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.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**

```r
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: $H_0: Z \sim N(\mu, \Sigma_0), H_1: Z \sim N(\mu, \Sigma_1)$. Let $M = \text{Inverse}(\Sigma_0)^*\Sigma_1$, the likelihood ratio test statistic $G^2$, is, $n^*\text{trace}(M) - \text{log}(\text{det}(M)) - (m1+m2)$. Under the $H_0$, $G^2$ follows a chi-square distribution with $2m1*m2$ degree of freedom.

In some cases, the determinant of $M$ is negative so that the log of (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 if 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**

```r
library("CORREP")
library("MASS")
Sigma <- matrix(c(1, 0.8, .5,.5, 0.8, 1,
                 0.5,0.5,0.5,0.5,1,0.6,0.5,0.5,0.6,1),4,4)
dat <- mvrnorm(50, mu=c(0,0,0,0), Sigma)
dat.std <- apply(dat,2,function(x) x/sd(x))
cor.LRtest(t(dat.std), m1=2, m2=2)
```
cor.LRtest.std

Maximum Likelihood Ratio Test for Multivariate Correlation Estimator

Description
Performs LRT to test if multivariate correlation vanishes. Note this code ‘standardizes’ estimated correlation matrix to make sure its determinant is positive.

Usage
cor.LRtest.std(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 vector of the data is iid sample. 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, $n[\text{trace}(M)-\log(\det(M))-(m1+m2)]$. 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)) 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
cor.balance

Examples

```r
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, 0.5,
                 0.5, 0.5, 0.5, 0.5, 0.6, 0.6), 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.balance

Multivariate Correlation Estimator (Equal Number of Replicates)

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

**cor.bootci**  

**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
## profile 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)

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)

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 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
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
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gal1RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal1RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal1RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal2RG1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal2RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal2RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal2RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal3RG1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal3RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal3RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal3RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal4RG1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal4RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal4RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal4RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal5RG1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal5RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal5RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal5RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal6RG1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal6RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal6RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal6RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal7RG1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal7RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal7RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal7RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal10RG1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal10RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal10RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal10RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal80RG1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal80RG2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal80RG3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal80RG4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>wtR1</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>wtR2</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>wtR3</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>wtR4</td>
<td>a numeric vector</td>
</tr>
<tr>
<td>gal1R1</td>
<td>a numeric vector</td>
</tr>
</tbody>
</table>
Details

The 205 genes have been classified into four functional classes based on their GO annotations. In the data example 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.
permutest

Source


References


Examples

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

permutest

Permutation Test P-value for Multivariate Correlation

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

### 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) ...

## Sample data
```r
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)
```
Index

*Topic* cluster
- cor.balance, 4
- cor.bootci, 5
- cor.LRtest, 1
- cor.LRtest.std, 3
- cor.unbalance, 6
- CORREP-package, 1
- permutest, 10

*Topic* datasets
- gal_all, 7
- true.member, 11

*Topic* htest
- cor.balance, 4
- cor.bootci, 5
- cor.LRtest, 1
- cor.LRtest.std, 3
- cor.unbalance, 6
- CORREP-package, 1
- permutest, 10

*Topic* models
- cor.balance, 4
- cor.bootci, 5
- cor.LRtest, 1
- cor.LRtest.std, 3
- cor.unbalance, 6
- CORREP-package, 1
- permutest, 10

*Topic* multivariate
- cor.balance, 4
- cor.bootci, 5
- cor.LRtest, 1
- cor.LRtest.std, 3
- cor.unbalance, 6
- CORREP-package, 1
- permutest, 10

- cor, 5, 7
- cor.balance, 4, 5, 6
- cor.bootci, 5
- cor.LRtest, 1, 3, 6, 11
- cor.LRtest.std, 2, 3, 6, 11
- cor.test, 2, 3, 6, 11
- cor.unbalance, 6, 6, 7
- CORREP (CORREP-package), 1

12