

# Package ‘ABSSeq’

January 22, 2025

**Type** Package

**Title** ABSSeq: a new RNA-Seq analysis method based on modelling absolute expression differences

**Version** 1.61.0

**Author** Wentao Yang

**Maintainer** Wentao Yang <[wyang@zoologie.uni-kiel.de](mailto:wyang@zoologie.uni-kiel.de)>

**Description** Inferring differential expression genes by absolute counts difference between two groups, utilizing Negative binomial distribution and moderating fold-change according to heterogeneity of dispersion across expression level.

**License** GPL (>= 3)

**biocViews** DifferentialExpression

**Imports** locfit, limma

**Depends** R (>= 2.10), methods

**Suggests** edgeR

**RoxygenNote** 6.0.1

**git\_url** <https://git.bioconductor.org/packages/ABSSeq>

**git\_branch** devel

**git\_last\_commit** 15bbde7

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2025-01-22

## Contents

ABSDataSet . . . . .	2
ABSSeq . . . . .	3
ABSSeqlm . . . . .	4
aFoldcomplexDesign . . . . .	6
callDEs . . . . .	7

callParameter	8
callParameterwithoutReplicates	9
counts	10
estimateSizeFactorsForMatrix	11
excunts	12
genAFold	12
groups	13
LevelstoNormFC	14
maxRates	15
minimalDispersion	16
minRates	17
normalFactors	17
normMethod	18
paired	19
plotDiffToBase	20
qtotalNormalized	21
ReplaceOutliersByMAD	22
results	23
sFactors	24
simuN5	25

## Index 26

---

ABSDataSet	<i>ABSDataSet object and constructors</i>
------------	---

---

### Description

ABSDataSet object and constructors

### Usage

```
ABSDataSet(counts, groups, normMethod = c("user", "qtotal", "total", "quartile",
"geometric", "TMM"), sizeFactor = 0, paired = FALSE, minDispersion = NULL, minRates = 0.1,
maxRates = 0.3, LevelstoNormFC = 100)
```

### Arguments

counts	a matrix or table with at least two columns and one row,
groups	a factor with two groups, whose length should be equal with sample size
normMethod	method for estimating the size factors, should be one of 'user', 'qtotal', 'total', 'quartile', 'geometric' and 'TMM'. See <a href="#">normalFactors</a> for description.
sizeFactor	size factors for 'user' method, self-defined size factors by user.
paired	switch for differential expression detection in paired samples.
minDispersion	a positive double for user-defined penalty of dispersion estimation
minRates	low bounder rate of baseline estimation for counts difference, default is 0.1

maxRates	up bounder rate of baseline estimation for counts difference, default is 0.3. Setting minRates equal with maxRates will result in a testing on user-define rate,
LevelstoNormFC	maximal level of average standard deviation in fold-change normalization according to expression level, default is 100.

### Details

The function constructs an `ABSDataSet` object with counts table and groups. It also checks the structure of counts and groups. The `ABSDataSet` is a class, used to store the input values, intermediate calculations and results of an analysis of differential expression. It also contains information for the running time of an analysis.

### Value

An `ABSDataSet` object.

### Examples

```
counts <- matrix(1:4, ncol=2)
groups <- factor(c("a", "b"))
obj <- ABSDataSet(counts, groups)
obj <- ABSDataSet(counts, groups, paired=TRUE)
```

---

ABSSeq

*Differential expression analysis based on the total counts difference.*

---

### Description

This function performs a default analysis by calling, in order, the functions: `normalFactors`, `callParameter`, `callDEs`.

### Usage

```
ABSSeq(object, adjmethod = "BH", replaceOutliers = TRUE, useaFold = FALSE,
       quiet = FALSE, ...)
```

### Arguments

object	an <code>ABSDataSet</code> object, contains the reads count matrix, groups and normalization method.
adjmethod	default is 'BH', method for p-value adjusted, see <code>p.adjust.methods</code> for details
replaceOutliers	default is TRUE, switch for outlier replacement.
useaFold	default is FALSE, switch for DE detection through fold-change, see <code>callDEs</code> for details
quiet	default is FALSE, whether to print messages at each step
...	parameters passed to <code>ReplaceOutliersByMAD</code> and <code>genAFold</code> from <code>callParameter</code>

**Details**

The differential expression analysis models the total counts difference by a Negative binomial distribution

$$NB(\mu, r)$$

:

**Value**

an ABSDataSet object with additional elements, which can be retrieved by `results`: Amean and Bmean, mean of log2 normalized reads count for group A and B, foldChange, shrinked (expression level and gene-specific) log2 of fold-change, B - A, rawFC, raw log2 of fold-change, B-A (without shrinkage), lowFC, expression level corrected log2 fold-change, pvalue, pvalue from NB distribution model, adj.pvalue, adjusted p-value used p.adjust method.

