Package 'acde'

January 19, 2026

Type Package

Title Artificial Components Detection of Differentially Expressed Genes

Version 1.40.0

Date 2015-02-25

Author Juan Pablo Acosta, Liliana Lopez-Kleine

Maintainer Juan Pablo Acosta <jpacostar@unal.edu.co>

Description This package provides a multivariate inferential analysis method for detecting differentially expressed genes in gene expression data. It uses artificial components, close to the data's principal components but with an exact interpretation in terms of differential genetic expression, to identify differentially expressed genes while controlling the false discovery rate (FDR). The methods on this package are described in the vignette or in the article 'Multivariate Method for Inferential Identification of Differentially Expressed Genes in Gene Expression Experiments' by J. P. Acosta, L. Lopez-Kleine and S. Restrepo (2015, pending publication).

License GPL-3

LazyData yes

Depends R(>= 3.3), boot(>= 1.3)

Imports stats, graphics

Suggests BiocGenerics, RUnit

Encoding UTF-8

biocViews DifferentialExpression, TimeCourse, PrincipalComponent, GeneExpression, Microarray, mRNAMicroarray

NeedsCompilation no

git_url https://git.bioconductor.org/packages/acde

git_branch RELEASE_3_22

git_last_commit 95c7bd4

git_last_commit_date 2025-10-29

Repository Bioconductor 3.22

Date/Publication 2026-01-19

Contents

---

acde-package *Artificial Components Detection of Differentially Expressed Genes*

Description

This package provides a multivariate inferential analysis method for detecting differentially expressed genes in gene expression data. It uses artificial components, close to the data's principal components but with an exact interpretation in terms of differential genetic expression, to identify differentially expressed genes while controlling the false discovery rate (FDR). The methods on this package are described in the article *Multivariate Method for Inferential Identification of Differentially Expressed Genes in Gene Expression Experiments* by Acosta (2015).

Details

Package:	acde
Type:	Package
Version:	1.0
Date:	2015-02-25
License:	GLP-3
LazyData:	yes
Depends:	R(>= 3.1), ade4(>= 1.6), boot(>= 1.3)
Encoding:	UTF-8
Built:	R 3.1.2; 2015-05-01; unix

Index:

ac	Artificial Components for Gene Expression Data
acde-package	Artificial Components Detection of Differentially Expressed Genes
bcaFDR	BCa Confidence Upper Bound for the FDR.
fdr	False Discovery Rate Computation
phytophthora	Gene Expression Data for Tomato Plants

	Inoculated with _Phytophthora infestans_
plot.STP	Plot Method for Single Time Point Analysis
plot.TC	Plot Method for Time Course Analysis
print.STP	Print Method for Single Time Point Analysis
print.TC	Print Method for Time Course Analysis
qval	Q-Values Computation
stp	Single Time Point Analysis for Detecting Differentially Expressed Genes
tc	Time Course Analysis for Detecting Differentially Expressed Genes

Author(s)

Juan Pablo Acosta, Liliana Lopez-Kleine

Maintainer: Juan Pablo Acosta <jpacostar@unal.edu.co>

References

Acosta, J. P. (2015) *Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data*. Unpublished MS thesis. Universidad Nacional de Colombia, Bogot\'a.

Examples

```
## Single time point analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
n <- 500; p <- 20; p1 <- 10
des <- c(rep(1, p1), rep(2, (p-p1)))
mu <- as.matrix(rexp(n, rate=1))
Z <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### 5 up regulated genes
Z[1:5,1:p1] <- Z[1:5,1:p1] + 5
### 10 down regulated genes
Z[6:15,(p1+1):p] <- Z[6:15,(p1+1):p] + 4

resSTP <- stp(Z, des)
resSTP
plot(resSTP)

## Time course analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
tPts <- c("h0", "12h", "24h")
n <- 500; p <- 20; p1 <- 10
Z <- vector("list", 3)
des <- vector("list", 3)
for(tp in 1:3){ des[[tp]] <- c(rep(1, p1), rep(2, (p-p1))) }
mu <- as.matrix(rexp(n, rate=1))
### h0 time point (no diff. expr.)
Z[[1]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### h12 time point (diff. expr. begins)
Z[[2]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
#### Up regulated genes
Z[[2]][1:5,1:p1] <- Z[[2]][1:5,1:p1] +
    matrix(runif(5*p1, 1, 3), nrow=5)
#### Down regulated genes
Z[[2]][6:15,(p1+1):p] <- Z[[2]][6:15,(p1+1):p] +
    matrix(runif(10*(p-p1), 1, 2), nrow=10)
```

```
### h24 time point (maximum differential expression)
Z[[3]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
#### 5 up regulated genes
Z[[3]][1:5,1:p1] <- Z[[3]][1:5,1:p1] + 5
#### 10 down regulated genes
Z[[3]][6:15,(p1+1):p] <- Z[[3]][6:15,(p1+1):p] + 4

resTC <- tc(Z, des)
resTC
summary(resTC)
plot(resTC)
```

