Package ‘metaSeq’

January 14, 2017

Type  Package
Title  Meta-analysis of RNA-Seq count data in multiple studies
Version  1.14.0
Date  2013-12-2
Author  Koki Tsuyuzaki, Itoshi Nikaido
Maintainer  Koki Tsuyuzaki <k.t.the-answer@hotmail.co.jp>
Depends  R (>= 2.13.0), NOISeq, snow, Rcpp
Description  The probabilities by one-sided NOISeq are combined by Fisher’s method or Stouffer’s method
License  Artistic-2.0
biocViews  RNASeq, DifferentialExpression, Sequencing
NeedsCompilation  no

R topics documented:

metaSeq-package .................................................. 2
Accelerate.NOISeq .............................................. 2
BreastCancer ...................................................... 3
Fisher.test ......................................................... 4
meta.oneside.noiseq ............................................. 5
meta.readData .................................................... 7
other.oneside.pvalues .......................................... 9
pvals .............................................................. 10
Reset.Accelerate.NOISeq ....................................... 10
Result.Meta ........................................................ 11
Stouffer.test ...................................................... 11
StudyA ............................................................ 12
text.busca_unix ................................................. 13
text.busca_win .................................................. 13
text.n.menor_unix ............................................. 13
text.n.menor_win ............................................... 13

Index  14
Description

Meta-analysis for multiple studies’s RNA-Seq count data. In this package, probability of gene differential is calculated by NOISeq, which is customized for one-sided test. One-sided probabilities are integrated by Fisher’s method (without weighting) or Stouffer’s method (with weighting by sample-size). P-values or probabilities calculated by non-NOISeq methods are also applicable by other.oneside.pvalues.

Details

Package: metaSeq
Type: Package
Version: 1.3.2
Date: 2013-12-2
License: Artistic-2.0

Author(s)

Koki Tsuyuzaki, Itoshi Nikaido

Maintainer: Koki Tsuyuzaki <k.t.the-answer@hotmail.co.jp>

References


See Also

readData, noiseq

Accelerate.NOISeq

Accelerating NOISeq

Description

Exporting C++ re-implanted NOISeq to NOISeq namespace. Please use as Accelerate.NOISeq(OS="Unix") or Accelerate.NOISeq(OS="Windows") according to each OS.

Author(s)

Koki Tsuyuzaki, Itoshi Nikaido
**BreastCancer**

**Multiple RNA-Seq count data designed as Breast Cancer cell lines vs Normal cells**

**Description**

A data frame with 23368 rows (genes) with following 18 columns (samples).

All reads were measured by Illumina Genome Analyzer II or IIX, trimmed as 36 base, and mapped to the human reference genome hg19 as single-end. Each experimental design was restricted as Breast cancer cell vs Normal cell. Quality Control was performed by FastQC and samples whose quality scores were at least over 20 were choosed. Counts are quantified by HTSeq.

**Usage**

```r
data(BreastCancer)
```

**Details**

**StudyA**: SRP008746

- A_1: Breast Cancer (HCC1937), SRX099961, SRR350976
- A_2: Breast Cancer (HCC3153), SRX101334, SRR353604_1
- A_3: Breast Cancer (SUM131502), SRX101335, SRR353948_1
- A_4: Normal (MCF10A), SRX099963, SRR353603_1
- A_5: Normal (HCC2337), SRX101336, SRR353602_1

**StudyB**: SRP006726

- B_1: Breast Cancer (HCC1954), SRX061997, SRR201983
- B_2: Normal (HMEC), SRX061998, SRR201986

**StudyC**: SRP005601

- C_1: Breast Cancer (BT20), SRX040501, SRR097786_1
- C_2: Breast Cancer (BT474), SRX040502, SRR097787_1
- C_3: Breast Cancer (MCF7), SRX040504, SRR097789_1
- C_4: Breast Cancer (MDAMB231), SRX040505, SRR097790_1
- C_5: Breast Cancer (MDAMB468), SRX040506, SRR097791_1
- C_6: Breast Cancer (T47D), SRX040507, SRR097792_1
- C_7: Breast Cancer (ZR751), SRX040508, SRR097793_1
- C_8: Normal (MCF10A), SRX040503, SRR097788_1

**StudyD**: ERP000992

- D_1: Breast Cancer (MCF7), ERX030989, ERR053953
- D_2: Breast Cancer (T47D), ERX031000, ERR053958
- D_3: Normal (Ishikawa), ERX030994, ERR053948
Fisher.test

Source

https://trace.ddbj.nig.ac.jp/DRASearch/study?acc=SRP008746
http://trace.ddbj.nig.ac.jp/DRASearch/study?acc=SRP006726
http://trace.ddbj.nig.ac.jp/DRASearch/study?acc=SRP005601
http://trace.ddbj.nig.ac.jp/DRASearch/study?acc=ERP000992

References


See Also

StudyA, pvals.

