Package ‘metaRNASeq’

Type Package
Title Meta-Analysis of RNA-Seq Data
Version 1.0.5
Date 2021-03-11
Author Guillemette Marot [aut, cre], Andrea Rau [aut, cre], Florence Jaffrezic [aut], Samuel Blanck [ctb]
Maintainer Guillemette Marot <guillemette.marot@univ-lille.fr>
Depends R (>= 3.5.0)
Suggests DESeq2 (>= 1.0.17), VennDiagram (>= 1.6.20)
Description Implementation of two p-value combination techniques (inverse normal and Fisher methods). A vignette is provided to explain how to perform a meta-analysis from two independent RNA-seq experiments.
License GPL
LazyLoad yes
biocViews HighThroughputSequencing, RNAseq, DifferentialExpression
NeedsCompilation no
Repository CRAN
Date/Publication 2021-03-15 14:20:02 UTC

R topics documented:

metaRNASeq-package .............................................. 2
adjpval .............................................................. 3
dispFuncs ........................................................... 3
FC ................................................................. 5
fishercomb ......................................................... 5
IDD.IRR ............................................................ 7
invnorm .............................................................. 8
param ............................................................... 10
rawpval ............................................................. 11
sim.function ....................................................... 11

Index 14
Description

Implementation of two p-value combination techniques (inverse normal and Fisher methods). A vignette is provided to explain how to perform a meta-analysis from two independent RNA-seq experiments.

Details

Package: metaRNASeq
Type: Package
Version: 1.0.2
Date: 2015-01-26
License: GPL

Author(s)

Andrea Rau, Guillemette Marot, Florence Jaffrezic
Maintainer: Guillemette Marot <guillemette.marot@inria.fr>

References


See Also

invnorm fishercomb

Examples

#An User's guide with detailed examples can be downloaded in interactive R sessions
if(interactive()){
  vignette("metaRNASeq")
}
adjpval

Simulated adjusted p-values

Description
The adjusted p-values provided here result from the following procedure: 1) simulation of two RNA-seq experiments with four replicates in each condition via the `sim.function`, 2) analysis of differentially expressed tags using the DESeq package.

Usage
```
data(adjpval)
```

Format
List of length 2, where each list is a vector containing the adjusted p-values for 14,456 genes from individual differential analyses (obtained using DESeq v1.8.3) of each of the simulated RNA-seq datasets.

Details
It is possible to reproduce these adjusted p-values using the procedure described in the package vignette.

References

Examples
```
data(adjpval)
## Maybe str(adjpval)
```

dispFuncs

Gamma regression parameters describing the mean-dispersion relationship for two real datasets.

Description
Gamma regression parameters describing the mean-dispersion relationship for each of the two real datasets considered in the associated paper, as estimated using the DESeq package version 1.8.3 (Anders and Huber, 2010).

Usage
```
data(dispFuncs)
```


**Format**

List of length 2, where each list is a vector containing the two estimated coefficients ($\alpha_0$ and $\alpha_1$) for the gamma regression in each study (see details below).

**Details**

The `dispFuncs` object contains the estimated coefficients from the parametric gamma regressions describing the mean-dispersion relationship for the two real datasets considered in the associated paper. The gamma regressions were estimated using the DESeq package version 1.8.3 (Anders and Huber, 2010).

Briefly, after estimating a per-gene mean expression and dispersion values, the DESeq package fits a curve through these estimates. These fitted values correspond to an estimation of the typical relationship between mean expression values $\mu$ and dispersions $\alpha$ within a given dataset. By default, this relationship is estimated using a gamma-family generalized linear model (GLM), where two coefficients $\alpha_0$ and $\alpha_1$ are found to parameterize the fit as $\alpha = \alpha_0 + \alpha_1 / \mu$.

For the first dataset (F078), the estimated mean-dispersion relationship is described using the following gamma-family GLM:

$$\alpha = 0.024 + 14.896 / \mu.$$  

For the second dataset (F088), the estimated mean-dispersion relationship is described using the following gamma-family GLM:

$$\alpha = 0.00557 + 1.54247 / \mu.$$  

These gamma-family GLMs describing the mean-dispersions relationship in each of the two datasets are used in this package to simulate data using dispersion parameters that are as realistic as possible.

**References**


**See Also**

`sim.function`

**Examples**

```r
data(dispFuncs)
```
**Simulated fold changes (FC)**

**Description**

The FC provided here result from the following procedure: 1) simulation of two RNA-seq experiments with four replicates in each condition via the `sim.function`, 2) analysis of differentially expressed tags using the DESeq package.

**Usage**

```r
data(FC)
```

**Format**

List of length 2, where each list is a vector containing the FC for 14,456 genes from individual differential analyses (obtained using DESeq v1.8.3) of each of the simulated RNA-seq datasets.

**Details**

It is possible to reproduce these FC using the procedure described in the package vignette.

**References**


**Examples**

```r
data(FC)
## Maybe str(FC)
```

---

**fishercomb**

*p-value combination using Fisher’s method*

**Description**

Combines one sided p-values using Fisher’s method.

