of random time series to be created for each “original” nodal time course. This number equals the amount of iterations for generating the pairwise null-model distributions against which the original r value is tested. A high number of iterations is advisable to obtain valid and reliable non-parametric p-values.

2. The Network Construction panel:

2.1. The network construction parameters selected from the network construction panel will be applied to every subject. This means that each subjects association matrix will accordingly be transformed into subject specific networks (e.g. by applying several network thresholds) and/or several subject specific null-model networks (also refer to "[Appendix 2:](#)" for how to change the threshold range/steps). As many graph topological measures are influenced by the polarity (i.e., positive versus negative) of edge connection weights, GraphVar additionally offers to transform all weights to positive values, absolute values or to set all negative weights to zero.

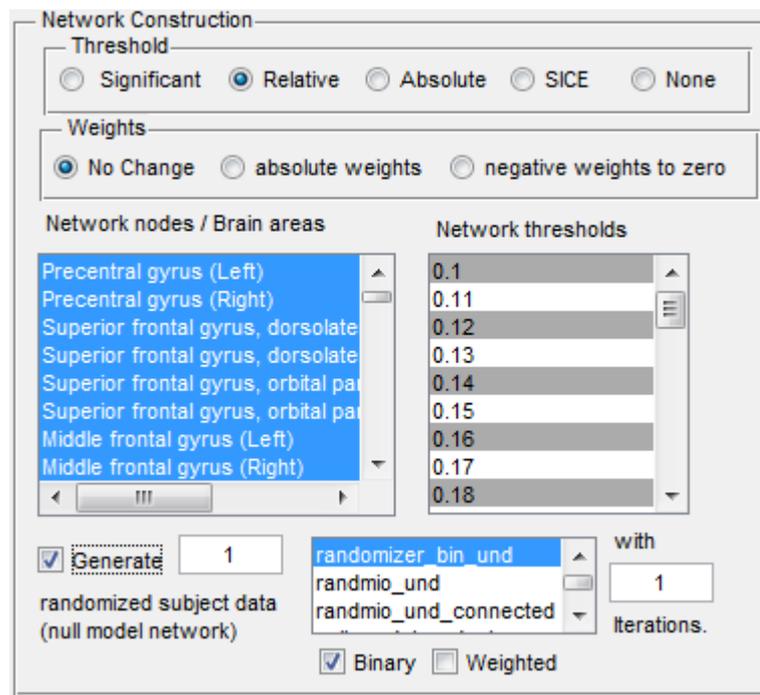


Figure 4. Network Construction panel

2.2. **Network nodes/Brain areas:** displays the labeling of the brain parcellation scheme (e.g. AAL atlas) that you defined with "[The "BrainRegions" file](#)". By default all nodes that are indicated with a "1" in the first column ("[BrainRegions](#)" file) will be selected as being part of the network. However, you can simply redefine your network (e.g. for sub-network analyses) by selecting nodes of your choice with the mouse cursor (mouse + ctrl).

2.3. **Network thresholds:** lets you construct adjacency matrices based on *Significant*, *Relative*, or *Absolute* thresholding. It is advisable to select a range of thresholds (ctrl+mouse) as assumptions about a relationship of independent variables (e.g. age) on only one threshold are not very meaningful (for most graph metrics the "Results viewer" does not allow selection of only one threshold to be displayed). To change the threshold range/ steps please refer to "[Appendix 2:](#)".

- 2.3.1. **Significant:** When selecting this checkbox, a selection window appears which demands you to select the array in which the p-values generated from the “*Create random time series*” option are stored (GraphVar will create arrays “PValMatrix”/ “RandPValMatrix” and save it in the subjects .mat file when generating correlation matrices out of time series signals). Please refer to the “*Create random time series*” option to learn how the p-values are generated.
- 2.3.2. **Relative:** when selecting this option the association matrices will be thresholded in a proportional way (i.e., normalization by “wiring cost” – e.g. 0.1 indicates that 10% of the strongest connections will be maintained as links) to derive the adjacency matrices.
- 2.3.3. **Absolute:** by selection of this option the association matrices will be thresholded in an absolute fashion (i.e., the selected value indicates the correlation coefficient considered as the minimum to display a connection between two nodes – e.g. 0.1 indicates that everything below will not be considered as a links in the adjacency matrices).
- 2.3.4. **None:** GraphVar allows choosing “no thresholding”, which may be used in combination with particular graph topological measures as e.g. modularity, global clustering, etc.
- 2.3.5. **SICE:** constructs binary graphs with predefined densities using sparse inverse covariance estimation “SICE” (please refer to the regular article for more description)
- 2.4. **Generate null-model networks:** allows creating subject specific null-model networks (with the same settings as the original networks – i.e. threshold range, number of nodes) that can be used for normalization of the network topological measures (including calculation of small-worldness), and for non-parametric testing of the significance of correlations among network measures with user defined variables (see “*Test against random networks (graph metrics)*”). The number of subject specific random networks to be created and the function used to generate such a network depends on your input (*randomizer_bin_und*, *randmio_und*, *randmio_und_connected*, *null_model_und_signed*; please refer to the description of these functions on the website of the brain connectivity toolbox; please also refer to “*Appendix 2:*” for how to include your own functions for null-model network generation into GraphVar). The number of iterations refers to the second argument in the randomization functions (e.g., to “*swap_bins*” for the “*null_model_und_sign*” function). For reliable and valid assumptions using null-model networks it is advisable to generate a high number of reference networks. Reference networks created with this option are not used as reference association-matrices as needed for computing network based statistics (see “*The Raw Matrix (link wise) panel*” for info on network based statistics).
- 2.4.1. **Binary/weighted:** This option only refers to the null-model networks and will not impact the nature of the “original” subjects adjacency matrices (their nature is specified by the selection of the version of the graph metric (binary or weighted; exception: small-worldness); see “*Network Calculations panel*”). You will have to choose the nature of these null-model networks (as stated above this choice depends on the nature of the network topological measures you want to compute).

3. The Network Calculations panel:

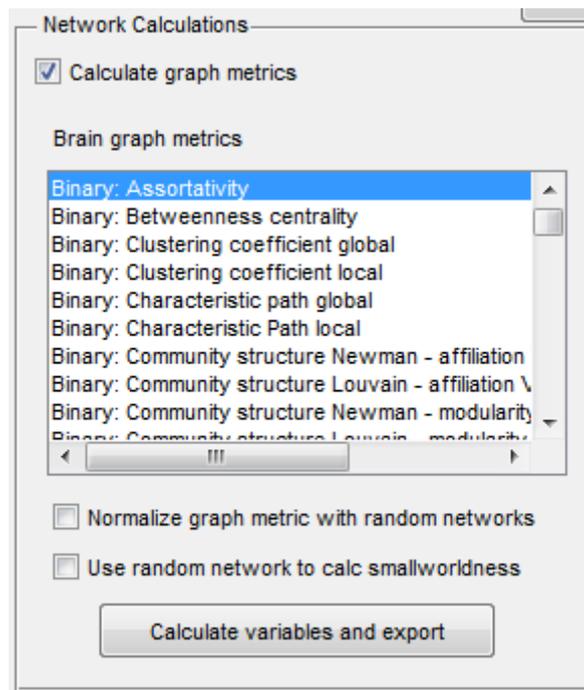


Figure 5. Network Calculations panel

3.1. **Calculate graph metrics:** when selecting this option GraphVar will compute the brain graph metrics that you have chosen in the selection field below (multiple selection is possible; mouse + ctrl). Most of the network topological features are available in their binary or weighted form and the type of the network (binary/weighted) is accordingly generated based on this selection, while you are free to choose both types at the same time (please refer to “[Appendix 1](#)” for the default output of the graph metrics and to “[Appendix 2:](#)” for how to include your own functions into GraphVar).

3.1.1. **TIP:** GraphVar entails a “mouse-over” function and will give you the definition of each graph metric by putting the mouse cursor on the respective function.