---

ac *Artificial Components for Gene Expression Data*

Description

Computes the artificial components for gene expression data between two conditions for a single time point.

Usage

```
ac(Z, design)

ac2(Z, design)
```

Arguments

Z	a numeric matrix or data.frame with n rows and p columns representing genes' expression levels. The rows of Z correspond to the genes in the experiment, and the columns correspond to the replicates. Treatment replicates are to the left, control replicates to the right.
design	a vector of length p with 1's for the treatment replicates and 2's for the control replicates $(1, \ldots, 1, 2, \ldots, 2)$.

Details

This function computes the artificial components of Z, based on the specified design vector. First, the function scales Z so that its columns have zero mean and unit variance. Then computation of the artificial components ψ_1 and ψ_2 is performed as $\psi_1 = Z\mathbf{v_1}$, where $\mathbf{v_1} = (1, \ldots, 1)/\sqrt{p}$, and $\psi_2 = Z\mathbf{v_2}$, where $\mathbf{v_2} = (1, \ldots, 1, -1, \ldots, -1)/\sqrt{pp_1(p-p_1)}$. Here, p_1 is the number of treatment replicates, and $\mathbf{v_2}$ has p_1 positive and $p - p_1$ negative entries.

Value

ac returns a matrix with the artificial components ψ_1 and ψ_2 in the columns.

ac2 returns a matrix with the second artificial component ψ_2 in the only column.

Author(s)

Juan Pablo Acosta (<jpacostar@unal.edu.co>).

References

Acosta, J. P. (2015) *Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data*. Unpublished MS thesis. Universidad Nacional de Colombia, Bogot\'a.

Examples

```
## Computes the artificial components for the
## phitophthora infestans data at 60 hai.
psi <- ac(phytophthora[[4]], c(rep(1,8), rep(2,8)))
plot(x=psi[,1], y=psi[,2])
```

---

bcaFDR *BCa Confidence Upper Bound for the FDR*

Description

For internal use in function `stp`. Computes a BCa confidence upper bound for the FDR following *Algorithm 2* in the vignette.

Usage

```
bcaFDR(Z, design, th = NULL, B = 100,
    lambda = 0.5, PER = FALSE, R = 1000,
    gamma = 0.95, Q = NULL, ...)
```

Arguments

`Z`	a matrix or data.frame representing genes' expression levels. The rows of Z correspond to the genes in the experiment, and the columns correspond to the replicates. Treatment replicates are to the left, control replicates to the right.
`design`	a vector of length equal to the number of columns in `Z` with 1's for the treatment replicates and 2's for the control replicates $(1, \ldots, 1, 2, \ldots, 2)$.
`th`	Threshold values for estimating the FDR. If `NULL`, the values from `abs(ac2(Z,design))` are used.
`B`	Number of bootstrap or permutation replications for estimating the FDR at each iteration (as passed from `stp`).
`lambda`	Parameter for the estimation of π_0 and the FDR as passed from `stp` (see Storey, 2002).
`R`	Number of bootstrap replications for the computation of the FDR's BCa confidence upper bound (as passed from `stp`).
`gamma`	Confidence level for the FDR's BCa upper confidence bound (as passed from `stp`).
`PER`	If `FALSE` (default), bootstrap replications are used to estimate the FDR. If `TRUE`, permutation replications are used instead (as passed from `stp`).
`Q`	Estimated FDR as returned in object `\$Q` from `fdr` function (passed from call to `stp`). For internal use.
`...`	additional arguments for parallel computation in `boot` function as passed from `stp` (see `stp` help page for details).

