

# Package: NetworkComparr (via r-universe)

September 9, 2024

**Title** Statistical Comparison of Networks

**Version** 0.0.0.9

**Imports** dplyr, qgraph, igraph, reshape2, networktools, gdata

**Description** A permutation-based hypothesis test for statistical comparison of two networks based on the invariance measures of the R package 'NetworkComparisonTest' by van Borkulo et al. (2022), <[doi:10.1037/met0000476](https://doi.org/10.1037/met0000476)>: network structure invariance, global strength invariance, edge invariance, and various centrality measures. Edgelist from dependent or independent samples are used as input. These edgelist are generated from concept maps and summed into two comparable group networks. The networks can be directed or undirected.

**License** GPL-2

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Lara Trani [aut, cre]  
(<<https://orcid.org/0000-0002-8517-7168>>), Maike Sauer [aut]  
(<<https://orcid.org/0000-0002-1418-1194>>)

**Maintainer** Lara Trani <lara.trani@rptu.de>

**Repository** CRAN

**Date/Publication** 2023-08-11 08:30:02 UTC

## Contents

CEN-Methods . . . . .	2
CompareEdgelistNetworks . . . . .	2
NetworkComparr . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

**Description**

This function can summary, print and plot results from '[CompareEdgelistNetworks](#)'

**Usage**

```
## S3 method for class 'CEN'
summary(object, ...)

## S3 method for class 'CEN'
print(x, ...)

## S3 method for class 'CEN'
plot(x, what = c("strength", "network", "edge", "centrality"), ...)
```

**Arguments**

object	output of ' <a href="#">CompareEdgelistNetworks</a> ', object of class "CEN"
...	Arguments only used in plot method
x	output of ' <a href="#">CompareEdgelistNetworks</a> ', object of class "CEN"
what	defines what has to be plotted: results pertaining to test on invariance of global strength ("strength"), network structure ("network"), edge strength ("edge"), or specific centrality measure ("centrality")

**Value**

CEN-Methods contains three functions to display the output from '[CompareEdgelistNetworks](#)'. [summary.CEN](#) summarizes the test statistics of the compared networks in detail. [print.CEN](#) briefly summarizes the test statistics of the compared networks. [plot.CEN](#) graphically presents the significance tests of the various variance measures in form of histograms.

## Description

This permutation-based hypothesis test assesses the differences between two networks based on the invariance measures of the R package 'NetworkComparisonTest' by van Borkulo et al. (2022): global strength invariance, network structure invariance, edge invariance, and various centrality measures. The global strength invariance represents the differences in the total number of all edges. The network structure invariance involves only the edge that differs the most in its count. The edge invariance covers the differences of the specified individual edges. The centrality measures covers the differences regarding the importance of the individual nodes. Edgelist of two comparable group networks from dependent or independent samples are used as input. These edgelist can be generated from individual graphical network representations, such as concept maps. The networks can be directed or undirected.

## Usage

```
CEN(EL1, EL2, noc, it = 10000,
    paired = FALSE, directed = FALSE, abs = TRUE, plot = TRUE,
    test.edges = TRUE, edges = "all", progressBar = TRUE,
    p.adjust.methods = c("none", "holm", "hochberg", "hommel",
                        "bonferroni", "BH", "BY", "fdr"),
    test centrality = TRUE, centrality = c("all"), cen.nodes = "all",
    test.bridge centrality = FALSE, bridge centrality = c("all"),
    brg.nodes = "all", communities = NULL, useCommunities = "all",
    verbose = TRUE)
```

## Arguments

- |     |  |
|-----|--|
| EL1 | Edgelist 1. One of two edgelist containing the subject number and the coding of the network. The object type is "data.frame", the columns must be labeled "person", "from" and "to". "Person" contains the subject number. For dependent samples, the same person must have the same label in both edgelist. For independent samples, the same subject number cannot occur twice in the two edgelist. "From" indicates where an edge begins, "to" indicates where it ends. For undirected networks, an edge can be specified in one direction only. The direction is irrelevant (e.g., from 1 to 2, or from 2 to 1). The nodes must be numbered and must not have labels of the class character. The example serves as an orientation for the structure of the data frame.       |
| EL2 | Edgelist 2. The other of two edgelist containing the subject number and the coding of the network. The object type is "data.frame", the columns must be labeled "person", "from" and "to". "Person" contains the subject number. For dependent samples, the same person must have the same label in both edgelist. For independent samples, the same subject number cannot occur twice in the two edgelist. "From" indicates where an edge begins, "to" indicates where it ends. For undirected networks, an edge can be specified in one direction only. The direction is irrelevant (e.g., from 1 to 2, or from 2 to 1). The nodes must be numbered and must not have labels of the class character. The example serves as an orientation for the structure of the data frame. |
| noc | Number of nodes. Total number of nodes that occur in the networks.   |