**Author(s)**

Wentao Yang

**References**

Wentao Yang, Philip Rosenstiel & Hinrich Schulenburg: ABSSeq: a new RNA-Seq analysis method based on modelling absolute expression differences

**Examples**

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
obj <- ABSSeq(obj)
res <- results(obj,c("Amean", "Bmean", "foldChange", "pvalue", "adj.pvalue"))
head(res)
```

---

ABSSeq1m

*Differential expression analysis for complex design.*

---

**Description**

This function performs a default analysis by calling, in order, the functions: `normalFactors`, `aFoldcomplexDesign`,

**Usage**

```
ABSSeq1m(object, design, condA, condB = NULL, lmodel = TRUE,
preval = 0.05, qforkappa = 0, adjmethod = "BH", scale = FALSE,
quiet = FALSE, ...)
```

**Arguments**

object	a <a href="#">ABSDataSet</a> object (not need 'groups' information).
design	a numeric matrix for experiment, with samples and factors in rows and columns, respectively. Design represents the saturated model.
condA	a vector of factors for DE analysis, which could be redundant, see <a href="#">aFoldcomplexDesign</a> .
condB	a vector of factors for DE analysis, which could be redundant, default is null, if not provide, the DE analysis will switch to assess difference across factors in condA (analysis of variance). If provide, DE analysis will focus on contrast between condB and condA (condB-condA). See <a href="#">aFoldcomplexDesign</a> . The unique factors in condA+condB represents the reduced model.
lmodel	switch of fit linear model from limma-lmFit under design, default is TRUE. If TRUE, a gene-specific residual variance will be estimated from (saturated model - reduced model). Saturated model includes all factors in design matrix and reduced model includes factors in condA+condB. if saturated model == reduced model, the DE analysis performs pairwise comparison or one-way analysis of variance. See <a href="#">aFoldcomplexDesign</a> .
preval	parameter for <a href="#">aFoldcomplexDesign</a> , prior value for controlling of variance scale in case over-scaled, default is 0.05,
qforkappa	parameter for <a href="#">aFoldcomplexDesign</a> , quantile for estimating kappa(>=qforkappa), default is 0 (no trimming of data).
adjmethod	default is 'BH', method for p-value adjusted, see <a href="#">p.adjust.methods</a> for details
scale	switch for scaling fold change according to common SD under log2 transformation, default is FALSE.
quiet	default is FALSE, whether to print messages at each step
...	parameters passed to lmFit in limma

**Details**

This function uses a linear model (limma-lmFit) to infer DE under complex design.

**Value**

a result table with additional elements, including: basemean, log of basemean, foldChange, shrinked (expression level and gene-specific) log2 of fold-change, B - A, or (SDs under log2 for analysis of variance) pvalue, pvalue from NB distribution model, p.adj, adjusted p-value used p.adjust method. scaledlogFC, scaled logFC if scale=TRUE.

**Author(s)**

Wentao Yang

**References**

Wentao Yang, Philip Rosenstiel & Hinrich Schulenburg: ABSSeq: a new RNA-Seq analysis method based on modelling absolute expression differences

**Examples**

```

data(simuN5)
groups=factor(simuN5$groups)
obj <- ABSDataSet(counts=simuN5$counts)
design <- model.matrix(~0+groups)
res <- ABSSeqLm(obj,design,condA=c("groups0"),condB=c("groups1"))
head(res)

```

---

aFoldcomplexDesign	<i>Calculate parameters for differential expression test base on absolute counts differences</i>
--------------------	--

---

**Description**

Calculate aFold for each gene and general sd

**Usage**

