Package 'quantro'

September 5, 2025

Title A test for when to use quantile normalization

Version 1.43.0

Imports Biobase, minfi, doParallel, foreach, iterators, ggplot2, methods, RColorBrewer

Depends R (>= 4.0)

Suggests rmarkdown, knitr, RUnit, BiocGenerics, BiocStyle

VignetteBuilder knitr

RoxygenNote 6.0.1

Encoding UTF-8

Description A data-driven test for the assumptions of quantile normalization using raw data such as objects that inherit eSets (e.g. ExpressionSet, MethylSet). Group level information about each sample (such as Tumor / Normal status) must also be provided because the test assesses if there are global differences in the distributions between the user-defined groups.

License GPL-3

biocViews Normalization, Preprocessing, MultipleComparison, Microarray, Sequencing

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

git_branch devel

git_last_commit 711c398

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-09-04

Author Stephanie Hicks [aut, cre] (ORCID: <https://orcid.org/0000-0002-7858-0231>), Rafael Irizarry [aut] (ORCID: <https://orcid.org/0000-0002-3944-4309>)

Maintainer Stephanie Hicks <shicks19@jhu.edu>

Contents

anova 2
flowSorted 3

matboxplot 3
matdensity 4
MSbetween 5
MSwithin 5
quantro-class 6
quantroPlot 6
quantroPvalPerm 7
quantroStat 8
quantroStatPerm 8
show 9
summary 11

Index **12**

---

anova *Accessors for the 'anova' slot of a quantro object.*

Description

Accessors for the 'anova' slot of a quantro object.

Usage

```
anova(object, ...)

## S4 method for signature 'quantro'
anova(object)

## S4 method for signature 'quantro'
anova(object)
```

Arguments

object	a quantro object
...	other

Examples

```
library(minfi)
data(flowSorted)
pd <- pData(flowSorted)
qtest <- quantro(flowSorted, groupFactor = pd$CellType)
anova(qtest)
```

flowSorted *A subset of FlowSorted.DLPFC.450k data set*

Description

This is the script used to create a subset of the FlowSorted.DLPFC.450k data set from Bioconductor. The purpose is to create an example object for the man pages and vignette in this package.

The object was created using the script in /inst and located in the /data folder.

Format

A MethylSet object with 1e4 rows (probes) and 58 columns (samples).

matboxplot *Box plots of columns in a matrix*

Description

Box plots of the columns of a matrix, but the columns are ordered and colored by a group-level variable

Usage

```
matboxplot(
  object,
  groupFactor,
  las = 3,
  brewer.n = 8,
  brewer.name = "Dark2",
  ...
)
```

Arguments

object	object an object which is inherited from an eSet such as an ExpressionSet or MethylSet object. The object can also be a data frame or matrix with observations (e.g. probes or genes) on the rows and samples as the columns.
groupFactor	a factor variable representing which group each column in object belongs to. It is important that values in groupFactor be in the same order as the columns in object.
las	a numeric in (0, 1, 2, 3) to orient the axis labels. Default is 3 (always vertical).
brewer.n	the number of colors in the palette from the RColorBrewer package. Default is 8.
brewer.name	the name of the palette from the RColorBrewer package. Default is "Dark2".
...	other arguments that can be passed to the codeboxplot function.

Value

A box plot for each column in `object`

Examples

```
library(minfi)
data(flowSorted)

p <- getBeta(flowSorted, offset = 100)
pd <- pData(flowSorted)
matboxplot(object = p, groupFactor = pd$CellType)
```

`matdensity` *Density plots of columns in a matrix*

Description

Plots the density of the columns of a matrix

Usage

```
matdensity(
  object,
  groupFactor = NULL,
  type = "l",
  lty = 1,
  brewer.n = 8,
  brewer.name = "Dark2",
  ...
)
```

Arguments

`object`	object an object which is inherited from an `eSet` such as an `ExpressionSet` or `MethylSet` object. The `object` can also be a data frame or matrix with observations (e.g. probes or genes) on the rows and samples as the columns.
`groupFactor`	an optional factor variable representing which group each column in `object` belongs to. It is important that values in `groupFactor` be in the same order as the columns in `object`.
`type`	the type of lines to plot. Default type is line ("l").
`lty`	the line type. Default is the solid line.
`brewer.n`	the number of colors in the palette from the RColorBrewer package. Default is 8.
`brewer.name`	the name of the palette from the RColorBrewer package. Default is "Dark2".
`...`	other arguments that can be passed to the codematplot function.

Value

A density plot for each column in `object`

Examples

```
library(minfi)
data(flowSorted)
p <- getBeta(flowSorted, offset = 100)
pd <- pData(flowSorted)
matdensity(object = p, groupFactor = pd$CellType, xlab = "beta values",
ylab = "density")
```

MSbetween *Accessors for the 'MSbetween' slot of a quantro object.*

Description

Accessors for the 'MSbetween' slot of a quantro object.

