

# Package ‘ANTs’

October 12, 2022

**Date** 2022-06-29

**Type** Package

**Title** Animal Network Toolkit Software

**Version** 0.0.16

**Description** How animals interact and develop social relationships in face of sociodemographic and ecological pressures is of great interest. New methodologies, in particular Social Network Analysis (SNA), allow us to elucidate these types of questions. However, the different methodologies developed to that end and the speed at which they emerge make their use difficult. Moreover, the lack of communication between the different software developed to provide an answer to the same/different research questions is a source of confusion. The R package Animal Network Toolkit 'ANTs' was developed with the aim of implementing in one package the different social network analysis techniques currently used in the study of animal social networks. Hence, ANT is a toolkit for animal research allowing among other things to: 1) measure global, dyadic and nodal networks metrics; 2) perform data randomization: pre- and post-network (node and link permutations); 3) perform statistical permutation tests as correlation test (<doi:10.2307/2332226>), t-test (<doi:10.1037/h0041412>), General Linear Model (<doi:10.2307/2346786>), General Linear Mixed Model (<doi:10.2307/2346786>), deletion simulation (<doi:10.1098/rsbl.2003.0057>), 'Matrix TauKr correlations' (<doi:10.1016/S0022-5193(05)80036-0>). The package is partially coded in C++ using the R package 'Rcpp' for an optimal coding speed. The package gives researchers a workflow from the raw data to the achievement of statistical analyses, allowing for a multilevel approach (<doi:10.1007/978-3-319-47829-6\_1882-1>): from the individual's position and role within the network, to the identification of interaction patterns, and the study of the overall network properties. Furthermore, ANT also provides a guideline on the SNA techniques used: 1) from the appropriate randomization technique according to the data collected; 2) to the choice, the meaning, the limitations and advantages of the network metrics to apply, 3) and the type of statistical tests to run. The ANT project is multi-collaborative, aiming to provide access to advanced social network analysis techniques and to create new ones that meet researchers' needs in future versions. The ANT project is multi-collaborative, aiming to provide access to advanced social network analysis techniques and to create new ones that meet researchers' needs in future versions.

**License** GPL (>= 3)

**URL** [www.s-sosa.com/softwares](http://www.s-sosa.com/softwares) or <https://github.com/SebastianSosa/ANTs>

**Encoding** UTF-8

**LazyData** true  
**Depends** R (>= 3.5.0)  
**Imports** lme4,methods,Kendall,gtools,rstudioapi  
**LinkingTo** Rcpp,RcppArmadillo,RcppEigen  
**SystemRequirements** C++11  
**RoxygenNote** 7.2.0  
**BugReports** <https://github.com/SebastianSosa/ANTs>  
**Suggests** testthat, knitr, rmarkdown, markdown  
**VignetteBuilder** knitr  
**NeedsCompilation** yes  
**Author** Sosa Sebastian [aut, cre],  
Puga-Gonzalez Ivan [aut],  
Hu Feng He [aut],  
Pansanel Jerome [aut],  
Xiaohua Xie [aut],  
Sueur Cedric [aut]  
**Maintainer** Sosa Sebastian <s.sosa@live.fr>  
**Repository** CRAN  
**Date/Publication** 2022-07-02 22:20:02 UTC

## R topics documented:

|                             |    |
|-----------------------------|----|
| ant . . . . .               | 4  |
| assoc.gfi . . . . .         | 5  |
| assoc.indices . . . . .     | 6  |
| convert.socprog . . . . .   | 7  |
| df.create . . . . .         | 8  |
| df.to.gbi . . . . .         | 8  |
| df.to.mat . . . . .         | 9  |
| gbi.to.df . . . . .         | 10 |
| import.df . . . . .         | 11 |
| import.mat . . . . .        | 12 |
| mat.to.edgl . . . . .       | 13 |
| met.affinity . . . . .      | 14 |
| met.assortativity . . . . . | 15 |
| met.betweenness . . . . .   | 16 |
| met.ci . . . . .            | 18 |
| met.degree . . . . .        | 19 |
| met.density . . . . .       | 20 |
| met.diameter . . . . .      | 21 |
| met.disparity . . . . .     | 23 |
| met.eigen . . . . .         | 24 |
| met.ge . . . . .            | 25 |

|                                    |    |
|------------------------------------|----|
| met.geodesic . . . . .             | 27 |
| met.indegree . . . . .             | 28 |
| met.instrength . . . . .           | 29 |
| met.lp . . . . .                   | 31 |
| met.outdegree . . . . .            | 32 |
| met.outstrength . . . . .          | 33 |
| met.reach . . . . .                | 34 |
| met.ri . . . . .                   | 36 |
| met.strength . . . . .             | 37 |
| perm.double.focal . . . . .        | 38 |
| perm.double.focal.single . . . . . | 40 |
| perm.double.grp . . . . .          | 41 |
| perm.double.grp.single . . . . .   | 43 |
| perm.ds.focal . . . . .            | 44 |
| perm.ds.grp . . . . .              | 45 |
| perm.net.degree . . . . .          | 46 |
| perm.net.lk . . . . .              | 47 |
| perm.net.lk.w . . . . .            | 48 |
| perm.net.nl . . . . .              | 49 |
| perm.net.nl.str . . . . .          | 50 |
| sampling.effort . . . . .          | 51 |
| sampling.robustness . . . . .      | 53 |
| sampling.uncertainty . . . . .     | 56 |
| sim.df . . . . .                   | 58 |
| sim.focal.directed . . . . .       | 58 |
| sim.focal.undirected . . . . .     | 59 |
| sim.gbi . . . . .                  | 59 |
| sim.gbi.att . . . . .              | 60 |
| sim.grp . . . . .                  | 60 |
| sim.m . . . . .                    | 61 |
| sim.socprog . . . . .              | 61 |
| stat.cor . . . . .                 | 62 |
| stat.deletions . . . . .           | 63 |
| stat.deletionsPlot . . . . .       | 65 |
| stat.glm . . . . .                 | 66 |
| stat.glmm . . . . .                | 68 |
| stat.glmm.no.first.model . . . . . | 69 |
| stat.glmm.parallel . . . . .       | 71 |
| stat.lm . . . . .                  | 72 |
| stat.t . . . . .                   | 74 |
| stat.tauKr . . . . .               | 75 |
| vis.post.distribution . . . . .    | 76 |
| vis.post.distribution2 . . . . .   | 77 |
| which.metric . . . . .             | 79 |
| which.protocol . . . . .           | 79 |

---

**ant***ANT diagnostic for permuted statistics*

---

**Description**

ANT method to make a diagnostic of all ANT permutation tests. This method adapts the diagnostic results according to the data input. The output is adapted to the type of test run. However, some outputs are common to all tests.

**Usage**

```
ant(x, progress = FALSE)
```

**Arguments**

|          |  |
|----------|--|
| x        | an ANT object from functions: stat.t, stat.c, stat.lm, stat.glm, stat.glmm or a numeric vector or a data frame with only numeric values. |
| progress | a Boolean indicating if functions output should be printed in addition to the return object.   |

**Value**

A list of two elements for stat.cor, stat.t functions:

- A data frame with the permuted p-values (left side and right side), the confidence interval (25, 50 or 95) and the mean of the posterior distribution of the statistics of interest according to the statistical test (coefficient of correlation, t of students, or estimate(s))
- An histogram of the posterior distribution of the statistics of interest according to the statistical test.

#' A list of four elements for stat.lm, stat.glm, stat.glmm functions:

- A data frame with the original stats of the model, and permuted p-values (left side and right side), the confidence interval (25, 50 or 95) and the mean of the posterior distribution of the statistics of interest according to the statistical test (coefficient of correlation, t of students, or estimate(s))
- Diagnostic plot of the original model
- An histogram of the posterior distribution of the statistics of interest according to the statistical test.
- a vector of the permutations that generates errors and for which new permutations were performed.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**Examples**

```
t=met.strength(sim.m,sim.df,1) # Computing network metric
t=perm.net.nl(t,labels='age',rf=NULL,nperm=10,progress=FALSE) # Node label permutations
r.c=stat.cor(t,'age','strength',progress=FALSE) # Permuted correlation test
r=ant(r.c)
```

assoc.gfi

*Generalized affiliation index***Description**

Computes generalized affiliation indices based on a matrix of interactions or associations and a confounding factor.

**Usage**

```
assoc.gfi(M1, M2, fr = TRUE, sym = FALSE, erase.diag = TRUE, index = "sri")
```

**Arguments**

|            |  |
|------------|--|
| M1         | a square adjacency matrix representing individual interactions or associations. In the latter case, associations must be in the form of a gbi.   |
| M2         | a square adjacency matrix representing individual values of confounding factors.   |
| fr         | if <i>true</i> , it considers the argument M1 as an adjacency matrix representing interaction frequencies between individuals. Otherwise, it considers the argument M1 as an adjacency matrix representing associations between individuals.   |
| sym        | if <i>true</i> , it considers the argument M1 as an adjacency matrix representing symmetric interactions/associations.   |
| erase.diag | if <i>true</i> , it omits the diagonal of the matrix.  |
| index      | a string indicating the association index to compute: <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x/x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/\text{sqr}((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |

**Details**

Generalized affiliation indices allow to control for individual associations by a given confounding factor (such as temporal or spatial overlaps, gregariousness, social unit membership, kinship...). The principle is to perform a Generalized Linear Regression (GLR) on both matrices (one representing individual interactions/associations and the other one representing the confounding factor) and to use GLR residuals as association indices. For an adjacency matrix representing individual interactions, the GLR belongs to the Poisson family. For an adjacency matrix representing individual associations, the GLR belongs to the Binomial family. High positive values suggest strong associations between two individuals and negative values suggest avoidance between two individuals.

**Value**

a square adjacency matrix representing the generalized affiliation index between individuals.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

Whitehead, H., & James, R. (2015). Generalized affiliation indices extract affiliations from social network data. *Methods in Ecology and Evolution*, 6(7), 836-844.

**Examples**

```
assoc.gfi(sim.gbi,sim.gbi.att, fr = FALSE)
```

---

|               |                            |
|---------------|----------------------------|
| assoc.indices | <i>Association indices</i> |
|---------------|----------------------------|

---

**Description**

Compute several association indices.

**Usage**

```
assoc.indices(gbi, index = "sri")
```

**Arguments**

|       |  |
|-------|--|
| gbi   | a group by individual matrix.  |
| index | a string indicating the association index to compute: <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x/x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/\text{sqr}((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |

**Value**

A square matrix of individual association indices.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

**References**

Whitehead, H. A. L. (1997). Analysing animal social structure. *Animal behaviour*, 53(5), 1053-1067.

**Examples**

```
sim.gbi
assoc.indices(gbi=sim.gbi, index='sri')
```

---

|                 |                                    |
|-----------------|------------------------------------|
| convert.socprog | <i>Converts Socprog data frame</i> |
|-----------------|------------------------------------|

---

**Description**

Converts socprog 'Group mode' data frame (one scan per line with several individuals observed during the scan)

**Usage**

```
convert.socprog(df, id, scan, sep = ";")
```

**Arguments**

|      |   |
|------|---|
| df   | a 'dyadic' or 'group' data frame socprog format                                   |
| id   | an integer or string indicating the column of ids                                 |
| scan | an integer or string vector indicating the column of the scans                    |
| sep  | a character indicating the type of separation between individuals inside one scan |

**Value**

a data frame in socprog format.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

**Examples**

```
head(sim.socprog)
convert.socprog(sim.socprog, id=3, scan=c(1,2), sep=";")
```

---

`df.create`*Creates an empty data frame or a list of empty data frames*

---

**Description**

Creates an empty data frame with as many columns as in the corresponding matrix.

**Usage**

```
df.create(M, names = TRUE)
```

**Arguments**

|       |  |
|-------|--|
| M     | a square adjacency matrix or a list of square adjacency matrices.  |
| names | a boolean, if <i>true</i> then a column is added in the data frame with the names of the matrix columns. |

**Value**

- if argument M is a matrix, then it creates a single empty data frame with as many rows as columns of the corresponding matrix.
- if the argument M is a list of matrices, it creates a list of empty data frames with as many rows as columns of the corresponding matrix in the list.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**Examples**

```
sim.m  
df.create(sim.m)
```

---

`df.to.gbi`*Data frame to GBI.*

---

**Description**

Converts a data frame of individual associations into a group by individual matrix.

**Usage**

```
df.to.gbi(df, scan, id)
```



**Arguments**

|      |  |
|------|--|
| df   | a data frame of individual associations.   |
| scan | a numeric or character vector representing one or more columns used as scan factors. |
| id   | a numeric or character vector indicating the column holding ids of individuals.      |

**Details**

Several association indices are computed on GBI.

**Value**

A group by individual matrix.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**See Also**

[assoc.indices](#)

**Examples**

```
head(sim.grp)
df.to.gbi(sim.grp,scan=c('location','time'),id='ID')
```

---

df.to.mat

*Data frame to a matrix*

---

**Description**

Converts a data frame into a matrix.

