Package ‘GeneNet’

September 15, 2013

Version 1.2.8
Date 2013-09-15
Title Modeling and Inferring Gene Networks
Author Juliane Schaefer, Rainer Opgen-Rhein, and Korbinian Strimmer.
Maintainer Korbinian Strimmer <strimmer@uni-leipzig.de>
Depends R (= 2.15.1), corpcor (= 1.6.6), longitudinal (= 1.1.8), fdrtool (= 1.2.11), igraph (= 0.6)
Suggests
Description GeneNet is a package for analyzing gene expression (time series) data with focus on the inference of gene networks. In particular, GeneNet implements the methods of Schaefer and Strimmer (2005a,b,c) and Opgen-Rhein and Strimmer (2006, 2007) for learning large-scale gene association networks (including assignment of putative directions).
License GPL (= 3)
URL http://strimmerlab.org/software/genenet/
NeedsCompilation no
Repository CRAN
Date/Publication 2013-09-15 09:50:09

R topics documented:

GeneNet-package .................................................. 2
arth800 .......................................................... 2
cor0.test ......................................................... 3
ecoli .............................................................. 5
ggm.estimate.pcor ............................................... 6
ggm.make.igraph ................................................ 7
ggm.simulate.data .............................................. 9
The GeneNet package

Description

GeneNet is a package for analyzing gene expression (time series) data with focus on the inference of gene networks. In particular, GeneNet implements the methods of Sch"afer and Strimmer (2005a,b,c) and Opgen-Rhein and Strimmer (2006, 2007) for learning large-scale gene association networks (including assignment of putative directions).

Author(s)

Juliane Sch"afer, Rainer Opgen-Rhein, and Korbinian Strimmer (http://strimmerlab.org/)

References

See website: http://strimmerlab.org/software/GeneNet/

See Also

ggm.estimate.pcor, network.test.edges, extract.network, network.make.dot.

Time Series Expression Data for 800 Arabidopsis Thaliana Genes

Description

This data set describes the temporal expression of 800 genes of A. thaliana during the diurnal cycle. The 800 genes are a subset of the data presented in Smith et al. (2004) and were selected for periodicity according to the method implemented in the R package GeneCycle (http://cran.r-project.org/web/packages/GeneCycle/).

Usage
data(arth800)
Format

arth800.expr is a longitudinal object with repetitions, and contains the log2 transformed expression data.
arth800.mexpr is a longitudinal object, and contains the mean expression levels of arth800.expr.
arth800.descr, arth800.name, arth800.probe, arth800.symbol are vectors containing additional information about each gene.

Source

The microarray experiments were performed in the laboratory of S. Smith (Edinburgh). The data are available from the NASCArrays database (http://affymetrix.arabidopsis.info/ under experiment reference number NASCARRAYS-60.

References


Examples

# load GeneNet library
library("GeneNet")

# load data set
data(arth800)
is.longitudinal(arth800.expr)
summary(arth800.expr)

# plot first nine time series
plot(arth800.expr, 1:9)

---

cor0.test  
Test of Vanishing (Partial) Correlation

description

cor0.test computes a p-value for the two-sided test with the null hypothesis H0: rho == 0 versus the alternative hypothesis HA: rho != 0.

If method="student" is selected then the statistic t=r*sqrt((kappa-1)/(1-r*r)) is considered which under H0 is student-t distributed with df=kappa-1. This method is exact.

If method="dcor0" is selected then the p-value is computed directly from the distribution function pcor0. This method is also exact.

If method="ztransform" is selected then the p-value is computed using the z-transform (see z.transform), i.e. using a suitable chosen normal distribution. This method returns approximate p-values.
Usage

cor0.test(r, kappa, method=c("student", "dcor0", "ztransform"))

Arguments

r  observed correlation
kappa  degree of freedom of the null-distribution
method  method used to compute the p-value

Value

A p-value.

Author(s)


See Also

dcor0, kappa2n, z.transform.

Examples

# load GeneNet library
library("GeneNet")

# covariance matrix
m.cov <- rbind(
c(3,1,1,0),
c(1,3,0,1),
c(1,0,2,0),
c(0,1,0,2)
)

# compute partial correlations
m.pcor <- cor2pcor(m.cov)
m.pcor

# corresponding p-values
# assuming a sample size of 25, i.e. kappa=22
kappa2n(22, 4)
cor0.test(m.pcor, kappa=22)
cor0.test(m.pcor, kappa=22) < 0.05

# p-values become smaller with larger r
cor0.test(0.7, 12)
cor0.test(0.8, 12)
cor0.test(0.9, 12)

# comparison of various methods
cor0.test(0.2, 45, method="student")
Microarray Time Series Data for 102 E. Coli Genes

Description
This data set describes the temporal expression of 102 genes of *E. Coli* after induction of the expression of SOD (recombinant human superoxide dismutase).