Value

cbound	BCa upper confidence bound for the FDR for each threshold value in th.
warnings	warning messages generated from use of boot.ci function from package boot.

Author(s)

Juan Pablo Acosta (<jpacostar@unal.edu.co>).

References

Acosta, J. P. (2015) *Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data*. Unpublished MS thesis. Universidad Nacional de Colombia, Bogot\'a.

Storey, J. D. (2002) *A direct approach to false discovery rates.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), **64(3):** 479–498.

Efron B. and Tibshirani R. J. (1994) *An Introduction to the Bootstrap.* Chapman & Hall/CRC, 1993.

See Also

stp.

Examples

```
## Single time point analysis for 50 genes with 10 treatment
## replicates and 10 control replicates
n <- 50; p <- 20; p1 <- 10
des <- c(rep(1, p1), rep(2, (p-p1)))
mu <- as.matrix(rexp(n, rate=1))
Z <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### 5 up regulated genes
Z[1:5,1:p1] <- Z[1:5,1:p1] + 5
### 10 down regulated genes
Z[6:15,(p1+1):p] <- Z[6:15,(p1+1):p] + 5

resFdr <- fdr(Z, des)
bca <- bcaFDR(Z, des, Q=resFdr$Q, B=50, R=500)
plot(resFdr$th, resFdr$Q, type="l", col="blue")
lines(resFdr$th, bca$cbound, col="green")
legend(x="topright", legend=c("FDR", "BCa upper bound"),
    lty=c(1,1), col=c("blue", "green"))
## Note: Discontinuities in the BCa upper bound are due to warnings
## generated during computations with function \code{boot.ci}
## from package \code{boot}.
```

---

fdr *False Discovery Rate Computation*

Description

For internal use in functions stp and bcaFDR. Computes steps 2.1 to 2.4 from *Algorithm 1* in the vignette.

Usage

```
fdr(Z, design, th = NULL, B = 100, lambda = 0.5, PER = FALSE, ...)
```

Arguments

Z	a matrix or data.frame representing genes' expression levels. The rows of Z correspond to the genes in the experiment, and the columns correspond to the replicates. Treatment replicates are to the left, control replicates to the right.
design	a vector of length equal to the number of columns in Z with 1's for the treatment replicates and 2's for the control replicates $(1, \ldots, 1, 2, \ldots, 2)$.
th	Threshold values for estimating the FDR. If NULL, the values from abs(ac2(Z,design)) are used.
B	Number of bootstrap or permutation replications for estimating the FDR (as passed from stp and bcaFDR).
lambda	Parameter for the estimation of π_0 and the FDR as passed from stp and bcaFDR (see Storey, 2002).
PER	If FALSE (default), bootstrap replications are used to estimate the FDR. If TRUE, permutation replications are used instead (as passed from stp and bcaFDR).
...	additional arguments for parallel computation in boot function as passed from stp (see stp help page for details).

Value

Q	Estimations of the FDR using each value in th as the threshold.
th	Threshold values used for estimating the FDR.
pi0	Estimation of π_0, the true proportion of non differentially expressed genes in the experiment.
B	Number of bootstrap or permutation replications used for estimating the FDR.
lambda	Parameter used for the estimation of π_0 and the FDR.
call	The matched call.

Author(s)

Juan Pablo Acosta (<jpacostar@unal.edu.co>).

References

Acosta, J. P. (2015) *Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data*. Unpublished MS thesis. Universidad Nacional de Colombia, Bogot\'a.

Storey, J. D. (2002) *A direct approach to false discovery rates.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), **64(3):** 479–498.

See Also

stp.

Examples

```
## Single time point analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
n <- 500; p <- 20; p1 <- 10
des <- c(rep(1, p1), rep(2, (p-p1)))
mu <- as.matrix(rexp(n, rate=1))
Z <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### 5 up regulated genes
Z[1:5,1:p1] <- Z[1:5,1:p1] + 5
### 10 down regulated genes
Z[6:15,(p1+1):p] <- Z[6:15,(p1+1):p] + 4

res <- fdr(Z, des)
plot(res$th, res$Q, type="l", col="blue")
legend(x="topright", legend="FDR", lty=1, col="blue")
```

---

phytophthora *Gene Expression Data for Tomato Plants Inoculated with* Phytophthora infestans

---

Description

Gene expression data for 16 tomato plants (line IL6-2) in field conditions. 8 of these plants were inoculated with *Phytophthora infestans*, and the other 8 were mock-inoculated with sterile water. Leaf tissue samples from each replicate were taken at 12 hours before and 12, 36 and 60 hours after inoculation. We refer to 12 hours before inoculation as the h0 time point. Expression levels were obtained for 13440 genes.