**Usage**

```
df.to.mat(
  df,
  actor,
  receiver,
  weighted = NULL,
  tobs = NULL,
  sym = FALSE,
  num.ids = FALSE
)
```

**Arguments**

|          |   |
|----------|---|
| df       | a data frame of interactions or associations.   |
| actor    | an integer or a string indicating the column of the individuals performing the behaviour.   |
| receiver | an integer or a string indicating the column of the individuals receiving the behaviour.  |
| weighted | an integer or a string indicating the column of the weights of interactions. If NULL, interactions are given a weight of 1.                           |
| tobs     | an numeric vector or a list of numeric vectors of each individual time of observation. It has to be alphabetically ordered following unique id names. |
| sym      | a boolean if <i>true</i> , interactions or associations are considered symmetric.   |
| num.ids  | a boolean if <i>true</i> , it returns ordered matrix according to the numeric ids.  |

**Details**

Converts a data frame of interactions or associations into an adjacency matrix of interactions or associations.

**Value**

An adjacency matrix (undirected if argument *sym* is true, directed otherwise).

**Examples**

```
head(sim.focal.directed)
df.to.mat(df=sim.focal.directed,actor='actor', receiver='receiver')
```

---

gbi.to.df

*Group by individual matrix to linear data frame of association*


---

**Description**

Convert Linear gbi to a linear data frame of association

**Usage**

```
gbi.to.df(gbi)
```

**Arguments**

gbi a a group (N) by individual (I) matrix

**Value**

a linear data frame of association.

**Author(s)**

Sebastian Sosa

**Examples**

```
gbi.to.df(sim.gbi)
```

---

import.df

---

*Imports data frames*


---

**Description**

Imports all current directory files in the format of a list of data frames.

**Usage**

```
import.df(
  header = TRUE,
  sep = ",",
  quote = "\"",
  dec = ".",
  fill = TRUE,
  comment.char = "",
  ...
)
```

**Arguments**

|              |  |
|--------------|--|
| header       | a logical value indicating whether the file contains the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns. |
| sep          | the field separator character. Values in each line of the file are separated by this character. If sep = "" (the default for read.table) the separator is 'white space, that is one or more spaces, tabs, newlines or carriage returns.                              |
| quote        | the set of quoting characters. To disable quoting altogether, use quote = "". See scan for the behaviour on quotes embedded in quotes. Quoting is only considered for columns read as character, which is all of them unless colClasses is specified.                |
| dec          | the character used in the file for decimal points.   |
| fill         | logical If TRUE then in case the rows have unequal lengths, blank fields are implicitly added. See 'Details'.  |
| comment.char | a character vector of length 1 containing a single character or an empty string. Use "" to turn off the interpretation of comments altogether.   |
| ...          | Further arguments to be passed to <a href="#">read.csv</a> .   |

**Details**

After selecting the directory, this function returns a single data frame if there is only one file in the folder or a list of data frames in the same order as in the original folder if there are several files in the folder. by default it considers that the data frame has header= TRUE and row names= 1.

**Value**

a single data frame or a list of data frames according to the number of files in the selected folder.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**See Also**

[read.csv](#)

---

|            |                         |
|------------|-------------------------|
| import.mat | <i>Imports matrices</i> |
|------------|-------------------------|

---

**Description**

Imports all current directory files in the format of a list of matrices.

**Usage**

```
import.mat(header = TRUE, sep = ",", row.names = 1, ...)
```

**Arguments**

|           |   |
|-----------|---|
| header    | a logical value indicating whether the file contains the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns.  |
| sep       | the field separator character. Values on each line of the file are separated by this character. If sep = "" (the default for read.table) the separator is 'white space', that is one or more spaces, tabs, newlines or carriage returns.  |
| row.names | a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names. If there is a header and the first row contains one fewer field than the number of columns, the first column in the input is used for the row names. Otherwise if row.names is missing, the rows are numbered. Using row.names = NULL forces row numbering. Missing or NULL row.names generate row names that are considered to be 'automatic' (and not preserved by as.matrix). |
| ...       | Further arguments to be passed to code 'read.csv'.  |

**Details**

returns a single matrix if there is only one file in the folder or a list of matrices in the same order as in the original folder. By default, it considers that the matrix has header and row names=1. It also orders the matrix according to row names.

**Value**

a single matrix or a list of matrices according to the number of files in the selected folder.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

|             |                            |
|-------------|----------------------------|
| mat.to.edgl | <i>Matrix to edge list</i> |
|-------------|----------------------------|

---

**Description**

Converts a square adjacency matrix into a data frame of three columns representing an edge list. Columns are: actor, receiver and weight.

**Usage**

```
mat.to.edgl(M, sym = FALSE, erase.diag = TRUE)
```

**Arguments**

|            |   |
|------------|---|
| M          | a square adjacency matrix.  |
| sym        | if <i>TRUE</i> , it extracts the lower triangle of the matrix only. |
| erase.diag | if <i>TRUE</i> , it omits diagonals.                                |

**Value**

a data frame representing an edge list.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

**Examples**

```
mat.to.edgl(sim.m, sym = FALSE, erase.diag = TRUE)
```

---

|              |                 |
|--------------|-----------------|
| met.affinity | <i>Affinity</i> |
|--------------|-----------------|

---

### Description

Calculates the node metric *affinity* for all the nodes of the network

### Usage

```
met.affinity(M, df = NULL, dfid = NULL, binary = FALSE)
```

### Arguments

|               |  |
|---------------|--|
| <i>M</i>      | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <i>df</i>     | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| <i>dfid</i>   | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |
| <i>binary</i> | a boolean, if <i>TRUE</i> , it calculates the binary version of the affinity.  |

### Details

Affinity is a second-order metric that evaluates how alters of node *i* are connected. The binary version is the average degree of alters of node *i*. The weighted version is the ratio between the metric reach and the strength of node *i*. A high affinity reveals that nodes tend to be connected to alters with high degrees or strengths. Thus, this metric informs on node assortativity by vertex *met.degree*, i.e. connections between nodes with similar degrees or strengths.

### Value

- An integer vector of nodes *affinity* if argument *df* is *NULL*.
- A list of integer vectors of nodes *affinity* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *affinity* if argument *df* is not *NULL*. The name of the column is adapted according to argument value *binary*.
- A list of arguments *df* with a new column for nodes *affinity* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The names of the column of each element of the list is adapted according to argument value *binary*.

### Author(s)

Sebastian Sosa, Ivan Puga-Gonzalez.

## References

- Whitehead, H. A. L. (1997). Analysing animal social structure. *Animal behaviour*, 53(5), 1053-1067.
- Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

## Examples

```
met.affinity(sim.m)
head(sim.df)
met.affinity(sim.m,df=sim.df)
```

---

|                   |                      |
|-------------------|----------------------|
| met assortativity | <i>Assortativity</i> |
|-------------------|----------------------|

---

## Description

Calculates the binary or weighted version of vertices Newman's assortativity for categorical or continuous attributes.

## Usage

```
met assortativity(
  M,
  attr,
  se = FALSE,
  weighted = TRUE,
  df = NULL,
  perm.nl = TRUE,
  nperm = NULL
)
```

## Arguments

- |          |  |
|----------|--|
| M        | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| attr     | a factor vector of attributes for categorical assortativity. a numeric vector of attributes for continuous assortativity.  |
| se       | a boolean, if <i>TRUE</i> it computes the assortativity standard error.  |
| weighted | a boolean, if <i>TRUE</i> it computes the weighted assortativity version of the network.   |
| df       | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| perm.nl  | a boolean, if <i>TRUE</i> it permutes argument <i>attr</i> .   |
| nperm    | an integer indicating the number of permutations wanted.   |

**Details**

Assortativity allows the study of homophily (preferential interaction between nodes with similar attributes) and heterophily (the preferential interaction between nodes with different attributes). Attributes can be individual characteristics such as sex or age, or individual node metrics such as the met.degree.

**Value**

- a double representing the assortativity index of the network if argument *M* is a square matrix.
- A list of doubles if argument *M* is a list of matrices and if argument *df* is *NULL*. Each double representing the assortativity index of the corresponding matrix of the list.
- A list of arguments *df* with a new column of network assortativity if argument *df* is not *NULL* and if argument *M* is a list of matrices. The name of the column is adapted according to arguments values *weighted* and *attr*.
- A list of arguments *df* with a new column of network assortativity if argument *df* is not *NULL*, if argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and if argument *df* is a list of data frames of same length as argument *M*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

**References**

- Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*, 67(2), 026126.
- Farine, D. R. (2014). Measuring phenotypic assortment in animal social networks: weighted associations are more robust than binary edges. *Animal Behaviour*, 89, 141-153.
- Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

---

met.betweenness

*Betweenness centrality*


---

**Description**

Computes node betweenness centrality of all nodes of the network.

**Usage**

```
met.betweenness(
  M,
  binary = FALSE,
  shortest.weight = FALSE,
  normalization = TRUE,
```



```

    sym = FALSE,
    out = TRUE,
    df = NULL,
    dfid = NULL
  )

```

### Arguments

|                              |  |
|------------------------------|--|
| <code>M</code>               | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <code>binary</code>          | ia boolean, if <i>TRUE</i> , it calculates the binary version of the betweenness centrality.   |
| <code>shortest.weight</code> | if <i>FALSE</i> , it considers the highest <i>strength</i> as the shortest path.   |
| <code>normalization</code>   | normalizes the weights of the links i.e. divides them by the average strength of the network. Argument <i>normalization</i> can't be <i>TRUE</i> when argument <i>binary</i> is <i>FALSE</i> .                     |
| <code>sym</code>             | if <i>TRUE</i> , then it symmetrizes the matrix. Otherwise, it calculates geodesic distances and diameter according to the directionality of the links.  |
| <code>out</code>             | if <i>TRUE</i> , it considers outgoing ties to compute shortest paths.   |
| <code>df</code>              | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| <code>dfid</code>            | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |

### Details

Betweenness is the number of times a node is included in the shortest paths (geodesic distances) between all the potential combinations of edges of the other nodes. As it directly derives from the geodesic distance, it is important to pay attention to how the investigator intends to calculate geodesic distances (binary or weighted, directed or undirected, and using the lowest or the highest strength as the shortest path). Betweenness provides a specific centrality measure insofar that it informs on the role of a node in the transmission of information as nodes with high betweenness are likely to constitute bridges that connect subgroups.

### Value

- An integer vector of nodes *betweenness* if argument *df* is *NULL*.
- A list of integer vectors of nodes *betweenness* if argument *M* is a list and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *betweenness* if argument *df* is not *NULL*. The name of the column is adapted according to arguments *binary*, *shortest.weight*, *normalization*, *sym* and *out*.
- A list of arguments *df* with a new column for nodes *betweenness* if argument *df* is not *NULL*, if argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and if argument *df* is a list of data frames of same length as argument *M*.

**Author(s)**

Hu Feng He, Sebastian Sosa, Ivan Puga-Gonzalez, Xiaohua Xie.

**References**

- Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3), 215-239.
- Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

**Examples**

```
met.betweenness(sim.m)
head(sim.df)
met.betweenness(sim.m,df=sim.df)
```

---

met.ci

*Centralisation index*

---

**Description**

Computes network Centralisation index

**Usage**

```
met.ci(M, df = NULL)
```

**Arguments**

- |    |  |
|----|--|
| M  | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| df | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |

**Details**

Centralisation index of a network is based on eigenvector centrality.

**Value**

#'

- a double representing the centralisation index of the network if argument *M* is a square matrix.
- A list of doubles if argument *M* is a list of matrices and if argument *df* is *NULL*. Each double represents the centralisation index of the corresponding matrix of the list.

- A list of arguments *df* with a new column of network centralisation index if argument *df* is not *NULL* and if argument *M* is a list of matrices. The name of the column is adapted according to arguments values *binary* and *sym*.
- A list of arguments *df* with a new column of network centralisation index if argument *df* is not *NULL*, if argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and if argument *df* is a list of data frames of same length as argument *M*.

### Author(s)

Sebastian Sosa, Ivan Puga-Gonzalez

### References

Pasquaretta, C., Levé, M., Claidiere, N., Van De Waal, E., Whiten, A., MacIntosh, A. J., ... & Crofoot, M. C. (2014). Social networks in primates: smart and tolerant species have more efficient networks. *Scientific reports*, 4, 7600.

### Examples

```
met.ci(sim.m)
```

---

|            |               |
|------------|---------------|
| met.degree | <i>Degree</i> |
|------------|---------------|

---

### Description

Calculates the node metric degree for all vertices.

### Usage

```
met.degree(M, sym = TRUE, df = NULL, dfid = NULL)
```

### Arguments

|             |   |
|-------------|---|
| <i>M</i>    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .   |
| <i>sym</i>  | if <i>TRUE</i> , then it symmetrizes the matrix, and computes the sums of the presence/absence of an edge between the focal node and it's alters without accounting for direction (i.e. it computes the number of alters). Otherwise, it computes the sum of the presence/absence of the indegrees and outdegrees between the focal node and it's alters. |
| <i>df</i>   | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <i>dfid</i> | an integer or a string indicating the column with individual ids in argument <i>df</i> .  |

**Details**

Node degree  $i$  is the sum of edges of  $i$ , it represents the number of alters.