Usage
data(ecoli)

Format
caulobacter is a longitudinal object containing the data from the Schmidt-Heck et al. (2004) experiment. Essentially, this is a matrix with 102 columns (=genes) and 9 rows (=time points). All expression levels are given in log2-ratios with respect to the first time point (i.e. the induction at time 0).

Source
The microarray experiment was performed at the Institute of Applied Microbiology, University of Agricultural Sciences of Vienne. The data and the experiment is described in Schmidt-Heck et al. (2004).

References

Examples

```r
# load GeneNet library
library("GeneNet")

# load data set
data(ecoli)
is.longitudinal(ecoli)

# how many samples and how many genes?
dim(ecoli)
summary(ecoli)
get.time.repeats(ecoli)
```
# plot first nine time series
plot(ecoli, 1:9)

---

**ggm.estimate.pcor**  
*Graphical Gaussian Models: Small Sample Estimation of Partial Correlation*

**Description**

`ggm.estimate.pcor` offers an interface to two related shrinkage estimators of partial correlation. Both are fast, statistically efficient, and can be used for analyzing small sample data.

The default method "statics" employs the function `pcor.shrink` whereas the "dynamic" method relies on `dyn.pcor`. The difference between the two estimators is that the latter takes the spacings between time points into account if the input are multiple time course data (these must be provided as `longitudinal` object).

**Usage**

```r
ggm.estimate.pcor(x, method = c("static", "dynamic"), ...)
```

**Arguments**

- `x` data matrix (each rows corresponds to one multivariate observation)
- `method` method used to estimate the partial correlation matrix. Available options are "static" (the default) and "dynamic" - both are shrinkage methods.
- `...` options passed to `pcor.shrink` and to `dyn.pcor`.

**Details**

For details of the shrinkage estimators we refer to Opgen-Rhein and Strimmer (2006a,b) and Sch"afer and Strimmer (2005), as well as to the manual pages of `pcor.shrink` and `dyn.pcor`.

Previously, this function offered several furthers options. The old option called "shrinkage" corresponds to the present "static" option. The other old options "observed.pcor", "partial.bagged.cor", and "bagged.pcor" are now considered obsolete and have been removed.

**Value**

An estimated partial correlation matrix.

**Author(s)**

Rainer Opgen-Rhein, Juliane Sch"afer, and Korbinian Strimmer ([http://strimmerlab.org](http://strimmerlab.org)).
References


See Also

ggm.simulate.data, ggm.estimate.pcor, pcor.shrink, and dyn.pcor.

Examples

## Not run:

# load GeneNet library
library("GeneNet")

# generate random network with 40 nodes
# it contains 780=40*39/2 edges of which 5 percent (=39) are non-zero
true.pcor <- ggm.simulate.pcor(40)

# simulate data set with 40 observations
m.sim <- ggm.simulate.data(40, true.pcor)

# simple estimate of partial correlations
estimated.pcor <- cor2pcor( cor(m.sim) )

# comparison of estimated and true values
sum((true.pcor-estimated.pcor)^2)

# a slightly better estimate ...
estimated.pcor.2 <- ggm.estimate.pcor(m.sim)
sum((true.pcor-estimated.pcor.2)^2)

## End(Not run)
Description

`ggm.make.igraph` converts an edge list as obtained by `ggm.test.edges` into an igraph object. `network.make.igraph` is just an alias to `network.make.igraph`. `ggm.make.dot` converts an edge list as obtained by `ggm.test.edges` into a "dot" file that can directly be used for plotting the network with graphviz. `network.make.dot` is just an alias to `ggm.make.dot`.