```

aFoldcomplexDesign(nncounts, design, condA, condB = NULL, lmodel = TRUE,
  preval = 0.05, qforkappa = 0, priorgenesd, ...)

```

**Arguments**

nncounts	matrix for read count.
design	a numeric matrix for experiment, with samples and factors in rows and columns, respectively.
condA	a vector of factors for DE analysis, which could be redundant.
condB	a vector of factors for DE analysis, which could be redundant, default is null. If not provide, the DE analysis will switch to assess difference across factors in condA (analysis of variance). If provide, DE analysis will focus on contrast between condB and condA (condB-condA).
lmodel	switch of fit linear model from limma-lmFit under design, default is TRUE. If TRUE, a gene-specific residual variance will be estimated from (saturated model - reduced model). Saturated model includes all factors in design matrix and reduced model includes factors in condA+condB.
preval	pre-defined scale control for variance normalization, default is 0.05, a large value generally increases the fold-changes (decreases penalty of variances) under low expression.
qforkappa	quantile for estimating kappa( $\geq$ qforkappa), default is 0 (without trimming of data). Please set up a value in [0,1) if you want to trim the low expressed data.
priorgenesd	prior value for general SD of fold change, if provided, the estimation of general SD will be replaced by this value.
...	parameters passed to lmFit in limma

**Details**

shifted and calculate a set of parameters from normalized counts table

**Value**

A list with log2 foldchange, general SD (gene-specific SD if lmodel is TRUE) for calculating pvalue, variance stablized counts and basemean

**Note**

This function should run after [normalFactors](#).

**Examples**

```
data(simuN5)
groups=factor(simuN5$groups)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
mtx <- counts(obj,TRUE)
design <- model.matrix(~0+groups)
aFold <- aFoldcomplexDesign(mtx,design,condA=c("groups0"),condB=c("groups1"))
hist(aFold[[1]])
```

---

 callDEs

---

*Testing the differential expression by counts difference*


---

**Description**

Using NB distribution to calculate p-value for each gene as well as adjust p-value

**Usage**

```
callDEs(object, adjmethod = "BH", useaFold = FALSE)
```

**Arguments**

object	an <a href="#">ABSDataSet</a> object.
adjmethod	the method for adjusting p-value, default is 'BH'. For details, see <a href="#">p.adjust.methods</a> .
useaFold	switch for DE detection through fold-change, which will use a normal distribution (N(0,sd)) to test the significance of log2 fold-change. The sd is estimated through a quantile function of gamma distribution at <a href="#">callParameter</a> .

**Details**

This function firstly calls p-value used [pnbinom](#) to call pvalue based on sum of counts difference between two groups or used [pnorm](#) to call pvalue via log2 fold-change, then adjusts the pvalues via [p.adjust](#) method. In addition, it also shrink the log2 fold-change towards a common dispersion after pvalue calling.

**Value**

an [ABSDataSet](#) object with additional elements: shrunked log2 fold-change, pvalue and adjusted p-value, denoted by foldChange pvalue and adj-pvalue, respectively. Use the [results](#) method to get access it.

**Note**

this function should run after [callParameter](#)

**Examples**

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
obj <- normalFactors(obj)
obj <- callParameter(obj)
obj <- callDEs(obj)
head(results(obj))
```

---

callParameter	<i>Calculate parameters for differential expression test base on absolute counts differences</i>
---------------	--

---

**Description**

Calculate parameters for each gene (the moderating basemean, dispersions, moderated fold-change and general sd)

**Usage**

```
callParameter(object, replaceOutliers = TRUE, ...)
```

**Arguments**

object	a <a href="#">ABSDataSet</a> object.
replaceOutliers	switch for outlier replacement, default is TRUE.
...	parameters past to <a href="#">ReplaceOutliersByMAD</a>

**Details**

shifted and calculate a set of parameters from normalized counts table before [callDEs](#)

**Value**

A [ABSDataSet](#) object with absolute differences, basemean, mean of each group, variance, log2 of foldchange, named as 'absD', 'baseMean', 'Amean', 'Bmean', 'Variance' and 'foldChange', respectively. Use the [results](#) to get access it and [plotDiffToBase](#) to plot it.



**Note**

This function should run after [normalFactors](#) or providing size factors.

