Package ‘CORREP’

March 27, 2024

Type  Package
Title  Multivariate Correlation Estimator and Statistical Inference Procedures.
Version 1.68.0
Date  2007-03-19
Author Dongxiao Zhu and Youjuan Li
Maintainer Dongxiao Zhu <doz@stowers-institute.org>
Imports e1071, stats
Suggests cluster, MASS
Description Multivariate correlation estimation and statistical inference. See package vignette.
License GPL (>= 2)
biocViews Microarray, Clustering, GraphAndNetwork
git_url https://git.bioconductor.org/packages/CORREP
git_branch RELEASE_3_18
git_last_commit fc35354
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-03-27

R topics documented:

CORREP-package .................................................. 2
cor.balance ....................................................... 2
cor.bootci ........................................................ 4
cor.LRtest ........................................................ 5
cor.LRtest.std ..................................................... 6
cor.unbalance ..................................................... 7
gal_all ............................................................ 9
permutest .......................................................... 12
true.member ....................................................... 13
Description

This package implements multivariate correlation estimator for data with equivalent number of replicates as well as inequivalent number of replicates. For small sample data, this package provides permutation test for test if the true correlation vanishes, and for large sample data, this package provides Likelihood Ratio Test (LRT) to test if the true correlation vanishes. A function to calculate Bootstrap confidence interval was also included in the package.

Details

Package: CORREP
Type: Package
Version: 1.0
Date: 2007-03-19
License: GPL 2.0 or newer

Author(s)

Dongxiao Zhu and Youjuan Li

Maintainer: Dongxiao Zhu <doz@stowers-institute.org>

References


cor.balance

Description

cor.balance estimates correlation matrix from replicated data assuming equal number of replicates. The data must be formatted in the right format (rows correspond to replicates, columns correspond to conditions, see example below) and the variance of each row of the data MUST equal to 1 (see example below).

Usage

cor.balance(x, m, G)
Arguments

- **x**: data matrix, column represents samples (conditions), and row represents variables (genes), see example below for format information.
- **m**: number of replicates for each variable (gene)
- **G**: number of variables (genes)

Details

The multivariate correlation estimator assumes replicated omics data are iid samples from the multivariate normal distribution. It is derived by maximizing the likelihood function. Note that each off-diagonal element in the returned correlation matrix (G by G) is the average of off-diagonals of MLE of correlation matrix of a pair of variables (m by m).

Value

A correlation matrix estimated for G variables (genes)

Author(s)

Dongxiao Zhu and Youjuan Li

References


See Also

`cor.balance`, `cor`

Examples

```r
library("CORREP")
d0 <- NULL
for(l in 1:10)
d0 <- rbind(d0, rnorm(100))
## The simulated data corresponds to the real-world data of 25 genes and 10 conditions, each gene expression profiles was replicated 4 times.
d0<- t(d0)
## This step is to make the standard deviation of each replicate equals to 1
## so that we can model the covariance matrix as correlation matrix.
d0.std <- apply(d0, 1, function(x) x/sd(x))
M <- cor.balance(t(d0.std), m=4, G=25)
```
cor.bootci

**Bootstrap Confidence Interval for Multivariate Correlation**

**Description**

This function calculates Bootstrap confidence interval for multivariate correlation. The procedure is very similar to those used to calculate Bootstrap CI's for other parameters, such as mean and correlation. See manuscript for detail.

**Usage**

```r
cor.bootci(x, y = NULL, m, G, alpha)
```

**Arguments**

- `x`: data matrix, column represents samples (conditions), and row represents variables (genes), see example below for format information
- `y`: optional, used when `x` and `y` are vectors
- `m`: number of replicates
- `G`: number of genes
- `alpha`: significant level

**Details**

See manuscript.

