Package ‘snha’

May 9, 2026

Type Package

Title Creating Correlation Networks using St. Nicolas House Analysis

Version 0.1.3

Date 2023-05-13

Maintainer Detlef Groth <`dgroth@uni-potsdam.de`>

Description Create correlation networks using St. Nicolas House Analysis ('SNHA').
The package can be used for visualizing multivariate data similar to Principal
Component Analysis or Multidimensional Scaling using a ranking approach.
In contrast to 'MDS' and 'PCA', 'SNHA' uses a network approach to explore
interacting variables.
For details see 'Hermanussen et. al. 2021', <doi:10.3390/ijerph18041741>.

URL `https://github.com/mittelmark/snha`

BugReports `https://github.com/mittelmark/snha/issues`

Depends R (>= 3.5.0)

Imports MASS

Suggests knitr, rmarkdown

VignetteBuilder knitr

License MIT + file LICENSE

LazyData yes

Language en-US

Encoding UTF-8

NeedsCompilation no

Collate ll.R asgp.R priv.R snha.R

Author Detlef Groth [aut, cre] (ORCID:
<`https://orcid.org/0000-0002-9441-3978`>)

Repository CRAN

Date/Publication 2023-03-14 11:40:02 UTC

Contents

snha-package 2
as.list.snha 3
decathlon88 4
plot.snha 5
snha 7
snha_get_chains 9
snha_graph2data 9
snha_layout 10
snha_ll 12
snha_rsquare 12

Index 14

`snha-package` *snha package - association chain graphs from correlation networks*

Description

The snha package can be used to construct association chain graphs based on the St. Nicolas House Analysis (SNHA) algorithm as described in Groth et. al. 2019. and Hermanussen et. al. 2021.

Details

The package provides the following functions: Function for graph generation from data:

snha(data) applys the SNHA method on the data and returns a new snha graph object

S3 methods for snha graphs:

plot.snha(x) plots a snha graph

as.list.snha(x) return a list representation of a snha graph object

Utility functions:

snha_get_chains(g) returns the chains found by the algorithm as matrix

snha_graph2data(A) create for the given adjacency matrix some data with the appropiate correlations

snha_layout(g) calculate layout coordinates for the given graph or adjacency matrix

snha_ll(g,chain) calculate log-likelihood for the given chain of the snha graph

snha_rsquare(data,g) for given data and graph or adjacency matrix calculate linear model r-square value

Value

No return value

Author(s)

Detlef Groth <dgroth@uni-potsdam.de>

References

- Groth, D., Scheffler, C., & Hermanussen, M. (2019). Body height in stunted Indonesian children depends directly on parental education and not via a nutrition mediated pathway - Evidence from tracing association chains by St. Nicolas House Analysis. *Anthropologischer Anzeiger*, 76 No. 5 (2019), p. 445 - 451. doi: 10.1127/anthranz/2019/1027
- Hermanussen, M., Assmann, & Groth, D. (2021). Chain Reversion for Detecting Associations in Interacting Variables - St. Nicolas House Analysis. *International Journal of Environmental Research and Public Health*. 18, 4 (2021). doi: 10.3390/ijerph18041741.
- Novine, M., Mattsson, C. C., & Groth, D. (2021). Network reconstruction based on synthetic data generated by a Monte Carlo approach. *Human Biology and Public Health*, 3:26. doi: 10.52905/hbph2021.3.26

Examples

```
library(MASS)
data(birthwt)
as=snha(birthwt[,-1])
plot(as)
as$theta
ls(as)
data(decathlon88)
head(decathlon88)
dec=snha(decathlon88,method="spearman",alpha=0.1)
plot(dec,layout='sam')
```

`as.list.snha` *return a list representation for an snha graph object*

Description

The function ‘as.list.snha‘ provides a S3 method to convert a snha graph object into a list object which can be for instance used to write a report into an XLSX file using the library openxlsx.

Usage

```
## S3 method for class 'snha'
as.list(x,...)
```

Arguments

`x`	snha graph object created with the snha function
`...`	additional arguments, delegated to the list command

Value

list object with the components: 'chains' (the association chain), 'data' (original data), 'theta' (adjacency matrix, 'sigma' (correlations), 'p.value' (correlation p-values)

See Also

plot.snha, snha

Examples

```
data(swiss)
as=snha(swiss,method="spearman",alpha=0.1)
result=as.list(as)
ls(result)
result$settings
# can be writte as xlsx file for instance like:
# library(openxlsx)
# write.xlsx(result,file="some-result.xlsx")
```

`decathlon88` *Men Decathlon data from the 1988 Olympics*

Description

A subset of data from the Decathlon from the 1988 Olympic games. Included are all athletes which finished with more than 7000 points.

