Package ‘Mulcom’

September 19, 2024

Type Package

Title Calculates Mulcom test

Version 1.55.0

Date 2011-10-08

Author Claudio Isella

Maintainer Claudio Isella <claudio.isella@ircc.it>

Depends R (>= 2.10), Biobase

Imports graphics, grDevices, stats, methods, fields

Description Identification of differentially expressed genes and false discovery rate (FDR) calculation by Multiple Comparison test.

License GPL-2

LazyLoad yes

biocViews StatisticalMethod, MultipleComparison, Microarray, DifferentialExpression, GeneExpression

NeedsCompilation yes

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

git_branch devel

git_last_commit 80b4bae

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-09-18

Contents

Affy 2
AffyIlmn 2
harmonicMean 3
Illumina 3
Ilmn 4
limmaAffySymbols 4
limmaIlmnSymbols 4
mulCalc 5
mulCAND 5
MULCOM-class 6

mulcomGeneListIlmn 7
MULCOM_P-class 7
mulDELTA 8
mulDiff 8
mulFSG 9
mulIndex 10
mulInt 11
mulMSE 11
mulOpt 12
mulOptPars 13
mulOptPlot 13
mulParOpt 14
mulPerm 15
mulPermC 16
mulScores 17
mulSSE 17
samAffySymbols 18
samIlmnSymbols 18
samOptPars 19

Index **20**

`Affy` *Affy Dataset*

Description

Affy Dataset

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

`AffyIlmn` *cross mapping table*

Description

cross mapping table

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

harmonicMean — *MulCom Harmonic Mean*

Description

Computes harmonic means across groups replicate Should not be called directly

Usage

```
harmonicMean(index)
```

Arguments

index	a numeric vector with the groups labels of the samples. 0 are the control samples. Number must be progressive

Details

harmonicMean calculates harmonic means across groups replicate for the estimation of Mulcom Test

Value

a numeric vector

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

References

<claudio.isella@ircc.it>

Illumina — *Illumina Dataset*

Description

Illumina Dataset

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Ilmn *Ilmn Dataset*

Description

Ilmn Dataset

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

limmaAffySymbols *significant gene list with limma in Affymetrix*

Description

significant gene list with limma in Affymetrix

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

limmaIlmnSymbols *significant gene list with limma in Illumina*

Description

significant gene list with limma in Illumina

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

mulCalc — *MulCom Calculation*

Description

Calculates MulCom test score for given m and t parameters

Usage

```
mulCalc(Mulcom_P, m, t)
```

Arguments

`Mulcom_P`	an object of class MULCOM
`m`	m: a numeric value corresponding to log 2 ratio correction for MulCom Test
`t`	t: a numeric value corresponding to T values for MulCom Test

Details

`mulCalc` Calculate the Mulcom Score with m and t defined by the user

`Mulcom_P:` an object of class MULCOM_P

`m:` a number corresponding to log 2 ratio correction for MulCom Test

`t:` a number corresponding to T values for MulCom Test

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_scores <- mulScores(Affy, Affy$Groups)
mulcom_calc <- mulCalc(mulcom_scores, 0.2, 2)
```

mulCAND — *Identify the Mulcom candidate feature selection*

Description

Identify the Mulcom candidate feature selection by the m and T defined by the user

Usage

```
mulCAND(eset, Mulcom_P, m, t, ese = "T")
```

Arguments

eset	an AffyBatch
Mulcom_P	an object of class MULCOM
m	m: a numeric vector corresponding to log 2 ratio correction
t	t: a numeric vector corresponding to the MulCom T values
ese	True or False

Details

mulCAND Identify the Mulcom candidate feature selection by the m and T defined by the user

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10,2)
mulcom_cand <- mulCAND(Affy, mulcom_perm, 0.2, 2)
```

MULCOM-class *Class MulCom*

Description

This is a class representation MulCom test scores

Objects from the Class

Objects can be created using the function mulScores on ExpressionSet.

Slots

FC: Object of class numeric representing difference between all experimental groups and the reference groups

HM: Object of class numeric representing the harmonic means in all subgroups

MSE_Corrected: Object of class numeric representing the MulCom test estimation of mean square error as described in the formula of the Dunnett's t-test

Author(s)

Claudio Isella

Examples

```
data(benchVign)
mulcom_scores <- mulScores(Affy, Affy$Groups)
```

mulcomGeneListIlmn *significant gene list with limma in Illumina*

Description

significant gene list with limma in Illumina

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

MULCOM_P-class *Class MulCom Permutation*

Description

This is a class representation MulCom test scores permutation

Objects from the Class

Objects can be created using the function mulScores on ExpressionSet.