**Value**

- `upperCI`: Upper bound of CI
- `lowerCI`: Lower bound of CI

**Author(s)**

Dongxiao Zhu and Youjuan Li

**References**


**See Also**

- `cor.LRtest`, `cor.LRtest.std`, `cor.test`, `permutest`
Examples

library("CORREP")
d0 <- NULL
## sample size is set to 5
for(l in 1:5)
d0 <- rbind(d0, rnorm(8))
## data must have row variance of 1
d0.std <- apply(d0, 2, function(x) x/sd(x))
M <- cor.balance(t(d0.std), m = 2, G= 4)
pv.bootci <- cor.bootci(t(d0.std), m = 2, G= 4, alpha = 0.05)

---

cor.LRtest

Maximum Likelihood Ratio Test for Multivariate Correlation Estimator (Positive Determinants)

Description

Performs LRT to test if multivariate correlation vanishes. Note this code will return NaN’s if the matrix determinant is negative (see below).

Usage

cor.LRtest(x, m1, m2)

Arguments

- **x**: data matrix, column represents samples (conditions), and row represents variables (genes), see example below for format information
- **m1**: number of replicates for gene X
- **m2**: number of replicates for gene Y

Details

Under the multivariate normal distribution assumption, the column vectors of the data are iid samples. We test the following hypothesis: H0: Z ~ N(mu, Sigma0), H1 ~ N(mu, Sigma1). Let M = Inverse(Sigma0) * Sigma1, the likelihood ratio test statistic $G^2$, is, $n*[\text{trace}(M) - \log(\text{det}(M)) - (m1+m2)]$. Under the H0, $G^2$ follows a chi-square distribution with $2m1*m2$ degree of freedom. In some case, the determinant of M is negative so that the log(det(M)) returns NaN. There are two ways to deal with this problem, one, those M’s whose determinants are negative tend to consist of very small correlations that are biological irrelevant. Therefore, we can simply ignore those gene pairs that the determinants of correlation matrices M’s are negative. Second, we can ‘standardize’ the M to make the determinant positive (implemented in function cor.LRtest.std). In most of cases, we recommend using function cor.LRtest. Use cor.LRtest.std only you know what you are doing.

Value

- **p.value**: The p-value of the LRT
Author(s)

Youjuan Li and Dongxiao Zhu

References


See Also

cor.LRtest.std, cor.test

Examples

library("CORREP")
library("MASS")
Sigma <- matrix(c(1, 0.8, .5,.5, 0.8, 1,
                 0.5,0.5,0.5,0.5,0.5,1),4,4)
dat <- mvrnorm(50, mu=c(0,0,0,0), Sigma)
dat.std <- apply(dat,2,function(x) x/sd(x))
cor.LRtest(t(dat.std), m1=2, m2=2)
Under the multivariate normal distribution assumption, the column vector of the data is iid sample. We test the following hypothesis: $H_0: Z \sim N(\mu_0, \Sigma_0)$, $H_1: Z \sim N(\mu_1, \Sigma_1)$. Let $M = \text{Inverse}(\Sigma_0) \times \Sigma_1$, the likelihood ratio test statistic $G^2$, is, $n[\text{trace}(M) - \log(\det(M)) - (m_1 + m_2)]$. Under the $H_0$, $G^2$ follows a chi-square distribution with $2m_1 \times m_2$ degree of freedom. In some case, the determinant of $M$ is negative so that the log(\det(M)) return NaN. There are two ways to deal with this problem, one, those M’s whose determinant are negative tend to consist of very small correlations that are biological irrelevant. Therefore, we can simply ignore those gene pairs that the determinant correlation matrix $M$ is negative. Second, we can ‘standardize’ the $M$ to make the determinant positive (implemented in function cor.LRtest.std). In most of cases, we recommend using function cor.LRtest. Use cor.LRtest.std only you know what you are doing.