3.2. **Normalize graph metrics with random networks:** when you select this option the computed network measures are respectively normalized by the according network measures computed from the subject specific null-model networks (see “[Generate null-model networks](#)”), i.e. normalization by the mean of the null-model derived measures: $GraphVar_{normalized} = (GraphVar_{original_network} / mean(GraphVar_{random_networks}))$. When you use this option, you cannot “Test against random networks” when doing correlations.

3.3. **Use random network to calc smallworldness:** by selecting this feature the measure of small-worldness will be computed: $\sigma = (C_{net} / C_{rand}) / (l_{net} / l_{rand})$

3.4. **Calculate variables and export:** when pushing this button, the selected graph metrics will be computed for each subject (with respect to the settings in the “[Network Construction](#)”

panel) and stored in an excel/csv sheet. When generating subject specific null-model networks, GraphVar will not export each subjects null-model network derived graph metric (on how to retrieve these values, please refer to "[Appendix 2:](#)"), but will still use these values to compute and export the normalized metrics.

4. **The Raw Matrix (link wise) panel:**

4.1. **Introduction:** The "Raw Matrix (link wise)" panel offers to perform network based statistics (association matrix based statistics) by applying either correlational and/or group comparison analyses. The addition "link wise" means that the respective analysis is performed on each link between any two nodes in the $n \times n$ matrix. Thus, this feature is inherently a mass-univariate approach that examines the significance of each link with respect to the applied analysis. However, to deal with the problem of alpha inflation (i.e., multiple comparison problem), GraphVar makes use of so called Graph-Components (i.e., subnetworks in which all pairs of nodes are connected by significant links) that are identified by the BCT "get_components" function. The measure of interest with regard to Graph-Components is their size (this is similar to a contiguous cluster of voxels in fMRI). To evaluate, whether a Graph-Component with a certain size as a result of the respective analysis is non-random, it is compared against the amount and size of Graph-Components in "random" data (please refer to the respective analysis type described below). Based on this, a p-value for each non-random Graph-Component is computed similar as in AFNIs AlphaSim (for how to view the resulting Graph-Components and a more detailed description of the p-value computation for Graph-Components please refer to the "[Network Inspector](#)" section).

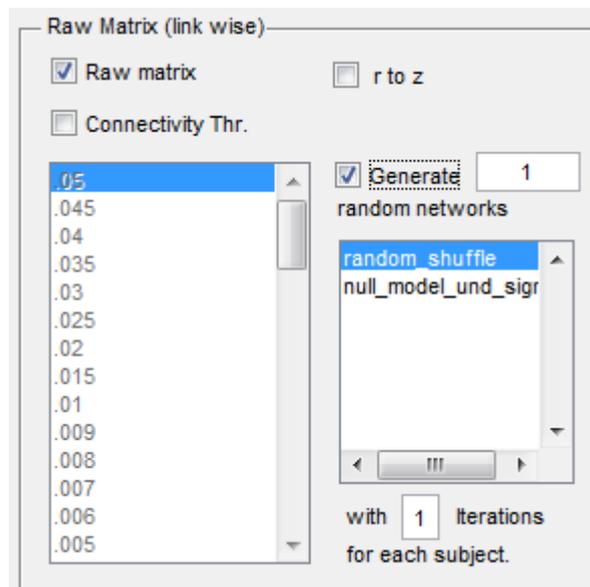


Figure 6. Raw Matrix panel

4.2. **Raw matrix:** when selecting this option GraphVar will perform the above described association matrix based computations.

- **TIP:** with the raw matrix function, it is also possible to examine correlations/group differences between a set of a-priori specified structures (e.g. correlation of age on insula-amygdala connectivity). For this function you may select the specific structures in the “**Network nodes/Brain areas**” selection window. When working with the entire matrix, you can also select specific regions using the “Results viewer” in the “The results selection box”.

4.3. **r to z:** when working with connectivity matrices that entail correlations (r) it is advisable to transform the correlation values in z values using Fisher’s r to z transformation as the r distribution is not a Gaussian distribution.

4.4. **Connectivity Thr.:** this option allows to restrict analyses on links that are significant in the first place. Significance values for each link have to be computed beforehand using the “Create random time series” option. When selecting this checkbox, a selection window appears which demands you to select the array in which the p-values are stored (GraphVar will create this array and save it in the subjects .mat file when generating correlation matrices out of time series signals).

- There are three options (accessible via script “src/calc/CalcCorMatrix.m”, line: “nan_method”) how GraphVar handles the connectivities between nodes that do not reach significance: “mean” (default: for every connection that is not significant the mean connectivity value of the entire sample is used for replacement); “nan” (missing connectivities are treated as Nan); “zero” (missing connectivities are treated as zero).

-> *keep this in mind when interpreting results derived by applying thresholds!*

4.5. **Generate random networks:** this feature is needed if you plan on doing correlational analyses in the raw matrix (for group comparisons refer to the statistics panel). As described in the “The Raw Matrix (link wise) panel” introduction, it is necessary to determine whether a Graph-Component with a certain size as a result of the correlational analysis is non-random. Thus, amount and size of Graph-Components of random networks (random association-matrices) need to be compared against the true Graph-Components. GraphVar offers two options to create subject specific random networks (see below). Subsequently, the subject specific variable selected for correlation (e.g. age) undergoes link wise correlational analyses on the original and the random networks. Only significant correlations of the user defined variable (e.g. age) with a network link are considered for Graph-Components (please refer to the “Network Inspector” section for determining the significance value of links considered for Graph-Components and for more info about p-values for true Graph-Components). To actually test against the random networks you have to select the “**Test against random networks (correlation)**” checkbox in the statistics panel. When selecting this option GraphVar simultaneously performs non-parametric testing of the correlation strength, which gives you the option to choose between parametric and non-parametric p-values for inspection of single links. However, significant links in the Graph-

Components are determined by the parametric p-values as it would be circular to create p-values from random data and then use the same random data again for Graph-Components (for more info refer to the “*Test against random networks (graph metrics)*” option in the “*Network Calculations*” panel).

4.5.1.1. **Random_shuffle:** by using this option the values comprising the subject’s association matrix are randomly shuffled using the “random_shuffle” function implemented in GraphVar (*src/calc/helper* folder) keeping the degree of positive and negative weights for the random networks the same as in the original network. Compared to the “null_model_und_signed” function this option is less conservative but very fast. Random_shuffle does not need an input for number of iterations.

4.5.1.2. **Null_model_und_sign:** uses the “null_model_und_sign” function of the BCT. The number of iterations refers to the “swap_bins” argument in the function.

5. The Statistics panel:

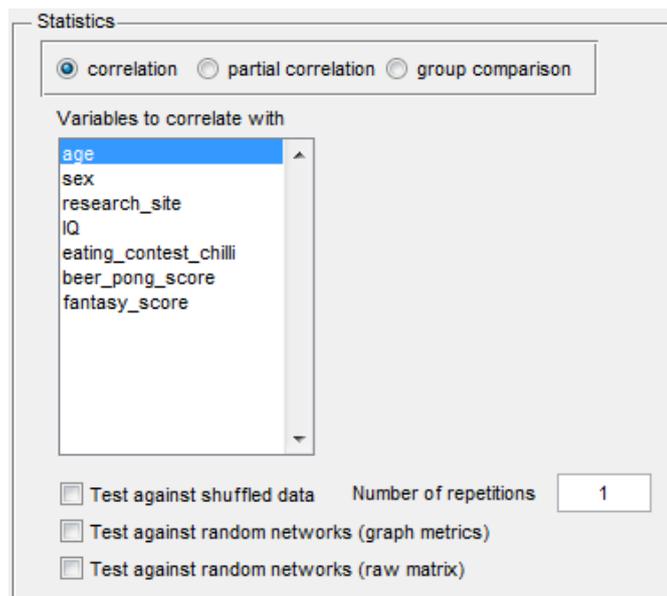


Figure 7. Correlate panel

5.1. **Do correlation/do partial correlation:** GraphVar enables you to compute correlations between specific graph metrics (see “*Network Calculations panel*”) and user defined variables (see “*Variables*”). GraphVar offers the selection between a regular correlation (pearson correlation) and a partial correlational analysis. When you select the “*Do partial correlation*” checkbox, you are demanded to select the respective variables in the selection box below that you want to control for (variables coded as string will not work). Selection of multiple variables for correlation and as covariates is possible. If you want to do non-parametric testing of the correlation strength (r) please refer to the “*Test against random networks (graph metrics)*” or “*test against shuffled data*” option.