Usage

```
decathlon88
```

Format

A data frame with 33 rows and 10 columns:

disc	discus results in m
high	high jump results in m
jave	javelin through results in m
long	long jump results in m
pole	pole vault results in m
shot	shot put results in m
X100	running speed over 100m in km/h
X110	running speed over 110m hurdles in km/h
X1500	running speed over 1500m in km/h
X400	running speed over 400m in km/h

Source

<https://en.wikipedia.org/wiki/Athletics_at_the_1988_Summer_Olympics_-_Men's_decathlon>

Examples

```
data(decathlon88)
head(decathlon88)
A=snha(decathlon88,method="spearman",alpha=0.1)
cols=rep("salmon",10)
cols[names(A$data) %in% c("jave","shot","disc","pole")]="skyblue"
plot(A,layout="sam",vertex.color=cols,vertex.size=8,cex=1.2,edge.width=5)
snha_rsquare(A)
```

`plot.snha` *display network or correlation matrices of snha graphs*

Description

The function ‘plot.snha‘ provides a simple display of network graphs correlation matrices using filled circles (vertices) to represent variables and edges which connect the vertices with high absolute. correlation values. Positive correlations are shown in black, negative correlations are shown in red. For more information see the details section.

Usage

```
## S3 method for class 'snha'
plot(
 x,
 type = "network",
 layout = "circle",
 vertex.color = "salmon",
 cex = 1,
 vertex.size = 5,
 edge.width = 2,
 edge.color = c("grey70", "red"),
 edge.text = NULL,
 edge.cex = 0.8,
 edge.pch = 0,
 noise = FALSE,
 hilight.chain = NULL,
 chain.color = c("black", "red"),
 star.center = NULL,
 plot.labels = TRUE,
 lty = 1,
 threshold = c(0.25, 0.5, 0.75),
 interactive = FALSE,
 ...
)
```

Arguments

`x`	snha graph object usually created with the ’snha’ function or an adjacency matrix
`type`	character string specifying the plot type either ’network’ or ’ cor’, default: ’network’
`layout`	graph layout for plotting one of ’circle’, ’sam’, ’samd’, ’grid’, ’mds’, ’mdsd’, ’star’, default: ’circle’
`vertex.color`	default color for the vertices, either a single value, all vertices have hen this color or a vector of values, for different colors for the nodes, default: ’salmon’
`cex`	size of the vertex labels which are plotted on the vertices, default: 1
`vertex.size`	number how large the vertices should be plotted, default: 5
`edge.width`	number on how strong the edges should be plotted, if edge.width=0, then the number is based on the correlation values, default: 2
`edge.color`	color to be plotted for edges. Usually vector of length two. First color for positive correlations, second color for negative correlations. Default: c(’grey’,’red’)
`edge.text`	optional matrix to give edge labels, default: NULL
`edge.cex`	character expansion for edge labels, default: 0.8
`edge.pch`	plotting character which should be placed below the edge.text, default: 0
`noise`	should be noise added to the layout. Sometimes useful if nodes are too close. Default: FALSE
`hilight.chain`	which chain should be highlighted, default: NULL (no chain highlight)
`chain.color`	which color for chain edges, default: black
`star.center`	the centered node if layout is ’start’, must be a character string for the node name, default: NULL
`plot.labels`	should node labels plotted, default: TRUE
`lty`	line type for standard edges in the graph, default: 1
`threshold`	cutoff values for bootstrap probabilities for drawing edges as dotted, broken lines and solid lines, default: c(0.25,0.5,0.75)
`interactive`	switch into interactive mode where you can click in the graph and move nodes with two clicks, first selecting the node, second click gives the new coordinates for the node, default: FALSE
`...`	currently not used

Details

This is a plot function to display networks or correlation matrices of ’snha’ graph objects. In case of bootstrapping the graph by using the ‘snha‘ function with the ‘prob=TRUE‘ option lines in style full, broken and dotted lines are drawn if they are found in more than 75, 50 or 25 percent of all re-samplings. You can change these limits by using the ‘threshold‘ argument.