**Examples**

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
obj <- normalFactors(obj)
obj <- callParameter(obj)
head(results(obj,c("foldChange","absD","baseMean")))
plotDiffToBase(obj)
```

---

callParameterwithoutReplicates

*Calculate parameters for differential expression test base on absolute counts differences without replicates*

---

**Description**

Calculate parameters for each gene (the moderating basemean and dispersions), without replicates

**Usage**

```
callParameterwithoutReplicates(object)
```

**Arguments**

object            a [ABSDataSet](#) object.

**Details**

buliding a pseudo group to esitmate parameter by mean difference. shifted and calculate a set of parameters from normalized counts table before [callDEs](#)

**Value**

A [ABSDataSet](#) object with absolute differences, basemean, mean of each group, variance, log2 of foldchange, named as 'absD', 'baseMean', 'Amean', 'Bmean', 'Variance' and 'foldChange', respectively. Use the [results](#) to get access it

**Note**

This function should run after [normalFactors](#) or providing size factors. This function firstly constructs an expression level depended fold-change cutoffs and then separate the data into two groups. The group with fold-change less than cutoffs is used to training the dispersion. However, the cutoff might be too small when applied on data set without or with less DEs. To avoid it, we set a prior value (0.5) to it.

**Examples**

```

data(simuN5)
obj <- ABSDataSet(counts=(simuN5$counts)[,c(1,2)], groups=factor(c(1,2)))
obj <- normalFactors(obj)
obj <- callParameterwithoutReplicates(obj)
obj <- callDEs(obj)
head(results(obj))

```

---

counts

*Accessors for the 'counts' slot of a ABSDataSet object.*


---

**Description**

Accessors for the 'counts' slot of a ABSDataSet object, return a matrix

**Usage**

```

## S4 method for signature 'ABSDataSet'
counts(object,norm=FALSE)

## S4 replacement method for signature 'ABSDataSet,matrix'
counts(object)<-value

```

**Arguments**

object	a ABSDataSet object.
norm	logical indicating whether or not to normalize the counts before returning
value	an numeric matrix

**Details**

The counts slot holds the count data as a matrix of non-negative integer count values, rows and columns for genes and samples, respectively.

**See Also**

[sFactors](#), [normalFactors](#)

**Examples**

```

data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
head(counts(obj))
counts(obj) <- matrix(1:50,nrow=5,ncol=10)
head(counts(obj))

```

---

`estimateSizeFactorsForMatrix`*Low-level function to estimate size factors with robust regression.*

---

**Description**

This function is borrowed from DESeq.

**Usage**

```
estimateSizeFactorsForMatrix(counts, locfunc = median)
```

**Arguments**

<code>counts</code>	a matrix or data frame of counts, i.e., non-negative integer values
<code>locfunc</code>	a function to compute a location for a sample. By default, the median is used.

**Details**

Given a matrix or data frame of count data, this function estimates the size factors as follows: Each column is divided by the geometric means of the rows. The median (or, if requested, another location estimator) of these ratios (skipping the genes with a geometric mean of zero) is used as the size factor for this column. Typically, you will not call this function directly.

**Value**

a vector with the estimates size factors, one element per column

**Author(s)**

Simon Anders

**References**

Simon Anders, Wolfgang Huber: Differential expression analysis for sequence count data. *Genome Biology* 11 (2010) R106, <http://dx.doi.org/10.1186/gb-2010-11-10-r106>

**Examples**

```
data(simuN5)
dat <- simuN5
estimateSizeFactorsForMatrix(dat$counts)
```

---

excounnts	<i>Accessors for the 'excounnts' slot of a ABSDataSet object.</i>
-----------	---

---

**Description**

Accessors for the 'excounnts' slot of a ABSDataSet object, return a matrix

**Usage**

```
## S4 replacement method for signature 'ABSDataSet,matrix'
excounnts(object)<-value
```

**Arguments**

object	a ABSDataSet object.
value	an numeric matrix

**Details**

The excounnts slot holds the nomarlized (trimmed or not) count data as a matrix of non-negative integer count values, rows and columns for genes and samples, respectively.

**See Also**

[ABSDataSet](#), [ReplaceOutliersByMAD](#)

**Examples**

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
obj <- normalFactors(obj)
obj <- ReplaceOutliersByMAD(obj)
head(excounnts(obj))
```

---

genAFold	<i>Calculate parameters for differential expression test base on absolute counts differences</i>
----------	--

---

**Description**

Calculate aFold for each gene and general sd

**Usage**

```
genAFold(nncounts, cond, preval = 0.05, qforkappa = 0, pair = FALSE,
priorgenesd)
```

**Arguments**

nncounts	matrix for read count.
cond	factor for conditions. If provide only one condition, fold-change estimation will be suppressed.
preval	pre-defined scale control for variance normalization, default is 0.05, a large value generally increases the fold-changes (decreases penalty of variances) under low expression.
qforkappa	quantile for estimating kappa( $\geq$ qforkappa), default is 0 (without trimming of data). Please set up a value in [0,1) if you want to trim the low expressed data.
pair	switch for paired samples, default is false
priorgenesd	prior value for general SD of fold change, if provided, the estimation of general SD will be replaced by this value.