**Value**

- An integer vector of nodes *degree* if argument *df* is *NULL*.
- A list of integer vectors of nodes *degree* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *degree* titled 'degree', if argument *df* is not *NULL*.
- A list of arguments *df* with a new column for nodes *degree* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

- Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3), 215-239.
- Barrat, A., Barthelemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11), 3747-3752.
- Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

**Examples**

```
met.degree(sim.m)
head(sim.df)
met.degree(sim.m,df=sim.df)
```

---

met.density

*Density*

---

**Description**

Calculates network binary density.

**Usage**

```
met.density(M, df = NULL)
```

**Arguments**

|                 |  |
|-----------------|--|
| <code>M</code>  | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <code>df</code> | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |

**Details**

Binary network density is the ratio of existing links of a network in relation to all potential links.

**Value**

- a double representing the density of the network if argument *M* is a square matrix.
- A list of doubles if argument *M* is a list of matrices and if argument *df* is *NULL*. Each double represents the density of the corresponding matrix of the list.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

**Examples**

```
met.density(sim.m)
```

---

|                           |                 |
|---------------------------|-----------------|
| <code>met.diameter</code> | <i>Diameter</i> |
|---------------------------|-----------------|

---

**Description**

Calculates the network diameter .

**Usage**

```
met.diameter(
  M,
  df = NULL,
  weighted = TRUE,
  shortest.weight = FALSE,
  normalization = TRUE,
  directed = TRUE,
  out = TRUE
)
```

**Arguments**

|                              |  |
|------------------------------|--|
| <code>M</code>               | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <code>df</code>              | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| <code>weighted</code>        | if <i>FALSE</i> , it binarizes the square adjacency matrix <i>M</i> . Geodesic distances and diameter are based only on the presence or absence of edges.  |
| <code>shortest.weight</code> | if <i>false</i> , it considers the highest met.strength as the shortest path.  |
| <code>normalization</code>   | normalizes the weights of the links i.e. divides them by the average strength of the network. Argument <i>normalization</i> can't be <i>TRUE</i> when argument <i>weighted</i> is <i>FALSE</i> .                   |
| <code>directed</code>        | if <i>false</i> , then it symmetrizes the matrix. Otherwise, it calculates geodesic distances and diameter according to the directionality of the links.   |
| <code>out</code>             | if <i>true</i> , it considers outgoing ties.   |

**Details**

Diameter is the longer geodesic distance.

**Value**

- a double representing the diameter of the network if argument *M* is a square matrix.
- A list of doubles if argument *M* is a list of matrices and if argument *df* is *NULL*. Each double represents the diameter of the corresponding matrix of the list.
- A list of arguments *df* with a new column of network diameter if argument *df* is not *NULL* and if argument *M* is a list of matrices. The name of the column is adapted according to arguments values *.weighted*, *shortest.weight*, *normalization*, *directed* and *out*.
- A list of arguments *df* with a new column of network diameter if argument *df* is not *NULL*, if argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and if argument *df* is a list of data frames of same length as argument *M*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

- Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks*, 32(3), 245-251.
- Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

**Examples**

```
met.diameter(sim.m)
```

---

|               |                  |
|---------------|------------------|
| met.disparity | <i>Disparity</i> |
|---------------|------------------|

---

**Description**

Calculates the node metric disparity for all vertices.

**Usage**

```
met.disparity(M, df = NULL, dfid = NULL)
```

**Arguments**

|      |  |
|------|--|
| M    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| df   | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| dfid | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |

**Details**

Disparity measures the variation of the weight of a node. This metric compares a node's strength with its degree. It informs on the type of edges a node *i* has, *i.e.* few strong edges or many weak edges. Disparity of a vertex *i* is the sum of the squares of the division between the weighed edge between node *i* and *j* and the met.strength of node *i*

**Value**

- An integer vector of nodes *disparity* if argument *df* is *NULL*.
- A list of integer vectors of nodes *disparity* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *disparity* titled 'disparity', if argument *df* is not *NULL*.
- A list of arguments *df* with a new column for nodes *disparity* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

## References

- Barthélemy, M., Barrat, A., Pastor-Satorras, R., & Vespignani, A. (2005). Characterization and modeling of weighted networks. *Physica a: Statistical mechanics and its applications*, 346(1-2), 34-43.
- Whitehead, H. (2009). SOCPROG programs: analysing animal social structures. *Behavioral Ecology and Sociobiology*, 63(5), 765-778.
- Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

## Examples

```
met.disparity(sim.m)
head(sim.df)
met.disparity(sim.m,df=sim.df)
```

---

|           |                               |
|-----------|-------------------------------|
| met.eigen | <i>Eigenvector Centrality</i> |
|-----------|-------------------------------|

---

## Description

Calculates the node metric met.evcent centrality for all vertices.

## Usage

```
met.eigen(M, df = NULL, dfid = NULL, sym = TRUE, binary = FALSE, out = FALSE)
```

## Arguments

|        |  |
|--------|--|
| M      | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| df     | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| dfid   | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |
| sym    | if <i>TRUE</i> , then it symmetrizes the matrix. Otherwise, it calculates geodesic distances and diameter according to the directionality of the links.  |
| binary | a boolean, if <i>TRUE</i> , it calculates the binary version of the eigenvector centrality.  |
| out    | if <i>TRUE</i> , it considers outgoing ties to compute the shortest paths.   |

## Details

Eigenvector centrality is the first non-negative met.evcent value obtained through the linear transformation of an adjacency matrix. This centrality measure quantifies not only a node connectedness, but also the connections of the nodes to whom it is connected. Thus, a node can have a high met.evcent value by having a high met.degree or met.strength, or by being connected to nodes that have high degrees or strengths.



**Value**

- An integer vector of nodes *eigenvector centrality* if argument *df* is *NULL*.
- A list of integer vectors of nodes *eigenvector centrality* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *eigenvector centrality* if argument *df* is not *NULL*. The name of the column is adapted according to arguments value *binary*, *sym* and *out*.
- A list of arguments *df* with a new column for nodes *eigenvector centrality* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

- Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology*, 2(1), 113-120.
- Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

**Examples**

```
met.eigen(sim.m)
head(sim.df)
met.eigen(sim.m,df=sim.df)
```

---

met.ge

*Global efficiency*


---

**Description**

Calculates the global efficiency of a network.

**Usage**

```
met.ge(
  M,
  df = NULL,
  weighted = TRUE,
  shortest.weight = FALSE,
  normalization = TRUE,
  directed = TRUE,
  out = TRUE
)
```

**Arguments**

|                              |   |
|------------------------------|---|
| <code>M</code>               | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .   |
| <code>df</code>              | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <code>weighted</code>        | if <i>false</i> , it binarizes the square adjacency matrix <i>M</i> . Geodesic distances and diameter are based only on the presence or absence of edges.   |
| <code>shortest.weight</code> | if <i>false</i> , and <code>weighted</code> is <code>TRUE</code> , it considers the highest weights as the shortest path. The value returned is the average of inverse of the all shortest distances ( $1/d_{ij}$ ). Thus a higher value means a higher efficiency. Note that in contrast to unweighted graphs where global efficiency assumes values from 0 to 1, in weighted graphs the values depend on the weights associated to the links. It is therefore very useful to compare the global efficiency of a given weighted network with the global efficiency of a randomized version of the network or to a network of same size but different distribution of weights among the links. By itself this value is meaningless. |
| <code>normalization</code>   | normalizes the weights of the links i.e. divides them by the average strength of the network.   |
| <code>directed</code>        | if <i>false</i> , it symmetrizes the matrix. Otherwise, it calculates geodesic distances and diameter according to the directionality of the links.   |
| <code>out</code>             | if <i>true</i> , it considers outgoing ties.  |

**Details**

The global efficiency provides a measure of how efficiently information is exchanged in a network. It can be calculated in binary or weighted networks, and for undirected or directed networks. Depending on the problematic, care is needed for the choice of calculation method (binary or weighted, directed or undirected, and using the lowest or the highest `met.strength` as shortest path).

**Value**

- a double representing the global efficiency of the network if argument *M* is a square matrix.
- A list of doubles if argument *M* is a list of matrices and if argument *df* is `NULL`. Each double represents the global efficiency of the corresponding matrix of the list.
- A list of arguments *df* with a new column of network global efficiency if argument *df* is not `NULL` and if argument *M* is a list of matrices. The name of the column is adapted according to arguments values `.weighted`, `shortest.weight`, `normalization`, `directed` and `out`.
- A list of arguments *df* with a new column of network global efficiency if argument *df* is not `NULL`, if argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and if argument *df* is a list of data frames of same length as argument *M*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

**References**

Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

**Examples**

```
met.ge(sim.m)
```

---

```
met.geodesic
```

```
Geodesic distances
```

---

**Description**

Calculates the geodesic distances of a network.

**Usage**

```
met.geodesic(
  M,
  weighted = TRUE,
  shortest.weight = FALSE,
  normalization = TRUE,
  directed = TRUE,
  out = TRUE
)
```

**Arguments**

|                 |   |
|-----------------|---|
| M               | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .   |
| weighted        | if <i>true</i> , it binarizes the square adjacency matrix M. Geodesic distances and diameter are based only on the presence or absence of edges.                    |
| shortest.weight | if <i>false</i> , it considers the highest met.strength as the shortest path.   |
| normalization   | normalizes the weights of the links i.e. divides them by the average strength of the network. Argument normalization can't be TRUE when argument weighted is FALSE. |
| directed        | if <i>false</i> , then it symmetrizes the matrix. Otherwise, it calculates geodesic distances and diameter according to the directionality of the links.            |
| out             | if <i>true</i> , it considers outgoing ties.  |

**Details**

Binary network met.density is the ratio of existing links of a network in relation to all potential links.

**Value**

- a matrix representing the geodesic distances of the network if argument  $M$  is a square matrix.
- A list of matrices if argument  $M$  is a list of matrices. Each matrix represents the geodesic distances of the corresponding matrix of the list.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

- Doreian, P. (1974). On the connectivity of social networks. *Journal of Mathematical Sociology*, 3(2), 245-258.
- Burt, R. S. (1976). Positions in networks. *Social forces*, 55(1), 93-122.
- Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks*, 32(3), 245-251.
- Sosa, S. (2018). *Social Network Analysis*, in: *Encyclopedia of Animal Cognition and Behavior*. Springer.

**Examples**

```
met.geodesic(sim.m)
```

---

```
met.indegree
```

```
Indegree
```

---

**Description**

Calculates the node metric met.indegree for all vertices.

**Usage**

```
met.indegree(M, df = NULL, dfid = NULL)
```

**Arguments**

- |      |   |
|------|---|
| M    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .   |
| df   | a data frame of same length as the input matrix or a list of data frames if argument $M$ is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| dfid | an integer or a string indicating the column with individual ids in argument $df$ .   |

**Details**

Indegree of a node  $i$  is the sum of all edges directed towards node  $i$ .

**Value**

- An integer vector of nodes *indegree* if argument *df* is *NULL*.
- A list of integer vectors of nodes *indegree* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *indegree* titled 'indegree', if argument *df* is not *NULL*.
- A list of arguments *df* with a new column for nodes *indegree* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

- Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3), 215-239.
- Barrat, A., Barthelemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11), 3747-3752.
- Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

**Examples**

```
met.indegree(sim.m)
head(sim.df)
met.indegree(sim.m,df=sim.df)
```

---

|                |                   |
|----------------|-------------------|
| met.instrength | <i>Instrength</i> |
|----------------|-------------------|

---

**Description**

Calculates the node metric met.instrength for all vertices.

**Usage**

```
met.instrength(M, df = NULL, dfid = NULL)
```

**Arguments**

|                   |  |
|-------------------|--|
| <code>M</code>    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <code>df</code>   | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| <code>dfid</code> | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |

**Details**

Instrength of a node *i* is the sum of the strengths of all edges directed towards node *i*.

**Value**

- An integer vector of nodes *instrength* if argument *df* is *NULL*.
- A list of integer vectors of nodes *instrength* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *instrength* titled 'instrength', if argument *df* is not *NULL*.
- A list of arguments *df* with a new column for nodes *instrength* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

- Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3), 215-239.
- Barrat, A., Barthelemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11), 3747-3752.
- Sosa, S. (2018). Social Network Analysis, *in*: *Encyclopedia of Animal Cognition and Behavior*. Springer.

**Examples**

```
met.instrength(sim.m)
head(sim.df)
met.instrength(sim.m,df=sim.df)
```

---

met.lp

*Laplacian centrality*


---

**Description**

Calculates the symmetric version of Laplacian centrality for each vertex.

**Usage**

```
met.lp(M, df = NULL, dfid = NULL, binary = FALSE)
```

**Arguments**

|        |  |
|--------|--|
| M      | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| df     | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| dfid   | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |
| binary | a boolean indicating the version of the Laplacian index, if TRUE it computes the binary version.   |

**Details**

Laplacian centrality is the drop in the Laplacian energy of the graph when a vertex is removed. This version uses the degrees (for the binary version) or the strength (for the weighted version) to calculate Laplacian centrality.