Usage

```
MSbetween(object)

## S4 method for signature 'quantro'
MSbetween(object)

## S4 method for signature 'quantro'
MSbetween(object)
```

Arguments

object	a quantro object

Examples

```
library(minfi)
data(flowSorted)
pd <- pData(flowSorted)
qtest <- quantro(flowSorted, groupFactor = pd$CellType)
MSbetween(qtest)
```

MSwithin *Accessors for the 'MSwithin' slot of a quantro object.*

Description

Accessors for the 'MSwithin' slot of a quantro object.

Usage

```
MSwithin(object)

## S4 method for signature 'quantro'
MSwithin(object)

## S4 method for signature 'quantro'
MSwithin(object)
```

Arguments

object	a quantro object

Examples

```
library(minfi)
data(flowSorted)
pd <- pData(flowSorted)
qtest <- quantro(flowSorted, groupFactor = pd$CellType)
MSwithin(qtest)
```

---

quantro-class *quantro*

Description

This is the S4 class quantro container

---

quantroPlot *Plot results from* quantro *function.*

Description

This function plots the histogram of the null test statistics from permutation test in the quantro function.

Usage

```
quantroPlot(
  object,
  savePlot = FALSE,
  xLab = NULL,
  yLab = NULL,
  mainLab = NULL,
  binWidth = NULL
)
```

Arguments

object	a quantro object from quantro
savePlot	a TRUE/FALSE object argument determining if the plot will be saved for further use or immediately displayed on the screen.
xLab	label for x-axis
yLab	label for y-axis
mainLab	title of plot
binWidth	binwidth or histogram. Default is the stat_bin default.

Value

A histogram will be plotted containing the null test statistics when using boostrapped samples. The red line is the observed test statistic `quantroStat` from `quantro()`.

Examples

```
library(minfi)
data(flowSorted)
p <- getBeta(flowSorted, offset = 100)
pd <- pData(flowSorted)

library(doParallel)
registerDoParallel(cores=4)
qtest <- quantro(p, pd$CellType, B = 100)
quantroPlot(qtest)
```

---

quantroPvalPerm *Accessors for the 'quantroPvalPerm' slot of a quantro object.*

Description

Accessors for the 'quantroPvalPerm' slot of a quantro object.

Usage

```
quantroPvalPerm(object)

## S4 method for signature 'quantro'
quantroPvalPerm(object)

## S4 method for signature 'quantro'
quantroPvalPerm(object)
```

Arguments

`object`	a `quantro` object

Examples

```
library(minfi)
data(flowSorted)
pd <- pData(flowSorted)
qtest <- quantro(flowSorted, groupFactor = pd$CellType)
quantroPvalPerm(qtest)
```

quantroStat *Accessors for the 'quantroStat' slot of a quantro object.*

Description

Accessors for the 'quantroStat' slot of a quantro object.

Usage

```
quantroStat(object)

## S4 method for signature 'quantro'
quantroStat(object)

## S4 method for signature 'quantro'
quantroStat(object)
```

Arguments

object	a quantro object

Examples

```
library(minfi)
data(flowSorted)
pd <- pData(flowSorted)
qtest <- quantro(flowSorted, groupFactor = pd$CellType)
quantroStat(qtest)
```

quantroStatPerm *Accessors for the 'quantroStatPerm' slot of a quantro object.*

Description

Accessors for the 'quantroStatPerm' slot of a quantro object.

Usage

```
quantroStatPerm(object)

## S4 method for signature 'quantro'
quantroStatPerm(object)

## S4 method for signature 'quantro'
quantroStatPerm(object)
```

Arguments

object	a quantro object

Examples

```
library(minfi)
data(flowSorted)
pd <- pData(flowSorted)
qtest <- quantro(flowSorted, groupFactor = pd$CellType)
quantroStatPerm(qtest)
```

---

show *quantro*

Description

This is a function that tests for global differences between groups of distributions which asses whether global normalization methods such as quantile normalization should be applied. This function defines the quantro class and constructor.

Usage

```
quantro(
  object,
  groupFactor = NULL,
  B = 0,
  qRange = NULL,
  useMedianNormalized = TRUE,
  verbose = TRUE
)
```

Arguments

`object`	an object which is inherited from an `eSet` such as an `ExpressionSet` or `MethylSet` object. The `object` can also be a data frame or matrix with observations (e.g. probes or genes) on the rows and samples as the columns.
`groupFactor`	a group level factor associated with each sample or column in the `object`. The order of the `groupFactor` must match the order of the columns in `object`.
`B`	number of permutations to assess statistical significance in a permutation test. Default `B`=0.
`qRange`	the range of quantiles to consider. Default is `seq(0, 1, length.out = nrow(object))`.
`useMedianNormalized`	TRUE/FALSE argument specifying if the median normalized data should be used or not as input to test for global differences between distributions. Default is TRUE.
`verbose`	TRUE/FALSE argument specifying if verbose messages should be returned or not. Default is TRUE.