- 5.2. **Test against random networks (graph metrics):** this feature allows non-parametric testing of the correlation strength (r) between a network measure with any user defined subject variable (e.g. age) against the correlation between the graph theoretical measure derived from the subject specific null-model network (see “[Generate null-model networks](#)”) with the same subject variable (e.g. age). Based on the number of subject specific null-model networks a null-model distribution of these “random” correlations (prior transformation of r to z values using Fishers transformation) is generated from which a non-parametric p-value for the “original” correlation is retrieved by placing the “original” correlation value in the corresponding null-model distribution and determining its percentile position.
- 5.3. **Test against shuffled data:** Additionally to performing non-parametric testing on the network level (i.e., rewiring of connections in the network), GraphVar also provides generation of null-distributions on the metric level when performing correlational analyses. In this case the network resulting topological measures are randomly shuffled among individuals and the correlation procedure is performed repeatedly resulting in a non-parametric correlation distribution.
- 5.4. **Test against random Networks (raw matrix):** if you wish to perform correlational analyses on the raw matrices it is necessary to compute random networks (please refer to the description in the raw matrix panel)

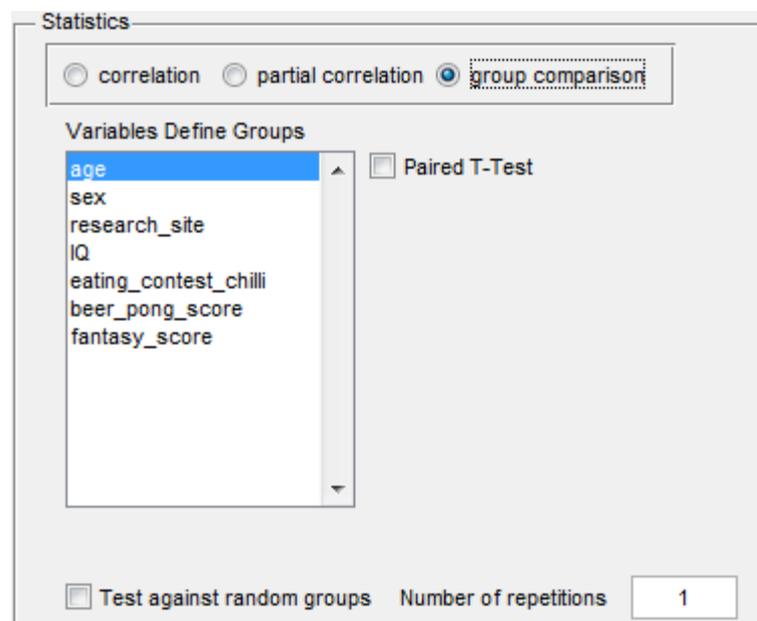


Figure 8. Group Comparison panel

- 5.5. **Do Group Comparison:** GraphVar enables you to compute group comparisons with specific graph metrics or the raw matrices as dependent variable (see “[Network Calculations panel](#)”), whereas the group membership of each subject is specified by the user defined variable, e.g. gender, genotype, etc. (see “[Variables](#)”). With GraphVar you have the possibility to compute comparisons between 2 groups (i.e., 2 sample t-test) or to find differences between more than two groups (i.e., ANOVA). GraphVar will automatically

detect how many groups are encoded in the selected variable and subsequently apply the appropriate test (i.e., two sample t-test or ANOVA). Selection of multiple variables for group comparisons is possible.

5.6. **Test against random groups:** when selecting this option GraphVar performs a non-parametric permutation test with a user defined amount of repetitions. For permutation testing, in each repetition, the network measures (or connectivity values in case of *"The Raw Matrix (link wise) panel"* calculations) of each subject are randomly reassigned to one of the groups so that each randomized group obtains the same number of subjects as the original groups. Then, the differences in network measures between randomized groups are calculated resulting in a permutation distribution of difference under the null hypothesis (for ANOVA also the F value deriving from each repetition is computed). The actual between-group difference in network measures (and for ANOVA also the F value) is then placed in the corresponding permutation distribution and a p-value is calculated based on its percentile position. For how to correct the result by applying the non-parametric p-values please refer to the *"Results"* section under *"Correction"*.

5.7. *It is also possible to perform group comparison analyses on the "raw" correlation/association matrices (i.e., association-matrix based statistics, where no graph topological measures are involved).*

5.8. **Raw Matrix – Group:** if you wish to perform group comparisons on the raw matrices it is necessary to select the *"Do Group Comparison"* checkbox in the *"Group Comparison"* panel.

5.8.1. As described in the *"The Raw Matrix (link wise) panel"* introduction, it is necessary to determine whether a Graph-Component with a certain size as a result of the correlational analysis is non-random. Thus, amount and size of Graph-Components resulting from "random data" are needed to be compared against the true Graph-Components. For group comparisons on the raw matrix GraphVar produces these "random data" by performing non-parametric permutation tests with the subjects underlying connectivity values for the respective groups (to get more info on this refer to *"Test against random groups"* option in the *"Group Comparison"* panel). Thus, to produce this random data you need to select the *"Test against random groups"* option in the group comparison panel and enter the amount of repetitions. When selecting this option GraphVar simultaneously performs parametric testing of the group difference on each link, which gives you the option to choose between parametric and non-parametric p-values for determining for inspection of single links. However, significant links in the Graph-Components are determined by the parametric p-values as it would be circular to create p-values from random data and then use the same random data again for Graph-Components (please refer to the *"Network Inspector"* section for determining the significance value of links considered for Graph-Components and for more info about p-values for true Graph-Components).

3. Results viewer

- The results viewer is basically divided into three panels. “*The results selection box*” and the “*general functions panel*” are located on the upper part of the results viewer, whereas an adaptive “*results display*” panel is located on the bottom of the viewer.

1. The results selection box:

- The results selection box contains of four selection windows: Variable, Graphvar, Threshold and Brain Areas.
- If you get the prompt “*Please deselect at least one filter*” it means to rethink the results selection (most likely it means that you have to choose at least two thresholds)