Examples

data(BreastCancer)

Fisher.test

Fisher’s combined probability method

Description

Fisher’s method combines multiple p-values which are calculated in each study.

Usage

Fisher.test(pvals, na.mode = "notignore")

Arguments

pvals
A matrix coming from meta.oneside.noiseq function or other.oneside.pvalues, which is used for any one-sided p-values or probability.

na.mode
A string indicating how to treat NA in pvals. "notignore" means that genes having at least one NA is regarded as NA, "ignore" means NA is ignored and remaining data is used. By default, na.mode = "notignore".

Author(s)

Koki Tsuyuzaki, Itoshi Nikaido

References

See Also

meta.readData, meta.oneside.noiseq, other.oneside.pvalues

Examples

data(BreastCancer)
library("snow")

# Experimental condition (1: BreastCancer, 0: Normal)
flag1 <- c(1,1,1,0,0, 1,0, 1,1,1,1,1,0, 1,1,0)

# Source of data
flag2 <- c("A","A","A","A", "B","B", "C","C","C","C","C","C", "D","D","D")

# readData function for meta-analysis
cds <- meta.readData(data = BreastCancer, factor = flag1, studies = flag2)

# oneside NOISeq for meta-analysis
# cl <- makeCluster(4, "SOCK")
# result <- meta.oneside.noiseq(cds, k = 0.5, norm = "tmm", replicates = "biological", factor = flag1, conditions = c(1, 0), studies = flag2)
# stopCluster(cl)

# Script above is very time-consuming step. Please use this pre-calculated result instead
data(Result.Meta)
result <- Result.Meta

# Fisher's method (without weighting)
F <- Fisher.test(result)
str(F)

# Stouffer's method (with weighting by sample-size)
S <- Stouffer.test(result)
str(S)

meta.oneside.noiseq

One-sided NOISeq for meta-analysis

Description

NOISeq customized for one-sided test in meta-analysis. Parallel computing is also available by snow package.

Usage

meta.oneside.noiseq(input, k = 0.5, norm = c("rpkm", "uqua", "tmm", "n"), replicates = c("technical", "biological"), factor = NULL, conditions = NULL, pnr = 0.2, nss = 5, v = 0.02, lc = 1, studies = NULL, cl = NULL)

Arguments

input Object of eSet class coming from readData function or other R packages such as DESeq.
k Counts equal to 0 are replaced by k. By default, k = 0.5.
norm Normalization method. It can be one of "rpkm" (default), "uqua" (upper quartile), "tmm" (trimmed mean of M) or "n" (no normalization).
replicates  In this argument, the type of replicates to be used is defined. Technical, biological or none. By default, technical replicates option is chosen. Note that "no" is automatically chosen against the study which has no replicate.

factor  A string indicating the name of factor whose levels are the conditions to be compared.

conditions  A vector containing the two conditions to be compared by the differential expression algorithm (needed when the factor contains more than 2 different conditions).

pnr  Percentage of the total reads used to simulated each sample when no replicates are available. By default, pnr = 0.2.

nss  Number of samples to simulate for each condition (nss>= 2). By default, nss = 5.

v  Variability in the simulated sample total reads. By default, v = 0.02. Sample total reads is computed as a random value from a uniform distribution in the interval \[(pnr-v)*sum(counts), (pnr+v)*sum(counts)\]

lc  Length correction is done by dividing expression by length^lc. By default, lc = 1.

studies  A vector specifying which column in data are measured in common study. Its length must be equal to the number of column in data.

cl  cluster object in snow package.

Author(s)
Koki Tsuyuzaki, Itoshi Nikaido

References

See Also
noiseq

Examples

data(BreastCancer)
library("snow")

# Experimental condition (1: BreastCancer, 0: Normal)
flag1 <- c(1,1,1,0,0, 1,0, 1,1,1,1,1,1,0, 1,1,0)

# Source of data

# readData function for meta-analysis
cds <- meta.readData(data = BreastCancer, factor = flag1, studies = flag2)

# oneside NOISeq for meta-analysis
# cl <- makeCluster(4, "SOCK")
# result <- meta.oneside.noiseq(cds, k = 0.5, norm = "tmm", replicates = "biological", factor = flag1, conditions = c(1, 0), studies = flag2, cl = cl)
# stopCluster(cl)

# Script above is very time-consuming step. Please use this pre-calculated result instead
data(Result.Meta)
result <- Result.Meta

# Fisher's method (without weighting)
F <- Fisher.test(result)
str(F)

# Stouffer's method (with weighting by sample-size)
S <- Stouffer.test(result)
str(S)

---

**meta.readData**

**readData function for meta-analysis**

**Description**

```
readData function in NOISeq package customized for one-sided test in meta-analysis. Parallel
computing is also available by snow package.
```

**Usage**

```
meta.readData(data = NULL, factors = NULL, length = NULL, biotype = NULL, chromosome = NULL, gc = NULL)
```

**Arguments**

- **data**
  - Matrix or data.frame containing the counts (or expression data) for each feature and sample. Features must be in rows and samples must be in columns.