Usage

```
data("phytophthora")
```

Format

A list with four matrices representing expression levels for 13440 genes (rows) in 16 replicates (columns) at time points "h0", "h12", "h36" and "h60". At each time point, the first 8 columns correspond to treatment (inoculated) replicates and the last 8 columns correspond to control (mock-inoculated) replicates. The names of the genes are the names of the rows in each matrix.

Details

For details about experimental conditions, see Restrepo et al. (2005) and Cai et al. (2013).

Source

Tomato Expression Database website (http://ted.bti.cornell.edu/), experiment E022 (Restrepo et al., 2005).

References

Restrepo, S., Cai, G., Fry, W. E. and Smart, C. D. (2005) *Gene expression profiling of infection of tomato by Phytophthora infestans in the field*. Phytopathology, **95(S88)**.

Cai, G., Restrepo, S., Myers, K., Zuluaga, P., Danies, G., Smart, C. and Fry W.E. (2013) *Gene profiling in partially resistant and susceptible near-isogenic tomatoes in response to late blight in the field*. Molecular plant pathology, **14(2):** 171–184.

Examples

```
for(tp in 1:4){
    cat(paste("Time Point:", names(phytophthora[tp]), "\n"))
    print(phytophthora[[tp]][1:10,])
    cat("...\n \n")
}
```

---

plot.STP *Plot Method for Single Time Point Analysis*

Description

a method for the plot generic. It is designed for displaying plots of the estimated FDR and the genes' classification when performing a Single Time Point Analysis for detecting differentially expressed genes in gene expression data.

Usage

```
## S3 method for class 'STP'
plot(x, FDR=TRUE, AC=TRUE, WARNINGS=FALSE, tp=NULL, ...)
```

Arguments

x	an object of class 'STP' as returned by function stp.
FDR	if TRUE, a plot of the estimated FDR is displayed.
AC	if TRUE, a plot of the differentially expressed genes in the artificial components is displayed.
WARNINGS	if TRUE and if a BCa confidence upper bound was computed for obtaining x, the threshold values for which an extreme order statistic was used in the BCa computations are shown (these warnings are produced in calls to boot.ci).
tp	a character string to be added at the end of the plot's title (used for adding time points in plot.TC).
...	further arguments passed to or from other methods.

Author(s)

Juan Pablo Acosta (<jpacostar@unal.edu.co>).

See Also

stp, print.STP.

Examples

```
## Single time point analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
n <- 500; p <- 20; p1 <- 10
des <- c(rep(1, p1), rep(2, (p-p1)))
mu <- as.matrix(rexp(n, rate=1))
Z <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### 5 up regulated genes
Z[1:5,1:p1] <- Z[1:5,1:p1] + 5
### 10 down regulated genes
Z[6:15,(p1+1):p] <- Z[6:15,(p1+1):p] + 4

resSTP <- stp(Z, des)
resSTP
plot(resSTP)
```

---

plot.TC *Plot Method for Time Course Analysis*

Description

a method for the plot generic. It is designed for displaying plots of the estimated FDR and the genes' classification when performing a Time Course Analysis for detecting differentially expressed genes in gene expression data.

Usage

```
## S3 method for class 'TC'
plot(x, iRatios=TRUE, FDR = TRUE, AC = TRUE,
    WARNINGS = FALSE, ...)
```

Arguments

x	if TRUE, a plot of inertia ratios for all the time points is displayed.
iRatios	an object of class 'TC' as returned by function tc.
FDR	if TRUE, a plot of the estimated FDRs are displayed for each time point.
AC	if TRUE, a plot of the differentially expressed genes in the artificial components is displayed for each time point.
WARNINGS	if TRUE and if a BCa confidence upper bound was computed for obtaining x, the threshold values for which an extreme order statistic was used in the BCa computations are shown (these warnings are produced in calls to boot.ci).
...	further arguments passed to or from other methods.