**Details**

shifted and calculate a set of parameters from normalized counts table before [callDEs](#)

**Value**

A list with log2 foldchange, general SD for calculating pvalue, variance stabilized counts and expression level adjusted counts (used for PCA analysis)

**Note**

This function should run after [normalFactors](#).

**Examples**

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
mtx <- counts(obj, TRUE)
aFold <- genAFold(mtx, factor(simuN5$groups))
hist(aFold[[1]])
```

---

groups

*Accessors for the 'groups' slot of a ABSDataSet object.*

---

**Description**

Accessor functions for the 'groups' information in a ABSDataSet object.

**Usage**

```
## S4 method for signature 'ABSDataSet'
groups(object)

## S4 replacement method for signature 'ABSDataSet,factor'
groups(object)<-value
```

**Arguments**

object            an ABSDataSet object.  
value            a factor object, includes two groups, equal with the number of samples

**Details**

The 'groups' is a factor object, contains the experiment design for differential expression analysis. Its length should be equal with the sample size.

**Examples**

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
groups(obj)
groups(obj) <- factor(rep(c("A","B"),c(5,5)))
groups(obj)
```

---

LevelstoNormFC

*Accessors for the 'LevelstoNormFC' slot of a ABSDataSet object.*


---

**Description**

Accessor functions for the 'LevelstoNormFC' slot of a ABSDataSet object.

**Usage**

```
## S4 method for signature 'ABSDataSet'
LevelstoNormFC(object)

## S4 replacement method for signature 'ABSDataSet,numeric'
LevelstoNormFC(object)<-value
```

**Arguments**

object            an ABSDataSet object.  
value            a positive numeric object

**Details**

The 'LevelstoNormFC' is maximal level of average standard deviation in fold-change normalization according to expression level.

**See Also**

[ABSDataSet](#), [callParameter](#)

**Examples**

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
LevelstoNormFC(obj)
LevelstoNormFC(obj) <- 200
LevelstoNormFC(obj)
```

---

maxRates

*Accessors for the 'maxRates' slot of a ABSDataSet object.*

---

**Description**

Accessor functions for the 'maxRates' slot of a ABSDataSet object.

**Usage**

```
## S4 method for signature 'ABSDataSet'
maxRates(object)

## S4 replacement method for signature 'ABSDataSet,numeric'
maxRates(object)<-value
```

**Arguments**

object            an ABSDataSet object.  
value            a positive numeric object

**Details**

The 'maxRates' is the upper bound of rate for baseline of counts difference estimation.

**See Also**

[callParameter](#), [ABSDataSet](#)

**Examples**

```

data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
maxRates(obj)
maxRates(obj) <- 0.4
maxRates(obj)

```

---

minimalDispersion      *Accessors for the 'minDispersion' slot of a ABSDataSet object.*

---

**Description**

Accessor functions for the 'minDispersion' slot of a ABSDataSet object.

**Usage**

```

## S4 method for signature 'ABSDataSet'
minimalDispersion(object)

## S4 replacement method for signature 'ABSDataSet,numeric'
minimalDispersion(object)<-value

```

**Arguments**

object	an ABSDataSet object.
value	a positive numeric object

**Details**

The 'minimalDispersion' is the penalty of dispersion estimation. User can set the penalty of dispersion by this function

**See Also**

[callParameter,ABSDataSet](#)

**Examples**

```

data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
minimalDispersion(obj)
minimalDispersion(obj) <- 0.2
minimalDispersion(obj)

```



---

minRates	<i>Accessors for the 'minRates' slot of a ABSDataSet object.</i>
----------	--

---

**Description**

Accessor functions for the 'minRates' slot of a ABSDataSet object.

**Usage**