- **factors**
  - A data.frame containing the experimental condition or group for each sample (columns in the data object).

- **length**
  - Optional argument. Vector, matrix or data.frame containing the length of each feature. In case of giving a vector, the names of the vector must be the feature names or ids with the same type of identifier used in data. If a matrix or a data.frame is provided, and it has two columns, it is expected that the feature names or ids are in the first column and the length of the features in the second. If it only has one column containing the length, the rownames of the object must be the feature names or ids.

- **biotype**
  - Optional argument. Vector, matrix or data.frame containing the biological group (biotype) for each feature. In case of giving a vector, the names of the vector must be the feature names or ids with the same type of identifier used in data. If a matrix or a data.frame is provided, and it has two columns, it is expected that the feature names or ids are in the first column and the biotypes of the features in the second. If it only has one column containing the biotypes, the rownames of the object must be the feature names or ids.

- **chromosome**
  - Optional argument. A matrix or data.frame containing the chromosome, start position and end position of each feature. The rownames must be the feature names or ids with the same type of identifier used in data.
Optional argument. Vector, matrix or data.frame containing the GC content of each feature. In case of giving a vector, the names of the vector must be the feature names or ids with the same type of identifier used in data. If a matrix or a data.frame is provided, and it has two columns, it is expected that the feature names or ids are in the first column and the GC content of the features in the second. If it only has one column containing the GC content, the rownames of the object must be the feature names or ids.

A vector specifying which column in data are measured in common study. Its length must be equal to the number of column in data.

Author(s)
Koki Tsuyuzaki, Itoshi Nikaido

References

See Also
readData

Examples
data(BreastCancer)
library("snow")

# Experimental condition (1: BreastCancer, 0: Normal)
flag1 <- c(1,1,1,0,0,1,0,1,1,1,1,1,1,0,1,1,0)

# Source of data
flag2 <- c("A","A","A","A","A","B","B","C","C","C","C","C","D","D","D")

# readData function for meta-analysis
cds <- meta.readData(data = BreastCancer, factor = flag1, studies = flag2)

# oneside NOISeq for meta-analysis
# cl <- makeCluster(4, "SOCK")
# result <- meta.oneside.noiseq(cds, k = 0.5, norm = "tmm", replicates = "biological", factor = flag1, conditions = c(1, 0), studies = flag2, cl = cl)
# stopCluster(cl)

# Script above is very time-consuming step. Please use this pre-calculated result instead
data(Result.Meta)
result <- Result.Meta

# Fisher's method (without weighting)
F <- Fisher.test(result)
str(F)

# Stouffer's method (with weighting by sample-size)
S <- Stouffer.test(result)
str(S)
other.oneside.pvalues  Optional function for non-NOISeq method

Description
Optional function for non-NOISeq method users. P-values or probability in one-sided test in positive and negative differentiation is integrated and converted as a input matrix for Fisher.test or Stouffer.test. Weight in each study can also be introduced.

Usage
other.oneside.pvalues(Upper, Lower, weight = NULL)

Arguments
- Upper: A matrix which means p-values or probability in one-sided test (positive differentiation). Its rows mean "gene" and its columns mean "study" that test was conducted.
- Lower: A matrix which means p-values or probability in one-sided test (negative differentiation). Its rows mean "gene" and its columns mean "study" that test was conducted.
- weight: A vector which means weight in each study. Its length must be equal to the number of column in Upper and Lower.

Author(s)
Koki Tsuyuzaki, Itoshi Nikaido

Examples
```r
## Assume these are one-sided p-value generated by non-NOISeq method (e.g., cufflinks)
upper <- matrix(runif(300), ncol=3, nrow=100)
lower <- 1 - upper
rownames(upper) <- paste0("Gene", 1:100)
rownames(lower) <- paste0("Gene", 1:100)
weight <- c(3,6,8)

# other.oneside.pvalues function return a matrix which can input Fisher.test or Stouffer.test
result <- other.oneside.pvalues(upper, lower, weight)

# Fisher's method (without weighting)
F <- Fisher.test(result)
str(F)

# Stouffer's method (with weighting by sample-size)
S <- Stouffer.test(result)
str(S)
```
pvals

P-values or probability calculated by DESeq, edgeR, baySeq, NOISeq, and DEGseq against StudyA

Description

P-values or probability calculated by DESeq, edgeR, baySeq, NOISeq, and DEGseq against StudyA, which was down-sampled simulation data (1, 1/2, 1/4, 1/8, 1/16, and 1/32).