**Value**

- An integer vector of nodes *Laplacian centrality* if argument *df* is *NULL*.
- A list of integer vectors of nodes *Laplacian centrality* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *Laplacian centrality* if argument *df* is not *NULL*. The name of the column is adapted according to argument value *binary*.
- A list of arguments *df* with a new column for nodes *Laplacian centrality* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

## References

- Qi, X., Fuller, E., Wu, Q., Wu, Y., & Zhang, C. Q. (2012). Laplacian centrality: A new centrality measure for weighted networks. *Information Sciences*, 194, 240-253.
- Qi, X., Duval, R. D., Christensen, K., Fuller, E., Spahiu, A., Wu, Q., ... & Zhang, C. (2013). Terrorist networks, network energy and node removal: A new measure of centrality based on Laplacian energy. *Social Networking*, 2(01), 19.
- Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

## Examples

```
met.lp(sim.m)
head(sim.df)
met.lp(sim.m,df=sim.df)
```

---

|               |                  |
|---------------|------------------|
| met.outdegree | <i>Outdegree</i> |
|---------------|------------------|

---

## Description

Calculates the node metric met.outdegree for all vertices.

## Usage

```
met.outdegree(M, df = NULL, dfid = NULL)
```

## Arguments

- |      |  |
|------|--|
| M    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| df   | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| dfid | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |

## Details

Outdegree of a node *i* is the sum of all the outgoing edges of a node *i*.

## Value

- An integer vector of nodes *outdegree* if argument *df* is *NULL*.
- A list of integer vectors of nodes *outdegree* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *outdegree* titled 'outdegree', if argument *df* is not *NULL*.



- A list of arguments *df* with a new column for nodes *outdegree* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

### Author(s)

Sebastian Sosa, Ivan Puga-Gonzalez.

### References

- Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3), 215-239.
- Barrat, A., Barthelemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11), 3747-3752.
- Sosa, S. (2018). Social Network Analysis, *in*: *Encyclopedia of Animal Cognition and Behavior*. Springer.

### Examples

```
met.outdegree(sim.m)
head(sim.df)
met.outdegree(sim.m,df=sim.df)
```

---

|                 |                    |
|-----------------|--------------------|
| met.outstrength | <i>Outstrength</i> |
|-----------------|--------------------|

---

### Description

Calculates the node metric *outstrength* for all vertices.

### Usage

```
met.outstrength(M, df = NULL, dfid = NULL)
```

### Arguments

- |             |  |
|-------------|--|
| <i>M</i>    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <i>df</i>   | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| <i>dfid</i> | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |

**Details**

outstrength of a node  $i$  is the sum of the strengths of all outgoing edges of a node  $i$ .

**Value**

- An integer vector of nodes *outstrength* if argument *df* is *NULL*.
- A list of integer vectors of nodes *outstrength* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *outstrength* titled 'outstrength', if argument *df* is not *NULL*.
- A list of arguments *df* with a new column for nodes *outstrength* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

- Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3), 215-239.
- Barrat, A., Barthelemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11), 3747-3752.
- Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

**Examples**

```
met.outstrength(sim.m)
head(sim.df)
met.outstrength(sim.m,df=sim.df)
```

---

met.reach

*Reach centrality*

---

**Description**

Calculates the weighted node metric 'reach centrality' for all vertices.

**Usage**

```
met.reach(M, df = NULL, dfid = NULL)
```

**Arguments**

|                   |  |
|-------------------|--|
| <code>M</code>    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| <code>df</code>   | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| <code>dfid</code> | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |

**Details**

Reach is the sum of the product of ego's and alters' strengths and alters' degrees.

**Value**

- An integer vector of nodes *reach centrality* if argument *df* is *NULL*.
- A list of integer vectors of nodes *reach centrality* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *reach centrality* titled 'reach centrality', if argument *df* is not *NULL*.
- A list of arguments *df* with a new column for nodes *reach centrality* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

Whitehead, H. A. L. (1997). Analysing animal social structure. *Animal behaviour*, 53(5), 1053-1067.

**Examples**

```
met.reach(sim.m)
head(sim.df)
met.reach(sim.m,df=sim.df)
```

met.ri

*R-Index***Description**

Calculates the node metric R-Index for all vertices.

**Usage**

```
met.ri(M, df = NULL, dfid = NULL)
```

**Arguments**

|      |  |
|------|--|
| M    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| df   | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| dfid | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |

**Details**

R-Index of vertex *i* is the outstrength of node *i* divided by the met.strength of node *i*. This node metric aims to measure if an individual is more likely to receive or emit edges.

**Value**

- An integer vector of nodes *R-index* if argument *df* is *NULL*.
- A list of integer vectors of nodes *R-index* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *R-index* titled 'R-index', if argument *df* is not *NULL*.
- A list of arguments *df* with a new column for nodes *R-index* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

- Liao, Z., Sosa, S., Wu, C., & Zhang, P. (2017). The influence of age on wild rhesus macaques' affiliative social interactions. *American journal of primatology*.
- Sosa, S. (2018). Social Network Analysis, *in*: *Encyclopedia of Animal Cognition and Behavior*. Springer.

**Examples**

```
met.ri(sim.m)
head(sim.df)
met.ri(sim.m,df=sim.df)
```

---

|              |                 |
|--------------|-----------------|
| met.strength | <i>Strength</i> |
|--------------|-----------------|

---

**Description**

Calculates the node metric strength for all vertices.

**Usage**

```
met.strength(M, df = NULL, dfid = NULL)
```

**Arguments**

|      |  |
|------|--|
| M    | a square adjacency matrix, or a list of square adjacency matrices, or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> .  |
| df   | a data frame of same length as the input matrix or a list of data frames if argument <i>M</i> is a list of matrices or an output of ANT functions <i>stat.ds.grp</i> , <i>stat.df.focal</i> , <i>stat.net.lk</i> . |
| dfid | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |

**Details**

Strength of a node *i* is the sum all edge values of the node *i*.

**Value**

- An integer vector of nodes *strength* if argument *df* is *NULL*.
- A list of integer vectors of nodes *strength* if argument *M* is a list of matrices and if argument *df* is *NULL*.
- A list of arguments *df* with a new column for nodes *strength* titled 'strength', if argument *df* is not *NULL*.
- A list of arguments *df* with a new column for nodes *strength* if 1) argument *df* is not *NULL*, 2) argument *M* is an output from ANT functions *stat.ds.grp*, *stat.df.focal*, *stat.net.lk* for multiple matrices permutations, and 3) argument *df* is a list of data frames of same length as argument *M*. The name of the column of each element of the list is adapted according to argument value *binary*.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

## References

- Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3), 215-239.
- Barrat, A., Barthelemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11), 3747-3752.
- Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

## Examples

```
met.strength(sim.m)
head(sim.df)
met.strength(sim.m,df=sim.df)
```

---

```
perm.double.focal      Data stream permutation for focal sampling data .
```

---

## Description

Warning, the original function (Farine 2017) uses a control factor, the number of focals and the ids of the focals.

## Usage

```
perm.double.focal(
  obs,
  ego,
  alters,
  focal,
  nperm,
  progress = FALSE,
  index = "sri",
  measure,
  test = "median",
  df = NULL,
  dfid = NULL,
  rf,
  ...
)
```

## Arguments

|        |   |
|--------|---|
| obs    | a data frame of focal observations.                           |
| ego    | an integer indicating the column of the focal id for the obs. |
| alters | an integer indicating the column of focal's alters in obs.    |

|          |  |
|----------|--|
| focal    | a numeric vector indicating the focal number in obs.   |
| nperm    | an integer indicating the number of permutations to performed.   |
| progress | a boolean indicating if the permutation process must be visible.   |
| index    | Which type of index of associations to calculate: <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x \div x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/sqr((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |
| measure  | a character indicating the social network measure to compute (Only those available in ANTs)  |
| test     | a character indicating the test to realize to account for the social network measure   |
| df       | a data frame of individual characteristics in which store permutations.  |
| dfid     | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |
| rf       | an integer (column id) or a string (column name) indicating the column holding the factor grouping multiple networks in argument <i>df</i> .   |
| ...      | Additional arguments related to the social network measure to compute (argument measure).  |

## Details

Pre-network permutation for focal sampling data, and for symmetric behaviour only.

## References

- Farine, D. R. (2017). A guide to null models for animal social network analysis. *Methods in ecology and evolution*, 8(10), 1309-1320.
- Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

## Examples

```
# Single network without data frame-----
head(sim.focal.undirected)
t=perm.double.focal(obs = sim.focal.undirected, ego = 3, alters = 4,
focal = 1, nperm = 10, progress = FALSE, measure = "met.strength")
# Multiple networks with data frames-----
d1 = data.frame("id" = names(t[[1]]), "period" = 1)
d2 = data.frame("id" = names(t[[1]]), "period" = 2)
t = list(d1, d2)
obs = list(sim.focal.undirected, sim.focal.undirected)
t =perm.double.focal(obs = obs, ego = 3, alters = 4, focal = 1, nperm = 10,
measure = "met.strength", df = t, dfid = "id", rf = "period")
```

---

 perm.double.focal.single

*Double permutation approach for focal sampling*


---

## Description

Double permutation approach for focal sampling

## Usage

```
perm.double.focal.single(
  obs,
  ego,
  alters,
  focal,
  nperm,
  progress = TRUE,
  index = "sri",
  measure,
  test = "median",
  df = NULL,
  dfid = NULL,
  ...
)
```

## Arguments

|          |   |
|----------|---|
| obs      | a data frame of focal observations.   |
| ego      | an integer indicating the column of the focal id for the obs.   |
| alters   | an integer indicating the column of focal's alters in obs.  |
| focal    | a numeric vector indicating the focal number in obs.  |
| nperm    | an integer indicating the number of permutations to performed.  |
| progress | a boolean indicating if the permutation process must be visible.  |
| index    | Which type of index of associations to calculate: <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x \div x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/\text{sqr}((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |
| measure  | a character indicating the social network measure to compute (Only those available in ANTs)   |
| test     | a character indicating the test to realize to account for the social network measure  |
| df       | a data frame of individual characteristics in which store permutations.   |
| dfid     | an integer or a string indicating the column with individual ids in argument <i>df</i> .  |
| ...      | Additional arguments related to the social network measure to compute (argument measure).   |



**Details**

Output need to be incorporated in a data frame and node label permutations with ANTs function perm.net.nl need to be performed before using any ANTs functions "stat".

**Value**

A numeric vector of individuals social measure corrected by double permutation approach (node label permutation can the be performed on this output)

**References**

Farine, D. R., & Carter, G. G. (2022). Permutation tests for hypothesis testing with animal social network data: Problems and potential solutions. *Methods in Ecology and Evolution*, 13, 144- 156. <https://doi.org/10.1111/2041-210X.13741>

---

 perm.double.grp

---

*Data stream permutation for association data*


---

**Description**

Pre-network permutation on association data for gambit of the group data collection protocol. The data frame must have a column named 'ID'.

**Usage**

```
perm.double.grp(
  obs,
  scan,
  ctrlf = NULL,
  nperm,
  progress = TRUE,
  index = "sri",
  measure,
  test = "median",
  df = NULL,
  dfid = NULL,
  rf,
  ...
)
```

**Arguments**

**obs** a data frame of gambit of the group observations. The data frame must have a column named 'ID'.

**scan** an integer indicating the column of scans of individual associations in obs.

**ctrlf** A confounding factor by which to control group associations in obs.

|          |  |
|----------|--|
| nperm    | number of permutations to perform.   |
| progress | a boolean indicating if the permutation process must be visible.   |
| index    | Which type of index of associations to calculate: <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x \div x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/sqr((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |
| measure  | a character indicating the social network measure to compute (Only those available in ANTs)  |
| test     | a character indicating the test to realize to account for the social network measure   |
| df       | a data frame of individual characteristics in which store permutations.  |
| dfid     | an integer or a string indicating the column with individual ids in argument <i>df</i> .   |
| rf       | an integer (column id) or a string (column name) indicating the column holding the factor grouping multiple networks in argument <i>df</i> .   |
| ...      | Additional arguments related to the social network measure to compute (argument measure).  |

### Details

Data stream permutation is a pre-network permutation approach. It is used on association data based on the gambit of the group.

### Value

list of square association index matrices. The first element of the list is the non-permuted association index matrix.

### Author(s)

Sebastian Sosa, Ivan Puga-Gonzalez.

### References

- Whitehead, H. A. L. (1997). Analysing animal social structure. *Animal behaviour*, 53(5), 1053-1067.
- Farine, D. R. (2017). A guide to null models for animal social network analysis. *Methods in Ecology and Evolution*.
- Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

### Examples

```
head(sim.grp)
t=perm.double.grp(sim.grp, 'location', 'time', 10, measure = "met.strength")
```

---

 perm.double.grp.single

*Double permutation approach for gambit of the group*


---

### Description

Double permutation approach for gambit of the group

### Usage

```
perm.double.grp.single(
  obs,
  scan,
  ctrlf,
  nperm,
  progress = TRUE,
  index = "sri",
  measure,
  test = "median",
  df = NULL,
  dfid = NULL,
  ...
)
```

### Arguments

|          |   |
|----------|---|
| obs      | a data frame of gambit of the group observations. The data frame must have a column named 'ID'.   |
| scan     | an integer indicating the column of scans of individual associations in obs.  |
| ctrlf    | A confounding factor by which to control group associations in obs.   |
| nperm    | number of permutations to perform.  |
| progress | a boolean indicating if the permutation process must be visible.  |
| index    | Which type of index of associations to calculate: <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x \div x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/\text{sqr}((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |
| measure  | a character indicating the social network measure to compute (Only those available in ANTs)   |
| test     | a character indicating the test to realize to account for the social network measure  |
| df       | a data frame of individual characteristics in which store permutations.   |
| dfid     | an integer or a string indicating the column with individual ids in argument <i>df</i> .  |
| ...      | Additional arguments related to the social network measure to compute (argument measure).   |

## Details

Output need to be incorporated in a data frame and node label permutations with ANTs function perm.net.nl need to be performed before using any ANTs functions "stat".