<code>it</code>	Iterations. The number of iterations (permutations). Defaults to 10.000. The ideal number of permutations can be calculated in advance using the R package 'coin'.
<code>paired</code>	Logical. Can be TRUE or FALSE to indicate whether the samples are dependent or not. If paired is TRUE, relabeling is performed within each pair of observations. If paired is FALSE, relabeling is not restricted to pairs of observations. Note that, currently, dependent data is assumed to entail one group measured twice. Defaults to FALSE.
<code>directed</code>	Logical. Can be TRUE or FALSE to indicate whether the networks are directed or not. Undirected networks have edges without directions. For directed networks, the edges are often represented with arrows in one direction, however, the arrow can also point in both directions. Defaults to FALSE.
<code>abs</code>	Logical. Should global strength consider the absolute value of edge weights, or the raw value (i.e., global expected influence)? Important if the edgelist contains negative values. Defaults to TRUE.
<code>plot</code>	Logical. Should the networks be displayed graphically? Defaults to TRUE.
<code>test.edges</code>	Logical. Can be TRUE or FALSE to indicate whether or not differences in individual edges should be tested. Defaults to TRUE.
<code>edges</code>	Character or list. When 'all', differences between all individual edges are tested. When provided a list with one or more pairs of nodes, the provided edges are tested. The list must contain the two nodes that delimit the edge, e.g., <code>edges = list(c(1,2), c(3,2))</code> .
<code>progressbar</code>	Logical. Should the progressbar be plotted in order to see the progress of the permutation procedure? Defaults to TRUE.
<code>p.adjust.methods</code>	Character. Can be one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", or "none". To control (or not) for testing of multiple edges. Defaults to "none".
<code>test.centrality</code>	Logical. Should centrality metrics be compared across networks? Defaults to FALSE.
<code>centrality</code>	Type of centrality metrics to test. Can be either "all" or individually selected. For undirected networks it can be any of c("Closeness", "Betweenness", "Strength", "ExpectedInfluence"). For directed networks it can be any of c("Closeness", "Betweenness", "InStrength", "OutStrength", "InExpectedInfluence", "OutExpectedInfluence"). For more information see ' <a href="#">centrality</a> '.
<code>cen.nodes</code>	Specific nodes for centrality tests. Indicated by the node numbers, e.g., <code>c(1,3)</code> , to test node 1 and 3. Only used if <code>test.centrality = TRUE</code> .
<code>test.bridge.centrality</code>	Logical. Should bridge centrality metrics be compared across networks? Defaults to FALSE.
<code>bridge.centrality</code>	Type of bridge centrality metrics to test. Currently, only "all" can be selected.
<code>brg.nodes</code>	Specific nodes for bridge centrality tests. Indicated by the node numbers, e.g., <code>c(1,3)</code> , to test node 1 and 3. Only used if <code>test.bridge.centrality = TRUE</code> .

<code>communities</code>	A character vector of community assignments for each node (e.g., <code>c("Comm1", "Comm1", "Comm2", "Comm2")</code> ). The ordering of this vector should correspond to the order of the nodes. Can also be in list format (e.g., <code>list("Comm1"=c(1:3), "Comm2"=c(4:6))</code> ). For more information see <code>'bridge'</code> .
<code>useCommunities</code>	A character vector specifying which communities should be included. Default set to "all". For more information see <code>'bridge'</code> .
<code>verbose</code>	Logical: Should warnings and notes be printed?

## Details

This function is predominantly based on the R package `'NetworkComparisonTest'` by van Borkulo et al. (2022). By changing the input in the form of existing edgelists, the usage has been changed.

## Value

`CompareEdgelistNetworks` returns an object of class "CEN" containing the following items:

**`glstrinv.real`** The difference in global strength between the networks of the observed data sets.

**`glstrinv.sep`** The global strength values of the individual networks.

**`glstrinv.pval`** The p value resulting from the permutation test concerning difference in global strength.

**`glstrinv.perm`** The difference in global strength between the networks of the permuted data sets.

**`nwinv.real`** The value of the maximum difference in edge weights of the observed networks.

**`nwinv.pval`** The p value resulting from the permutation test concerning the maximum difference in edge weights.

**`nwinv.perm`** The values of the maximum difference in edge weights of the permuted networks.

**`edges.tested`** The edges, that are called to be tested. Only if `test.edges = TRUE`.

**`einv.real`** The values of the differences in edge count of the observed networks. This can be either of all edges or of the selected edge(s). Only if `test.edges = TRUE`.

**`einv.pvals`** p-values (corrected for multiple testing or not according to `'p.adjust.methods'`) per edge from the permutation test concerning differences in edges count. Only returned if `test.edges = TRUE`.

**`einv.perm`** The values of the differences in edge count of the permuted networks. Only if `test.edges = TRUE`.

**`diffcen.real`** The values of the difference in centralities of the observed networks. Only if `test.centrality = TRUE`.

**`diffcen.pval`** p-values (corrected for multiple testing or not according to `'p.adjust.methods'`) per node from the permutation test concerning differences in centralities. Only if `test.centrality = TRUE`.

**`diffcen.perm`** The values of the difference in centralities of the permuted networks. Only if `test.centrality = TRUE`.