### Value

- **p.value**: The p-value of the LRT

### Author(s)

Youjuan Li and Dongxiao Zhu

### References


### See Also

- cor.LRtest, cor.test

### Examples

```r
library("CORREP")
library("MASS")
Sigma <- matrix(c(1,0.8,.5,.5,0.8,1,.5,.5,.5,.5,.5,.5,0.6,.5,.5,0.6,1),4,4)
dat <- mvrnorm(50, mu=c(0,0,0,0), Sigma)
dat.std <- apply(dat,2,function(x) x/sd(x))
cor.LRtest.std(t(dat.std), m1=2, m2=2)
```

## cor.unbalance

### Description

cor.unbalance estimates correlation from replicated data of unequal number of replicates. different from cor.balance, cor.unbalance takes a pair of variables at a time because of unequal number of replicates. the variance of each row of the data MUST equal to 1 (see example below)
cor.unbalance

Usage

cor.unbalance(x, m1, m2)

Arguments

x  data matrix, column represents samples (conditions), and row represents variables (genes), see example below for format information
m1  number of replicates for one variable (gene)
m2  number of replicates for another variable (gene)

Details

The multivariate correlation estimator assumes replicated omics data are iid samples from the multivariate normal distribution. It is derived by maximizing the likelihood function. Note that the off-diagonal elements in the returned correlation matrix (G by G) is the average of off-diagonals of MLE of correlation matrix of a pair of variables (m1+m2 by m1+m2).

Value

A correlation matrix containing only one distinct correlation coefficient for the pair of variables (genes)

Author(s)

Dongxiao Zhu and Youjuan Li

References


See Also

cor.unbalance, cor

Examples

library("CORREP")
d0 <- NULL
for(l in 1:10)
d0 <- rbind(d0, rnorm(8))
## The simulated data corresponds to the real-world data of 2 genes and 10 conditions, gene expression profiles were replicated 3 and 5 times.
## Note this function can only take calculate correlation matrix between two genes at a time.
d0<- t(d0)
## This step is to make the standard deviation of each replicate equal to 1
## so that we can model the covariance matrix as correlation matrix.
d0.std <- apply(d0, 1, function(x) x/sd(x))
M <- cor.unbalance(t(d0.std), m1=3, m2=5)
**gal_all**

---

**Galactose Metabolism Data**

**Description**

The data is compiled by Mario Medvedovic et al, 2003 based on the original full data reported in Ideker et al, 2001. There are a total of 205 rows (genes), 20 experiments, and 4 repeated measurements in the data. There are 4 classes (which correspond to functional categories). The data contains approximately 8% of missing data. The missing values were filled by applying k-nearest neighbor (k = 12) to impute all the missing values.

**Usage**

```r
data(gal_all)
```

**Format**

A data frame with 205 variables on the following 80 replicated observations.

- `wtRG1` a numeric vector
- `wtRG2` a numeric vector
- `wtRG3` a numeric vector
- `wtRG4` a numeric vector
- `gal1RG1` a numeric vector
- `gal1RG2` a numeric vector
- `gal1RG3` a numeric vector
- `gal1RG4` a numeric vector
- `gal2RG1` a numeric vector
- `gal2RG2` a numeric vector
- `gal2RG3` a numeric vector
- `gal2RG4` a numeric vector
- `gal3RG1` a numeric vector
- `gal3RG2` a numeric vector
- `gal3RG3` a numeric vector
- `gal3RG4` a numeric vector
- `gal4RG1` a numeric vector
- `gal4RG2` a numeric vector
- `gal4RG3` a numeric vector
- `gal4RG4` a numeric vector
- `gal5RG1` a numeric vector
- `gal5RG2` a numeric vector
gal5RG3  a numeric vector
gal5RG4  a numeric vector
gal6RG1  a numeric vector
gal6RG2  a numeric vector
gal6RG3  a numeric vector
gal6RG4  a numeric vector
gal7RG1  a numeric vector
gal7RG2  a numeric vector
gal7RG3  a numeric vector
gal7RG4  a numeric vector
gal10RG1 a numeric vector
gal10RG2 a numeric vector
gal10RG3 a numeric vector
gal10RG4 a numeric vector
gal80RG1 a numeric vector
gal80RG2 a numeric vector
gal80RG3 a numeric vector
gal80RG4 a numeric vector
wtR1   a numeric vector
wtR2   a numeric vector
wtR3   a numeric vector
wtR4   a numeric vector
gal1R1  a numeric vector
gal1R2  a numeric vector
gal1R3  a numeric vector
gal1R4  a numeric vector
gal2R1  a numeric vector
gal2R2  a numeric vector
gal2R3  a numeric vector
gal2R4  a numeric vector
gal3R1  a numeric vector
gal3R2  a numeric vector
gal3R3  a numeric vector
gal3R4  a numeric vector
gal4R1  a numeric vector
gal4R2  a numeric vector
gal4R3  a numeric vector
The 205 genes have been classified into four functional classes based on their GO annotations. In the data examaple provided in the vignette, we assume the four classes as true memberships (external knowledge) and use it to evaluate the performances of different correlation measured based clustering methods.

