Package 'BioQC'

April 14, 2017

Type Package

Title Detect tissue heterogeneity in expression profiles with gene sets

Version 1.2.0

Date 2015-11-24

Author Jitao David Zhang <jitao_david.zhang@roche.com>, with inputs from Laura Badi

Maintainer Jitao David Zhang <jitao_david.zhang@roche.com>

Description BioQC performs quality control of high-throughput expression data based on tissue gene signatures

Depends Rcpp, Biobase

Suggests testthat

biocViews GeneExpression,QualityControl,StatisticalMethod

License LGPL (>=2)

NeedsCompilation yes

R topics documented:

`absLog10p` *Absolute base-10 logarithm of p-values*

Description

The function returns the absolute values of base-10 logarithm of p-values.

Usage

```
absLog10p(x)
```

Arguments

x	Numeric vector or matrix

Details

The logarithm transformation of p-values is commonly used to visualize results from statistical tests. Although it may cause misunderstanding and therefore its use is disapproved by some experts, it helps to visualize and interpret results of statistical tests intuitively.

The function transforms p-values with base-10 logarithm, and returns its absolute value. The choice of base 10 is driven by the simplicity of interpreting the results.

Value

Numeric vector or matrix.

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

Examples

```
testp <- runif(1000, 0, 1)
testp.al <- absLog10p(testp)

print(head(testp))
print(head(testp.al))
```

---

entropy *Shannon entropy and related concepts*

Description

These functions calculate Shannon entropy and related concepts, including diversity, specificity, and specialization. They can be used to quantify gene expression profiles.

Usage

```
entropy(vector)
entropyDiversity(mat, norm=FALSE)
entropySpecificity(mat, norm=FALSE)
sampleSpecialization(mat, norm=TRUE)
```

Arguments

vector	A vector of numbers, or characters. Discrete probability of each item is calculated and the Shannon entropy is returned.
mat	A matrix (usually an expression matrix), with genes (features) in rows and samples in columns.
norm	Logical value. If set to TRUE the scores will be normalized between 0 and 1.

Details

Shannon entropy can be used as measures of gene expression specificity, as well as measures of tissue diversity and specialization. See references below.

We use 2 as base for the entropy calculation, because in this base the unit of entropy is *bit*.

Value

`entropy` returns one entropy value. `entropyDiversity` and `sampleSpecialization` returns a vector as long as the column number of the input matrix. `entropySpecificity` returns a vector of the length of the row number of the input matrix, namely the specificity score of genes.

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

References

Martinez and Reyes-Valdes (2008) Defining diversity, specialization, and gene specificity in transcriptomes through information theory. PNAS 105(28):9709–9714

Examples

```
myVec0 <- 1:9
entropy(myVec0) ## log2(9)
myVec1 <- rep(1, 9)
entropy(myVec1)

myMat <- rbind(c(3,4,5),c(6,6,6), c(0,2,4))
entropySpecificity(myMat)
entropySpecificity(myMat, norm=TRUE)
entropyDiversity(myMat)
entropyDiversity(myMat, norm=TRUE)
sampleSpecialization(myMat)
sampleSpecialization(myMat,norm=TRUE)

myRandomMat <- matrix(runif(1000), ncol=20)
entropySpecificity(myRandomMat)
entropySpecificity(myRandomMat, norm=TRUE)
entropyDiversity(myRandomMat)
entropyDiversity(myRandomMat, norm=TRUE)
sampleSpecialization(myRandomMat)
sampleSpecialization(myRandomMat,norm=TRUE)
```

---

filterPmat *Filter rows of p-value matrix under the significance threshold*

Description

Given a p-value matrix and a threshold value, `filterPmat` removes rows where there is no p-values lower than the given threshold.

Usage

```
filterPmat(x, threshold)
```

Arguments

x	A matrix of p-values. It must be raw p-values and should not be transformed (e.g. logarithmic).
threshold	A numeric value, the minimal p-value used to filter rows. If missing, given the values of NA, NULL or number 0, no filtering will be done and the input matrix will be returned.