```
## S4 method for signature 'ABSDataSet'  
minRates(object)  
  
## S4 replacement method for signature 'ABSDataSet,numeric'  
minRates(object)<-value
```

**Arguments**

object	an ABSDataSet object.
value	a positive numeric object

**Details**

The 'minRates' is the lower bound of rate for baseline of counts difference estimation.

**See Also**

[callParameter,ABSDataSet](#)

**Examples**

```
data(simuN5)  
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))  
minRates(obj)  
minRates(obj) <- 0.3  
minRates(obj)
```

---

normalFactors	<i>Estimating size factors from the reads count table</i>
---------------	---

---

**Description**

Function for estimating size factors

**Usage**

```
normalFactors(object)
```

**Arguments**

`object` a ABSSeq object with element of 'counts' and 'normMethod', see the constructor functions [ABSDataSet](#).

**Details**

Given a matrix of count data, this function estimates the size factors by selected method. It also provides four different methods for normalizing according to user-defined size factors, total reads, up quantile (75

**Value**

a ABSDataSet object with the estimates size factors, one element per column. Use the [sFactors](#) to show it.

**Examples**

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
obj <- normalFactors(obj)
sFactors(obj)
```

---

normMethod

*Accessors for the 'normMethod' slot of a ABSDataSet object.*


---

**Description**

Accessor functions for the 'normMethod' information in a ABSDataSet object.

**Usage**

```
## S4 method for signature 'ABSDataSet'
normMethod(object)

## S4 replacement method for signature 'ABSDataSet,character'
normMethod(object)<-value
```

**Arguments**

`object` an ABSDataSet object.

`value` a character object, should be one of 'user', 'qtoatl', 'total', 'quartile' and 'geometric'. See [normalFactors](#)

**Details**

The 'normMethod' is the method for calculating the size factors. Currently, Four methods: 'user', 'qtoatl', 'total', 'quartile' and 'DESeq' are available.

**Examples**

```

data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
normMethod(obj)
normMethod(obj) <- "geometric"
normMethod(obj)

```

---

paired	<i>Accessors for the 'paired' slot of a ABSDataSet object.</i>
--------	--

---

**Description**

Accessors for the 'paired' slot of a ABSDataSet object, return a logical value

**Usage**

```

## S4 method for signature 'ABSDataSet'
paired(object)

## S4 replacement method for signature 'ABSDataSet,logical'
paired(object)<-value

```

**Arguments**

object	a ABSDataSet object.
value	value a boolean object, should be either TRUE or FALSE.

**Details**

The 'paired' is the switch for differential expression detection among paired samples, with a boolean value: TRUE or FALSE (default). When "paired" is TRUE, the replicates in each group should be equal.

**Examples**

```

data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
paired(obj)
paired(obj) <- TRUE
paired(obj)

```

---

plotDiffToBase	<i>Plot absolute log2 fold-change against base mean of expression</i>
----------------	---

---

### Description

Plot absolute differences against expression levels

### Usage

```
plotDiffToBase(object, foldname = "foldChange", adj.pcut = 0.05,  
  cols = c("black", "red"), pch = 16, xlab = "log2 of Expression level",  
  ylab = "log2 fold-change", ...)
```

### Arguments

object	a ABSDataSet
foldname	indicates kind of fold-change in plotting, default is 'foldChange', see results
adj.pcut	cutoff for differential expressed genes, marked by different color, default is 0.05
cols	the colors to mark the non-DE and DE genes, default is black and red, respectively
pch	pch, default is 16
xlab	xlab, default is 'log2 of Expression level'
ylab	ylab, default is 'log2 fold-change'
...	further arguments to plot

### Details

Plot absolute differences against expression levels and mark the gene with a color at a given cutoff of fold-change

### Examples

```
data(simuN5)  
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))  
obj <- ABSSeq(obj)  
plotDiffToBase(obj)
```

---

qtotalNormalized      *Estimating size factors from the reads count table via ranking*

---

## Description

Function of qtotal for estimating size factors

## Usage

```
qtotalNormalized(ma, qper = 0.95, qst = 0.1, qend = 0.95, qstep = 0.01,  
qbound = 0.05, mcut = 4, qc1 = 1.5)
```

