Package 'networkR'

May 9, 2026

Type Package

Title Network Analysis and Visualization

Version 0.1.5

Date 2025-09-22

Maintainer Claus Thorn Ekstrøm <ekstrom@sund.ku.dk>

Description Collection of functions for fast manipulation, handling, and analysis of large-scale networks based on family and social data. Functions are utility functions used to manipulate data in three ``formats'': sparse adjacency matrices, pedigree trio family data, and pedigree family data. When possible, the functions should be able to handle millions of data points quickly for use in combination with data from large public national registers and databases. Kenneth Lange (2003, ISBN:978-8181281135).

Imports Rcpp, data.table, fastmatch, Matrix

Suggests testthat

LinkingTo Rcpp, RcppArmadillo

License GPL (>= 2)

Encoding UTF-8

ByteCompile true

RoxygenNote 7.3.3

NeedsCompilation yes

Author Claus Thorn Ekstrøm [aut, cre], Bendix Carstensen [ctb]

Repository CRAN

Date/Publication 2025-09-23 06:40:16 UTC

Contents

adjacency 2
hits 3
make_family_id 4

make_parental_chain 5
mksib 6
validate_trio_consistency 7

Index **9**

`adjacency` *Create adjacency matrix*

Description

Create an adjacency matrix from a set of nodes and edges.

Usage

```
adjacency(from, to, weight = 1, directed = TRUE)
```

Arguments

`from`	a vector of nodes where the edges originate
`to`	a vector of nodes where the edges point to
`weight`	a numeric vector of weights
`directed`	logical. Are the edges directed (TRUE, the default) or bidirected(FALSE).

Value

Returns a sparse adjacency matrix

Author(s)

Claus Ekstrom <ekstrom@sund.ku.dk>

Examples

```
from <- c("A", "A", "A", "B", "C")
to <- c("B", "C", "D", "D", "E")
adjacency(from, to)

from <- c("A", "A", "A", "B", "C")
to <- c("B", "C", "D", "D", "E")
weights <- c(1, .5, 1, .7, 1)
adjacency(from, to, weights)
```

hits *Hyperlink-induced topic search*

Description

Hyperlink-induced topic search (HITS) is a link analysis algorithm that is also known as hubs and authorities. It rates nodes by comparing arrows pointing in and out of nodes in an asymmetrical graph.

Usage

```
hits(adjmatrix, maxiter = 100L, tol = 1e-05)
```

Arguments

adjmatrix	an adjacency matrix
maxiter	non-negative integer
tol	positive numeric value to be used as tolerance threshold for convergence

Details

Hubs are nodes with a lot of arrows pointing out while authorities are node with a lot of arrows pointing in.

Value

Returns a list with three elements: authorities (a vector) of and hubs (a vector), and number of iterations used.

Author(s)

Claus Ekstrom <ekstrom@sund.ku.dk>

References

Kleinberg, Jon (1999). "Authoritative sources in a hyperlinked environment" (PDF). Journal of the ACM. 46 (5): 604–632. doi:10.1145/324133.324140

Examples

```
from <- c("A", "A", "A", "B", "C")
to <- c("B", "C", "D", "D", "E")
hits(adjacency(from, to))
```

make_family_id *Construct family id vector from pedigree trio information*

Description

Create a vector of length n, giving the family id of each subject. If the pedigree is totally connected, then everyone will end up in tree 1, otherwise the tree numbers represent the disconnected subfamilies. Singleton subjects each have unique family numbers.

No check is done to ensure that the id, fid, and mid actually refer to proper family structure. References to ids in the fid and mid arguments that are not part of the id vector are considered founders and are thus replaced by NA or 0s after being used to group full and half-sibs.

Usage

```
make_family_id(id, fid, mid)
```

Arguments

id	Numeric vector of ids
fid	Numeric vector of ids of the father. This should be NA or 0 for a founder.
mid	Numeric vector of ids of the mother. This should be NA or 0 for a founder.

Value

Returns an integer vector giving the family index of each individual

Author(s)

Claus Ekstrom <ekstrom@sund.ku.dk>

Examples

```
id <- 1:11
fid <- c(NA, NA, 1, 1, NA, 23, 45, 5, 5, 7, NA)
mid <- c(NA, NA, 2, 2, 65, NA, 46, 6, 6, 6, 0)
make_family_id(id, fid, mid)
```

`make_parental_chain` *Construct parental chain id vector from pedigree trio information*

Description

Create a vector of length n, giving the id of parental chains. A parental chain is set of individuals that are all linked because they have children together. If the pedigree is totally connected, then everyone will end up in tree 1, otherwise the tree numbers represent the disconnected subfamilies. Singleton subjects each have unique family numbers.