Slots

FC: Object of class numeric representing delta between all experimental groups and the reference groups

MSE_Corrected: Object of class numeric representing the MulCom test estimation of mean square error as described in the formula of the Dunnett's t-test

FCp: Object of class numeric representing delta between all experimental groups and the reference groups in permutated data

MSE_Correctedp: Object of class numeric representing the MulCom test estimation of mean square error as described in the formula of the Dunnett's t-test in permutated data

Author(s)

Claudio Isella

Examples

```
data(benchVign)
mulcom_scores <- mulScores(Affy, Affy$Groups)
```

mulDELTA *MulCom Delta*

Description

Computes Delta for all the experimental points in the datasets in respect to control Should not be called directly

Usage

```
mulDELTA(vector, index)
```

Arguments

vector	vector: numeric vector with data measurements
index	a numeric vector with the labels of the samples. 0 are the control samples. number must be progressive

Details

mulDELTA An internal function that should not be called directly. It calculates differential expression in the groups defined in the index class vector, in respect to the 0 groups

Value

vector	a numeric vector with data measurements
index	a numeric vector with the labels of the samples. 0 are the control samples. number must be progressive

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_delta <- mulDELTA(exprs(Affy[1,]),Affy$Groups)
```

mulDiff *MulCom Test Differential analysis*

Description

Identify the differentially expressed features for a specific comparison with given m and t value

Usage

```
mulDiff(eset, Mulcom_P, m, t, ind)
```

Arguments

eset	An ExpressionSet object from package Biobase
Mulcom_P	An object of class Mulcom_P
m	the m values for the analysis
t	the t values for the analysis
ind	and index refeing to te comparison, should be numeric

Value

eset	An ExpressionSet object from package Biobase
Mulcom_P	An object of class Mulcom_P
m	the m values for the analysis
t	the t values for the analysis
ind	and index refeing to te comparison, should be numeric

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10, 7)
mulcom_diff <- mulDiff(Affy, mulcom_perm, 0.2, 2)
```

---

mulFSG *MulCom False Significant Genes*

Description

Calculate the False Significant Genes for m and t defined by the user

Usage

```
mulFSG(Mulcom_P, m, t)
```

Arguments

Mulcom_P	an object of class MULCOM
m	m: a numeric value corresponding to log 2 ratio correction for MulCom Test
t	t: a numeric value corresponding to t values for MulCom Test

Details

mulFDR evaluate the False Significant genes on the Mulcom_P object according to specific m and t parameters. For each permutation it is calculated the number of positive genes. An estimation of the false called genes is evaluated with the median for each experimental subgroups

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10, 7)
mulcom_fsg <- mulFSG(mulcom_perm, 0.2, 2)
```

---

mulIndex *Mulcom Index for Monte Carlo Simlation*

Description

Random assebly of the groups indices for Monte Carlo Simulation

Usage

```
mulIndex(index, np, seed)
```

Arguments

index	the vector with the groups of analysis, must be numeric and 0 correspond to the reference.
np	number of permutation in the simulation
seed	seed for permtations

Details

'mulIndex' generates random index for the function mulPerm. it is not directly called by the user.

Value

A matrix with all indices permutations

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_scores <- mulIndex(Affy$Groups, 5, 7)
```

mulInt — *generates a consensus matrix from list of genes*

Description

generates a consensus matrix from list of genes

Usage

```
mulInt(...)
```

Arguments

...	the function requires vector files as imputs

Details

mulCAND generates a consensus matrix from list of genes

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10,2)
mulcom_opt <- mulOpt(mulcom_perm, vm = seq(0,0.5, 0.1), vt = seq(1,3, 0.1))

h1_opt <- mulParOpt(mulcom_perm, mulcom_opt, ind = 1, th = 0.05)
h2_opt <- mulParOpt(mulcom_perm, mulcom_opt, ind = 1, th = 0.05)

int <- mulInt(h1_opt, h2_opt)
```

mulMSE — *MulCom Mean Square Error*

Description

Computes Mean Square Error for all the experimental points in the datasets in respect to control. should not be called directly

Usage