Usage

data(pvals)

Source

https://trace.ddbj.nig.ac.jp/DRASearch/study?acc=SRP008746

See Also

StudyA, BreastCancer.

Examples

data(pvals)
names(pvals)

Reset.Accelerate.NOISeq

Reset Accelerate.NOISeq function

Description

Resetting the result of Accelerate.NOISeq function and making NOISeq as normal mode. Just type Reset.Accelerate.NOISeq() after running Accelerate.NOISeq().

Author(s)

Koki Tsuyuzaki, Itoshi Nikaido
Result.Meta

Result of meta.oneside.noiseq against Breast Cancer data

Description

A matrix which containing the probability of oneside.noiseq in each study.

Usage

data(Result.Meta)

Source

https://trace.ddbj.nig.ac.jp/DRASearch/study?acc=SRP008746
http://trace.ddbj.nig.ac.jp/DRASearch/study?acc=SRP006726
http://trace.ddbj.nig.ac.jp/DRASearch/study?acc=SRP005601
http://trace.ddbj.nig.ac.jp/DRASearch/study?acc=ERP000992

References


See Also

*BreastCancer, meta.oneside.noiseq.*

Examples

data(Result.Meta)

Stouffer.test

*Stouffer's weighted Z-score method (Inverse normal method)*

Description

Stouffer’s method combines multiple weighted Z-scores which are calculated in each study. Although many weight can be introduced but weighting by sample-size is used in `meta.oneside.noiseq`.

Usage

Stouffer.test(pvals, na.mode = "notignore")
Arguments

pvals  A matrix coming from `meta.oneside.noiseq` function or `other.oneside.pvalues`, which is used for any one-sided p-values or probability.

na.mode  A string indicating how to treat NA in pvals. "notignore" means that genes having at least one NA is regarded as NA. "ignore" means NA is ignored and remaining data is used. By default, na.mode = "notignore".

Author(s)

Koki Tsuyuzaki, Itoshi Nikaido

References


Examples

data(BreastCancer)
library("snow")

# Experimental condition (1: BreastCancer, 0: Normal)
flag1 <- c(1,1,1,0,0,1,0,1,1,1,1,1,0,1,1,0)

# Source of data
flag2 <- c("A","A","A","A","A","B","B","C","C","C","C","C","C","D","D","D")

# readData function for meta-analysis
cds <- meta.readData(data = BreastCancer, factor = flag1, studies = flag2)

# oneside NOISeq for meta-analysis
# cl <- makeCluster(4, "SOCK")
# result <- meta.oneside.noiseq(cds, k = 0.5, norm = "tmm", replicates = "biological", factor = flag1, conditions = c(1, 0), studies = flag2, cl = cl)
# stopCluster(cl)

# Script above is very time-consuming step. Please use this pre-calculated result instead
data(Result.Meta)
result <- Result.Meta

# Fisher's method (without weighting)
F <- Fisher.test(result)
str(F)

# Stouffer's method (with weighting by sample-size)
S <- Stouffer.test(result)
str(S)
Description

Count data of SRP008746 used for simulation study. Original count data (BreastCancer) are down-sampled repeatedly in accordance with distributions of binomial (the probability equals 0.5). 1, 1/2, 1/4, 1/8, 1/16, and 1/32 data are prepared.

Usage
data(StudyA)

Source

https://trace.ddbj.nig.ac.jp/DRASearch/study?acc=SRP008746

See Also

pvals, BreastCancer.

Examples
data(StudyA)

---

Description

This object is imported to namespace of NOISeq in Unix machine

Description

This object is imported to namespace of NOISeq in Windows machine

---

Description

This object is imported to namespace of NOISeq in Unix machine

Description

This object is imported to namespace of NOISeq in Windows machine
Index

∗Topic **datasets**

- BreastCancer, 3
- pvals, 10
- Result.Meta, 11
- StudyA, 12
- text.busca_unix, 13
- text.busca_win, 13
- text.n.menor_unix, 13
- text.n.menor_win, 13

∗Topic **package**

- metaSeq-package, 2
- Accelerate.NOISeq, 2
- BreastCancer, 3, 10, 11, 13
- Fisher.test, 4
- meta.oneside.noiseq, 4, 5, 5, 11, 12
- meta.readData, 5, 7
- metaSeq (metaSeq-package), 2
- metaSeq-package, 2
- noiseq, 2, 6
- other.oneside.pvalues, 4, 5, 9, 12
- pvals, 4, 10, 13
- readData, 2, 7, 8
- Reset.Accelerate.NOISeq, 10
- Result.Meta, 11
- Stouffer.test, 11
- StudyA, 4, 10, 12
- text.busca_unix, 13
- text.busca_win, 13
- text.n.menor_unix, 13
- text.n.menor_win, 13