Author(s)

Juan Pablo Acosta (<jpacostar@unal.edu.co>).

See Also

tc, print.TC, summary.TC.

Examples

```
## Time course analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
tPts <- c("h0", "12h", "24h")
n <- 500; p <- 20; p1 <- 10
Z <- vector("list", 3)
des <- vector("list", 3)
for(tp in 1:3){ des[[tp]] <- c(rep(1, p1), rep(2, (p-p1))) }
mu <- as.matrix(rexp(n, rate=1))
### h0 time point (no diff. expr.)
Z[[1]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### h12 time point (diff. expr. begins)
Z[[2]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
#### Up regulated genes
Z[[2]][1:5,1:p1] <- Z[[2]][1:5,1:p1] +
    matrix(runif(5*p1, 1, 3), nrow=5)
#### Down regulated genes
Z[[2]][6:15,(p1+1):p] <- Z[[2]][6:15,(p1+1):p] +
    matrix(runif(10*(p-p1), 1, 2), nrow=10)
### h24 time point (maximum differential expression)
Z[[3]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
#### 5 up regulated genes
Z[[3]][1:5,1:p1] <- Z[[3]][1:5,1:p1] + 5
#### 10 down regulated genes
Z[[3]][6:15,(p1+1):p] <- Z[[3]][6:15,(p1+1):p] + 4

resTC <- tc(Z, des)
resTC
summary(resTC)
plot(resTC)

## Not run:
## Phytophthora Infestans Time Course Analysis (takes time...)
dataPI <- phytophthora
desPI <- vector("list", 4)
for(tp in 1:4){ desPI[[tp]] <- c(rep(1, 8), rep(2, 8)) }
resPI <- tc(dataPI, desPI)
resPI
summary(resPI)
plot(resPI)

## End(Not run)
```

`print.STP` *Print Method for Single Time Point Analysis*

Description

a method for the print generic. It prints relevant results when performing a Single Time Point Analysis for detecting differentially expressed genes in gene expression data.

Usage

```
## S3 method for class 'STP'
print(x, headerSTP = TRUE, ...)
```

Arguments

x	an object of class 'STP' as returned by function stp.
headerSTP	if FALSE, the header is omitted (used for print.TC).
...	further arguments passed to or from other methods.

Details

If the desired FDR level was achieved (i.e. x$astar <= x$alpha), the results are printed for the differentially expressed genes and 10 more rows only. If the desired FDR level was not achieved, only ten rows are displayed.

See Also

stp, plot.STP.

Examples

```
## Single time point analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
n <- 500; p <- 20; p1 <- 10
des <- c(rep(1, p1), rep(2, (p-p1)))
mu <- as.matrix(rexp(n, rate=1))
Z <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### 5 up regulated genes
Z[1:5,1:p1] <- Z[1:5,1:p1] + 5
### 10 down regulated genes
Z[6:15,(p1+1):p] <- Z[6:15,(p1+1):p] + 4

resSTP <- stp(Z, des)
resSTP
plot(resSTP)
```

---

print.TC *Print Method for Time Course Analysis*

Description

methods for the print and summary generics that print relevant results when performing a Time Course Analysis for detecting differentially expressed genes in gene expression data.

Usage

```
## S3 method for class 'TC'
print(x, ...)

## S3 method for class 'TC'
summary(object, ...)
```

Arguments

x	an object of class 'TC' as returned by function tc.
object	an object of class 'TC' as returned by function tc.
...	further arguments passed to or from other methods.

Details

With `print`, at each time point, if the desired FDR level was achieved (i.e. `x$astar <= x$alpha`), the results are printed for the differentially expressed genes and 10 more rows only. If the desired FDR level was not achieved, only ten rows are displayed.

`summary` prints a more concise version of the results.

See Also

`tc`, `plot.TC`.