Usage

```
ggm.make.igraph(edge.list, node.labels, show.edge.labels=FALSE)
network.make.igraph(edge.list, node.labels, show.edge.labels=FALSE)
ggm.make.dot(filename, edge.list, node.labels, main=NULL, show.edge.labels=FALSE)
network.make.dot(filename, edge.list, node.labels, main=NULL, show.edge.labels=FALSE)
```

Arguments

- `filename` name of file containing the "dot" commands for graphviz
- `edge.list` a data frame, as obtained by `ggm.test.edges`, listing all edges to be included in the graph
- `node.labels` a vector with labels for each node (will be converted to type character)
- `main` title included in plot
- `show.edge.labels` show partial correlation as edge labels (default: FALSE)
- `...` options passed to `plot` functions

Details


`ggm.make.dot` and `network.make.dot` produce 'dot' files for use with the graphviz software - see [http://www.graphviz.org](http://www.graphviz.org).

In the resulting plots, dotted lines indicate negative partial correlation. The strength of the partial correlation is visualized by the line width and the color of the edge: the strongest 20 percent of all edges are shown with thick black lines, whereas the weakest 20 percent are shown in thin grey lines.

Value

- `ggm.make.dot` produces a "dot" network description file that can directly be fed into the 'graphviz' in order to produce a plot of a network.
- `ggm.make.igraph` returns a graph object, suitable for plotting with functions from the `igraph` R package.

Author(s)

Korbinian Strimmer ([http://strimmerlab.org](http://strimmerlab.org)).
ggm.simulate.data

Graphical Gaussian Models: Simulation of Data

Description

`ggm.simulate.data` takes a positive definite partial correlation matrix and generates an i.i.d. sample from the corresponding standard multinormal distribution (with mean 0 and variance 1). If the input matrix `pcor` is not positive definite an error is thrown.

See Also

`ggm.test.edges`, `plot.igraph`.

Examples

```r
# load GeneNet library
library("GeneNet")

# generate random network with 20 nodes and 10 percent edges (=19 edges)
true.pcor <- ggm.simulate.pcor(20, 0.1)

# convert to edge list
dge.list <- ggm.list.edges(true.pcor)[1:19,]

nlab <- LETTERS[1:20] # node labels

######## use igraph R package produce a plot ##########
igr1 <- ggm.make.igraph(dge.list, nlab)
plot(igr1, main = "A Random Graph")

igr2 <- ggm.make.igraph(dge.list, nlab, show.edge.labels=TRUE)
plot(igr2, main = "A Random Graph with Partial Correlations")

# igraph allows to fine-tune the plot
# e.g. smaller edge labels and red nodes:
plot(igr2, main = "A Random Graph with Partial Correlations",
edge.label.cex=0.7, vertex.color="red")

######## use graphviz to produce a plot ##########
# uncomment for actual use!
# nlab <- LETTERS[1:20]
# ggm.make.dot(filename="randomnet.dot", edge.list, nlab, main = "A graph")
# system("fdp -T svg -o randomnet.svg randomnet.dot") # SVG format
# system("fdp -T jpg -o randomnet.jpg randomnet.dot") # JPG format
```
Usage

```r
ggm.simulate.data(sample.size, pcor)
```

Arguments

- `sample.size`: sample size of simulated data set
- `pcor`: partial correlation matrix

Value

A multinormal data matrix.

Author(s)


References


See Also

- `ggm.simulate.pcor`
- `ggm.estimate.pcor`

Examples

```r
# load GeneNet library
library("GeneNet")

# generate random network with 40 nodes
# it contains 780=40*39/2 edges of which 5 percent (=39) are non-zero
true.pcor <- ggm.simulate.pcor(40)

# simulate data set with 40 observations
m.sim <- ggm.simulate.data(40, true.pcor)

# simple estimate of partial correlations
estimated.pcor <- cor2pcor( cor(m.sim) )

# comparison of estimated and true values
sum((true.pcor-estimated.pcor)^2)

# a slightly better estimate ...
estimated.pcor.2 <- ggm.estimate.pcor(m.sim)
sum((true.pcor-estimated.pcor.2)^2)
```
ggm.simulate.pcor

Description

ggm.simulate.pcor generates a random matrix of partial correlations that corresponds to a GGM network of a given size (num.nodes) with a specified fraction of non-zero edges.

Usage

ggm.simulate.pcor(num.nodes, etaA=/zero.noslash./zero.noslash5)

Arguments

num.nodes number of nodes in the network
etaA fraction of edges with non-zero partial correlation (default: 0.05)

Details

The output of ggm.simulate.pcor is always positive definite. This is ensured by using diagonally dominant matrices when generating the random GGM model. For the full algorithm see Sch\"afer and Strimmer (2005).

Value

A positive definite partial correlation matrix.