## Value

A numeric vector of individuals social measure corrected by double permutation approach (node label permutation can the be perfomed on this output)

## References

Farine, D. R., & Carter, G. G. (2022). Permutation tests for hypothesis testing with animal social network data: Problems and potential solutions. *Methods in Ecology and Evolution*, 13, 144- 156. <https://doi.org/10.1111/2041-210X.13741>

---

|               |  |
|---------------|--|
| perm.ds.focal | <i>Data stream permutation for focal sampling data .</i> |
|---------------|--|

---

## Description

Warning, the original function (Farine 2017) uses a control factor, the number of focals and the ids of the focals.

## Usage

```
perm.ds.focal(
  df,
  focal,
  ctrl = NULL,
  alters,
  nperm,
  progress = T,
  index = "sri"
)
```

## Arguments

|          |   |
|----------|---|
| df       | a data frame.   |
| focal    | an integer indicating the column of the focal.  |
| ctrl     | a numeric vector indicating the control factors.  |
| alters   | an integer indicating the column of focal's alters.   |
| nperm    | an integer indicating the number of permutations.   |
| progress | a boolean indicating if the permutation process must be visible.  |
| index    | a string indicating the association index to compute: <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x/x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/sqr((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |

**Details**

Pre-network permutation for focal sampling data, and for symmetric behaviour only.

**References**

Farine, D. R. (2017). A guide to null models for animal social network analysis. *Methods in ecology and evolution*, 8(10), 1309-1320.

Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

**Examples**

```
head(sim.focal.undirected)
t=perm.ds.focal(sim.focal.undirected, focal=3, ctrl=1, alters=4, nperm=10, progress=TRUE, index='sri')
```

---

 perm.ds.grp

*Data stream permutation for association data*


---

**Description**

Pre-network permutation on association data for gambit of the group data collection protocol. The data frame must have a column named 'ID'.

**Usage**

```
perm.ds.grp(df, scan, ctrlf = NULL, index = "sri", nperm, progress = TRUE)
```

**Arguments**

|          |   |
|----------|---|
| df       | A data frame. The data frame must have a column named 'ID'.   |
| scan     | an integer indicating the column of scans of individual associations.   |
| ctrlf    | A confounding factor by which to control group associations.  |
| index    | Which type of index of associations to calculate: <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x \div x + yAB + yA + yB</math></li> <li>• 'hwi' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sqri' for Square root index: <math>x/\text{sqrt}((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |
| nperm    | number of permutations to perform.  |
| progress | a boolean indicating if the permutation process must be visible.  |

**Details**

Data stream permutation is a pre-network permutation approach. It is used on association data based on the gambit of the group.

**Value**

list of square association index matrices. The first element of the list is the non-permuted association index matrix.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

Whitehead, H. A. L. (1997). Analysing animal social structure. *Animal behaviour*, 53(5), 1053-1067.

Farine, D. R. (2017). A guide to null models for animal social network analysis. *Methods in Ecology and Evolution*.

Sosa, S. (2018). Social Network Analysis, *in: Encyclopedia of Animal Cognition and Behavior*. Springer.

**Examples**

```
head(sim.grp)
t=perm.ds.grp(df = sim.grp, scan = 'location', ctrlf = 'time', nperm = 10, index = 'sri')
```

---

perm.net.degree

*Link permutation keeping the structure*

---

**Description**

Permutes the rows and/or the columns of a matrix to maintain the link structure during the permutation process.

**Usage**

```
perm.net.degree(M, nperm)
```

**Arguments**

|       |   |
|-------|---|
| M     | a square adjacency matrix or a list of square adjacency matrices. |
| nperm | number of permutations wanted.                                    |

**Details**

Link permutation maintaining the link structure, is usually used for matrix correlation or regression.

**Value**

- A list of permuted matrices, if M is a single square adjacency matrix.
- A list of a list of permuted matrices if M is a list of square adjacency matrices.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**Examples**

```
t=perm.net.degree(sim.m, nperm=10)
```

---

 perm.net.lk

---

*Matrix links permutations*


---

**Description**

Permutes matrix links.

**Usage**

```
perm.net.lk(M, sym = FALSE, erase.diag = TRUE, nperm, progress = TRUE)
```

**Arguments**

|            |   |
|------------|---|
| M          | a square adjacency matrix or a list of square adjacency matrices. |
| sym        | if <i>true</i> it vectorizes the lower triangle only .            |
| erase.diag | if <i>true</i> it erases the diagonal of the matrix.              |
| nperm      | number of permutations wanted.                                    |
| progress   | a boolean indicating if the permutation process must be visible.  |

**Details**

Edge permutations can be used to create random networks based on the observed network. Such permutation method is useful when analysing patterns of interaction such as assortativity.

**Value**

list of square matrices of length nperm + 1.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

**Examples**

```
t=perm.net.lk(sim.m, sym = FALSE, erase.diag = TRUE, nperm=10, progress=TRUE)
```

---

perm.net.lk.w      *Links weigths permutations*

---

## Description

Permute network links weigths while keep same network structure.

## Usage

```
perm.net.lk.w(M, nperm, progress = TRUE)
```

## Arguments

|          |  |
|----------|--|
| M        | a square adjacency matrix, or a list of square adjacency matrices. |
| nperm    | an integer indicating the number of permutations wanted.           |
| progress | a boolean indicating the visualization of the permutation process. |

## Details

Permute network links weigths while keep same network structure (density, modularity, binary global clustering coefficient).

## Value

a list of matrices of length nperm + 1 with the first matrix beeing the original one and the other elements, the permuted ones.

## Author(s)

Sebastian Sosa

## Examples

```
test = perm.net.lk.w(sim.m, nperm = 2)
test[[1]][test[[1]] > 0] == test[[2]][test[[2]] > 0]
```



---

perm.net.nl                      *Node label permutations with or without random factor(s)*

---

### Description

Performs node label permutations.

### Usage

```
perm.net.nl(ldf, labels, rf = NULL, nperm, progress = TRUE)
```

### Arguments

|          |   |
|----------|---|
| ldf      | a list of data frames, with each data frame representing a subset of the random factors (e.g. groups and/or periods)  |
| labels   | a numeric or character vector representing the labels to permute.   |
| rf       | an integer (column id) or a string (column name) indicating the column holding the factor grouping multiple networks. |
| nperm    | an integer indicating the number of permutations wanted.  |
| progress | a boolean indicating the visualization of the permutation process.  |

### Details

Node label permutations for GLMM models need to permute labels within a control factor.

### Value

A list of data frames. Each data frame is the merger of the input lists of data frames. The first element of the list is the original input data, the other elements are the different permutations.

### Author(s)

Sebastian Sosa, Ivan Puga-Gonzalez.

### References

Croft, D. P., James, R., & Krause, J. (2008). Exploring animal social networks. Princeton University Press.

Croft, D. P., Madden, J. R., Franks, D. W., & James, R. (2011). Hypothesis testing in animal social networks. *Trends in Ecology & Evolution*, 26(10), 502-507.

Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.

### Examples

```
df=met.strength(sim.m,df=sim.df)
t=perm.net.nl(df,labels='sex',rf=NULL,nperm=1000,progress=TRUE)
```

---

perm.net.nl.str      *Nodes labels permutation keeping network structure*

---

**Description**

Permute node metrics while keeping their dependency

**Usage**

```
perm.net.nl.str(df, labels, rf = NULL, nperm, progress = TRUE)
```

**Arguments**

|          |   |
|----------|---|
| df       | a data frame with nodes informations  |
| labels   | a numeric or string vector indicating the columns (labels) to permute.  |
| rf       | an integer (column id) or a string (column name) indicating the column holding the factor grouping multiple networks. |
| nperm    | an integer indicating the number of permutations wanted.  |
| progress | a boolean indicating the visualization of the permutation process.  |

**Details**

Permute network links weights while keep same network structure (density, modularity, binary global clustering coefficient).

**Value**

a list of data frames of length nperm + 1 with the first element of the list being the original data frame and the other elements, the permuted ones.

**Author(s)**

Sebastian Sosa

**Examples**

```
df = met.strength(sim.m, df = sim.df, dfid = 1)
df = met.eigen(sim.m, df = df, dfid = 1)
head(df)
perm.net.nl.str(df, labels = c('strength', 'eigen'), nperm = 2)
```

---

|                 |                        |
|-----------------|------------------------|
| sampling.effort | <i>Sampling effort</i> |
|-----------------|------------------------|

---

**Description**

Visualize metric variation through period of observations

**Usage**

```
sampling.effort(
  df,
  col.time,
  cumulative = TRUE,
  metric = "met.strength",
  assoc.indices = FALSE,
  actor = NULL,
  receiver = NULL,
  sym = FALSE,
  scan = NULL,
  id = NULL,
  index = "sri",
  ...
)
```

**Arguments**

|                            |  |
|----------------------------|--|
| <code>df</code>            | a data frame of interactions or associations.  |
| <code>col.time</code>      | an integer or string indicating the column with the time/period information  |
| <code>cumulative</code>    | a boolean, if TRUE, it computes the argument metric declared for each step of periods keeping previous periods   |
| <code>metric</code>        | a string to call an ANTs function of class 'met.XXX'.  |
| <code>assoc.indices</code> | a boolean, if TRUE, it creates matrices of associations according to argument 'index' and argument 'df' must be a data frame of associations, see <code>df.to.gbi</code> . Otherwise, it creates a matrix of interactions and argument 'df' must be a data frame of interactions type (see <code>df.to.mat</code> ). |
| <code>actor</code>         | an integer or a string indicating the column of the individuals performing the behaviour. This argument must be declared if argument 'assoc.indices' is equal to FALSE.  |
| <code>receiver</code>      | an integer or a string indicating the column of the individuals receiving the behaviour. This argument must be declared if argument 'assoc.indices' is equal to FALSE.   |
| <code>sym</code>           | a boolean if true, interactions or associations are considered symmetric. This argument must be declared if argument 'assoc.indices' is equal to FALSE.  |

|       |   |
|-------|---|
| scan  | a numeric or character vector representing one or more columns used as scan factors. This argument must be declared if argument 'assoc.indices' is equal to FALSE.  |
| id    | a numeric or character vector indicating the column holding ids of individuals.   |
| index | a string indicating the association index to compute:   |
| ...   | additional argument related to the computation of the metric declared. <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x/x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/\text{sqr}((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |

### Details

This function allows to visualize metric (nodal and global) variation through periods of observation. Studies have highlighted the need to assess their stability. Metric stability can be assessed by a sigmoide curve reaching a plateau. While the function doesn't give you any statistical test, it allows to visualize if the plateau is reached or not. For this approach, argument cumulative must be set to TRUE.

### Value

A list of two elemnts:

- 'df', a data frame with metric evolution through time
- plot a plot of the metric evolution through time

### Author(s)

Sebastian Sosa

### References

Farine, D. R., & Strandburg-Peshkin, A. (2015). Estimating uncertainty and reliability of social network data using Bayesian inference. *Royal Society open science*, 2(9), 150367.

### Examples

```
df <- sim.focal.directed
df$period <- rep(c("a", "b", "c", "d", "e"))
# Node measures non cumulative example
sampling.effort(df, col.time = "period", cumulative = FALSE,
               metric = "met.strength", actor = "actor", receiver = "receiver")

# Node measures cumulative example
sampling.effort(df, col.time = "period", cumulative = TRUE,
               metric = "met.strength", actor = "actor", receiver = "receiver")

# Node measures with extra arguments example
sampling.effort(df, col.time = "period", actor = "actor",
               receiver = "receiver", metric = "met.affinity")
```

```

sampling.effort(df, col.time = "period", actor = "actor",
               receiver = "receiver", metric = "met.affinity", binary = TRUE)

# Example of how to test global network metric with non cumulative version
sampling.effort(df, col.time = "period", cumulative = FALSE,
               metric = "met.density", actor = "actor", receiver = "receiver")

# Example of how to test global network metric with cumulative version
sampling.effort(df, col.time = "period", cumulative = TRUE,
               metric = "met.density", actor = "actor", receiver = "receiver")

# Same example with gambit of the group data collection protocol-----
# Node measures non cumulative example
sampling.effort(sim.grp, col.time = "day", cumulative = TRUE,
               metric = "met.strength", assoc.indices = TRUE,
               scan = c("time", "location"), id = "ID", index = "sri")

# Node measures non cumulative example
sampling.effort(sim.grp, col.time = "day", cumulative = FALSE,
               metric = "met.strength", assoc.indices = TRUE,
               scan = c("time", "location"), id = "ID", index = "sri" )

# Node measures with extra arguments example
sampling.effort(sim.grp, col.time = "day", cumulative = FALSE,
               metric = "met.affinity", assoc.indices = TRUE,
               scan = c("time", "location"), id = "ID", index = "sri")

sampling.effort(sim.grp, col.time = "day", cumulative = FALSE,
               metric = "met.affinity", assoc.indices = TRUE,
               scan = c("time", "location"), id = "ID",
               index = "sri", binary = TRUE)

# Example of how to test global network metric with non cumulative version
sampling.effort(df = sim.grp, col.time = "day", cumulative = FALSE,
               metric = "met.density", assoc.indices = TRUE,
               scan = c("time", "location"), id = "ID", index = "sri")

# Example of how to test global network metric with cumulative version
sampling.effort(df = sim.grp, col.time = "day", cumulative = TRUE,
               metric = "met.density", assoc.indices = TRUE,
               scan = c("time", "location"), id = "ID", index = "sri")

```

---

sampling.robustness     *Metric robustness*

---

### Description

Performs a matrix bootstrapping approach to estimate the confidence intervals surrounding each pairwise association.