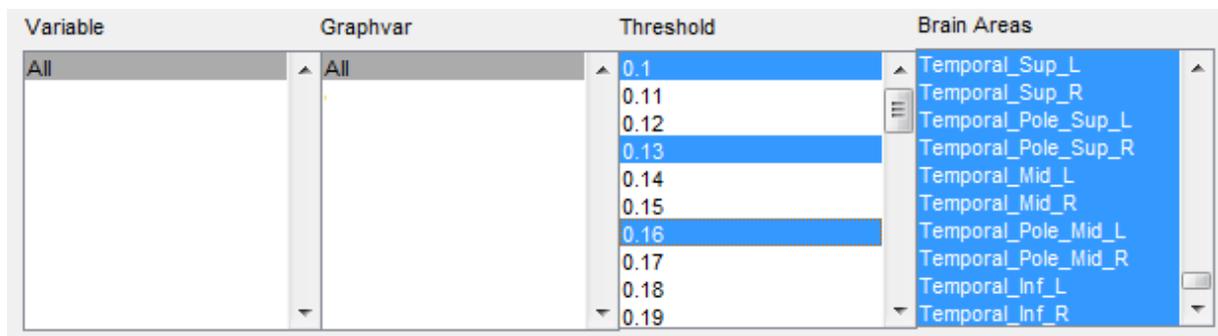


Figure 9. Results selection box

- 1.2. **Variable:** this field contains the independent variables you selected for analyses (i.e., the subjects variable as e.g. age). Selection of a specific variable will determine the results displayed. For “*one-dimensional*” graph metrics (as e.g. the global clustering coefficient) a simultaneous selection of multiple (ctrl+mouse) independent variables is possible (e.g. to explore the correlation of several variables to the graph topological measure). However, for viewing results of a group analysis it is recommended to only select one independent variable (e.g. sex) at a time. For “*more-dimensional*” graph variables as the local clustering coefficient, selection of more than one independent variable is not possible.
- 1.3. **Graphvar:** this selection window contains the dependent variables (i.e., graph metrics) you selected for analyses (also refer to “*Appendix 1*”). Selection of a specific variable will determine the results displayed. For “*one-dimensional*” graph metrics a simultaneous selection (ctrl+mouse) of multiple graph metrics is possible (e.g. to explore the relation of an independent variable to the different graph topological measures). For “*more-dimensional*” graph variables, selection of more than one dependent variable is not possible. If doing analyses on the raw connectivity matrices, the graph variable will be displayed as “*corr_area*” (when computing statistical analyses on a “*regular*” graph variable and *corr_area* simultaneously, you will have to double click *corr_area* for displaying results in the results display.

- **“one-dimensional”** : 1. dimension= network thresholds (metrics: e.g. all global metrics as global clustering coefficient, global path length, transitivity, etc.).
- **“two-dimensional”** : 1. dimension= network thresholds, 2. dimension= network nodes (metrics: e.g. all local metrics as local clustering coefficient, nodal path length, local efficiency).
- **“three-dimensional”**: 1. dimension= network thresholds, 2. dimension and 3. dimension= network nodes by network nodes as n x n (metrics: e.g. edge betweenness centrality, topological overlap, raw connectivity matrix).

1.4. **Threshold:** this field contains the different network thresholds selected in the “Network Construction” panel, on which the respective analyses were performed. For all graph topological measures (except “three-dimensional” metrics) it is mandatory to select at least two thresholds (when selecting one threshold no results will be displayed – this is to prevent you to only examine relationships of variables to the graph metrics on a certain threshold). For “three-dimensional” metrics it is only possible to select one threshold at the time. In general you are not obliged to select the full range of thresholds that were computed but can also select just specific thresholds by using ctrl+mouse.

1.5. **Brain Areas:** this display contains all the network nodes/brain areas that were selected via the “Network Construction” panel. For “one-dimensional” graph metrics selection of nodes in this window will not affect the displayed results. However, when “more-dimensional” graph variables are displayed the content of the results window will change according the highlighted nodes. For example if you have an a-priori idea about specific nodes in your network with respect to some graph metrics and their relation to independent variables, or want to examine the relationship of e.g. age on amygdala-insula “raw” connectivity, then simply select the desired nodes from the menu (ctrl+mouse).

2. The results display:

- The results display is adaptive to what you select in the “The results selection box” and the displayed “X- and Y-axes” will change according to the dimensionality of the graph metrics (see **figures 9-11**).

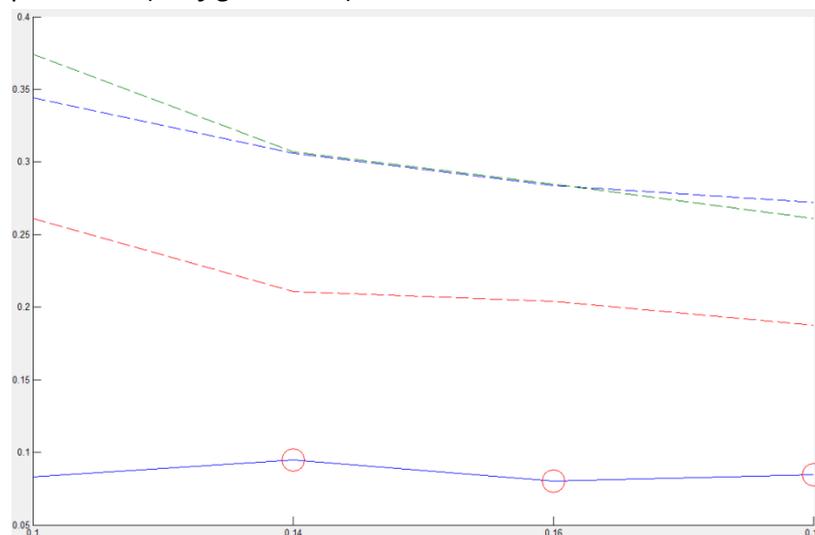


Figure 10. Results display for “one-dimensional” graph variables

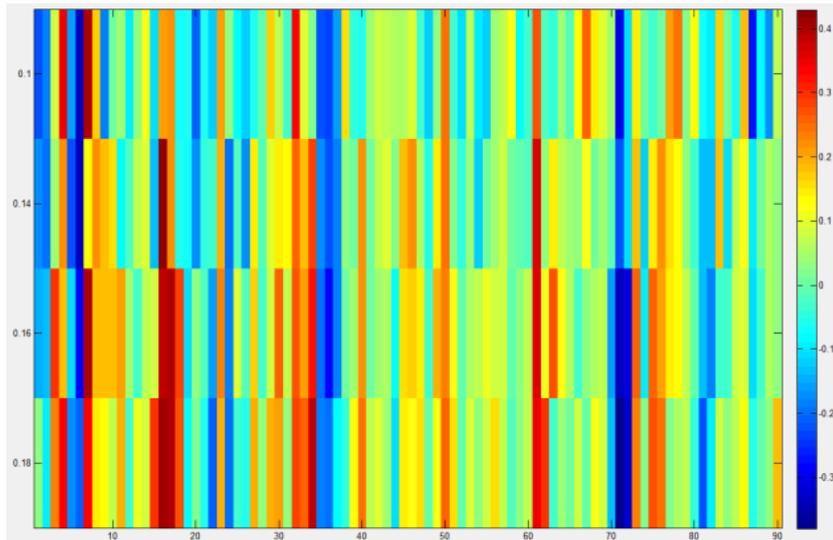


Figure 11. Results display for “two-dimensional” graph variables

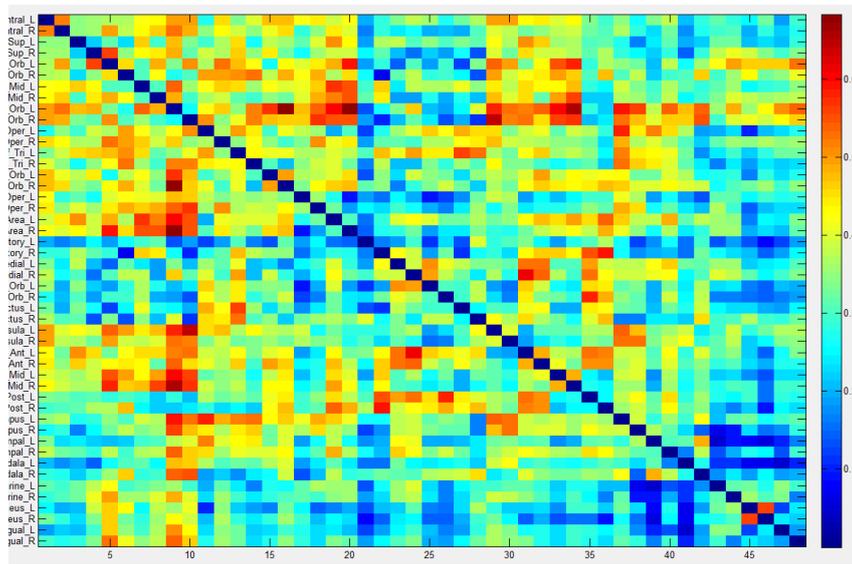


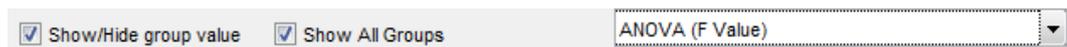
Figure 12. Results display for “three-dimensional” graph variables

- Moveable legend-boxes (drag and drop style for “one-dimensional” graph metrics) or a mouse-over display (“more-dimensional” graph metrics; **Figure 12**) indicate what is respectively displayed.