Author(s)


References


See Also

ggm.simulate.data, ggm.estimate.pcor.

Examples

## Not run:

# load GeneNet library
library("GeneNet")

# generate random network with 40 nodes
# it contains 780=40*39/2 edges of which 5 percent (=39) are non-zero
true.pcor <- ggm.simulate.pcor(40)

# simulate data set with 40 observations
m.sim <- ggm.simulate.data(40, true.pcor)

# simple estimate of partial correlations
estimated.pcor <- cor2pcor(cor(m.sim))

# comparison of estimated and true values
sum((true.pcor-estimated.pcor)^2)

# a slightly better estimate ...
estimated.pcor.2 <- ggm.estimate.pcor(m.sim)
sum((true.pcor-estimated.pcor.2)^2)

## End(Not run)

---

**ggm.test.edges**

**Graphical Gaussian Models: Assess Significance of Edges (and Directions)**

**Description**

`ggm.test.edges` returns a data frame containing all edges listed in order of the magnitude of the partial correlation associated with each edge. If `fdr=TRUE` then in addition the p-values, q-values and posterior probabilities (=1 - local fdr) for each potential edge are computed.

`network.test.edges` is the same function as `ggm.test.edges`.

`extract.network` returns a data frame with a subset of significant edges.

**Usage**

```r
ggm.test.edges(r.mat, fdr=TRUE, direct=FALSE, plot=TRUE, ...) network.test.edges(r.mat, fdr=TRUE, direct=FALSE, plot=TRUE, ...) extract.network(network.all, method.ggm=c("prob", "qval","number"),
               cutoff.ggm=0.8, method.dir=c("prob","qval","number", "all"),
               cutoff.dir=0.8, verbose=TRUE)
```

**Arguments**

- `r.mat` matrix of partial correlations
- `fdr` estimate q-values and local fdr
- `direct` compute additional statistics for obtaining a partially directed network
- `plot` plot density and distribution function and (local) fdr values
- `...` parameters passed on to `fdrtool`
network.all list with partial correlations and fdr values for all potential edges (i.e. the output of network.test.edges

method.ggm determines which criterion is used to select significant partial correlations (default: prob)

cutoff.ggm default cutoff for significant partial correlations

method.dir determines which criterion is used to select significant directions (default: prob)

cutoff.dir default cutoff for significant directions

verbose print information on the number of significant edges etc.

Details

For assessing the significance of edges in the GGM a mixture model is fitted to the partial correlations using fdrtool. This results in (i) two-sided p-values for the test of non-zero correlation, (ii) corresponding posterior probabilities (= 1-local fdr), as well as (iii) tail area-based q-values. See Sch\"afer and Strimmer (2005) for details.

For determining putatative directions on this GGM log-ratios of standardized partial variances re estimated, and subsequently the corresponding (local) fdr values are computed - see Opgen-Rhein and Strimmer (2007).

Value

ggm.test.edges and network.test.edges return sorted data frame with the following columns:

pcor correlation (from r.mat)

node1 first node connected to edge

node2 second node connected to edge

pval p-value

qval q-value

prob probability that edge is nonzero (= 1-local fdr

log.spvar log ratio of standardized partial variance (determines direction)

pval.dir p-value (directions)

qval.dir q-value (directions)

prob.dir 1-local fdr (directions)

Each row in the data frame corresponds to one edge, and the rows are sorted according the absolute strength of the correlation (from strongest to weakest)

extract.network processes the above data frame containing all potential edges, and returns a data frame with a subset of edges. If applicable, an additional last column (11) contains additional information on the directionality of an edge.

Author(s)

Rainer Opgen-Rhein, Juliane Sch\"afer, Korbinian Strimmer (http://strimmerlab.org).
References


See Also

cor0.test, fdrtool, ggm.estimate.pcor.

Examples

```r
# load GeneNet library
library("GeneNet")

# ecoli data
data(ecoli)

# estimate partial correlation matrix
inferred.pcor <- ggm.estimate.pcor(ecoli)

# p-values, q-values and posterior probabilities for each potential edge
#
test.results <- ggm.test.edges(inferred.pcor)

# show best 20 edges (strongest correlation)
test.results[1:20,]

# extract network containing edges with prob > 0.9 (i.e. local fdr < 0.1)
net <- extract.network(test.results, cutoff.ggm=0.9)
net

# how many are significant based on FDR cutoff Q=0.05 ?
num.significant.1 <- sum(test.results$qval <= 0.05)
test.results[1:num.significant.1,]

# how many are significant based on "local fdr" cutoff (prob > 0.9) ?
num.significant.2 <- sum(test.results$prob > 0.9)
test.results[test.results$prob > 0.9,]

# parameters of the mixture distribution used to compute p-values etc.
c <- fdrtool(sm2vec(inferred.pcor), statistic="correlation")
c$param
```
kappa2n

Relationship Between Sample Size and the Degree of Freedom of Correlation Distribution

Description

The function kappa2n returns the sample size that corresponds to a given degree of freedom kappa, whereas n2kappa converts sample size to the corresponding degree of freedom.