Source


References


Examples

data(gal_all)
## maybe str(gal_all) ; plot(gal_all) ...
Description

This function calculates p-values of the multivariate correlation estimator by enumerating all permutations. We recommend using Likehood Ratio Test implemented in function cor.LRtest if your data has moderate to large sample size (>5) The procedure is same as those permutation tests for Pearson correlation coefficient or other parameters. Since the approximation of null distribution requires enumerating all permutations. The computational burden increases in $n^2$.

Usage

permutest(x, y=NULL, m, G)

Arguments

x   data matrix, column represents samples (conditions), and row represents variables (genes), see example below for format information
y   optional, used when x and y are vectors
m   number of replicates
G   number of genes

Details

See manuscript.

Value

PV   P-values of permutation tests

Author(s)

Dongxiao Zhu and Youjuan Li

References


See Also

cor.LRtest, cor.LRtest.std, cor.test
true.member

Examples

```r
library("CORREP")
library("e1071")
d0 <- NULL
## sample size is set to 5, it takes about a min to finish
for(l in 1:5)
d0 <- rbind(d0, rnorm(100))
## data must have row variance of 1
d0.std <- apply(d0, 2, function(x) x/sd(x))
M <- cor.balance(t(d0.std), m = 4, G= 25)
M.pv <- permutest(t(d0.std), m = 4, G= 25)
```

true.member | The Pre-defined Class Memberships of 205 Genes According to GO Annotation

Description

The four classes are: Biosynthesis; Energy pathways; Nucleobase and Transport It is used as true clusters (external knowledge) to compare performance of different clustering methods.

Usage

data(true.member)

Source


References


Examples

data(true.member)
## maybe str(true.member) ; plot(true.member) ...
Index

* cluster
  cor.balance, 2
  cor.bootci, 4
  cor.LRtest, 5
  cor.LRtest.std, 6
  cor.unbalance, 7
  CORREP-package, 2
  permutest, 12

* datasets
  gal_all, 9
  true.member, 13

* htest
  cor.balance, 2
  cor.bootci, 4
  cor.LRtest, 5
  cor.LRtest.std, 6
  cor.unbalance, 7
  CORREP-package, 2
  permutest, 12

* models
  cor.balance, 2
  cor.bootci, 4
  cor.LRtest, 5
  cor.LRtest.std, 6
  cor.unbalance, 7
  CORREP-package, 2
  permutest, 12

* multivariate
  cor.balance, 2
  cor.bootci, 4
  cor.LRtest, 5
  cor.LRtest.std, 6
  cor.unbalance, 7
  CORREP-package, 2
  permutest, 12

  cor, 3, 8
  cor.balance, 2, 3, 7
  cor.bootci, 4
  cor.LRtest, 4, 5, 7, 12
  cor.LRtest.std, 4, 6, 6, 12
  cor.test, 4, 6, 7, 12
  cor.unbalance, 7, 7, 8
  CORREP (CORREP-package), 2
  CORREP-package, 2
  gal_all, 9
  permutest, 4, 12
  true.member, 13