Package ‘Mulcom’

January 30, 2017

Type Package
Title Calculates Mulcom test
Version 1.24.0
Date 2011-10-08
Author Claudio Isella
Maintainer Claudio Isella <claudio.isella@ircc.it>
Depends R (>= 2.10), fields, Biobase
Imports graphics, grDevices, stats, methods
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

R topics documented:

Affy ......................................................... 2
AffyIlmn ..................................................... 2
harmonicMean ............................................. 3
Ilumina ...................................................... 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
### AffyILmn

<table>
<thead>
<tr>
<th>mulOpt</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>mulOptPars</td>
<td>13</td>
</tr>
<tr>
<td>mulOptPlot</td>
<td>13</td>
</tr>
<tr>
<td>mulParOpt</td>
<td>14</td>
</tr>
<tr>
<td>mulPerm</td>
<td>15</td>
</tr>
<tr>
<td>mulPermC</td>
<td>16</td>
</tr>
<tr>
<td>mulScores</td>
<td>17</td>
</tr>
<tr>
<td>mulSSE</td>
<td>17</td>
</tr>
<tr>
<td>samAffySymbols</td>
<td>18</td>
</tr>
<tr>
<td>samIlmnSymbols</td>
<td>18</td>
</tr>
<tr>
<td>samOptPars</td>
<td>19</td>
</tr>
</tbody>
</table>

---

#### Affy

**Description**

Affy Dataset

**Author(s)**

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

---

#### AffyILmn

**Description**

cross mapping table

**Author(s)**

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

**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

**Description**
Illumina Dataset

**Author(s)**
Claudio Isella, <claudio.isella@ircc.it>
Ilmn Dataset

Description

Ilmn Dataset

Author(s)

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

limmaAffySymbols

Description

significant gene list with limma in Affymetrix

Author(s)

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

limmaIlmnSymbols

Description

significant gene list with limma in Illumina

Author(s)

Claudio Isella, <claudio.isella@ircc.it>
\textit{mulCalc}  

\textit{MulCom Calculation}

\textbf{Description}

Calculates \textit{MulCom} test score for given m and t parameters

\textbf{Usage}

\texttt{mulCalc(Mulcom\_P, m, t)}

\textbf{Arguments}

- \texttt{Mulcom\_P}: an object of class MULCOM
- \texttt{m}: a numeric value corresponding to log 2 ratio correction for MulCom Test
- \texttt{t}: a numeric value corresponding to T values for MulCom Test

\textbf{Details}

\texttt{mulCalc} Calculate the Mulcom Score with m and t defined by the user

\texttt{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

\textbf{Author(s)}

Claudio Isella, \texttt{<claudio.isella@ircc.it>}

\textbf{Examples}

\begin{verbatim}
  data(benchVign)
  mulcom_scores <- mulScores(Affy, Affy$Groups)
  mulcom_calc <- mulCalc(mulcom_scores, 0.2, 2)
\end{verbatim}

\textit{mulCAND}  

\textit{Identify the Mulcom candidate feature selection}

\textbf{Description}

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

\textbf{Usage}

\texttt{mulCAND(eset, Mulcom\_P, m, t, ese = "T")}
MULCOM-class

Arguments

- **eset**: an AffyBatch
- **Mulcom_P**: an object of class MULCOM
- **m**: a numeric vector corresponding to log 2 ratio correction
- **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

```r
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

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

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

**Arguments**

- `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` is 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**

```r
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**

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eset</td>
<td>An ExpressionSet object from package affy</td>
</tr>
<tr>
<td>Mulcom_P</td>
<td>An object of class Mulcom_P</td>
</tr>
<tr>
<td>m</td>
<td>the m values for the analysis</td>
</tr>
<tr>
<td>t</td>
<td>the t values for the analysis</td>
</tr>
<tr>
<td>ind</td>
<td>and index referring to the comparison, should be numeric</td>
</tr>
</tbody>
</table>

Value

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eset</td>
<td>An ExpressionSet object from package affy</td>
</tr>
<tr>
<td>Mulcom_P</td>
<td>An object of class Mulcom_P</td>
</tr>
<tr>
<td>m</td>
<td>the m values for the analysis</td>
</tr>
<tr>
<td>t</td>
<td>the t values for the analysis</td>
</tr>
<tr>
<td>ind</td>
<td>and index referring to the comparison, should be numeric</td>
</tr>
</tbody>
</table>

Author(s)

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

Examples

```r
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

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mulcom_P</td>
<td>an object of class MULCOM</td>
</tr>
<tr>
<td>m</td>
<td>m: a numeric value corresponding to log 2 ratio correction for MulCom Test</td>
</tr>
<tr>
<td>t</td>
<td>t: a numeric value corresponding to t values for MulCom Test</td>
</tr>
</tbody>
</table>

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.
mulIndex

Mulcom Index for Monte Carlo Simulation

Description
Random assembly 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 permutations

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

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 inputs

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 measurements
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>
mulOptPars

mulOptPars  MulCom Parameter Optimization

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))

mulOptPlot  MulCom optimization Plot

Description

MulCom optimization Plot to identify best configuration parameters

Usage

mulOptPlot(M.Opt, ind, th, smooth = "NO")
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 permuted 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 paramethere 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>
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**

```r
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**

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

```r
mulPermC
```
mulScores

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

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

muSSE 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 maximum number of genes with a user defined FDR

**Usage**

```r
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>