**Usage**

```
sampling.robustness(
  df,
  subsampling = c(5, 10, 20, 30, 40, 50),
  metric = "met.strength",
  assoc.indices = FALSE,
  actor,
  receiver,
  scan,
  id,
  index = "sri",
  progress = TRUE,
  ...
)
```

**Arguments**

|               |   |
|---------------|---|
| df            | a data frame of individual associations   |
| subsampling   | a vector of integers indicating the percentage of data to remove  |
| metric        | the network metric to compute   |
| assoc.indices | a boolean indicating if association indices must be used  |
| actor         | If argument assoc.indices is FALSE, fill this argument, an integer or a string indicating the column of the individuals performing the behaviour.   |
| receiver      | If argument assoc.indices is FALSE, fill this argument, an integer or a string indicating the column of the individuals receiving the behaviour.  |
| scan          | If argument assoc.indices is TRUE, fill this argument, a numeric or character vector representing one or more columns used as scan factors.   |
| id            | If argument assoc.indices is TRUE, fill this argument, a numeric or character vector indicating the column holding ids of individuals.  |
| index         | a string indicating the association index to compute:   |
| progress      | a boolean indicating if function should print progress.   |
| ...           | additional argument related to the computation of the metric declared. <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x/x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/\text{sqr}((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |

**Details**

This process evaluates network metrics robustness by performing a bootstrap process to remove a certain percent of the data collected and recomputing the network metric of interest. This has been used in Balasubramaniam et al. 2018 to assess sensitivity of sampling effort of global network metrics in primate networks.

**Value**

3 elements:

- A matrix in which each columns represent a node metric variation through bootstrapping, with the first row representing the original metric.
- A summary of bootstrap distribution for each nodes.
- A plot of metric variations through deletions

**Author(s)**

Sebastian Sosa

**References**

Balasubramaniam, K. N., Beisner, B. A., Berman, C. M., De Marco, A., Duboscq, J., Koirala, S., ... & Ogawa, H. (2018). The influence of phylogeny, social style, and sociodemographic factors on macaque social network structure. *American journal of primatology*, 80(1), e22727.

**Examples**

```
#Example for node measures
test <- sampling.robustness(sim.focal.directed, actor = "actor",
  receiver = "receiver", metric = "met.strength")

# objects returned by the function
test$metrics
test$summary
test$plot

# Examples with metric extra arguments
sampling.robustness(sim.focal.directed, actor = "actor",
  receiver = "receiver", metric = "met.affinity")
sampling.robustness(sim.focal.directed, actor = "actor",
  receiver = "receiver", metric = "met.affinity", binary = FALSE)

# Examples with association data
test2 <- sampling.robustness(df = sim.grp, assoc.indices = TRUE,
  scan = c("time", "location"), id = "ID", metric = "met.strength")

# Example of how to test global network metric robustness
#by removing 10% of the observations simulated 100 times
test <- sampling.robustness(sim.focal.directed, subsampling = rep(10, 100),
  actor = "actor", receiver = "receiver", metric = "met.diameter")
```

---

sampling.uncertainty *Metric uncertainty*

---

### Description

Perform a matrix bootstrapping approach to estimate the confidence intervals surrounding each pairwise association.

### Usage

```
sampling.uncertainty(
  df,
  nboot,
  metric = "met.strength",
  assoc.indices = FALSE,
  actor = NULL,
  receiver = NULL,
  scan = NULL,
  id = NULL,
  index = "sri",
  progress = TRUE,
  ...
)
```

### Arguments

|               |  |
|---------------|--|
| df            | a data frame of individual interactions or associations  |
| nboot         | an integer indicating the number of bootstrap wanted.  |
| metric        | the network metric to compute  |
| assoc.indices | a boolean indicating if association indices must be used   |
| actor         | If argument assoc.indices is FALSE, fill this argument, an integer or a string indicating the column of the individuals performing the behaviour.  |
| receiver      | If argument assoc.indices is FALSE, fill this argument, an integer or a string indicating the column of the individuals receiving the behaviour.   |
| scan          | If argument assoc.indices is TRUE, fill this argument, a numeric or character vector representing one or more columns used as scan factors.  |
| id            | If argument assoc.indices is TRUE, fill this argument, a numeric or character vector indicating the column holding ids of individuals.   |
| index         | a string indicating the association index to compute.  |
| progress      | a boolean indicating if function should print progress.  |
| ...           | additional argument related to the computation of the metric declared. <ul style="list-style-type: none"> <li>• 'sri' for Simple ratio index: <math>x/x + yAB + yA + yB</math></li> <li>• 'hw' for Half-weight index: <math>x/x + yAB + 1/2(yA + yB)</math></li> <li>• 'sr' for Square root index: <math>x/sqr((x + yAB + yA)(x + yAB + yB))</math></li> </ul> |



## Details

This process evaluates network metrics uncertainty by performing a bootstrap with replacement on the data frame of associations and recomputing the network metric of interest.

## Value

3 elements:

- A matrix in which each column represents a node metric variation through bootstrapping, with the first row representing the original metric.
- A summary of bootstrap distribution for each node.
- A plot of metric variations through bootstrap

## Author(s)

Sebastian Sosa

## References

Lusseau, D., Whitehead, H., & Gero, S. (2009). Incorporating uncertainty into the study of animal social networks. arXiv preprint arXiv:0903.1519.

## Examples

```
test <- sampling.uncertainty(df = sim.focal.directed, nboot = 100,
                           actor = "actor", receiver = "receiver",
                           metric = "met.strength")

# objects returned by the function
test$metrics
test$summary
test$plot

# Example with metric extra arguments
sampling.uncertainty(df = sim.focal.directed, nboot = 100,
                   actor = "actor", receiver = "receiver",
                   metric = "met.affinity", binary = FALSE)
sampling.uncertainty(df = sim.focal.directed, nboot = 100,
                   actor = "actor", receiver = "receiver",
                   metric = "met.affinity", binary = TRUE)

# Example with individual associations
sampling.uncertainty(df = sim.grp, nboot = 100, assoc.indices = TRUE,
                   scan = c("day", "location", "time"), id = "ID")
```

---

|        |                             |
|--------|-----------------------------|
| sim.df | <i>Simulated data frame</i> |
|--------|-----------------------------|

---

**Description**

This is a simulated data frame of individual characteristics

**Usage**

```
sim.df
```

**Format**

An object of class `data.frame` with 20 rows and 3 columns.

**Value**

a data frame.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

|                    |                                     |
|--------------------|-------------------------------------|
| sim.focal.directed | <i>Focal sampling directed data</i> |
|--------------------|-------------------------------------|

---

**Description**

This is a simulated data frame of directed behaviour collected through focal sampling.

**Usage**

```
sim.focal.directed
```

**Format**

An object of class `data.frame` with 100 rows and 4 columns.

**Value**

a data frame

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

sim.focal.undirected    *Focal sampling undirected data*

---

**Description**

This is a simulated data frame of undirected behaviour collected through focal sampling.

**Usage**

```
sim.focal.undirected
```

**Format**

An object of class `data.frame` with 100 rows and 4 columns.

**Value**

a data frame.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

sim.gbi    *Group by individual (GBI) matrix*

---

**Description**

This is a simulated Group by individual (GBI) matrix collected through gambit of the group sampling.

**Usage**

```
sim.gbi
```

**Format**

An object of class `matrix` (inherits from `array`) with 6 rows and 10 columns.

**Value**

a matrix

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

`sim.gbi.att`*Dyadic attributes for sim.gbi data*

---

**Description**

This is a simulated matrix of dyadic attributes to use as argument M2 in `assoc.gfi` function.

**Usage**`sim.gbi.att`**Format**

An object of class `matrix` (inherits from `array`) with 10 rows and 10 columns.

**Value**

a data frame.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

`sim.grp`*Simulated data collected through gambit of the group sampling*

---

**Description**

This is a simulated data set collected through gambit of the group sampling.

**Usage**`sim.grp`**Format**

An object of class `data.frame` with 100 rows and 5 columns.

**Value**

a data frame

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

|       |   |
|-------|---|
| sim.m | <i>Simulated Matrix of interactions</i> |
|-------|---|

---

**Description**

This is a simulated Matrix of interaction.

**Usage**

sim.m

**Format**

An object of class `matrix` (inherits from `array`) with 20 rows and 20 columns.

**Value**

a matrix

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

|             |                                      |
|-------------|--------------------------------------|
| sim.socprog | <i>Simulated Socprog data format</i> |
|-------------|--------------------------------------|

---

**Description**

This is a simulated Socprog data format.

**Usage**

sim.socprog

**Format**

An object of class `data.frame` with 100 rows and 3 columns.

**Value**

a data frame.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez

---

`stat.cor`*Correlation test on permuted data.*

---

**Description**

This function is the first step in the process to create a correlation test in permuted data. For more details about correlation tests, see R documentation.

**Usage**

```
stat.cor(ant, var1, var2, method = "pearson", progress = TRUE)
```

**Arguments**

|                       |  |
|-----------------------|--|
| <code>ant</code>      | an ant data object originating from permutations   |
| <code>var1</code>     | an integer or string indicating the column in the data frames that corresponds to the first variable to correlate. This variable must be numerical.  |
| <code>var2</code>     | an integer or string indicating the column in the data frames that corresponds to the second variable to correlate. This variable must be numerical.   |
| <code>method</code>   | a string vector indicating which type of correlation to perform: <ul style="list-style-type: none"><li>• 'pearson' for pearson correlation</li><li>• 'kendall' for kendall correlation</li><li>• 'spearman' for spearman correlation</li></ul> |
| <code>progress</code> | a boolean indicating the visualization of the permutation process.   |

**Details**

Performs correlations tests in a data frame and extracts the t statistic.

**Value**

a vector of 3 elements: the t statistic, the degree of freedom, and the estimate.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**References**

D. J. Best & D. E. Roberts (1975), Algorithm AS 89: The Upper Tail Probabilities of Spearman's rho. *Applied Statistics*, 24, 377-379.

Myles Hollander & Douglas A. Wolfe (1973), *Nonparametric Statistical Methods*. New York: John Wiley & Sons. Pages 185-194 (Kendall and Spearman tests).

**Examples**

```
t=met.strength(sim.m,sim.df,1) # Computing network metric
t=perm.net.nl(t,labels='age',rf=NULL,nperm=10,progress=FALSE) # Node label permutations
r.c=stat.cor(t,'age','strength',progress=FALSE) # Permuted correlation test
```

---

stat.deletions                      *Network target & random deletion simulations*

---

**Description**

Performs a knockout analysis according to specific vertex attributes and a specific percentage of nodes to delete

**Usage**

```
stat.deletions(
  M,
  attr,
  target,
  ndel,
  nsim,
  weighted = TRUE,
  shortest.weight = FALSE,
  normalization = FALSE,
  directed = TRUE,
  out = TRUE,
  progress = TRUE,
  return.mat = FALSE
)
```

**Arguments**

|          |  |
|----------|--|
| M        | a square adjacency M.  |
| attr     | a vector of categorical (factor or character) or numeric (continuous) attributes of the nodes. The vector must have the same length and order as the nodes in the square adjacency matrix 'M'.   |
| target   | Indicates the nodes that will be the target of deletion. If the argument 'attr' is categorical, then 'target' indicates the attribute of the node target of deletion. If the argument 'attr' is numeric, then 'target' can take one of two character elements 1) 'decreasing' or 2) 'increasing' indicating whether the target of deletions are nodes with the greatest or lowest attribute's values respectively. |
| ndel     | an integer indicating the number of deletions to perform in each simulation.   |
| nsim     | an integer indicating the number of simulations, <i>i.e.</i> how many times to perform <i>ndel</i> deletions.  |
| weighted | if <i>true</i> , it binarizes the square adjacency matrix M. Geodesic distances and diameter are based only on the presence or absence of edges.   |

|                 |   |
|-----------------|---|
| shortest.weight | if <i>false</i> , it considers the higher met.strength as the shortest path.  |
| normalization   | normalizes the weights of the links i.e. divides them by the average strength of the network. Argument normalization can't be TRUE when argument weighted is FALSE.       |
| directed        | if <i>false</i> , then it symmetrizes the matrix. Otherwise, it calculates geodesic distances and diameter according to the directionality of the links.                  |
| out             | if <i>true</i> , it considers outgoing ties for the diameter and the geodesic distances.  |
| progress        | a boolean if <i>TRUE</i> it prints the progress of the simulations.   |
| return.mat      | a boolean if <i>TRUE</i> it returns a list of two elements : 1) a list of matrix deletions through target deletion; 2) a list of matrix deletions through random deletion |

### Details

Knockout analysis allows the study of resilience properties of networks when specific nodes are removed. It is usually compared with random deletions.