Examples

```
## Time course analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
tPts <- c("h0", "12h", "24h")
n <- 500; p <- 20; p1 <- 10
Z <- vector("list", 3)
des <- vector("list", 3)
for(tp in 1:3){ des[[tp]] <- c(rep(1, p1), rep(2, (p-p1))) }
mu <- as.matrix(rexp(n, rate=1))
### h0 time point (no diff. expr.)
Z[[1]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### h12 time point (diff. expr. begins)
Z[[2]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
#### Up regulated genes
Z[[2]][1:5,1:p1] <- Z[[2]][1:5,1:p1] +
    matrix(runif(5*p1, 1, 3), nrow=5)
#### Down regulated genes
Z[[2]][6:15,(p1+1):p] <- Z[[2]][6:15,(p1+1):p] +
    matrix(runif(10*(p-p1), 1, 2), nrow=10)
### h24 time point (maximum differential expression)
Z[[3]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
#### 5 up regulated genes
Z[[3]][1:5,1:p1] <- Z[[3]][1:5,1:p1] + 5
#### 10 down regulated genes
Z[[3]][6:15,(p1+1):p] <- Z[[3]][6:15,(p1+1):p] + 4

resTC <- tc(Z, des)
resTC
summary(resTC)
plot(resTC)

## Not run:
## Phytophthora Infestans Time Course Analysis (takes time...)
dataPI <- phytophthora
desPI <- vector("list", 4)
for(tp in 1:4){ desPI[[tp]] <- c(rep(1, 8), rep(2, 8)) }
resPI <- tc(dataPI, desPI)
resPI
summary(resPI)
plot(resPI)

## End(Not run)
```

qval *Q-Values Computation*

Description

For internal use in function stp. Computes the genes' Q-Values in the Single Time Point Analysis according to *Algorithm 3* in the vignette.

Usage

```
qval(Q, psi2)
```

Arguments

Q	vector with the estimated FDRs when the threshold values used are abs(ac2(Z, design)).
psi2	vector with the second artificial component as returned by ac2.

Value

returns a vector with the computed Q-Values for each gene in the experiment.

Author(s)

Juan Pablo Acosta (<jpacostar@unal.edu.co>).

References

Acosta, J. P. (2015) *Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data*. Unpublished MS thesis. Universidad Nacional de Colombia, Bogot\'a.

Storey, J. D. (2002) *A direct approach to false discovery rates.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), **64(3):** 479–498.

See Also

stp, fdr, ac2.

Examples

```
## Single time point analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
n <- 500; p <- 20; p1 <- 10
des <- c(rep(1, p1), rep(2, (p-p1)))
mu <- as.matrix(rexp(n, rate=1))
Z <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### 5 up regulated genes
Z[1:5,1:p1] <- Z[1:5,1:p1] + 5
### 10 down regulated genes
Z[6:15,(p1+1):p] <- Z[6:15,(p1+1):p] + 4

res <- fdr(Z, des)
qValues <- qval(res$Q, ac2(Z, des))
```

```
plot(res$th, res$Q, type="l", col="blue")
lines(res$th, qValues[order(abs(ac2(Z, des)))], col="green")
legend(x="topright", legend=c("FDR", "Q Values"), lty=c(1,1),
    col=c("blue", "green"))
```

---

stp *Single Time Point Analysis for Detecting Differentially Expressed Genes*

Description

Performs the Single Time Point Analysis for detecting differentially expressed genes following Acosta (2015).

Usage

```
stp(Z, design, alpha = 0.05, B = 100, lambda = 0.5,
    th = NULL, PER = FALSE, BCa = FALSE, gamma = 0.95,
    R = 1000, ...)
```

Arguments

Z	a matrix or data.frame representing genes' expression levels. The rows of Z correspond to the genes in the experiment, and the columns correspond to the replicates. Treatment replicates are to the left, control replicates to the right.
design	a vector of length equal to the number of columns in Z with 1's for the treatment replicates and 2's for the control replicates $(1, \ldots, 1, 2, \ldots, 2)$.
alpha	between 0 and 1. Desired level for controlling the false discovery rate (FDR).
B	Number of bootstrap or permutation replications for estimating the FDR.
lambda	Parameter for the estimation of π_0 and of the FDR (see Storey, 2002).
th	Threshold values for estimating the FDR. If NULL, the values from abs(ac2(Z,design)) are used.
PER	If FALSE (default), bootstrap replications are used to estimate the FDR. If TRUE, permutation replications are used instead.
BCa	If TRUE, a BCa confidence upper bound for the FDR is computed (see Efron and Tibshirani, 1994).
gamma	Confidence level for the FDR's BCa confidence upper bound.
R	Number of bootstrap replications for the computation of the FDR's BCa confidence upper bound.
...	additional arguments for parallel computation in boot function (see Details).