No check is done to ensure that the id, fid, and mid actually refer to proper family structure. References to ids in the fid and mid arguments that are not part of the id vector are considered founders.

Usage

```
make_parental_chain(id, fid, mid)
```

Arguments

`id`	Numeric vector of ids
`fid`	Numeric vector of ids of the father. This should be NA or 0 for a founder.
`mid`	Numeric vector of ids of the mother. This should be NA or 0 for a founder.

Value

Returns an integer vector giving the family index

Author(s)

Claus Ekstrom <ekstrom@sund.ku.dk>

Examples

```
id <- 1:11
fid <- c(0,0,1,0,0,4,0,0,3,7,7)
mid <- c(0,0,2,0,0,5,0,0,6,6,8)
make_parental_chain(id, fid, mid)
```

mksib *Generate variables (or lists) of siblings from a file of ids of persons and their father and mother.*

Description

The function generates for each person lists of maternal half-sibs, paternal half-sibs and full sibs. Optionally these are expanded to separate columns in a data.table.

Usage

```
mksib(obj, ns = 3, expand.vars = TRUE)
```

Arguments

obj	A 3-column structure with column names id, pid (paternal id) and mid (maternal id).
ns	Integer. The maximal no of sibs of each type to include in the result if sibling ids are required in separate columns.
expand.vars	Logical. Should the sibling ids be returned in separate columns. If FALSE they will be returned i three columns of lists.

Details

There are no checks of persons being both mother and father, nor being its own parent and incest checks are not performed. In other words, the obj is assumed to be sane, but possibly immoral.

Value

A data.table with the columns of the obj and columns for ns maternal, paternal and full sibs, named ms1, ms2, ... ps1, ps2, ... fs1, fs2.

If expand.vars=FALSE there will instead be three columns of lists named msibs, psibs and fsibs.

Author(s)

Claus Thorn Ekstrøm, <ekstrom@sund.ku.dk>, Bendix Carstensen, <b@bxc.dk>

Examples

```
library( data.table )
id <- 1:12
pid <- c(NA,  1, 1, 1, NA, 23, 45, 5, 5, 7, 12, NA)
mid <- c(NA, NA, 2, 2, 12, NA, 46, 6, 6, 6, NA, 12)
indd <- data.table( id, mid, pid )
indata <- copy( indd )
indata

str( xx <- mksib( indata ) )
```

```
xx

zz <- mksib( indata, 2, e=FALSE )
zz
```

`validate_trio_consistency` *Validate pedigree trio information consistency*

Description

Simple tests to chech the consistency of the pedigree trio family data. Currently the following checks are undertaken: 1) that no duplicates ids are found; 2) that the primary id is not missing for anyone; 3) that founders have both the father and mother id missing; 4) that individuals are not both classified as male (fathers and mothers);

Usage

```
validate_trio_consistency(id, fid, mid, sex = NULL)
```

Arguments

`id`	Numeric. The id of the individual. These values should be unique
`fid`	Numeric. The father id. NA or 0 are used for missing.
`mid`	Numeric. The mother id. NA or 0 are used for missing.
`sex`	An optional numeric vector with the sex of the individual. Only four values should be present 1 (male), 2 (female), 0 or NA (missing)

Details

There are no checks of persons being both mother and father, nor being its own parent and incest checks are not performed. In other words, the `obj` is assumed to be sane, but possibly immoral.

Value

Throws an error if an inconsistency is found. Otherwise returns TRUE.

Author(s)

Claus Thorn Ekstrøm, <ekstrom@sund.ku.dk>

Examples

```
library("data.table")
id <- 1:12
fid <- c(NA,  0, 1, 1, NA, 23, 45, 5, 5, 7, 10, 10)
mid <- c(NA, NA, 2, 2,  0, 56, 46, 6, 6, 6, 9, 11)

validate_trio_consistency(id, fid, mid)
```

Index

∗ **manip**
adjacency, 2
hits, 3
make_family_id, 4
make_parental_chain, 5
mksib, 6
validate_trio_consistency, 7

adjacency, 2

hits, 3

make_family_id, 4
make_parental_chain, 5
mksib, 6

validate_trio_consistency, 7