Usage

kappa2n(kappa, p=2)
n2kappa(n, p=2)

Arguments

kappa     degree of freedom
p         number of variables (p=2 corresponds to simple correlation)
n         sample size

Details

The degree of freedom kappa of the sample distribution of the empirical correlation coefficient depends both on the sample size n and the number p of investigated variables, i.e. whether simple or partial correlation coefficients are being considered. For p=2 (simple correlation coefficient) the degree of freedom equals kappa = n-1, whereas for arbitrary p (with p-2 variables eliminated in the partial correlation coefficient) kappa = n-p+1 (see also dcor0).

Value

The sample size n corresponding to a given kappa, or the degree of freedom kappa corresponding to a given p.

Author(s)


See Also
dcor0.

Examples

# load GeneNet library
library("GeneNet")

# sample sizes corresponding to kappa=7
z.transform  

Variance-Stabilizing Transformations of the Correlation Coefficient

Description

`z.transform` implements Fisher’s (1921) first-order and Hotelling’s (1953) second-order transformations to stabilize the distribution of the correlation coefficient. After the transformation the data follows approximately a normal distribution with constant variance (i.e. independent of the mean).

The Fisher transformation is simply $z = \text{atanh}(r)$.

Hotelling’s transformation requires the specification of the degree of freedom $\kappa$ of the underlying distribution. This depends on the sample size $n$ used to compute the sample correlation and whether simple or partial correlation coefficients are considered. If there are $p$ variables, with $p-2$ variables eliminated, the degree of freedom is $\kappa = n-p+1$. (cf. also `dcor`).

Usage

```r
z.transform(r)
hotelling.transform(r, kappa)
```

Arguments

- `r`: vector of sample correlations
- `kappa`: degrees of freedom of the distribution of the correlation coefficient

Value

The vector of transformed sample correlation coefficients.

Author(s)

Korbinian Strimmer (http://strimmerlab.org).

References


z.transform

See Also
dcor0, kappa2n.

Examples

# load GeneNet library
library("GeneNet")

# small example data set
r <- c(-0.26074194, 0.47251437, 0.23957283,-0.02187209,-0.07699437, -0.03809433,-0.06010493, 0.01334491,-0.42383367,-0.25513041)

# transformed data
z1 <- z.transform(r)
z2 <- hotelling.transform(r,7)
z1
z2
Index

*Topic **datasets**
  arth800, 2
  ecoli, 5
*Topic **hplot**
  ggm.make.igraph, 7
*Topic **htest**
  cor0.test, 3
  ggm.estimate.pcor, 6
  ggm.test.edges, 12
*Topic **multivariate**
  GeneNet-package, 2
  ggm.simulate.data, 9
  ggm.simulate.pcor, 11
*Topic **univar**
  kappa2n, 15
  z.transform, 16

arth800, 2
cor0.test, 3, 14
dcor0, 4, 15–17
dyn.pcor, 6, 7
ecoli, 5
extract.network, 2
extract.network (ggm.test.edges), 12
fdrtool, 12–14

GeneNet-package, 2
  ggm.estimate.pcor, 2, 6, 7, 10, 11, 14
  ggm.make.dot (ggm.make.igraph), 7
  ggm.make.igraph, 7
  ggm.simulate.data, 7, 9, 11
  ggm.simulate.pcor, 10, 11
  ggm.test.edges, 8, 9, 12
hotelling.transform (z.transform), 16
kappa2n, 4, 15, 17

longitudinal, 3, 5, 6
n2kappa (kappa2n), 15
network.make.dot, 2
network.make.dot (ggm.make.igraph), 7
network.make.igraph (ggm.make.igraph), 7
network.test.edges, 2
network.test.edges (ggm.test.edges), 12
pcor.shrink, 6, 7
pcor0, 3
z.transform, 3, 4, 16