Details

For details on the computations performed in this function, see Acosta (2015).

Additional parameters in the '...' argument are used for parallel computation in bootstrap calculations. These are supplied to calls to the boot function in package boot. With this in mind, the use of additional arguments must be restricted to arguments parallel and ncpus from function boot.

Value

stp returns an object of class 'STP', which is a list with components:

dgenes	factor with the classification of each gene in Z. Classes: "up-reg.", "down-reg.", "no-diff.".
tstar	Threshold value used to identify differentially expressed genes.
astar	Achieved FDR level.
Q	Estimations of the FDR using each value in th as threshold.
th	Threshold values used for estimating the FDR.
qvalues	Estimated Q-Values for the genes in the analysis. If argument th!=NULL, these are not computed.
pi0	Estimation of π_0, the true proportion of non differentially expressed genes in the experiment.
B	Number of bootstrap or permutation replications used for estimating the FDR.
lambda	Parameter used for the estimation of π_0 and the FDR.
ac	Artificial components of Z.
gNames	Gene names (by default the row names in Z).
iRatio	Inertia ratio $Var(\psi_2)/\lambda_1$, where λ_1 is the first eigenvalue of Z's Principal Components Analysis.
bca	BCa upper confidence bounds for the FDR using each value in th as the threshold.
gamma	Confidence level used in the computation of the BCa upper bounds.
alpha	Desired FDR level.
call	The matched call.

Warning

If argument BCa=TRUE, computations may take a considerable amount of time.

Author(s)

Juan Pablo Acosta (<jpacostar@unal.edu.co>).

References

Acosta, J. P. (2015) *Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data*. Unpublished MS thesis. Universidad Nacional de Colombia, Bogot\'a.

Storey, J. D. (2002) *A direct approach to false discovery rates.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), **64(3):** 479–498.

Efron B. and Tibshirani R. J. (1994) *An Introduction to the Bootstrap.* Chapman & Hall/CRC, 1993.

See Also

tc for Time Course Analysis; plot.STP, print.STP.

Examples

```
## Single time point analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
n <- 500; p <- 20; p1 <- 10
des <- c(rep(1, p1), rep(2, (p-p1)))
mu <- as.matrix(rexp(n, rate=1))
Z <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### 5 up regulated genes
Z[1:5,1:p1] <- Z[1:5,1:p1] + 5
### 10 down regulated genes
Z[6:15,(p1+1):p] <- Z[6:15,(p1+1):p] + 4

resSTP <- stp(Z, des)
resSTP
plot(resSTP)


## Not run:
## Phytophthora Infestans Single Time Point Analysis (takes time...)
dataPI <- phytophthora[[4]]
desPI <- c(rep(1,8), rep(2,8))
resPI <- stp(dataPI, desPI)
resPI
plot(resPI, tp="60 hai")

## End(Not run)
```

`tc` *Time Course Analysis for Detecting Differentially Expressed Genes*

Description

Performs the Time Course Analysis from Acosta (2015) for detecting differentially expressed genes in time course experiments for gene expression data.

Usage

```
tc(data, designs, tPoints = NULL,
    method = c("active vs complementary", "groups conformation"),
    activeTP = NULL, alpha = 0.05, B = 100, lambda = 0.5,
    PER = FALSE, BCa = FALSE, gamma = 0.95, R = 1000, ...)
```

Arguments

`data`	a list with matrices or data.frames representing genes' expression levels at each time point. The rows of each matrix correspond to the genes in the experiment, and the columns correspond to the replicates. Treatment replicates are to the left, control replicates to the right.
`designs`	a list with a vector for each time point, of length equal to the number of columns in the respective matrix or data.frame in `data`, with 1's for the treatment replicates and 2's for the control replicates.
`tPoints`	a character vector with the names of the timepoints.

method	if "active vs complementary", an analysis following the *active vs complementary time points* approach (Acosta, 2015) is performed. If "groups conformation", an analysis following the *groups conformation through time* approach (Acosta, 2015) is performed. The default is both.
activeTP	numeric. The index of the active timepoint in tPoints for the *active vs complementary time points* approach. If NULL (default), the active time point is selected via inertia ratios.
alpha	between 0 and 1. Desired level for controlling the false discovery rate (FDR).
B	Number of bootstrap or permutation replications for estimating the FDR.
lambda	Parameter for the estimation of π_0 and, further, the estimation of the FDR (see Storey, 2002).
PER	If FALSE (default), bootstrap replications are used to estimate the FDR. If TRUE, permutation replications are used instead.
BCa	If TRUE, a BCa confidence upper bound for the FDR is computed (see Efron and Tibshirani, 1994).
gamma	Confidence level for the FDR's BCa upper confidence bound.
R	Number of bootstrap replications for the computation of the FDR's BCa upper confidence bound.
...	additional arguments for parallel computation in boot function (see Details).