Value

Matrix of p-values. If no line is left, a empty matrix of the same dimension as input will be returned.

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

Examples

```
set.seed(1235)
testMatrix <- matrix(runif(100,0,1), nrow=10)

## filtering
(testMatrix.filter <- filterPmat(testMatrix, threshold=0.05))
## more strict filtering
(testMatrix.strictfilter <- filterPmat(testMatrix, threshold=0.01))
## no filtering
(testMatrix.nofilter <- filterPmat(testMatrix))
```

---

gini *Calculate Gini Index of a numeric vector*

Description

Calculate the Gini index of a numeric veector

Usage

```
gini(x, na.rm=FALSE)
```

Arguments

x	A numeric vector.
na.rm	Logical. If set to TRUE, NA values are omitted.

Details

The Gini index (Gini coefficient) is a measure of statistical dispersion. A Gini coefficient of zero expresses perfect equality where all values are the same. A Gini coefficient of one expresses maximal inequality among values.

Value

A numeric value between 0 and 1.

Author(s)

Jitao David Zhang

References

Gini. C. (1912) *Variability and Mutability*, C. Cuppini, Bologna 156 pages.

Examples

```
testValues <- runif(100)
gini(testValues)
```

`readGmt` *Read in gene sets from a GMT file*

Description

Read in gene sets from a GMT file

Usage

```
readGmt(filename)
```

Arguments

`filename`	GMT file name

Value

A gene set list, wrapped in a S3-class `gmtlist`. Each list item is a list with three items: gene set name (`name`), gene set description (`desc`), and gene list (a character vector, `genes`).

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

Examples

```
gmt_file <- system.file("extdata/exp.tissuemark.affy.roche.symbols.gmt", package="BioQC")
gmt_list <- readGmt(gmt_file)
```

wmwTest *Wilcoxon-Mann-Whitney rank sum test for high-throughput expression profiling data*

Description

We have implemented an highly efficient Wilcoxon-Mann-Whitney rank sum test for high-throughput expression profiling data. For datasets with more than 100 features (genes), the function performs almost identical to its R implementations (`wilcox.test` in `stats`, or `rankSumTestWithCorrelation` in `limma`) can be more than 1000 times faster.

Usage

```
wmwTest(x, ind.list, alternative = c("greater", "less", "two.sided",
"U","abs.log10.greater","log10.less","abs.log10.two.sided","Q"), simplify = TRUE)
```

Arguments

`x`	A numeric matrix. All other data types (e.g. numeric vectors or `ExpressionSet` objects) are coerced into matrix.
`ind.list`	A list of integer indices (starting from 1) indicating signature genes. Can be of length zero. Other data types (e.g. a list of numeric or logical vectors, or a numeric or logical vector) are coerced into such a list. See `details` below for a special case using GMT files.
`alternative`	The value type to be returned, allowed values include `greater` and `less` (one-sided tests), `two.sided`, and `U` statistic, and their log10 transformation variants. See details below.
`simplify`	Logical. If not, the returning value is in matrix format; if set to `TRUE`, the results are simplified into vectors when possible (default).

Details

The basic application of the function is to test the enrichment of gene sets in expression profiling data or differentially expressed data (the matrix with feature/gene in rows and samples in columns).

A special case is when `x` is an `eSet` object (e.g. `ExpressionSet`), and `ind.list` is a list returned from `readGmt` function. In this case, the only requirement is that one column named `GeneSymbol` in the `featureData` contain gene symbols used in the GMT file. See the example below.

Besides the conventional alternatives such as 'greater', 'less', 'two.sided', and 'U', `wmwTest` (from version 0.99-1) provides further alternatives: `abs.log10.greater` and `log10.less` perform log10 transformation on respective *p*-values and give the transformed value a proper sign (positive for greater than, and negative for less than); `abs.log10.two.sided` transforms two-sided *p*-values to non-negative values; and `Q` score reports absolute log10-transformation of *p*-value of the two-side variant, and gives a proper sign to it, depending on whether it is rather greater than (positive) or less than (negative).