### Value

A list of two elements:

- The first element is the diameter of the network according to the option specified (weighted or not, directed or not, through lowest weights or greatest weights)
- The second element is the geodesic distances between all nodes according to the option specified (weighted or not, directed or not, through lowest weights or greatest weights)

### Author(s)

Sebastian Sosa, Ivan Puga-Gonzalez.

### References

- Lusseau D. 2003. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London Series B: Biological Sciences* 270(Suppl 2):S186-S188.
- Manno TG. 2008. Social networking in the Columbian ground squirrel, *Spermophilus columbianus*. *Animal Behaviour* 75(4):1221-1228.
- Kanngiesser P, Sueur C, Riedl K, Grossmann J, Call J. 2011. Grooming network cohesion and the role of individuals in a captive chimpanzee group. *American journal of primatology* 73(8):758-767.
- Sosa S. 2014. Structural Architecture of the Social Network of a Non-Human Primate (*Macaca sylvanus*): A Study of Its Topology in La Foret des Singes, Rocamadour. *Folia Primatologica* 85(3):154-163.
- Sosa, S. (2018). Social Network Analysis, *in*: Encyclopedia of Animal Cognition and Behavior. Springer.



**Examples**

```

# Simulating data
m <- matrix(sample(c(0:5), 50 * 50, TRUE), 50, 50)
diag(m) <- 0
mb <- mat.binaryzation(m)

# Weighted categorical attribute example
attr <- sample(c("a", "b"), 50, TRUE)
t <- stat.deletions(m, attr = attr, target = "a", nsim = 2, ndel = 10)
t <- stat.deletions(mb, attr = attr, target = "a", nsim = 2, ndel = 10)
# continuous attribute example
attr <- c(sample(c(1:10), 50, TRUE))

t <- stat.deletions(m, attr = attr, target = "decreasing", nsim = 2, ndel = 4)
t <- stat.deletions(mb, attr = attr, target = "decreasing", nsim = 2, ndel = 4)

```

---

stat.deletionsPlot      *Plot for network deletion simulations*

---

**Description**

Plot mean and standard deviation of the deletion simulation in the form of dual line plots.

**Usage**

```
stat.deletionsPlot(ant, col.target = NULL, col.random = NULL)
```

**Arguments**

|            |  |
|------------|--|
| ant        | an ant object returned by function sta.deletion          |
| col.target | a character indicating the color of the target deletions |
| col.random | a character indicating the color of the random deletions |

**Details**

knockout analysis allows the study of resilience properties of networks when specific nodes are removed. It is usually compared with random deletions.

**Value**

a plot.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

## Examples

```
M=matrix(sample(c(1:20),100,TRUE),ncol=10,nrow=10)
diag(M)=0
attr=sample(c('a','b'),10,TRUE)
target='a' #categorical target node attributes
t=stat.deletions(M,attr = attr,target = target,nsim = 2,ndel=4)
stat.deletionsPlot(t)
```

---

stat.glm

*Permuted Generalized Linear Model*


---

## Description

Performs Generalized Linear Models tests and computes permuted p-values

## Usage

```
stat.glm(
  ant,
  oda,
  formula,
  family = "gaussian",
  progress = TRUE,
  start = NULL,
  control = list(...),
  model = TRUE,
  method = "glm.fit",
  x = FALSE,
  y = TRUE,
  contrasts = NULL,
  ...
)
```

## Arguments

- |         |  |
|---------|--|
| ant     | an output of ANT function <a href="#">perm.net.nl</a> without any random factor declared, or output of ANT 'met' category functions in which output of ANT functions <a href="#">perm.ds.focal</a> , <a href="#">perm.ds.grp</a> or <a href="#">perm.net.lk</a> where single matrices were used. |
| oda     | the original data frame of associations when argument ant is obtained with <a href="#">perm.ds.focal</a> or <a href="#">perm.ds.grp</a> ANT functions.   |
| formula | an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.   |

|           |  |
|-----------|--|
| family    | a description of the error distribution and link function to be used in the model. For glm this can be a character string naming a family function, a family function or the result of a call to a family function. For glm.fit only the third option is supported, see <a href="#">glm</a> and <a href="#">family</a> . |
| progress  | a boolean indicating the visualization of the permutation process.   |
| start     | starting values for the parameters in the linear predictor.  |
| control   | a list of parameters for controlling the fitting process.  |
| model     | a logical value indicating whether model frame should be included as a component of the returned value.  |
| method    | the method to be used in fitting the model. The default method "glm.fit" uses iteratively reweighted least squares (IWLS); the alternative "model.frame" returns the model frame and does no fitting.  |
| x, y      | For glm: logical values indicating whether the response vector and model matrix used in the fitting process should be returned as components of the returned value.  |
| contrasts | an optional list. See the contrasts.arg of model.matrix.default.   |
| ...       | Extra arguments for <a href="#">glm</a> function only.   |

### Details

This function is the first step in the process to create a t-test in permuted data. For more details on t-tests, see R documentation.

### Value

Returns a list of 3 elements :

- An object of class inheriting from "glm" which inherits from the class "lm".
- A data frame if the estimates of the permuted models.
- A vector of integers indicating the permutations that returned model errors or warnings (e.g. model convergence issues) and for which new permutations were done.

### Author(s)

Sebastian Sosa, Ivan Puga-Gonzalez.

### References

- Dobson, A. J. (1990) An Introduction to Generalized Linear Models. London: Chapman and Hall.
- Hastie, T. J. and Pregibon, D. (1992) Generalized linear models. Chapter 6 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.
- McCullagh P. and Nelder, J. A. (1989) Generalized Linear Models. London: Chapman and Hall.
- Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. New York: Springer.

### See Also

[glm](#)

**Examples**

```
t=met.degree(sim.m, sym = TRUE,sim.df,1) # Computing network metric
t=perm.net.nl(t,labels='age',rf=NULL,nperm=10,progress=FALSE) # Node label permutations
r.glm=stat.glm(ant = t,formula = degree ~ age,progress=FALSE) # Permuted GLM
```

---

|           |   |
|-----------|---|
| stat.glmm | <i>Extracts statistical measures of interest in Generalized Linear Mixed Models</i> |
|-----------|---|

---

**Description**

Performs Generalized Linear Mixed Models tests

**Usage**

```
stat.glmm(ant, formula, family, oda = NULL, progress = TRUE, ncores = 1, ...)
```

**Arguments**

|          |   |
|----------|---|
| ant      | an output of ANT function <a href="#">perm.net.nl</a> with random factor stated, or output of ANT 'met' categories functions in which output of ANT functions <a href="#">perm.ds.focal</a> , <a href="#">perm.ds.grp</a> or <a href="#">perm.net.lk</a> where multiple matrices have been used.  |
| formula  | two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. Random-effects terms are distinguished by vertical bars ( ) separating expressions for design matrices from grouping factors. Two vertical bars (  ) can be used to specify multiple uncorrelated random effects for the same grouping variable. (Because of the way it is implemented, the   -syntax works only for design matrices containing numeric (continuous) predictors; to fit models with independent categorical effects, see <a href="#">dummy</a> or the <a href="#">lmer_alt</a> function from the <a href="#">afex</a> package.). |
| family   | a GLM family, see <a href="#">glm</a> and <a href="#">family</a> .  |
| oda      | the original data frame of associations when argument ant is obtained with <a href="#">perm.ds.grp</a> or <a href="#">perm.ds.focal</a> ANT functions.  |
| progress | a boolean indicating the visualization of the permutation process.  |
| ncores   | an integer indicating the number of jobs to create for parallelization.   |
| ...      | Extra arguments for <a href="#">lmer</a> or <a href="#">glmer</a> function only.  |

**Details**

GLMM with permutation data.

**Value**

Returns a list of 3 elements :

- An object of class `merMod` (more specifically, an object of subclass `lmerMod` or `glmerMod`), for which many methods are available (e.g. `methods(class="merMod")`).
- A data frame if the estimates of the permuted models.
- A vector of integers indicating the permutations that returned model errors or warnings (e.g. model convergence issues) and for which new permutations were done.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**See Also**

[lmer](#) or [glmer](#)

**Examples**

```
# Creating temporal data-----
m2=matrix(sample(sim.m),20,20)
diag(m2)=0
colnames(m2)=colnames(sim.m)
row.names(m2)=row.names(sim.m)
df2=sim.df
df2$age=df2$age+1
df1=sim.df
df1$period=rep(1,nrow(df1))
df2$period=rep(2,nrow(df2))
# Data structure for multiple matrices analytical protocol-----
sim.lm=list(sim.m,m2)
sim.ldf=list(df1,df2)
# Computing network metric-----
t=met.strength(sim.lm,sim.ldf,1)
# Node label permutations-----
t=perm.net.nl(t,labels='age',rf="period",nperm=10,progress=FALSE)
# Permuted GLMM-----
r.glmm=stat.glmm(ant = t,formula = strength ~ age + (1|id),family = gaussian(), progress=TRUE)
# Rstudio parallelization.-----
## Not run: r.glmm=stat.glmm(t,formula = strength ~ age + (1|id),family = gaussian(), ncores = 10)
```

---

stat.glmm.no.first.model

*Extracts statistical measures of interest in Generalized Linear Mixed Models*

---

**Description**

Performs Generalized Linear Mixed Models tests

**Usage**

```
stat.glm.no.first.model(
  ant,
  formula,
  family,
  oda = NULL,
  progress = TRUE,
  odf,
  ...
)
```

**Arguments**

|                       |  |
|-----------------------|--|
| <code>ant</code>      | an output of ANT function <code>perm.net.nl</code> with random factor stated, or output of ANT 'met' categories functions in which output of ANT functions <code>perm.ds.focal</code> , <code>perm.ds.grp</code> or <code>perm.net.lk</code> where multiple matrices have been used.   |
| <code>formula</code>  | two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a <code>~</code> operator and the terms, separated by <code>+</code> operators, on the right. Random-effects terms are distinguished by vertical bars ( <code> </code> ) separating expressions for design matrices from grouping factors. Two vertical bars ( <code>  </code> ) can be used to specify multiple uncorrelated random effects for the same grouping variable. (Because of the way it is implemented, the <code>  </code> -syntax works only for design matrices containing numeric (continuous) predictors; to fit models with independent categorical effects, see <code>dummy</code> or the <code>lmer_alt</code> function from the <code>afex</code> package.). |
| <code>family</code>   | a GLM family, see <a href="#">glm</a> and <a href="#">family</a> .   |
| <code>oda</code>      | the original data frame of associations when argument <code>ant</code> is obtained with <code>perm.ds.grp</code> or <code>perm.ds.focal</code> ANT functions.  |
| <code>progress</code> | a boolean indicating the visualization of the permutation process.   |
| <code>odf</code>      | the original data frame  |
| <code>...</code>      | Extra arguments for <code>lmer</code> or <code>glmer</code> function only.   |

**Details**

GLMM with permutation data.

**Value**

Returns a list of 3 elements :

- An object of class `merMod` (more specifically, an object of subclass `lmerMod` or `glmerMod`), for which many methods are available (e.g. `methods(class="merMod")`).
- A data frame if the estimates of the permuted models.
- A vector of integers indicating the permutations that returned model errors or warnings (e.g. model convergence issues) and for which new permutations were done.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

**See Also**

[lmer](#) or [glmer](#)

---

|                    |   |
|--------------------|---|
| stat.glmm.parallel | <i>Extracts statistical measures of interest in Generalized Linear Mixed Models in parallel</i> |
|--------------------|---|

---

**Description**

Performs Generalized Linear Mixed Models tests in parallel

**Usage**

```
stat.glmm.parallel(  
  ant,  
  formula,  
  family,  
  oda = NULL,  
  progress = FALSE,  
  ncores = NULL,  
  ...  
)
```

**Arguments**

|          |  |
|----------|--|
| ant      | an output of ANT function <code>perm.net.nl</code> with random factor stated, or output of ANT 'met' categories functions in which output of ANT functions <code>perm.ds.focal</code> , <code>perm.ds.grp</code> or <code>perm.net.lk</code> where multiple matrices have been used.   |
| formula  | two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a <code>~</code> operator and the terms, separated by <code>+</code> operators, on the right. Random-effects terms are distinguished by vertical bars ( <code> </code> ) separating expressions for design matrices from grouping factors. Two vertical bars ( <code>  </code> ) can be used to specify multiple uncorrelated random effects for the same grouping variable. (Because of the way it is implemented, the <code>  </code> -syntax works only for design matrices containing numeric (continuous) predictors; to fit models with independent categorical effects, see <code>dummy</code> or the <code>lmer_alt</code> function from the <code>afex</code> package.). |
| family   | a GLM family, see <a href="#">glm</a> and <a href="#">family</a> .   |
| oda      | the original data frame of associations when argument <code>ant</code> is obtained with <code>perm.ds.grp</code> or <code>perm.ds.focal</code> ANT functions.  |
| progress | a boolean indicating the visualization of the permutation process.   |
| ncores   | an integer indicating the number of jobs to create for parallelization.  |
| ...      | Extra arguments for <code>lmer</code> or <code>glmer</code> function only.   |

**Details**

GLMM with permutation data.