**Usage**

```r
fishercomb(indpval, BHth = 0.05)
```
Arguments

indpval  List of vectors of one sided p-values to be combined.
BHth     Benjamini Hochberg threshold. By default, the False Discovery Rate is controlled at 5%.

Details

The test statistic for each gene \( g \) is defined as

\[
F_g = -2 \sum_{s=1}^{S} \ln(p_{gs})
\]

where \( p_{gs} \) corresponds to the raw \( p \)-value obtained for gene \( g \) in a differential analysis for study \( s \) (assumed to be uniformly distributed under the null hypothesis). Under the null hypothesis, the test statistic \( F_g \) follows a \( \chi^2 \) distribution with \( 2S \) degrees of freedom. Classical procedures for the correction of multiple testing, such as that of Benjamini and Hochberg (1995) may subsequently be applied to the obtained \( p \)-values to control the false discovery rate at a desired rate \( \alpha \).

Value

DEindices  Indices of differentially expressed genes at the chosen Benjamini Hochberg threshold.
TestStatistic  Vector with test statistics for differential expression in the meta-analysis.
rawpval     Vector with raw \( p \)-values for differential expression in the meta-analysis.
adjpval     Vector with adjusted \( p \)-values for differential expression in the meta-analysis.

References


See Also

metaRNASeq

Examples

data(rawpval)
fishcomb <- fishercomb(rawpval, BHth = 0.05)
DE <- ifelse(fishcomb$adjpval<=0.05,1,0)
hist(fishcomb$rawpval,nclass=100)

## A more detailed example is given in the vignette of the package:
## vignette("metaRNASeq")
**Description**

Calculates the gain or the loss of differentially expressed genes due to meta-analysis compared to individual studies.

**Usage**

IDD.IRR(meta_de, ind_de)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta_de</td>
<td>Vector of differentially expressed tags (or indices of these tags) with the meta-analysis</td>
</tr>
<tr>
<td>ind_de</td>
<td>List of vectors storing differentially expressed tags (or indices of these tags) in each individual study</td>
</tr>
</tbody>
</table>

**Value**

- **DE**: Number of Differentially Expressed (DE) genes
- **IDD**: Integrated Driven Discoveries: number of genes that are declared DE in the meta-analysis that were not identified in any of the individual studies alone.
- **Loss**: Number of genes that are declared DE in individual studies but not in meta-analysis.
- **IDR**: Integration-driven Discovery Rate: proportion of genes that are identified as DE in the meta-analysis that were not identified in any of the individual studies alone.
- **IRR**: Integration-driven Revision Rate: percentage of genes that are declared DE in individual studies but not in meta-analysis.

**Author(s)**

Guillemette Marot

**References**

Examples

data(rawpval)
adjpval<-lapply(rawpval, FUN=function(x) p.adjust(x, method="BH"))
ind_smalladjp<-lapply(adjpval, FUN=function(x) which(x <= 0.05))
#indicators corresponding to the inverse normal p-value combination
invnormcomb <- invnorm(rawpval,nrep=c(8,8), BHth = 0.05)
IDD.IRR(invnormcomb$DEindices,ind_smalladjp)
#indicators corresponding to the p-value combination with Fisher's method
fishcomb <- fishercomb(rawpval, BHth = 0.05)
IDD.IRR(fishcomb$DEindices,ind_smalladjp)

invnorm

P-value combination using the inverse normal method

Description

Combines one sided p-values using the inverse normal method.

Usage

invnorm(indpval, nrep, BHth = 0.05)

Arguments

indpval List of vectors of one sided p-values to be combined.
nrep Vector of numbers of replicates used in each study to calculate the previous one-sided p-values.
BHth Benjamini Hochberg threshold. By default, the False Discovery Rate is controlled at 5%.

Details

For each gene \( g \), let

\[ N_g = \sum_{s=1}^{S} \omega_s \Phi^{-1}(1 - p_{gs}), \]

where \( p_{gs} \) corresponds to the raw \( p \)-value obtained for gene \( g \) in a differential analysis for study \( s \) (assumed to be uniformly distributed under the null hypothesis), \( \Phi \) the cumulative distribution function of the standard normal distribution, and \( \omega_s \) a set of weights. We define the weights \( \omega_s \) as in Marot and Mayer (2009):

\[ \omega_s = \sqrt{\frac{\sum_c R_{cs}}{\sum_{c} \sum_{\ell} R_{c\ell}}}, \]

where \( \sum_c R_{cs} \) is the total number of biological replicates in study \( s \). This allows studies with large numbers of biological replicates to be attributed a larger weight than smaller studies.

Under the null hypothesis, the test statistic \( N_g \) follows a \( \text{N}(0,1) \) distribution. A unilateral test on the righthand tail of the distribution may then be performed, and classical procedures for the correction of multiple testing, such as that of Benjamini and Hochberg (1995), may subsequently be applied to the obtained \( p \)-values to control the false discovery rate at a desired level \( \alpha \).
**Value**

- **DEindices**: Indices of differentially expressed genes at the chosen Benjamini Hochberg threshold.
- **TestStatistic**: Vector with test statistics for differential expression in the meta-analysis.
- **rawpval**: Vector with raw p-values for differential expression in the meta-analysis.
- **adjpval**: Vector with adjusted p-values for differential expression in the meta-analysis.