Value

A numeric matrix or vector containing the statistic.

Note

The function has been optimized for expression profiling data. It avoids repetitive ranking of data as done by native R implementations and uses efficient C code to increase the performance and control memory use. Simulation studies using expression profiles of 22000 genes in 2000 samples and 200 gene sets suggested that the C implementation can be >1000 times faster than the R implementation. And it is possible to further accelerate by parallel calling the function with `mclapply` in the `multicore` package.

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

References

Barry, W.T., Nobel, A.B., and Wright, F.A. (2008). A statistical framework for testing functional categories in microarray data. _Annals of Applied Statistics_ 2, 286-315.

Wu, D, and Smyth, GK (2012). Camera: a competitive gene set test accounting for inter-gene correlation. _Nucleic Acids Research_ 40(17):e133

Zar, JH (1999). _Biostatistical Analysis 4th Edition_. Prentice-Hall International, Upper Saddle River, New Jersey.

See Also

`wilcox.test` in the `stats` package, and `rankSumTestWithCorrelation` in the `limma` package.

Examples

```
## R-native data structures
set.seed(1887)
rd <- rnorm(1000)
rl <- sample(c(TRUE, FALSE), 1000, replace=TRUE)
wmwTest(rd, rl, alternative="two.sided")
wmwTest(rd, which(rl), alternative="two.sided")
rd1 <- rd + ifelse(rl, 0.5, 0)
wmwTest(rd1, rl, alternative="greater")
wmwTest(rd1, rl, alternative="U")
rd2 <- rd - ifelse(rl, 0.2, 0)
wmwTest(rd2, rl, alternative="greater")
wmwTest(rd2, rl, alternative="two.sided")
wmwTest(rd2, rl, alternative="less")

## matrix forms
rmat <- matrix(c(rd, rd1, rd2), ncol=3, byrow=FALSE)
wmwTest(rmat, rl, alternative="two.sided")
wmwTest(rmat, rl, alternative="greater")

wmwTest(rmat, which(rl), alternative="two.sided")
wmwTest(rmat, which(rl), alternative="greater")

## other alternatives
wmwTest(rmat, which(rl), alternative="U")
wmwTest(rmat, which(rl), alternative="abs.log10.greater")
wmwTest(rmat, which(rl), alternative="log10.less")
wmwTest(rmat, which(rl), alternative="abs.log10.two.sided")
```

```
wmwTest(rmat, which(rl), alternative="Q")

## using ExpressionSet
data(sample.ExpressionSet)
testSet <- sample.ExpressionSet
fData(testSet)$GeneSymbol <- paste("GENE_",1:nrow(testSet), sep="")
mySig1 <- sample(c(TRUE, FALSE), nrow(testSet), prob=c(0.25, 0.75), replace=TRUE)
wmwTest(testSet, which(mySig1), alternative="greater")

## using integer
exprs(testSet)[,1L] <- exprs(testSet)[,1L] + ifelse(mySig1, 50, 0)
wmwTest(testSet, which(mySig1), alternative="greater")

## using lists
mySig2 <- sample(c(TRUE, FALSE), nrow(testSet), prob=c(0.6, 0.4), replace=TRUE)
wmwTest(testSet, list(first=mySig1, second=mySig2))

## using GMT file
gmt_file <- system.file("extdata/exp.tissuemark.affy.roche.symbols.gmt", package="BioQC")
gmt_list <- readGmt(gmt_file)

gss <- sample(unlist(sapply(gmt_list, function(x) x$genes)), 1000)
eset<-new("ExpressionSet",
          exprs=matrix(rnorm(10000), nrow=1000L),
          phenoData=new("AnnotatedDataFrame", data.frame(Sample=LETTERS[1:10])),
          featureData=new("AnnotatedDataFrame",data.frame(GeneSymbol=gss)))
esetWmwRes <- wmwTest(eset ,gmt_list, alternative="greater")
summary(esetWmwRes)
```

Index