**Value**

Returns a list of 3 elements :

- An object of class `merMod` (more specifically, an object of subclass `lmerMod` or `glmerMod`), for which many methods are available (e.g. `methods(class="merMod")`).
- A data frame of the estimates of the permuted models.
- A vector of integers indicating the permutations that returned model errors or warnings (e.g. model convergence issues) and for which new permutations were done.

**See Also**

[lmer](#) or [glmer](#)

---

stat.lm

*Extracts statistical measures of interest in Linear Model*

---

**Description**

Performs correlations Generalized Linear Models tests and extracts estimates of predictor factors in each permuted model.

**Usage**

```
stat.lm(  
  ant,  
  formula,  
  oda,  
  progress = TRUE,  
  method = "qr",  
  model = TRUE,  
  x = FALSE,  
  y = FALSE,  
  qr = TRUE,  
  singular.ok = TRUE,  
  contrasts = NULL,  
  ...  
)
```



**Arguments**

|             |  |
|-------------|--|
| ant         | an output of ANT function <code>perm.net.n1</code> without any random factor declared, or output of ANT 'met' categories functions in which output of ANT functions <code>perm.ds.focal</code> , <code>perm.ds.grp</code> or <code>perm.net.lk</code> where single matrices were used. |
| formula     | an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.   |
| oda         | the original data frame of associations when argument <code>ant</code> is obtained with <code>perm.ds.grp</code> or <code>perm.ds.focal</code> ANT functions.  |
| progress    | a boolean indicating the visualization of the permutation process.   |
| method      | the method to be used; for fitting, currently only <code>method = "qr"</code> is supported; <code>method = "model.frame"</code> returns the model frame (the same as with <code>model = TRUE</code> , see below).  |
| model       | logicals. If <code>TRUE</code> the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.   |
| x           | logicals. If <code>TRUE</code> the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.   |
| y           | logicals. If <code>TRUE</code> the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.   |
| qr          | logicals. If <code>TRUE</code> the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.   |
| singular.ok | logical. If <code>FALSE</code> (the default in S but not in R) a singular fit is an error.   |
| contrasts   | an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .  |
| ...         | Extra arguments for <code>lm</code> function only.   |

**Details**

This function is the first step for performing t-tests in permuted data. For more details on t-tests, see R documentation.

**Value**

Returns a list of 3 elements :

- An object of class "lm" or for multiple responses of class `c("mlm", "lm")`.
- A data frame if the estimates of the permuted models.
- A vector of integers indicating the permutations that returned model errors or warnings (e.g. model convergence issues) and for which new permutations were done.

**Author(s)**

Sebastian Sosa, Ivan Puga-Gonzalez.

## References

Chambers, J. M. (1992) Linear models. Chapter 4 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

Wilkinson, G. N. and Rogers, C. E. (1973) Symbolic descriptions of factorial models for analysis of variance. Applied Statistics, 22, 392-9.

## See Also

[lm](#)

## Examples

```
t=met.strength(sim.m,sim.df,1) # Computing network metric
t=perm.net.nl(t,labels='age',rf=NULL,nperm=10,progress=FALSE) # Node label permutations
r.lm=stat.lm(t,formula = strength ~ age,progress=FALSE) # Permuted LM
```

---

stat.t

*T-test on data frame*

---

## Description

Performs correlations T-test through observed and permuted data to calculate the p-value on the posterior distribution

## Usage

```
stat.t(
  ant,
  formula,
  alternative = "two.sided",
  na.action = na.omit,
  mu = 0,
  paired = FALSE,
  var.equal = FALSE,
  progress = TRUE
)
```

## Arguments

|             |   |
|-------------|---|
| ant         | an output of ANT function <a href="#">perm.net.nl</a> without any random factor declared, or output of ANT 'met' categories functions in which output of ANT functions <a href="#">perm.ds.focal</a> , <a href="#">perm.ds.grp</a> or <a href="#">perm.net.lk</a> where single matrix have been used. |
| formula     | a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs a factor with two levels giving the corresponding groups.  |
| alternative | a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.   |

|           |  |
|-----------|--|
| na.action | a function which indicates what should happen when the data contain NAs. Defaults to <code>getOption("na.action")</code> .   |
| mu        | a number indicating the true value of the mean (or difference in means if you are performing a two sample test).   |
| paired    | a logical indicating whether you want a paired t-test.   |
| var.equal | a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used. |
| progress  | a boolean indicating the visualization of the permutation process.   |

### Details

t-test on permuted data allows to extract the posterior distribution of the value of interest. The posterior distribution allows to calculate the p-value. For more details about t-tests, see R documentation.

### Value

a data frame with 2 columns: the t statistic, the met.degree of freedom, the confidence interval for var1 and 2, and the estimates for var1 and var2

### Author(s)

Sebastian Sosa, Ivan Puga-Gonzalez.

### See Also

[t.test](#)

### Examples

```
t=met.strength(sim.m,sim.df,1) # Computing network metric
t=perm.net.nl(t,labels='sex',rf=NULL,nperm=10,progress=FALSE) # Node label permutations
r.t=stat.t(t,formula = strength ~ sex,progress=FALSE) # Permuted t-test
```

---

stat.tauKr

*Matrix TauKr correlations*

---

### Description

Matrix TauKr correlation. Partial or not, with significant test or not.

### Usage

```
stat.tauKr(X, Y, Z = NULL, nperm = NULL, omitDiag)
```

**Arguments**

|          |   |
|----------|---|
| X        | a square matrix of individual interactions or associations  |
| Y        | a square matrix of individual interactions or associations  |
| Z        | a square matrix of individual interactions or attributes to use as a control in the matrix correlation between X and Y (only for partial correlation) |
| nperm    | an integer indicating the number of permutations wanted.  |
| omitDiag | boolean to omit or not the diagonals in the matrices if <i>TRUE</i> it does not consider diagonals  |

**Details**

returns a list of matrices in the same order as is the original folder.

**Value**

a list of matrices .

**Author(s)**

Ivan Puga-Gonzalez, Sebastian Sosa.

**References**

Hemelrijk, C. K. 1990. Models of, and tests for, reciprocity, unidirectional and other social interaction patterns at a group level. *Animal Behavior*, 39, 1013-1029

Hemelrijk, C. K. 1990. A matrix partial correlation test used in investigations of reciprocity and other social interaction patterns at a group level. *Journal of theoretical Biology*, 143, 405-420.

---

vis.post.distribution *Histogram of posterior distribution*

---

**Description**

Create histogram of posterior distribution and compute posterior distribution statistics (p-values and confidence interval).

**Usage**

```
vis.post.distribution(
  x,
  quantile = c(0.05, 0.95),
  background.color = "gray63",
  observe.value.color = "white",
  ci.lower.color = "white",
  ci.upper.color = "white",
  xlab = NULL,
```

```

    main = NULL,
    legend = TRUE,
    legend.position = "topright",
    record = TRUE
  )

```

### Arguments

`x` A numeric vector with the first value corresponding to the observed value.

`quantile` a numeric vector of length 2 to indicate the lower and upper confidence interval.

`background.color` A specification for the default histogram background color.

`observe.value.color` A specification for the default abline line color of the observed value.

`ci.lower.color` A specification for the default abline line color of the lower confidence interval.

`ci.upper.color` A specification for the default abline line color of the lower confidence interval.

`xlab` A string for x axis label.

`main` A string for histogram main title.

`legend` A boolean to print or not the legend

`legend.position` The x co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by `xy.coords`:

`record` A boolean indicating to return or not the histogram in a R object.

### Value

an histogram of posterior distribution.

### Examples

```

t=met.strength(sim.m,sim.df,1) # Computing network metric
t=perm.net.nl(t,labels='age',rf=NULL,nperm=1000,progress=FALSE) # Node label permutations
r.c=stat.cor(t,'age','strength',progress=FALSE) # Permuted correlation test
vis.post.distribution(r.c[,1])# Histogram of posterior distribution

```

---

```
vis.post.distribution2
```

*Histogram of posterior distribution*

---

### Description

Create histogram of posterior distribution and compute posterior distribution statistics (p-values and confidence interval).

**Usage**

```
vis.post.distribution2(
  x,
  quantile = c(0.05, 0.95),
  backgroud.color = "gray63",
  observe.value.color = "white",
  ci.lower.color = "white",
  ci.upper.color = "white",
  xlab = NULL,
  main = NULL,
  legend = TRUE,
  legend.position = "topright",
  record = TRUE
)
```

**Arguments**

|                                  |  |
|----------------------------------|--|
| <code>x</code>                   | A numeric vector with the first value corresponding to the observed value.   |
| <code>quantile</code>            | a numeric vector of length 2 to indicate the lower and upper confidence interval.  |
| <code>backgroud.color</code>     | A specification for the default histogram background color.  |
| <code>observe.value.color</code> | A specification for the default abline line color of the observed value.   |
| <code>ci.lower.color</code>      | A specification for the default abline line color of the lower confidence interval.  |
| <code>ci.upper.color</code>      | A specification for the default abline line color of the lower confidence interval.  |
| <code>xlab</code>                | A string for x axis label.   |
| <code>main</code>                | A string for histogram main title.   |
| <code>legend</code>              | A boolean to print or not the legend   |
| <code>legend.position</code>     | The x co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by <code>xy.coords</code> : |
| <code>record</code>              | A boolean indicating to return or not the histogram in a R object.   |

**Value**

an histogram of posterior distribution.

**Examples**

```
t=met.strength(sim.m,sim.df,1) # Computing network metric
t=perm.net.nl(t,labels='age',rf=NULL,nperm=1000,progress=FALSE) # Node label permutations
r.c=stat.cor(t,'age','strength',progress=FALSE) # Permuted correlation test
vis.post.distribution(r.c[,1])# Histogram of posterior distribution
```

---

|              |                               |
|--------------|-------------------------------|
| which.metric | <i>Which metric to choose</i> |
|--------------|-------------------------------|

---

**Description**

An interactive decision tree to choose the most appropriate network measure according to the research question and network analysis level.

**Usage**

```
which.metric()
```

**Details**

For more details on each of these metrics, see Sosa et al. 2020.

**Value**

a decision tree.

**Author(s)**

Sebastian Sosa

**References**

Sosa, S., Sueur, C., & Puga Gonzalez, I. (2020). Network measures in animal social network analysis: their strengths, limits, interpretations and uses. *Methods in Ecology and Evolution*.

---

|                |                               |
|----------------|-------------------------------|
| which.protocol | <i>Which metric to choose</i> |
|----------------|-------------------------------|

---

**Description**

An interactive decision tree to choose the most appropriate ANTs analytical protocol and related function according to the research question.

**Usage**

```
which.protocol()
```

**Value**

a decision tree.

**Author(s)**

Sebastian Sosa

# Index

## \* datasets

- sim.df, 58
- sim.focal.directed, 58
- sim.focal.undirected, 59
- sim.gbi, 59
- sim.gbi.att, 60
- sim.grp, 60
- sim.m, 61
- sim.socprog, 61

## \* export

- which.metric, 79
- which.protocol, 79

ant, 4

assoc.gfi, 5

assoc.indices, 6, 9

convert.socprog, 7

df.create, 8

df.to.gbi, 8

df.to.mat, 9

family, 67, 68, 70, 71

gbi.to.df, 10

glm, 67, 68, 70, 71

glmer, 68, 69, 71, 72

import.df, 11

import.mat, 12

lm, 73, 74

lmer, 68, 69, 71, 72

mat.to.edgl, 13

merMod, 69, 70, 72

met.affinity, 14

met.assortativity, 15

met.betweenness, 16

met.ci, 18

met.degree, 19

met.density, 20

met.diameter, 21

met.disparity, 23

met.eigen, 24

met.ge, 25

met.geodesic, 27

met.indegree, 28

met.instrength, 29

met.lp, 31

met.outdegree, 32

met.outstrength, 33

met.reach, 34

met.ri, 36

met.strength, 37

perm.double.focal, 38

perm.double.focal.single, 40

perm.double.grp, 41

perm.double.grp.single, 43

perm.ds.focal, 44, 66, 68, 74

perm.ds.grp, 45, 66, 68, 74

perm.net.degree, 46

perm.net.lk, 47, 66, 68, 74

perm.net.lk.w, 48

perm.net.nl, 49, 66, 68, 74

perm.net.nl.str, 50

read.csv, 11, 12

sampling.effort, 51

sampling.robustness, 53

sampling.uncertainty, 56

sim.df, 58

sim.focal.directed, 58

sim.focal.undirected, 59

sim.gbi, 59

sim.gbi.att, 60

sim.grp, 60

sim.m, 61



sim.socprog, [61](#)  
stat.cor, [62](#)  
stat.deletions, [63](#)  
stat.deletionsPlot, [65](#)  
stat.glm, [66](#)  
stat.glmm, [68](#)  
stat.glmm.no.first.model, [69](#)  
stat.glmm.parallel, [71](#)  
stat.lm, [72](#)  
stat.t, [74](#)  
stat.tauKr, [75](#)

t.test, [75](#)

vis.post.distribution, [76](#)  
vis.post.distribution2, [77](#)

which.metric, [79](#)  
which.protocol, [79](#)