## Arguments

ma	a count matrix
qper	quantile for assessing dispersion of data, default is 0.95, which serves to avoid outliers, should in (0,1]
qst	start of quantile for estimating cv ratio, should be in [0,1], default is 0.1
qend	end of quantile for estimating cv ratio, should be in [qbound,1-qbound], default is .95
qstep	step of quantile for estimating cv ratio (sliding window), should be in (0,1], default is 0.01
qbound	window size for estimating cv and shifted size factor, default is 0.05, a smaller window size is suitable if number of genes is large
mcut	cutoff of mean from sliding window to avoid abnormal cv, should $\geq 0$ , default is 4
qc1	scale for outlier detection, should $\geq 0$ , default is 1.5

## Details

Given a matrix of count data, this function estimates the size factors by qtotal method, which is based on assessing DE (CV) and ranking. The CV is estimated via sliding window.

## Value

a vector with the estimates size factors, one element per column

## Examples

```
data(simuN5)  
counts <- simuN5$counts  
qtotalNormalized(counts)
```

---

ReplaceOutliersByMAD *Replacing outliers by moderated MAD*

---

### Description

Function for replacing the outliers by MAD

### Usage

```
ReplaceOutliersByMAD(object, replaceOutlier = TRUE, cutoff = 2,
  baseMean = 100, limitMad = 0.707, spriors = 2, Caseon = TRUE, ...)
```

### Arguments

object	a ABSSeq object with element of 'counts' and 'normMethod', see the constructor functions <a href="#">ABSDataSet</a> .
replaceOutlier	switch for replacing, default is TRUE.
cutoff	cutoff of moderating MAD for outliers, default is 2
baseMean	parameter for limiting the trimming at low expression level by baseMean/(sample size), default is 100.
limitMad	the minimal prior for moderating MAD, default is set to 0.707, which is usually the highest standard deviation at expression level of 1
spriors	prior weight size for prior MAD, default is 2
Caseon	switch for dealing with outlier trimming at sample size of 2
...	reserved parameters

### Details

Given a matrix of count data, this function replacing the outliers by MAD. Noticeably, this function also provides part of parameters for DEs calling. It is called by [callParameter](#)

### Value

a ABSDataSet object with normalized counts after trimming (replaceOutlier=TRUE) or not (replaceOutlier=FALSE). Use the [excunts](#) to show it. Use [results](#) with name 'trimmed' to view the trimming status.

### Examples

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
obj <- normalFactors(obj)
obj <- ReplaceOutliersByMAD(obj)
head(excunts(obj))
head(results(obj,c("trimmed")))
```

---

results *Accessor functions for the result from a ABSDataSet*

---

## Description

Accessor functions for the result from a ABSDataSet by given names

## Usage

```
## S4 method for signature 'ABSDataSet'
results(object, cnames = c("Amean", "Bmean",
  "baseMean", "absD", "Variance", "rawFC", "lowFC", "foldChange", "pvalue",
  "adj.pvalue", "trimmed"))
```

## Arguments

object	a ABSDataSet
cnames	a vector of names for output, which are among: 'Amean', 'Bmean', log2 of mean counts for group A and B, "baseMean", estimated mean for absolute counts difference (absD), used for mu in <a href="#">pnbinom</a> 'absD', absolute counts difference in total 'Variance', pooled Variance for two groups 'rawFC', 'lowFC', 'foldChange', log2 fold-change of original (Bmean-Amean), corrected by expression level and corrected by both expression level and gene-specific dispersion 'pvalue', 'adj.pvalue', pvalue and adjusted pvalue 'trimmed', number of trimmed outliers

## Details

This function returns the result of ABSSeq as a table or a vector depended on the given names, see [ABSSeq](#)

## Value

a table according to canmes.

## See Also

[ABSSeq](#)

## Examples

```
data(simuN5)
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
obj <- normalFactors(obj)
obj <- callParameter(obj)
obj <- callDEs(obj)
head(results(obj))
```

---

sFactors	<i>Accessors for the 'sizeFactor' slot of a ABSDataSet object.</i>
----------	--

---

## Description

Accessor functions for the 'sizeFactor' slot of a ABSDataSet object.

## Usage

```
## S4 method for signature 'ABSDataSet'  
sFactors(object)  
  
## S4 replacement method for signature 'ABSDataSet,numeric'  
sFactors(object)<-value
```

## Arguments

object	an ABSDataSet object.
value	a numeric object, one for each sample

## Details

The sizeFactors vector assigns to each sample a value, used to normalize the counts in each sample according to selected normMethod.

## See Also

[normalFactors](#)

## Examples