Details

In the *active vs complementary time points* approach, the time point that maximizes the inertia ratio is selected as the *active* time point. Then, a Single Time Point Analysis (stp) is performed on this time point and plots of the behavior throughout the time course of the differentially expressed genes identified in this time point are displayed.

In the *groups conformation through time* approach, a Single Time Point Analysis (stp) is performed at each time point and plots are displayed showing the behaviour of the differential expression process throughout the time course; that is, how many genes are differentially expressed and how strong is the differential expression at each time point.

For details on the computations performed in this function, see Acosta (2015).

Additional parameters in the '. . .' argument are used for parallel computation in bootstrap calculations. These are supplied to calls to the boot function in package boot. With this in mind, the use of additional arguments must be restricted to arguments parallel and ncpus from function boot.

Value

tc returns an object of class 'TC', which is a list with components:

iRatios	inertia ratios for each time point.
gct	results for the *groups conformation through time* approach. A list with an object of class 'STP' for each timepoint.
act	results for the *active vs complementary time points* approach. A list with an object of class 'STP' for each timepoint.
activeTP	the index of the active timepoint in tPoints used in the *active vs complementary time points* approach.
tPoints	a character vector with the names of the timepoints.
call	The matched call.

Warning

If argument BCa=TRUE, computations may take a considerable amount of time.

Author(s)

Juan Pablo Acosta (<jpacostar@unal.edu.co>).

References

Acosta, J. P. (2015) *Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data*. Unpublished MS thesis. Universidad Nacional de Colombia, Bogot\'a.

Storey, J. D. (2002) *A direct approach to false discovery rates.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), **64(3):** 479–498.

Efron B. and Tibshirani R. J. (1994) *An Introduction to the Bootstrap.* Chapman & Hall/CRC, 1993.

See Also

stp for Single Time Point Analysis; plot.TC, print.TC, summary.TC.

Examples

```
## Time course analysis for 500 genes with 10 treatment
## replicates and 10 control replicates
tPts <- c("h0", "12h", "24h")
n <- 500; p <- 20; p1 <- 10
Z <- vector("list", 3)
des <- vector("list", 3)
for(tp in 1:3){ des[[tp]] <- c(rep(1, p1), rep(2, (p-p1))) }
mu <- as.matrix(rexp(n, rate=1))
### h0 time point (no diff. expr.)
Z[[1]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
### h12 time point (diff. expr. begins)
Z[[2]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
#### Up regulated genes
Z[[2]][1:5,1:p1] <- Z[[2]][1:5,1:p1] +
    matrix(runif(5*p1, 1, 3), nrow=5)
#### Down regulated genes
Z[[2]][6:15,(p1+1):p] <- Z[[2]][6:15,(p1+1):p] +
    matrix(runif(10*(p-p1), 1, 2), nrow=10)
### h24 time point (maximum differential expression)
Z[[3]] <- t(apply(mu, 1, function(mui) rnorm(p, mean=mui, sd=1)))
#### 5 up regulated genes
Z[[3]][1:5,1:p1] <- Z[[3]][1:5,1:p1] + 5
#### 10 down regulated genes
Z[[3]][6:15,(p1+1):p] <- Z[[3]][6:15,(p1+1):p] + 4

resTC <- tc(Z, des)
resTC
summary(resTC)
plot(resTC)

## Not run:
## Phytophthora Infestans Time Course Analysis (takes time...)
dataPI <- phytophthora
desPI <- vector("list", 4)
```

```
for(tp in 1:4){ desPI[[tp]] <- c(rep(1, 8), rep(2, 8)) }
resPI <- tc(dataPI, desPI)
resPI
summary(resPI)
plot(resPI)

## End(Not run)
```

Index