```
mulMSE(vector, index, tmp = vector())
```

Arguments

vector	a numeric vector with data mesurements
index	a numeric vector with the labels of the samples. 0 are the control samples. number must be progressive
tmp	a vector

Details

mulMSE An internal function that should not be called directly. It calculates within group means square error for the values defined in the x vector according to the index class vector

Value

vector	a numeric vector with data measurements
index	a numeric vector with the labels of the samples. 0 are the control samples. number must be progressive
tmp	a vector

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

---

mulOpt *MulCom optimization*

Description

The function systematically performs the calculation of significant genes and corresponding FDR for all the combination of given list of m and t values.

Usage

```
mulOpt(Mulcom_P, vm, vt)
```

Arguments

Mulcom_P	an object of class Mulcom_P
vm	a vector of m values to test
vt	a vector of t values to test

Details

mulOpt The function systematically performs the calculation of significant genes and corresponding FDR for all the combination of given list of m and t values.

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10, 7)
mulcom_opt <- mulOpt(mulcom_perm, seq(0.1, 0.5, 0.1), seq(1, 3, 0.1))
```

---

mulOptPars *MulCom Parameter Optimization*

Description

Function to optimize Mulcom parameter for maximim nuber of genes with a user defined FDR

Usage

```
mulOptPars(opt, ind, ths)
```

Arguments

opt	an MulCom optimization object
ind	index corresponding to the comparison
ths	a threshold for the FDR optimization, default is 0.05

Details

mulOptPars MulCom optimization function to identify best parameters

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10, 7)
#mulcom_opt <- mulOpt(mulcom_perm, seq(0.1, 0.5, 0.1), seq(1, 3, 0.1))
#optThs <- mulOptPars(mulcom_opt, 1, 0.05)
```

---

mulOptPlot *MulCom optimization Plot*

Description

MulCom optimization Plot to identify best configuration paramters

Usage

```
mulOptPlot(M.Opt, ind, th, smooth = "NO")
```

Arguments

M.Opt	an MulCom optimization object
ind	index corresponding to the comparison to plot
th	a threshold for the FDR plot
smooth	indicates whether the FDR plot will show a significant threshold or will be continuous.

Details

mulOptPlot MulCom optimization Plot

Value

a numeric vector

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10,2)
mulcom_opt <- mulOpt(mulcom_perm, vm=seq(0.1, 0.5, 0.1), vt=seq(1, 3,1))
mulOptPlot(mulcom_opt, 1, 0.05)
```

---

mulParOpt *MulCom Parameters Optimization*

Description

MulCom parameter optimization function to identify best combination of t and m providing maximum number of genes at a given FDR

Usage

```
mulParOpt(perm, M.Opt, ind, th, image = "T")
```

Arguments

perm	a object with permutated MulCom Scores
M.Opt	an MulCom optimization object
ind	index corresponding to the comparison to plot
th	a threshold for the FDR plot
image	default = "T", indicates is print the MulCom optimization plot

Details

`mulParOpt` The function mulParOpt is designed to identify the optimal m and t values combination leading to the maximum number of differentially regulated genes satisfying an user define FDR threshold. In case of equal number of genes, the combination of m and t with the lower FDR will be prioritized. In case of both identical number of genes and FDR, the function will chose the highest t. The function optionally will define a graphical output to visually inspect the performance of the test at given m and t parameters for a certain comparison.

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10,2)
mulcom_opt <- mulOpt(mulcom_perm, vm=seq(0.1, 0.5, 0.1), vt=seq(1, 3,1))
mulParOpt(mulcom_perm, mulcom_opt, 1, 0.05)
```

---

mulPerm *MulCom Permutation*

Description

Reiterate MulCom Test on permutated data to perform Montecarlo simulation

Usage

```
mulPerm(eset, index, np, seed, segm = "F")
```

Arguments

`eset`	An an `AffyBatch` object, each row of must correspond to a variable and each column to a sample.
`index`	a numeric vector of length ncol(data) with the labels of the samples. 0 are the reference samples.
`np`	a numeric values indicating the number of permutation to perform. It is set as default to 10
`seed`	set the seed of the permutaton, default is 1
`segm`	a default set to F. This parametheres requires to be setted to avoid segmentation fault of C subroutin in the case of very large datasets.

Details

`mulPerm`

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_perm <- mulPerm(Affy, Affy$Groups, 10,2)
```

---

mulPermC *MulCom Permutation*

Description

R pipe to C function not called directly by user that reiterate MulCom Test on permutated data to perform Monte Carlo simulation

