

Package ‘heterocop’

May 5, 2025

Type Package
Title Semi-Parametric Estimation with Gaussian Copula
Version 0.1.1
Description A method for generating random vectors which are linked by a Gaussian copula. It also enables to estimate the correlation matrix of the Gaussian copula in order to identify independencies within the data.
Imports mvtnorm, stats, igraph, matrixcalc, graphics, foreach, stringr, doSNOW, utils, huge
License GPL (>= 3)
Encoding UTF-8
LazyData true
RoxygenNote 7.3.2
Depends R (>= 2.10)
Suggests knitr, rmarkdown, kableExtra, dplyr
VignetteBuilder knitr
NeedsCompilation no
Author Julie Cartier [aut],
Florence Jaffrezic [aut],
Gildas Mazo [aut],
Ekaterina Tomilina [aut, cre]
Maintainer Ekaterina Tomilina <ekaterina.tomilina@inrae.fr>
Repository CRAN
Date/Publication 2025-05-05 10:10:09 UTC

Contents

CopulaSim	2
cor_network_graph	3
diag_block_matrix	3
gauss_gen	4

icgc_data	5
matrix_cor_ts	5
matrix_gen	6
omega_estim	6
rho_estim	7
Index	8

CopulaSim	<i>CopulaSim</i>
-----------	------------------

Description

This function enables the user to simulate data from a Gaussian copula and arbitrary marginal quantile functions

Usage

```
CopulaSim(n, R, qdist, random = FALSE)
```

Arguments

- n the number of observations
- R a correlation matrix of size dxd
- qdist a vector containing the names of the marginal quantile functions as well as the number of times they are present in the dataset
- random a boolean defining whether the order of the correlation coefficients should be randomized

Value

a list containing an nxd data frame, the shuffled correlation matrix R, and the permutation leading to the new correlation matrix

Examples

```
M <- diag_block_matrix(c(3,4,5),c(0.7,0.8,0.2))
CopulaSim(20,M,c(rep("qnorm(0,1)",6),rep("qexp(0.5)",4),rep("qbinom(4,0.8)",2)),random=TRUE)
```

cor_network_graph *cor_network_graph*

Description

This function enables the user to plot the graph corresponding to the correlations of the Gaussian copula

Usage

```
cor_network_graph(R, TS, binary = TRUE, legend)
```

Arguments

R	a correlation matrix of size dxd (d is the number of variables)
TS	a threshold for the absolute values of the correlation matrix coefficients
binary	a boolean specifying whether the coefficients should be binarized, TRUE by default (zero if the coefficient is less than the threshold in absolute value, 1 otherwise). If FALSE, the edge width is proportional to the coefficient value.
legend	a vector containing the type of each variable used to color the vertices

Value

a graph representing the correlations between the latent Gaussian variables

Examples

```
R <- diag_block_matrix(c(3,4,5),c(0.7,0.8,0.2))
data <- CopulaSim(20,R,c(rep("qnorm(0,1)",6),rep("qexp(0.5)",4),
rep("qbinom(4,0.8)",2)),random=FALSE)[[1]]
cor_network_graph(R,TS=0.3,binary=TRUE,legend=c(rep("Normal",6),
rep("Exponential",4),rep("Binomial",2)))
```

diag_block_matrix *diag_block_matrix*

Description

This function enables the user to generate a diagonal block-matrix with homogeneous blocks

Usage

```
diag_block_matrix(blocks, coeff)
```

Arguments

blocks	a vector containing the sizes of the blocks
coeff	a vector containing the coefficient corresponding to each block, the coefficients must be between 0 and 1

Value

a diagonal block-matrix containing the specified coefficients

Examples

```
diag_block_matrix(c(3,4,5),c(0.3,0.4,0.8))
```

gauss_gen

gauss_gen

Description

This function enables the user to generate gaussian vectors with correlation matrix R

Usage

```
gauss_gen(R, n)
```

Arguments

R	a correlation matrix of size dxd
n	the number of observations

Value

a nxd data frame containing n observations of the d variables

Examples

```
M <- diag_block_matrix(c(3,4,5),c(0.7,0.8,0.2))
gauss_gen(M,20)
```

icgc_data	<i>ICGC dataset</i>
-----------	---------------------

Description

Dataset containing RNA counts, protein expression and mutations measured on breast cancer tumors.