Frontal_Sup_L
 Threshold: 0.13
 1.5466
 0.21484

Figure 13. Mouse-over info box for “more-dimensional” graph metrics: The upper line indicates the network node/brain area; the second line indicates the respective threshold; third line: level/degree of the computed variable (for correlation: r ; for group comparison either F -value or group difference); fourth line: p -value.

- 2.1. **X- and Y-axes:** *“one-dimensional”* graph metrics: x-axis= threshold, y-axis= *“level/degree of the computed variable”*; *“two-dimensional”* metrics: x-axis= nodes/brain areas, y-axis= threshold, level/degree of the computed variable= color coded; *“three-dimensional”* metrics: x-axis and y-axis= network nodes/brain areas, threshold= to select in the *“Threshold”* window in the results selection box; level/degree of the computed variable= color coded.
- **INFO:** *level/degree of the computed variable* is dependent on what kind of statistical analysis you have computed and what you decide to display in the results window. For correlational analyses this refers to the correlation coefficient. For group analyses this can refer to the ANOVA F-value or the group difference.
 - **TIP:** you can easily modify the **properties of the color bar** located on the right of the results display by using a right mouse click. Besides different colormaps (e.g. gray scale), you can also make use of an interactive colormap shift.
- 2.2. **Group-comparison:** when selecting results of a group comparison analysis, GraphVar will enable new “options” just below the results display.



- 2.2.1. **Show/Hide group value:** (only available with *“one-dimensional”* graph metrics) in addition of displaying either the mean group difference on each threshold or the F-value resulting an ANOVA, selection of this check box will additionally show each groups mean of the computed graph metric in the selected threshold range. The Y-axis is subsequently rescaled. To show the group means when the F-value is selected to display, both check boxes (*“Show/Hide group value”* and *“Show All Groups”*) must be selected.
- 2.2.2. **Show All Groups:** (only available with *“one-dimensional”* graph metrics) when computing an ANOVA, this check box will show the mean values of all groups even if you have only two groups selected for display.
- 2.2.3. **Selection field:** this selection field is always present when displaying results of group comparisons and lets you select to either to display the F-value (in case of an ANOVA) or the respective group differences. Once selected this field with the mouse cursor, you can use the keyboard arrows to skip through the selection fields.

3. The general functions panel:

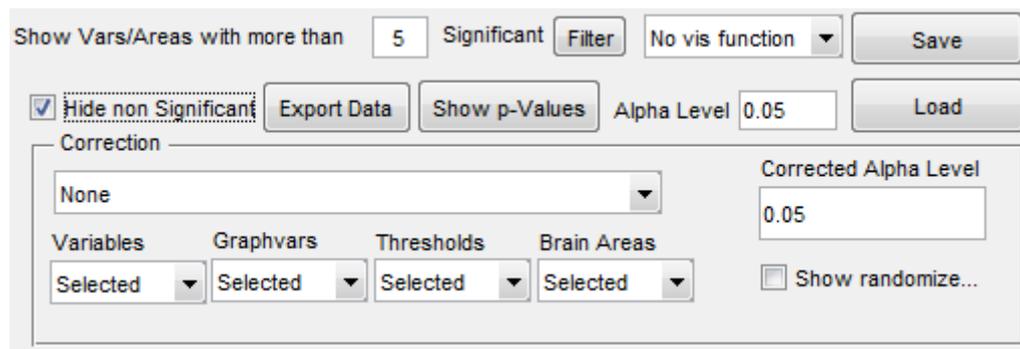


Figure 14. General functions panel

3.1. **Alpha Level:** here you can specify the level of significance you want to apply to the results. Depending on the applied "Correction" method, the field below ("Corrected Alpha Level") will change according to the reference Alpha level that you input. By default, for "one-dimensional" graph metrics, red circles indicate significance on each network threshold. For how to highlight significant results in "more-dimensional" graph metrics please refer to the "Hide non-significant" function. Important to keep in mind is the directedness of the significance testing (i.e., two-tailed vs. one-tailed):

- **Parametric testing derived p-values: are always two-tailed.**
- **Non-parametric testing derived p-values: one-tailed for group comparison and two-tailed for correlations. P-values will depend on the number of null-models (e.g., p-value for one-tailed testing with 100 null models can be minimally 0.01; two-tailed: 0.02).**
- **CAVEAT for Group comparisons:** p-values are one-tailed. Thus, when using non-parametric p-values without an a-priori hypothesis of the directedness of the group comparison it is necessary to change the "Alpha Level" accordingly (e.g. to 0.025 when planning to test two-tailed on 0.05). For ANOVA F-values the alpha level has not to be changed as the "real" F-value is always tested to be higher as F-values derived from randomness (i.e., one-sided).

3.2. **Hide non-significant:** this option is only available for "more-dimensional" graph metrics and will cause the results display to only show the computed values that are significant according to the specified alpha level. By default, for "one-dimensional" graph metrics, red circles indicate significance on each network threshold.

3.3. **Show p-values:** this function allows displaying p-values of computed measures. P-values for the "Correction" methods "None", "Bonferroni correction", "FDR correction" are naturally the same and are always of parametric nature. P-values for the correction method "Random Networks/Groups" are non-parametric and derive from "Test against random networks (graph metrics)" or "Test against random groups" (see "Analysis setup" section). When displaying non-parametric p-values for "one-dimensional" graph metrics, the parametric p-values are simultaneously displayed as a reference.

3.4. **Correction:** this function allows applying several correction methods to your results. Even if you have computed results by testing against random networks/random groups (see "[Test against random networks \(graph metrics\)](#)" and "[Test against random groups](#)") the correction methods "None", "Bonferroni correction", "FDR correction" will still apply to the parametric p-values which are simultaneously derived in GraphVars non-parametric testing.

3.4.1. **None:** parametric p-values without correction.

3.4.2. **Bonferroni correction:** when using this option a regular Bonferroni correction is applied with respect to the reference "[Alpha Level](#)". By default, when performing this correction, the number of comparisons accounted for is based on "everything you see in the results display"/selected in "[The results selection box](#)" (e.g. number of independent variables x number of graph metrics x number of thresholds). However, you can also specify the number of comparisons by making use of the drop-down fields below the correction drop-down menu ("[Variables](#)", "[Graphvars](#)", "[Thresholds](#)", "[Brain Areas](#)"). According to the Bonferroni correction, the field "[Corrected Alpha Level](#)" displays the new maximum Alpha Level and the "[Hide non-significant](#)" function will apply to it.

3.4.3. **FDR correction:** when using this option an FDR correction algorithm (*written by Arnaud Delorme, Salk Institute; Reference: Benjamini & Yekutieli (2001) The Annals of Statistics*) is carried out with respect to the reference "[Alpha Level](#)". By default, the number of significant results entering the FDR correction is based on "everything you see in the results display"/selected in "[The results selection box](#)". However, you can also change the number of significant results entering the FDR correction by making use of the drop-down fields below the correction drop-down menu ("[Variables](#)", "[Graphvars](#)", "[Thresholds](#)", "[Brain Areas](#)"). According to the FDR correction, the field "[Corrected Alpha Level](#)" displays the new maximum Alpha Level and the "[Hide non-significant](#)" function will apply to it.