Usage

```
mulPermC(eset, index, means, mse, n, m, nump, ngroups, reference)
```

Arguments

eset	An an AffyBatch object, each row of must correspond to a variable and each column to a sample.
index	a numeric vector of length ncol(data) with the labels of the samples. 0 are the reference samples.
means	entry for the means output.
mse	entry for the mean square errors output
n	number of rows in obext of class eset
m	number of columns
nump	number of permutation to perform
ngroups	a number corresponding to the number of groups in the analysis.
reference	reference for the comparisons. typically it is 0

Details

mulPerm

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
```

mulScores — *MulCom Score Calculation*

Description

Computes the scores for the MulCom test. The function calculates the numerator and the denominator of the test without the parameters m and t

Usage

```
mulScores(eset, index)
```

Arguments

eset	An an AffyBatch object, each row of must correspond to a variable and each column to a sample.
index	a numeric vector of length ncol(data) with the labels of the samples. 0 are the reference samples.

Details

'mulScore' computes the scores for the MulCom test for multiple point profile. The Mulcom test is designed to compare each experimental mean with the control mean and it is derived from the "Dunnett's test". Dunnett's test controls the Experiment-wise Error Rate and is more powerful than tests designed to compare each mean with each other mean. The test is conducted by computing a modified t-test between each experimental group and the control group.

Value

An Object of class MULCOM from Mulcom package

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Examples

```
data(benchVign)
mulcom_scores <- mulScores(Affy, Affy$Groups)
```

mulSSE — *MulCom Sum of Square Error*

Description

Computes sum of square errors for all the experimental points in the datasets Should not be called directly

Usage

```
mulSSE(vec, index)
```

Arguments

vec	a numeric vector with data measurements
index	a numeric vector with the labels of the samples. 0 are the control samples. number should be progressive

Details

mulSSE An internal function that should not be called directly. It calculates sum of square error in the groups defined in the index class vector.

Value

vec	a numeric vector with data measurements
index	a numeric vector with the labels of the samples. 0 are the control samples. number must be progressive

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

---

samAffySymbols *significant gene list with SAM in Affymetrix*

Description

significant gene list with SAM in Affymetrix

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

---

samIlmnSymbols *significant gene list with SAM in Illumina*

Description

significant gene list with SAM in Illumina

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

`samOptPars` *sam Parameter Optimization*

Description

Function to optimize Sam parameter for maximim nuber of genes with a user defined FDR

Usage

```
samOptPars(opt, ths)
```

Arguments

`opt`	an Sam optimization object
`ths`	a threshold for the FDR optimization

Value

a numeric vector

Author(s)

Claudio Isella, <claudio.isella@ircc.it>

Index

∗ **MulCom**
- Affy, 2
- AffyIlmn, 2
- harmonicMean, 3
- Illumina, 3
- Ilmn, 4
- limmaAffySymbols, 4
- limmaIlmnSymbols, 4
- mulCalc, 5
- mulCAND, 5
- mulcomGeneListIlmn, 7
- mulDELTA, 8
- mulDiff, 8
- mulFSG, 9
- mulIndex, 10
- mulInt, 11
- mulMSE, 11
- mulOpt, 12
- mulOptPars, 13
- mulOptPlot, 13
- mulParOpt, 14
- mulPerm, 15
- mulPermC, 16
- mulScores, 17
- mulSSE, 17
- samAffySymbols, 18
- samIlmnSymbols, 18
- samOptPars, 19

∗ **classes**
- MULCOM-class, 6
- MULCOM_P-class, 7

Affy, 2
AffyBatch, *6*, *15–17*
AffyIlmn, 2

class:MULCOM *(*MULCOM-class*)*, 6
class:MULCOM_P *(*MULCOM_P-class*)*, 7

harmonicMean, 3

Illumina, 3
Ilmn, 4

limmaAffySymbols, 4
limmaIlmnSymbols, 4

mulCalc, 5
mulCAND, 5
MULCOM *(*MULCOM-class*)*, 6
MULCOM-class, 6
MULCOM_P *(*MULCOM_P-class*)*, 7
MULCOM_P-class, 7
mulcomGeneListIlmn, 7
mulDELTA, 8
mulDiff, 8
mulFSG, 9
mulIndex, 10
mulInt, 11
mulMSE, 11
mulOpt, 12
mulOptPars, 13
mulOptPlot, 13
mulParOpt, 14
mulPerm, 15
mulPermC, 16
mulScores, *6*, *7*, 17
mulSSE, 17

samAffySymbols, 18
samIlmnSymbols, 18
samOptPars, 19