Details

Quantile normalization is one of the most widely used normalization tools for data analysis in genomics. Although it was originally developed for gene expression microarrays it is now used across many different high-throughput applications including RNAseq and ChIPseq. The methodology relies on the assumption that observed changes in the empirical distribution of samples are due

to unwanted variability. Because the data is transformed to remove these differences it has the potential to remove interesting biologically driven global variation. Therefore, applying quantile normalization, or other global normalization methods that rely on similar assumptions, may not be an appropriate depending on the type and source of variation.

This function can be used to test a priori to the data analysis whether global normalization methods such as quantile normalization should be applied. The `quantro` function uses the raw unprocessed high-throughput data to test for global differences in the distributions across a set of groups.

The `quantro` function will perform two tests:

1. An ANOVA to test if the medians of the distributions are different across groups. Differences across groups could be attributed to unwanted technical variation (such as batch effects) or real global biological variation. This is a helpful step for the user to verify if there is some unaccounted technical variation.

2. A test for global differences between the distributions across groups. The main output is a test statistic called `quantroStat`. This test statistic is a ratio of two variances and is similar to the idea of ANOVA. The main idea of the test is to compare the variability of distributions within the groups to the variability of distributions between the groups. If the variance between the groups is sufficiently larger than the variance within the groups, quantile normalization may not be an appropriate normalization technique depending on the source of variation (technical or biological variation). As a default, we perform this test on after a median normalization, but this option may be changed.

To assess the statistical significance of `quantroStat`, we use permutation testing. To perform a permutation test, set `B` to the number of permutations which will create a null distribution. If the number of samples is large, this number can be a large number such as 1000. This step can be very slow, but a parallelization has been implemented throught the `foreach` package. Register the number of cores using the `doParallel` package.

See the vignette for more details.

Value

A `quantro` S4 class object

`summary` Returns a list of three elements related to a summary of the experiment: (1) the number of groups (nGroups), (2) total number of samples (nTotSamples), (3) number of samples in each group (nSamplesinGroups).

`B` Number of permutations for permutation testing.

`anova` ANOVA to test if the medians of the distributions (averaged across groups) are different across groups.

`quantroStat` A test statistic which is a ratio of the mean squared error between groups of distributions to the mean squared error within groups of distributions (psuedo F-statistic).

`quantroStatPerm`
If `B` is not equal to 0, then a permutation test was performed to assess the statistical significance of `quantroStat`. These are the test statistics resulting from the permuted samples.

`quantroPvalPerm`
If `B` is not equal to 0, then this is the p-value associated with the proportion of times the test statistics resulting from the permuted samples were larger than `quantroStat`.

Examples

```
library(minfi)
data(flowSorted)
p <- getBeta(flowSorted, offset = 100)
pd <- pData(flowSorted)

qtest <- quantro(object = p, groupFactor = pd$CellType)
```

---

summary *Accessors for the 'summary' slot of a quantro object.*

Description

Accessors for the 'summary' slot of a quantro object.

Usage

```
summary(object, ...)

## S4 method for signature 'quantro'
summary(object)

## S4 method for signature 'quantro'
summary(object)
```

Arguments

object	a quantro object
...	other

Examples

```
library(minfi)
data(flowSorted)
pd <- pData(flowSorted)
qtest <- quantro(flowSorted, groupFactor = pd$CellType)
summary(qtest)
```

Index

anova, 2
anova,quantro-method *(anova)*, 2
anova.quantro *(anova)*, 2

flowSorted, 3

matboxplot, 3
matdensity, 4
MSbetween, 5
MSbetween,quantro-method *(MSbetween)*, 5
MSbetween.quantro *(MSbetween)*, 5
MSwithin, 5
MSwithin,quantro-method *(MSwithin)*, 5
MSwithin.quantro *(MSwithin)*, 5

quantro *(show)*, 9
quantro-class, 6
quantroPlot, 6
quantroPvalPerm, 7
quantroPvalPerm,quantro-method
(quantroPvalPerm), 7
quantroPvalPerm.quantro
(quantroPvalPerm), 7
quantroStat, 8
quantroStat,quantro-method
(quantroStat), 8
quantroStat.quantro *(quantroStat)*, 8
quantroStatPerm, 8
quantroStatPerm,quantro-method
(quantroStatPerm), 8
quantroStatPerm.quantro
(quantroStatPerm), 8

show, 9
summary, 11
summary,quantro-method *(summary)*, 11
summary.quantro *(summary)*, 11