3.4.4. **Random Networks/Groups:** this correction option assumes that you have computed results by testing against random networks/random groups. Please refer to "[Test against random networks \(graph metrics\)](#)" and "[Test against random groups](#)" in the "Analysis setup" section to learn how these non-parametric p-values are computed.

3.5. **Show randomize...:** this option is enabled if you have computed results by testing against random networks/random groups and is only available for "[one-dimensional](#)" graph metrics (an equivalent option exists/appears when calculating on the raw association matrices and relates to the "[Network Inspector](#)"). Hitting this button will show the results of the "random data" depending on the analysis type:

- **Correlation:** the gray dotted lines in the results display show the correlations of the subject specific null-model networks with the independent variable and can basically be understood as the null-model distribution from which the non-

parametric p-value is calculated (please refer to "*Test against random networks (graph metrics)*" for more information).

- **Group analysis:** Depending on your selection in the "*Group-comparison*" options below the results window, the meaning of the gray dotted lines in the results display varies. ANOVA-F: shows the F-values of the ANOVAs on the permuted groups. Group differences: shows the difference of the random permuted groups. Depending on the direction of the group comparison the dotted lines can be understood as the permutation distribution of difference under the null hypothesis from which the non-parametric p-value is calculated (please refer to "*Test against random groups*" for more information).

- 3.6. **Show Vars/Areas with more than __ Significant Filter:** this function allows filtering for variables or areas with a specified number of significant associations across the network threshold range. With "*one-dimensional*" graph variables and e.g. a correlational analysis of several independent variables (e.g. several scores of neuropsychological tests) this may allow to directly highlight the independent variables that show an association to the graph metric on a specified number of thresholds (however, the number does not imply that the thresholds must be adjacent to each other). With "more-dimensional" graph variables this filter will apply directly to the network nodes/brain areas and will automatically highlight the selected nodes in the "*Brain Areas*" selection window (the results display will adapt accordingly).
- 3.7. **Visualization function- "No vis function" :** this drop-down window entails some visualization functions as offered by the BCT. These functions should and can be applied only on the results of raw matrix (association matrix) computations (see also "*The Raw Matrix (link wise) panel*"). It is important to notice that the "*reorder_mod*" function, which reorders the nodes according to modules, only works with positive values. Thus, computed values (e.g., correlations) will be transformed to absolute values (it is the user's responsibility to make proper interpretations by applying these visualization functions). For a description of the implemented functions please refer to the BCT (<https://sites.google.com/site/bctnet/visualization>).
- 3.8. **Export Data:** by using this option, GraphVar will export the results shown in the results display in an excel/csv file. Depending on the results displayed you will receive different sheets per file covering F-value or group difference or correlation including the respective p-values.
- 3.9. **Save:** You can save the computed results for a later review in the results viewer. This is strongly recommended as it will save you a lot of computing time once all the desired calculations are finished.

4. Network Inspector

1. The Network Inspector option is enabled when doing association matrix based statistics (to learn more refer to "*The Raw Matrix (link wise) panel*" in the "Analysis setup" section). By hitting the "Show Network" button (figure 14), the Network Inspector window will open. The Network Inspector consists of three parts (results panel on the left; options bar at the top; network display on the right).

Figure 15. Show Network

2. The results panel:

| | Size of Network | Count in real data | Mean count in random data | Max Frequency | Alpha |
|----|-----------------|--------------------|---------------------------|---------------|--------|
| 1 | 1 | 30 | 27.7800 | 0 | 1.0000 |
| 2 | 2 | 2 | 3.8200 | 11 | 1.0000 |
| 3 | 3 | 1 | 1.0500 | 30 | 0.8900 |
| 4 | 4 | 0 | 0.3800 | 32 | 0.5900 |
| 5 | 5 | 1 | 0.1600 | 12 | 0.2700 |
| 6 | 6 | 0 | 0.1000 | 9 | 0.1500 |
| 7 | 7 | 0 | 0.0100 | 1 | 0.0600 |
| 8 | 8 | 0 | 0.0200 | 2 | 0.0500 |
| 9 | 9 | 0 | 0.0200 | 2 | 0.0300 |
| 10 | 10 | 0 | 0.0100 | 1 | 0.0100 |
| 11 | 11 | 0 | 0 | 0 | 0 |

Figure 16. Results panel

- As stated in the "*The Raw Matrix (link wise) panel*", GraphVar makes use of so called Graph-Components (i.e., subnetworks in which all pairs of nodes are connected by significant links) that are identified by the BCT "*get_components*" function. Links considered for Graph-Components are either based on significant correlations of the independent variables (e.g. age) or significant group differences (or F-values). The significance level of a Graph-Component is achieved through a combination of probability threshold (i.e., the alpha level of each link) and Graph-Component size threshold (i.e., the size of the sub-network). The initial alpha level has to be specified in the "*Alpha input-box*" on the top of the results table.

- The program output contains a table with five columns. The first column lists different Graph-Component sizes (in participating nodes). Graph-Components of size one are disconnected nodes (disconnected through the initial Alpha; see *“Alpha input-box”*). The second column lists the frequency of occurrences for Graph-Components in the real data of the given size in the first column. For example, in the real data a Graph-Component of size 5 occurred one time. The third column lists the mean count of graph-Components in the random data of size as given in column one. For example, in the 100 comparisons of connectivity differences in the entire network using random permuted groups, a Graph-Component with the size 5 occurred 16 times (i.e., mean count in random data of 0.16). The fourth column lists the number of times, out of the 100 comparisons on random data, that the maximum size of a Graph-Component in the network was as given by column 1. For example, in 12 of the 100 “random” comparisons, the largest Graph-Component in the entire network contained exactly 5 nodes. Therefore, if the Graph-Component size listed in column one is used as the minimum Graph-Component size threshold, then column five estimates the probability of a false detection occurring in the “real” network (in this example the Graph-Component of size 5 in the real data would not be significant).
- ANOVA Graph-Components:** Graph-Components derived from an ANOVA always refer to links with a significant F-value (initial Alpha: F-value). Thus, to skip through the links of the Graph-Component and to see which group contributed how to each link you can select the respective group comparison in the options bar. Links where the selected groups do not differ significantly are indicated by dotted lines (see figure 16).