Value

returns the layout of the plotted network or NULL if type is ‘corrplot‘ (invisible)

Examples

```
data(swiss)
sw.g=snha(swiss,method='spearman')
sw.g$theta
round(sw.g$sigma,2)
plot(sw.g,type='network',layout='circle')
plot(sw.g,type='network',layout='sam')
plot(sw.g,type='corplot')
# adding correlation values
plot(sw.g,edge.text=round(sw.g$sigma,2),edge.cex=1.2,edge.pch=15)
sw.g=snha(swiss,method='spearman',prob=TRUE)
sw.g$theta
sw.g$probabilities
plot(sw.g,type='network',layout='sam')
sw.g$chains
# plot chains for a node
plot(sw.g,layout="sam",lty=2,hilight.chain="Infant.Mortality",
 edge.width=3,edge.color=c("black","red"))
# an example for an adjacency matrix
M=matrix(rbinom(100,1, 0.2),nrow=10,ncol=10)
diag(M)=0
colnames(M)=rownames(M)=LETTERS[1:10]
plot.snha(M)
```

`snha` *Initialize a snha object with data.*

Description

The main entry function to initialize a snha object with data where variables are in columns and items are in rows

Usage

```
snha(
  data,
  alpha=0.05,
  method='pearson',
  threshold=0.01,
  check.singles=FALSE,
  prob=FALSE,
  prob.threshold=0.2,
  prob.n=25)
```

Arguments

`data`	a data frame where network nodes are the row names and data variables are in the columns.

`alpha`	confidence threshold for p-value edge cutting after all chains were generated, default: 0.05.
`method`	method to calculate correlation/association values, can be 'pearson', 'spearman' or 'kendall', default: 'pearson'.
`threshold`	R-squared correlation coefficient threshold for which r-square values should be used for chain generation, r=0.1 is r-square of 0.01, default: 0.01.
`check.singles`	should isolated nodes connected with sufficient high R^2 and significance, default: FALSE.
`prob`	should be probabilities computed for each edge using bootstrapping. Only in this case the parameters starting with prob are used, default: FALSE
`prob.threshold`	threshold to set an edge, a value of 0.5 means, that the edge must be found in 50% of all samplings, default: 0.2
`prob.n`	number of bootstrap samples to be taken, default: 25

Value

A snha graph data object with the folling components:

chains association chains building the graph

data representing the original input data

p.values matrix with p-values for the pairwise correlations

probabilities in case of re-samplings, the proportion how often the chain was found

sigma correlation matrix used for the algorithm

theta adjacency matrix found by the SNHA method

See Also

plot.snha

Examples

```
data(swiss)
sw.g=snha(swiss,method='spearman')
# what objects are there?
ls(sw.g)
sw.g$theta
round(sw.g$sigma,2)
sw.g=snha(swiss,method='spearman',check.singles=TRUE,prob=TRUE)
sw.g$theta
sw.g$probabilities
```

`snha_get_chains` *Return the chains of an snha graph as data frame*

Description

This is a utility function to return the chains which constructs the graph as a matrix.

Usage

```
snha_get_chains(graph)
```

Arguments

`graph`	a snha graph object

Value

matrix with one chain per row, shorter chains are filled up with empty strings

Examples

```
data(swiss)
sw.g=snha(swiss)
snha_get_chains(sw.g)
```

`snha_graph2data` *create correlated data for the given adjacency matrix representing a directed graph or an undirected graph*

Description

This function is a short implementation of the Monte Carlo algorithm described in Novine et. al. 2022.

Usage

```
snha_graph2data(
  A,
  n=100,
  iter=50,
  val=100,
  sd=2,
  prop=0.025,
  noise=1,
  method="mc"
  )
```

Arguments

`A`	an adjacency matrix
`n`	number of values, measurements per node, default: 100
`iter`	number of iterations, default: 50
`sd`	initial standard deviation, default: 2
`val`	initial node value, default: 100
`prop`	proportion of the target node value take from the source node, default: 0.025
`noise`	sd for the noise value added after each iteration using rnorm function with mean 0, default: 1
`method`	method for data generation, either 'mc' for using Monte Carlo simulation or 'pc' for using a precision matrix, default: 'mc'

Value

matrix with the node names as rows and samplings in the columns

References

- Novine, M., Mattsson, C. C., & Groth, D. (2021). Network reconstruction based on synthetic data generated by a Monte Carlo approach. *Human Biology and Public Health*, 3:26. doi: 10.52905/hbph2021.3.26

Examples

```
opar=par(mfrow=c(1,2),mai=rep(0.2,4))
A=matrix(0,nrow=6,ncol=6)
rownames(A)=colnames(A)=LETTERS[1:6]
A[1:2,3]=1
A[3,4]=1
A[4,5:6]=1
A[5,6]=1
plot.snha(A,layout="circle");
data=snha_graph2data(A)
round(cor(t(data)),2)
P=snha(t(data))
plot(P,layout="circle")
par(opar)
```

`snha_layout` *Determine graph layouts*

Description

This function returns xy coordinates for a given input adjacency matrix or snha graph. It is useful if you like to plot the same set of nodes with different edge connections in the same layout.