Usage

icgc_data

Format

A dataframe of 15 variables and 250 observations containing the following:
ACACA, AKT1S1, ANLN,ANXA1,AR RNA counts (discrete)
ACACA_P, AKT1S1_P, ANLN_P,ANXA_P,AR_P protein expression measurements (discrete)
MU5219,MU4468,MU7870,MU4842,MU6962 5 mutations (binary)

matrix_cor_ts	<i>matrix_cor_ts</i>
---------------	----------------------

Description

This function enables the user to threshold matrix coefficients

Usage

matrix_cor_ts(R, TS, binary = TRUE)

Arguments

R a correlation matrix
TS a threshold
binary a boolean specifying whether the coefficients should be binarized, TRUE by default (zero if the coefficient is less than the threshold in absolute value, 1 otherwise)

Value

the thresholded input matrix

Examples

```
M <- diag_block_matrix(c(3,4,5),c(0.7,0.8,0.2))
matrix_cor_ts(M,0.5)
```

matrix_gen	<i>matrix_gen</i>
------------	-------------------

Description

This function enables the user to generate a sparse, nonnegative definite correlation matrix via the Cholesky decomposition

Usage

```
matrix_gen(d, gamma)
```

Arguments

d	the number of variables
gamma	an initial sparsity parameter for the lower triangular matrices in the Cholesky decomposition, must be between 0 and 1

Value

a list containing the generated correlation matrix and its final sparsity parameter (ie the proportion of zeros)

Examples

```
matrix_gen(15,0.81)
```

omega_estim	<i>omega_estim</i>
-------------	--------------------

Description

This function enables the user estimate the precision matrix of the latent variables via gLasso inversion

Usage

```
omega_estim(data, Type, lambda, n)
```

Arguments

data	a dataset of size nxd or a correlation matrix R of size dxd
Type	a vector containing the type of the variables, "C" for continuous and "D" for discrete (in the case a data set is entered as the first parameter)
lambda	a grid of penalization parameters to be evaluated
n	the sample size used (in the case of a correlation matrix entered as the first parameter)

Value

a list containing the correlation matrix, the optimal precision matrix, the optimal lambda, the minimal HBIC

Examples

```
M <- diag_block_matrix(c(3,4,5),c(0.7,0.8,0.2))
data <- CopulaSim(20,M,c(rep("qnorm(0,1)",6),rep("qexp(0.5)",4),
rep("qbinom(4,0.8)",2)),random=FALSE)[[1]]
P <- omega_estim(data,c(rep("C",10),rep("D",2)),seq(0.01,1,0.05))
```

rho_estim

*rho_estim***Description**

This function enables the user to estimate the correlation matrix of the Gaussian copula for a given dataset

Usage

```
rho_estim(data, Type, ncores = 1)
```

Arguments

data	an nxd data frame containing n observations of d variables
Type	a vector containing the type of the variables, "C" for continuous and "D" for discrete
ncores	an integer specifying the number of cores to be used for parallel computation. "1" by default, leading to non-parallel computation.

Value

the dxd estimated correlation matrix of the Gaussian copula

Examples

```
M <- diag_block_matrix(c(3,4,5),c(0.7,0.8,0.2))
data <- CopulaSim(20,M,c(rep("qnorm(0,1)",6),rep("qexp(0.5)",4),
rep("qbinom(4,0.8)",2)),random=FALSE)[[1]]
rho_estim(data,c(rep("C",10),rep("D",2)))
```

Index

* **datasets**

icgc_data, [5](#)

CopulaSim, [2](#)

cor_network_graph, [3](#)

diag_block_matrix, [3](#)

gauss_gen, [4](#)

icgc_data, [5](#)

matrix_cor_ts, [5](#)

matrix_gen, [6](#)

omega_estim, [6](#)

rho_estim, [7](#)