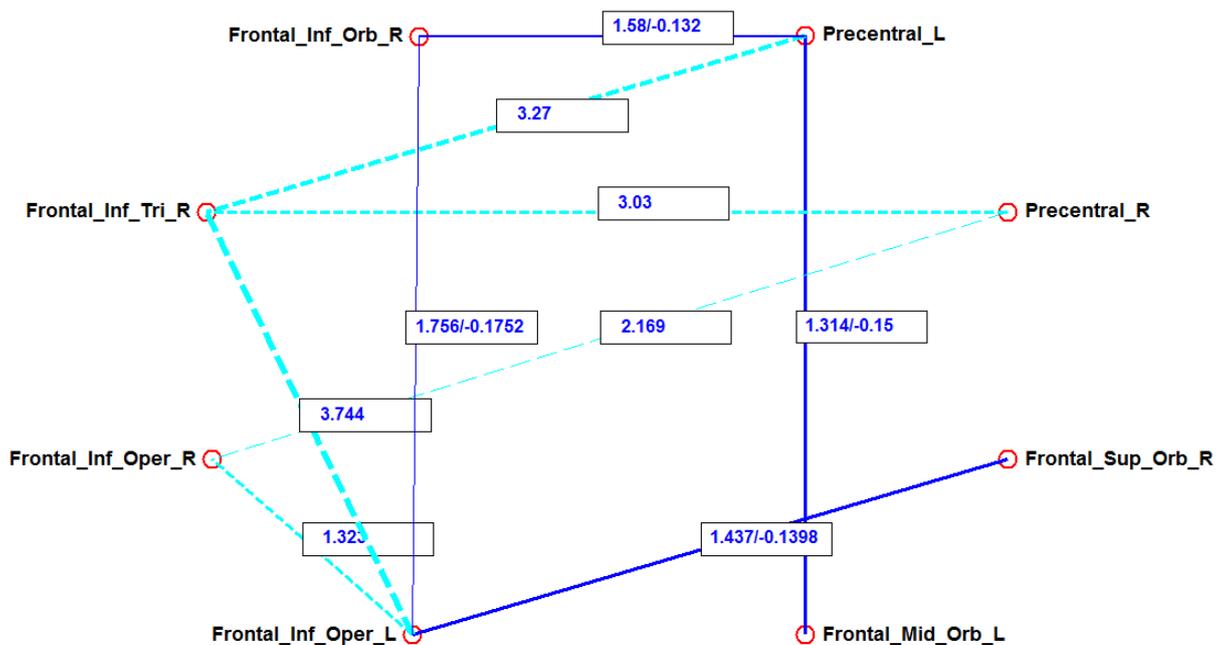


Figure 17. ANOVA Graph-Component

3. The options bar:

- The option bar entails several functions that directly apply to the Graph-Components.
 - 3..1. **Alpha input-box:** this is the initial significance level which is used to determine whether a link is considered for being part of Graph-Components. The p-values, which are used for link-thresholding are always parametric p-values as it would be circular to create p-values from random data and then use the same random data again to compare the size of Graph-Components.
 - 3..2. **Show all labels:** when selecting this check-box, the respective values of your computation are superimposed on the Graph-Component that is displayed in the network display. For correlational analysis r is displayed. For two-sample t-tests the respective group difference (according to the selected contrast) is displayed. For ANOVAs the first value denotes the F-value, whereas the second value denotes the group difference (according to the contrast; see figure 16).
 - 3..3. **Enable mouse-over:** when this check-box is selected, it is possible to show each nodes connections by pointing with the mouse on the respective node.
 - 3..4. **Previous/Next:** when there is only one node in the network, you can use these bottoms to toggle between the nodes.

4. The network display:

- The identified Graph-Components are shown in this window. The line thickness between nodes indicates the strength of the computed measure on the respective link.

Appendix 1: To update the BCT functions, simply replace the BCT folder under “GraphVar\src\ext”
 When doing so, keep in mind that GraphVar uses the first output (return value) of the functions for computations (also refer to Appendix 2 for more info).

| function name | binary/ weighted | function description | default output for calculations |
|---|-----------------------------|--|--|
| 'Assortativity', 'assortativity_bin2' | b | The assortativity coefficient is a correlation coefficient between the degrees of all nodes on two opposite ends of a link. A positive assortativity coefficient indicates that nodes tend to link to other nodes with the same or similar degree. | r, assortativity coefficient |
| 'Betweenness centrality', 'betweenness_bin', 'betweenness_wei' | b/w | Nodes with high values of betweenness centrality participate in a large number of shortest paths. | BC, node betweenness centrality vector. |
| 'Clustering coefficient global', 'clusterMean_bu', 'clusterMean_wu' | b/w | The clustering coefficient is the fraction of triangles around a node and is equivalent to the fraction of node's neighbors that are neighbors of each other. Mean value above all brain regions. | C, clustering coefficient vector (mean value for the network) |
| 'Clustering coefficient local', 'clustering_coef_bu', 'clustering_coef_wu' | b/w | The clustering coefficient is the fraction of triangles around a node and is equivalent to the fraction of node's neighbors that are neighbors of each other. | C, clustering coefficient vector |
| 'Characteristic path global', 'charpath_B', 'charpath_W' (helper function as distance matrix is required) | b/w | The characteristic path length is the average shortest path length in the network. | lambda, characteristic path length |
| 'Characteristic path local', 'nodalpath_bin', 'nodalpath_wei' (helper function as distance matrix is required) | b/w | The nodal characteristic path length is the average shortest path length of the node to the rest of the network | Lambda node, characteristic path length for each node |
| 'Community structure Newman Affiliation Vector', 'modularity_und' 'Community structure Louvain Affiliation Vector', 'modularity_louvain_und' | b/w | The optimal community structure is a subdivision of the network into nonoverlapping groups of nodes in a way that maximizes the number of within-group edges, and minimizes the number of between-group edges. The modularity is a statistic that quantifies the degree to which the network may be subdivided into such | Community number of each node (community vector C _i) |

| | | | |
|---|-----|---|--|
| 'Community structure Newman - modularity Output', 'modularity_QOut_und' | | clearly delineated groups. | Maximized modularity Q |
| 'Community structure Louvain - modularity Output', 'modularity_louvain_QOut_und' | | | |
| (helper function as community structure is required) | | | |
| 'Degree', 'degrees_und' | b/w | Node degree is the number of links connected to the node. Connection weights are ignored in calculations. | deg, node degree |
| 'Density', 'density_und' | b/w | Is the fraction of present connections to possible connections. Connection weights are ignored in calculations. | kden, density |
| 'Distance Matrix', 'distance_bin', 'distance_wei' | b/w | The distance matrix contains lengths of shortest paths between all pairs of nodes. | D, distance matrix |
| 'Eccentricity', 'charpath_B_ecc', 'charpath_W_ecc' | b/w | The eccentricity of a vertex v is the greatest geodesic distance between v and any other vertex. It can be thought of as how far a node is from the node most distant from it in the graph (average largest path length). | ecc, eccentricity (for each node) |
| (helper function as distance matrix is required) | | | |
| 'Edge betweenness centrality', 'edge_betweenness_bin', 'edge_betweenness_wei' | b/w | Edge betweenness centrality is the fraction of all shortest paths in the network that contain a given edge. Edges with high values of betweenness centrality participate in a large number of shortest paths. | EBC, edge betweenness centrality matrix |
| 'Efficiency global', 'efficiency_bin', 'efficiency_wei' | b/w | The global efficiency is the average inverse shortest path length in the network, and is inversely related to the characteristic path length. | Eglob, global efficiency (scalar) |
| 'Efficiency local', 'efficiency_local_bin', 'efficiency_local_wei' | b/w | The local efficiency is the global efficiency computed on node neighborhoods, and is related to the clustering coefficient. | Eloc, local efficiency (vector) |
| (helper functions) | | | |
| 'Eigenvector centrality', 'eigenvector_centrality_und' | b/w | Eigenvector centrality is a self-referential measure of centrality -nodes have high eigenvector | v, eigenvector associated with the largest |

| | | | |
|--|-----|--|---|
| | | centrality if they connect to other nodes that have high eigenvector centrality. | eigenvalue of the adjacency matrix CIJ |
| 'Flow coefficient local', 'flow_coef_bd' | b | The flow coefficient is similar to betweenness centrality, but computes centrality based on local neighborhoods. The flow coefficient is inversely related to the clustering coefficient. | fc, flow coefficient for each node |
| 'Flow coefficient global', 'flow_coef_bd_FC' (helper function) | b | Average flow coefficient of the network | FC, average flow coefficient over the network |
| 'Graph radius', 'charpath_B_radius', 'charpath_W_radius' (helper function as distance matrix is required) | b/w | The radius of the graph | radius, radius of graph |
| 'Graph diameter', 'charpath_B_diameter', 'charpath_W_diameter' (helper function as distance matrix is required) | b/w | The diameter of the graph | diameter, diameter of graph |
| 'K-coreness centrality', 'kcoreness_centrality_bu' | b | The k-core is the largest subgraph comprising nodes of degree at least k. The coreness of a node is k if the node belongs to the k-core but not to the (k+1)-core. | coreness, node coreness |
| 'Matching index', 'matching_ind_Mall' (helper function) | b | The matching index computes for any two nodes u and v, the amount of overlap in the connection patterns of u and v. Self-connections and u-v connections are ignored. The matching index is a symmetric quantity, similar to a correlation or a dot product. | Mall, matching index for all connections |
| 'Neighborhood overlap', 'edge_nei_overlap_bu' | b | These functions compute the overlap between the neighborhoods of pairs of nodes linked by edges. | EC, edge neighborhood overlap matrix |
| 'Rich Club Coefficient' (helper function) If klevel is not included the the maximum level will be set to the maximum degree of CIJ -> this may be changed. | b | The rich club coefficient, R, at level k is the fraction of edges that connect nodes of degree k or higher out of the maximum number of edges that such nodes might share. | vector of rich-club coefficients for levels 1 to k level. |
| 'Strength', 'strengths_und' | w | Node strength is the sum of weights of links connected to the node. | str, node strength |