**Note**

This function resembles the function `directpvalcombi` in the metaMA R package; there is, however, one important difference in the calculation of p-values. In particular, for microarray data, it is typically advised to separate under- and over-expressed genes prior to the meta-analysis. In the case of RNA-seq data, differential analyses from individual studies typically make use of negative binomial models and exact tests, which lead to one-sided, rather than two-sided, p-values. As such, we suggest performing a meta-analysis over the full set of genes, followed by an a posteriori check, and if necessary filter, of genes with conflicting results (over vs. under expression) among studies.

**References**


**See Also**

- `metaRNASeq`

**Examples**

```r
data(rawpval)
## 8 replicates simulated in each study
invnormcomb <- invnorm(rawpval,nrep=c(8,8), BHth = 0.05)
DE <- ifelse(invnormcomb$adjpval<=0.05,1,0)
hist(invnormcomb$rawpval,nclass=100)
```

## A more detailed example is given in the vignette of the package:

## vignette("metaRNASeq")
**Mean simulation parameters**

**Description**
Mean simulation parameters obtained from the analysis of a real dataset.

**Usage**
```r
data(param)
```

**Format**
A data frame with 26408 observations on the following 3 variables.

- **mucond1** a numeric vector with mean parameters for condition 1
- **mucond2** a numeric vector with mean parameters for condition 2
- **DE** a logical vector indicating which tags are differentially expressed (value 1)

**Details**
Mean parameters provided in this package are empirical means (obtained after normalization for library size differences) of real data described in the following references.

**Source**
Supplementary material of (Dillies et al., 2013) paper.

**References**


**Examples**
```r
data(param)
```
Simulated p-values

Description

The p-values provided here result from the following procedure: 1) simulation of two RNA-seq experiments with four replicates in each condition via the `sim.function`, 2) analysis of differentially expressed tags using the DESeq package.

Usage

data(rawpval)

Format

List of length 2, where each list is a vector containing the raw p-values for 14,456 genes from individual differential analyses (obtained using DESeq v1.8.3) of each of the simulated RNA-seq datasets.

Details

It is possible to reproduce these p-values using the procedure described in the package vignette.

References


Examples

data(rawpval)

## Maybe str(rawpval)

---

Simulation of multiple RNA-seq experiments

Description

Simulate data arising from multiple independent RNA-seq experiments

Usage

`sim.function(param, dispFuncs, nrep = 4, classes = NULL, inter.sd = 0.3)`
Arguments

`param`  Mean expression levels: `param` must be a data frame containing at least two columns named "mucond1" and "mucond2" and one row per gene.

`dispFuncs`  List of length equal to the number of studies to be simulated, containing the gamma regression parameters describing the mean-dispersion relationship for each one; these are the mean-dispersion functions linking mean and intra-study variability for each independent experiment.

`nrep`  Number of replicates to be simulated in each condition in each study. Ignored if `classes` is filled.

`classes`  List of class memberships, one per study (maximum 20 studies). Each vector or factor of the list can only contain two levels which correspond to the two conditions studied. If NULL, `classes` is built as a list of two vectors with `nrep` labels 1 (for condition 1) and `nrep` labels 2 (for condition 2).

`inter.sd`  Inter-study variability. By default, `inter.sd` is set to 0.3, which corresponds to a moderate inter-study variability in the case where `param` and `dispFuncs` parameters are used to simulate data.

Details

Details about the simulation procedure are given in the following paper:

Value

A matrix with simulated expression levels, one row per gene and one column per replicate. Names of studies are given in the column names of the matrix.

Note

If the `param` data provided in this package are not used to simulate data, one should check that the per-condition means in `param` are reasonable. Note also that for genes to be simulated as non-differentially expressed, the values of "mucond1" and "mucond2" in `param` should be equal.

References


See Also

`metaRNASeq`

Examples

```
## Load simulation parameters
data(param)
data(dispFuncs)

## Simulate data
```
matsim <- sim.function(param = param, dispFuncs = dispFuncs)
sim.conds <- colnames(matsim)
rownames(matsim) <- paste("tag", 1:dim(matsim)[1], sep="")

# extract simulated data from one study
simstudy1 <- extractfromsim(matsim, "study1")
head(simstudy1$study)
simstudy1$pheno
Index

* datasets
  adjpval, 3
  dispFuncs, 3
  FC, 5
  param, 10
  rawpval, 11
* methods
  fishercomb, 5
  IDD.IRR, 7
  invnorm, 8
  sim.function, 11
* models
  fishercomb, 5
  IDD.IRR, 7
  invnorm, 8
* package
  metaRNASeq-package, 2

adjpval, 3

dispFuncs, 3, 12

extractfromsim(sim.function), 11

FC, 5
fishercomb, 2, 5

IDD.IRR, 7
invnorm, 2, 8

metaRNASeq, 6, 9, 12
metaRNASeq (metaRNASeq-package), 2
metaRNASeq-package, 2

param, 10, 12

rawpval, 11

sim.function, 3–5, 11, 11