Usage

```
snha_layout(
   A,
   mode='sam',
   method='pearson',
   noise=FALSE,
   star.center=NULL,
   interactive=FALSE)
```

Arguments

`A`	an adjacency matrix or an snha graph object
`mode`	character string for the layout type, can be either 'mds' (mds on graph using shortest paths), 'mdsd' (mds on data) 'sam' (sammon on graph), 'samd' (sammon on data), 'circle', 'grid' or 'star', default: 'sam'
`method`	method for calculating correlation distance if mode is either 'mdsd' or 'samd', default: 'pearson'
`noise`	should some noise be added, default: FALSE
`star.center`	the centered node if layout is 'star', must be a character string for the node name, default: NULL
`interactive`	switch into interactive mode where you can click in the graph and move nodes with two clicks, first selecting the node, second click gives the new coordinates for the node, default: FALSE

Value

matrix with x and y columns for the layout

Examples

```
data(swiss)
sw.s=snha(swiss,method='spearman')
sw.p=snha(swiss,method='pearson')
lay=snha_layout(sw.s,mode='sam')
plot(sw.s,layout=lay)
plot(sw.p,layout=lay)
plot(sw.s,layout='star',star.center='Education')
rn1=rnorm(nrow(swiss))
nswiss=cbind(swiss,Rn1=rn1)
plot(snha(nswiss,method='spearman'),layout='sam')
plot(snha(nswiss,method='spearman'),layout='samd',
  vertex.size=2,vertex.color='beige')
```

`snha_ll` *log-likelihood for the given snha graph and the given chain*

Description

This function returns the log-likelihood for the given snha graph and the given chain. If the 'block.p.value' is lower than 0.05 than that the chain is not sufficient to capture the variable dependencies, p-values above 0.05 indicate a good coverage of the chain for the linear dependencies between the nodes.

Usage

```
snha_ll(graph,chain=NULL)
```

Arguments

`graph`	a snha graph object
`chain`	a chain object of a snha graph, if not given a data frame with the values is returned for all chains, default: NULL

Value

list with the following components: 'll.total', 'll.chain', 'll.rest', 'll.block', data frame 'df' with the columns 'chisq', 'p.value', 'block.df', 'block.ch', 'block.p.value'. If chain is not given an overall summary is made for all chains an returned as data frame.

Examples

```
data(swiss)
sw.g=snha(swiss)
snha_ll(sw.g,sw.g$chain$Catholic)
head(snha_ll(sw.g))
```

`snha_rsquare` *linear model based r-square values for given data and graph*

Description

The function 'snha_rsquare' calculates for given data and a graph the covered r-squared values by a linear model for each node. The linear model predicts each node by an additive mode using it's neighbor nodes in the graph.

Usage

```
snha_rsquare(data,graph=NULL)
```

Arguments

`data`	data matrix or data frame where variables are in columns and samples in rows or a snha graph
`graph`	graph object or adjacency matrix of an (un)directed graph, not needed if data is a snha graph, default: NULL.

Value

vector of rsquare values for each node of the graph

Examples

```
# random adjacency matrix
A=matrix(rbinom(100,1, 0.2),nrow=10,ncol=10)
diag(A)=0
colnames(A)=rownames(A)=LETTERS[1:10]
# random data
data=matrix(rnorm(1000),ncol=10)
colnames(data)=colnames(A)
snha_rsquare(data,A)
# real data
data(swiss)
sw.s=snha(swiss,method='spearman')
rsqs=snha_rsquare(sw.s)
plot(sw.s,main=paste(”r =”,round(mean(rsqs,2))),
   layout='star',star.center='Examination')
# some colors for r-square values
vcols=paste(”grey”,seq(80,40,by=-10),sep=””)
scols=as.character(cut(snha_rsquare(swiss,sw.s$theta),
   breaks=c(0,0.1,0.3,0.5,0.7,1),labels=vcols))
plot(sw.s,main=paste(”r =”,round(mean(snha_rsquare(swiss,sw.s$theta)),2)),
   vertex.color=scols ,layout='star',star.center='Examination',
   vertex.size=10,edge.color=c('black','red'),edge.width=3)
```

Index

∗ **correlation**
    snha, 7
∗ **datasets**
    decathlon88, 4
∗ **layout**
    snha_layout, 10
∗ **network**
    snha, 7
    snha_layout, 10

as.list.snha, 3
as.list.snha(x), *2*

decathlon88, 4

plot.snha, *4*, 5, *8*
plot.snha(x), *2*

snha, *4*, 7
snha(data), *2*
snha-package, 2
snha_get_chains, 9
snha_get_chains(g), *2*
snha_graph2data, 9
snha_graph2data(A), *2*
snha_layout, 10
snha_layout(g), *2*
snha_ll, 12
snha_ll(g,chain), *2*
snha_rsquare, 12
snha_rsquare(data,g), *2*