| | | | |
|---|-----|---|---|
| 'Subgraph centrality', 'subgraph_centrality' | b | The subgraph centrality of a node is a weighted sum of closed walks of different lengths in the network starting and ending at the node. | Cs, subgraph centrality |
| 'Shannon-entropy based diversity coefficient_POS', 'diversity_coef_sign_POS' (helper function as community structure is required) | w | The Shannon-entropy based diversity coefficient (based on positive connections) measures the diversity of intermodular connections of individual nodes and ranges from 0 to 1. | Hpos, diversity coefficient based on positive connections |
| 'Topological overlap', 'gtom2' (helper function as steps are required) ➔ Number of steps is set to 5 ➔ This may be changed in script | b | The m-th step generalized topological overlap measure quantifies the extent to which a pair of nodes have similar m-th step neighbors (nodes that are reachable by a path of at most length m). | gt, GTOM matrix |
| 'Transitivity', 'transitivity_bu', 'transitivity_wu' | b/w | The transitivity is the ratio of triangles to triplets in the network and is an alternative to the clustering coefficient. | T, transitivity scalar |
| 'Within-module degree z-score', 'module_degree_zscore2' (helper function as community structure is required) | b/w | The within-module degree z-score is a within-module version of degree centrality. This measure requires a previously determined community structure (accomplished by helper function). | Z, within-module degree z-score (with respect to the community affiliation vector) |
| 'Participation coefficient', 'participation_coefficient2' (helper function as community structure is required) | b/w | Participation coefficient is a measure of diversity of intermodular connections of individual nodes. This measure requires a previously determined community structure (accomplished by helper function). | P, participation coefficient (with respect to the community affiliation vector) |
| 'Louvain modularity algorithm (neg weights)', 'modularity_louvain_und_sign' 'Fine-tuning modularity algorithm (neg weights)', 'modularity_finetune_und_sign' _____ 'Louvain modularity algorithm (neg weights)', 'modularity_louvain_Qout_und_sign' 'Fine-tuning modularity algorithm (neg | w | The functions are supplementary information to the paper. Weight-conserving characterization of complex functional brain networks. Rubinov M, Sporns O (2011) NeuroImage, 56:2068-2079.; Also refer to 'Community structure Louvain' | Community number of each node (community vector C _i) _____ Maximized modularity Q |

| | | | |
|---|--|--|--|
| weights) 'modularity_finetune_QOut_und_sign' (helper function as community structure is required) | | | |
|---|--|--|--|

Appendix 2:

1. Add new null-model/randomizer function:

- Open GraphVar.fig (src/gui/graphVar) out of Matlab (right mouse click: open in GUIDE)
- On the GUI figure right click the box where the null-model/randomizer functions are located (in the network construction panel and/or in the raw matrix panel) and choose to open with “Property Inspector”
- Highlight the field “String” and click on the modify button next to it
- Write the name of your function into the string-window
- If your function is located somewhere in your GraphVar path that you have added to Matlab, GraphVar will automatically refer to it
- GraphVar will use the first argument of your function as the output. The second argument refers to the “Iterations” box in the GUI (e.g. bin_swaps in the null_model_und_sign function)

2. Add new graph/network functions (metrics) to the “Network Calculations” panel:

- Open the “getFunctions” script in the src folder
- Decide whether your function is binary, weighted or both
- Example:


```
'Characteristic path global', 'charpath_B', 1, ...
'Characteristic path length:<br>The characteristic path length is
the
average shortest path length in the network.'
```

 - The first word is the name of the function. It will appear as this in the GUI
 - The second word denotes the name of your functions script
 - The number denotes the dimensions of your function (refer to 1.3. in the results viewer section for the definition of the dimensions)
 - The second line entails the text displayed in the mouse-over box in the GUI
- GraphVar will use the first output (argument) of your function. Thus, if your function entails different outputs or additionally needs other measures as the community affiliation vector, you may want to specify this as a second helper function (src/calc/helper), which will then refer to the n-th argument.
 - Example: additional distance matrix required


```
function D = charpath_B(W)
D = distance_bin(W);
D = charpath(D);
```

- o Example: third output of the original function

```
function D = charpath_B_ecc(W)
    D = distance_bin(W);
    [~,~,D] = charpath(D);
```

- If your function is located somewhere in your GraphVar path that you have added to Matlab, GraphVar will automatically refer to it

3. Change the threshold range/steps (relative/absolute):

- Open the script "GraphVar_OpeningFcn.m" (src/gui/GraphVar folder)
- Search for the first "thr_str"
- Refers to proportional thresholding:

```
for i = 0.1:0.01:0.5
    thr_str{count} = num2str(i);
    count = count + 1 ;
end
```

- Refers to absolute thresholding:

```
count = 1;
for i = 0.05:0.01:1
    thr_str2{count} = num2str(i);
    count = count + 1 ;
end
```

4. Change the threshold range/steps (significant):

- Open GraphVar.fig (src/gui/graphVar) out of Matlab (right mouse click: open in GUIDE)
- On the GUI figure right click the box where significant thresholds are located (in the network construction panel and/or in the raw matrix panel) and choose to open with "Property Inspector"
- Highlight the field "String" and click on the modify button next to it
- Input your thresholds into the string-window
- save

5. Results (folder) structure in the workspace:

- The “results” folder contains initially empty subfolders (“CorrectedAlpha; CorrMatrix; CorrResults; RandomizedTimeSeries; Saved”) that will serve for storing the results of your computations.
- If the selection button “save interim results” is disabled in the setup menu, all computed graph variables (as well as random networks) will be stored in the subfolder “default” will be over written each time a new analysis is performed.
- If you opted to save the interim results, each time GraphVar performs any new analyses within the same workspace, interim results are written in a new folder (in the results folder) with ascending numbering. This allows loading these interim results and to do statistical analyses on them (this is especially useful if you have computed e.g. 1000 null models per subject over a large threshold range and want to use them in further analyses).
- If your data storage space is limited, it is advisable to delete the folders with the interim results from time to time manually.

6. Data structure and handling:

Results of the Stats function:

Results

| | |
|-------|---------------------------------|
| .desc | - A description of the contents |
| .type | - Type of this results struct |
| .n | - Number of Subjects included |

For Type Correlation & Partial_Correlation

| | |
|-------|---------------------------|
| .RHO | - r Result of Correlation |
| .PVAL | - p Value of Correlation |

Additional for Partial_Correlation

| | |
|--------------|--|
| .partialVars | - The Vars which were selected for Partial Correlation |
|--------------|--|

For type “groupCompair”

| |
|---|
| .anovaF: [size of GraphVar] |
| F Value of the Anova |
| .anovaP: [size of GraphVar] |
| p Value of the Anova |
| .groupMean: [size of GraphVar x number of groups] |
| Mean of the group ´ |

.groups

All unique group identifiers

Element wise T-Tests:

.group1: [size of GraphVar x number of ttest = nchoosek((n_groups-1),2)]

First group of T-Test

.group2: [size of GraphVar x number of ttest = nchoosek((n_groups-1),2)]

Second group of T-Test

. PVAL: [size of GraphVar x number of ttest = nchoosek((n_groups-1),2)]

p Value of the T-Test

.groupDiff: [size of GraphVar x number of ttest = nchoosek((n_groups-1),2)]

Group difference between those two groups