```
data(simuN5)  
obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))  
obj <- normalFactors(obj)  
sFactors(obj)  
sFactors(obj) <- runif(10,1,2)  
sFactors(obj)
```



---

`simuN5`*Simulated study with random outliers*

---

**Description**

Simulated study with random outliers, include five samples for two groups. It contains counts table, groups and defined differential expression genes.

**Usage**

```
data(simuN5)
```

**Format**

The format is: List of 3

\$ counts: integer, reads count matrix

\$ groups: two groups

\$ DEs : differential expression genes

**Details**

Multiple each gene with a value from 5-10 by chance at pvalue of 0.05.

**Source**

<http://bcf.isb-sib.ch/data/compcodeR/>

**References**

Soneson C, Delorenzi M: A comparison of methods for differential expression analysis of RNA-seq data. BMC Bioinformatics 2013, 14(1):91.

**Examples**

```
data(simuN5)
```

# Index

## \* datasets

- simuN5, 25
- [[<- , SumInfo, character, missing-method (ABSDataset), 2
- ABSDataset, 2, 3, 5, 7–9, 12, 15–18, 22
- ABSDataset-class (ABSDataset), 2
- ABSSeq, 3, 23
- ABSSeq1m, 4
- aFoldcomplexDesign, 4, 5, 6
- callDEs, 3, 7, 8, 9, 13
- callParameter, 3, 7, 8, 8, 15–17, 22
- callParameterwithoutReplicates, 9
- counts, 10
- counts, ABSDataSet-method (counts), 10
- counts<- (counts), 10
- counts<- , ABSDataSet, matrix-method (counts), 10
- estimateSizeFactorsForMatrix, 11
- excounds, 12, 22
- excounds, ABSDataSet-method (excounds), 12
- excounds<- (excounds), 12
- excounds<- , ABSDataSet, matrix-method (excounds), 12
- genAFold, 3, 12
- groups, 13
- groups, ABSDataSet-method (groups), 13
- groups<- (groups), 13
- groups<- , ABSDataSet, factor-method (groups), 13
- LevelstoNormFC, 14
- LevelstoNormFC, ABSDataSet-method (LevelstoNormFC), 14
- LevelstoNormFC<- (LevelstoNormFC), 14
- LevelstoNormFC<- , ABSDataSet, numeric-method (LevelstoNormFC), 14
- maxRates, 15
- maxRates, ABSDataSet-method (maxRates), 15
- maxRates<- (maxRates), 15
- maxRates<- , ABSDataSet, numeric-method (maxRates), 15
- minimalDispersion, 16
- minimalDispersion, ABSDataSet-method (minimalDispersion), 16
- minimalDispersion<- (minimalDispersion), 16
- minimalDispersion<- , ABSDataSet, numeric-method (minimalDispersion), 16
- minRates, 17
- minRates, ABSDataSet-method (minRates), 17
- minRates<- (minRates), 17
- minRates<- , ABSDataSet, numeric-method (minRates), 17
- normalFactors, 2–4, 7, 9, 10, 13, 17, 18, 24
- normMethod, 18
- normMethod, ABSDataSet-method (normMethod), 18
- normMethod<- (normMethod), 18
- normMethod<- , ABSDataSet, character-method (normMethod), 18
- p.adjust, 7
- p.adjust.methods, 3, 5, 7
- paired, 19
- paired, ABSDataSet-method (paired), 19
- paired<- (paired), 19
- paired<- , ABSDataSet, logical-method (paired), 19
- plotDifftoBase, 8, 20
- pnbinom, 7, 23
- pnorm, 7
- qtotalNormalized, 21

ReplaceOutliersByMAD, [3](#), [8](#), [12](#), [22](#)  
results, [4](#), [8](#), [9](#), [22](#), [23](#)  
results, ABSDataSet-method (results), [23](#)

sFactors, [10](#), [18](#), [24](#)  
sFactors, ABSDataSet-method (sFactors),  
[24](#)  
sFactors<- (sFactors), [24](#)  
sFactors<- , ABSDataSet, numeric-method  
(sFactors), [24](#)  
simuN5, [25](#)  
SumInfo-class (ABSDataSet), [2](#)