**`cen.nw1`** Values of the selected centrality measures of network 1.

**`cen.nw2`** Values of the selected centrality measures of network 2.

**`cen.plot1`** Visualization of the selected centrality measures of network 1.

**`cen.plot2`** Visualization of the selected centrality measures of network 2.

**diffbcen.real** The values of the difference in bridge centralities of the observed networks. Only if `test.bridge.centralicity = TRUE`.

**diffbcen.pval** p-values (corrected for multiple testing or not according to 'p.adjust.methods') per node from the permutation test concerning differences in bridge centralities. Only if `test.bridge.centralicity = TRUE`.

**diffbcen.perm** The values of the difference in bridge centralities of the permuted networks. Only if `test.bridge.centralicity = TRUE`.

**bridgecen.nw1** Values of the selected bridge centrality measures of network 1.

**bridgecen.nw2** Values of the selected bridge centrality measures of network 2.

**bridge.plot1** Visualization of the selected bridge centrality measures of network 1.

**bridge.plot2** Visualization of the selected bridge centrality measures of network 2.

**info** Returns the selected arguments of the CEN usage.

## Examples

```
###Simulate datasets with a dependent sample and directed networks
#For an example with an undependent sample and undirected networks see README
data1 <- dplyr::tibble(person = c("1", "1", "2", "2", "2", "3", "3", "3", "4", "4", "4", "4", "4", "5", "5"),
  from = c("1", "3", "3", "1", "4", "3", "1", "3", "1", "4", "3", "3", "3", "2", "1"),
  to = c("3", "4", "4", "3", "2", "1", "3", "4", "3", "3", "4", "2", "4", "4", "3"))
data2 <- dplyr::tibble(person = c("1", "2", "2", "3", "3", "3", "3", "3", "3", "3", "4", "4", "4", "5", "5", "5"),
  from = c("1", "1", "1", "1", "2", "1", "2", "4", "4", "3", "1", "3", "2", "1", "1", "3"),
  to = c("2", "2", "4", "2", "4", "4", "3", "3", "1", "4", "2", "2", "1", "4", "2", "4"))

### Compare networks of data sets using CEN
Res <- CEN(data1, data2, noc=4, it=50, paired=TRUE, directed=TRUE, abs=TRUE,
  test.edges=TRUE, edge=list(c(1,3),c(4,2),c(3,2)), p.adjust.methods= "none",
  test.centralicity=TRUE, centralicity=c("Closeness", "Betweenness"), cen.nodes="all",
  test.bridge.centralicity=FALSE, bridge.centralicity="all", brg.nodes=c(1,3),
  communities=c("1", "1", "2", "2"), useCommunities="all")

###See results
summary(Res)
print(Res)
Res$glstrinv.sep
Res$glstrinv.pval
Res$nwinv.real
Res$nwinv.pval
Res$einv.real
Res$einv.pvals
Res$diffcen.real
Res$diffcen.pval

###Plot results
plot(Res, what="network")
plot(Res, what="strength")
plot(Res, what="edge")
plot(Res, what="centralicity")
```

**Description**

A permutation-based hypothesis test for statistical comparison of two networks based on the invariance measures of the R package 'NetworkComparisonTest' by van Borkulo et al. (2022): global strength invariance, network structure invariance, edge invariance, and various centrality measures. Edgelists of two comparable group networks from dependent or independent samples are used as input. These edgelists can be generated from individual graphical network representations, such as concept maps. The networks can be directed or undirected.

Keywords: concept-maps, edgelists, comparing networks, permutation-test

**Author(s)**

Lara Trani [lara.trani@rptu.de](mailto:lara.trani@rptu.de) Maike Sauer [maike.sauer@rptu.de](mailto:maike.sauer@rptu.de)

**References**

- Salmaso, L., & Pesarin, F. (2010). Permutation tests for complex data: theory, applications and software. John Wiley & Sons. <https://doi.org/10.1002/9780470689516>
- van Borkulo, C. D., van Bork, R., Boschloo, L., Kossakowski, J. J., Tio, P., Schoevers, R. A., Borsboom, D., & Waldorp, L. J. (2022). Comparing network structures on three aspects: A permutation test. *Psychological methods*. <https://doi.org/10.1037/met0000476>.

# Index

- \* **comparing-networks**  
NetworkComparr, [7](#)
- \* **concept-maps**  
NetworkComparr, [7](#)
- \* **edgelist**  
NetworkComparr, [7](#)
- \* **permutation-test**  
NetworkComparr, [7](#)

bridge, [5](#)

CEN (CompareEdgelistNetworks), [2](#)  
CEN-Methods, [2](#)  
centrality, [4](#)  
CompareEdgelistNetworks, [2, 2](#)

NetworkComparr, [7](#)

plot.CEN, [2](#)  
plot.CEN (CEN-Methods), [2](#)  
print.CEN, [2](#)  
print.CEN (CEN-Methods), [2](#)

summary.CEN, [2](#)  
summary.CEN (CEN-Methods), [2](#)