

# Package ‘SCnorm’

May 18, 2026

**Title** Normalization of single cell RNA-seq data  
**Version** 1.35.0  
**Author** Rhonda Bacher  
**Maintainer** Rhonda Bacher <rbacher@ufl.edu>  
**Description** This package implements SCnorm — a method to normalize single-cell RNA-seq data.  
**Depends** R (>= 3.4.0),  
**Imports** SingleCellExperiment, SummarizedExperiment, stats, methods, graphics, grDevices, parallel, quantreg, cluster, moments, data.table, BiocParallel, S4Vectors, ggplot2, forcats, BiocGenerics  
**License** GPL (>= 2)  
**Encoding** UTF-8  
**LazyData** true  
**RoxygenNote** 7.3.2  
**Suggests** BiocStyle, knitr, rmarkdown, devtools  
**VignetteBuilder** knitr  
**biocViews** Normalization, RNASeq, SingleCell, ImmunoOncology  
**URL** <https://github.com/rhondabacher/SCnorm>  
**BugReports** <https://github.com/rhondabacher/SCnorm/issues>  
**git\_url** <https://git.bioconductor.org/packages/SCnorm>  
**git\_branch** devel  
**git\_last\_commit** 3ceb26a  
**git\_last\_commit\_date** 2026-04-28  
**Repository** Bioconductor 3.24  
**Date/Publication** 2026-05-17

## Contents

correctWithin . . . . .	2
evaluateK . . . . .	3
ExampleSimSCData . . . . .	4

generateEvalPlot . . . . .	4
getCounts . . . . .	5
getDens . . . . .	5
getSlopes . . . . .	6
GetTD . . . . .	7
normWrapper . . . . .	7
plotCountDepth . . . . .	8
plotWithinFactor . . . . .	10
quickReg . . . . .	11
redoBox . . . . .	12
results . . . . .	12
scaleNormMultCont . . . . .	13
SCnorm . . . . .	14
SCnormFit . . . . .	16
splitGroups . . . . .	17
<b>Index</b>	<b>18</b>

---

correctWithin	<i>correctWithin</i>
---------------	----------------------

---

## Description

Perform the correction within each sample (See loess normalization in original publication Risso et al., 2011 (BMC Bioinformatics)). Similar to function in EDASeq v2.8.0.

## Usage

```
correctWithin(y, correctFactor)
```

## Arguments

`y` gene to perform the regression on.  
`correctFactor` list of data needed for the regression.

## Details

Performs within sample normalization.

## Value

within-cell normalized expression estimates

---

`evaluateK`*Evaluate normalization using K slope groups*

---

**Description**

Median quantile regression is fit for each gene using the normalized gene expression values. A slope near zero indicate the sequencing depth effect has been successfully removed. Genes are divided into ten equally sized groups based on their non-zero median expression. Slope densities are plot for each group and estimated modes are calculated. If any of the ten group modes is larger than .1, the K is not sufficient to normalize the data.

**Usage**

```
evaluateK(  
  Data,  
  SeqDepth,  
  OrigData,  
  Slopes,  
  Name,  
  Tau,  
  PrintProgressPlots,  
  ditherCounts  
)
```

**Arguments**

<code>Data</code>	matrix of normalized expression counts. Rows are genes and columns are samples.
<code>SeqDepth</code>	vector of sequencing depths estimated as columns sums of un-normalized expression matrix.
<code>OrigData</code>	matrix of un-normalized expression counts. Rows are genes and columns are samples.
<code>Slopes</code>	vector of slopes estimated in the <code>GetSlopes()</code> function. Only used here to obtain the names of genes considered in the normalization.
<code>Name</code>	plot title
<code>Tau</code>	value of quantile for the quantile regression used to estimate gene-specific slopes (default is median, $\text{Tau} = .5$ ).
<code>PrintProgressPlots</code>	whether to automatically produce plot as <code>SCnorm</code> determines the optimal number of groups (default is <code>FALSE</code> , highly suggest using <code>TRUE</code> ). Plots will be printed to the current device.
<code>ditherCounts</code>	whether to dither/jitter the counts, may be used for data with many ties, default is <code>FALSE</code> .

**Value**

value of largest mode and a plot of the ten normalized slope densities.

**Author(s)**

Rhonda Bacher

---

ExampleSimSCData      *Example datasets for SCnorm*

---

### Description

Data generated as in SIM I from the manuscript with  $K = 4$ .

### Usage

```
ExampleSimSCData
```

### Format

data matrix

### Examples

```
data(ExampleSimSCData)
```

---

generateEvalPlot      *Internal plotting function.*

---

### Description

Genes are divided into NumExpressionGroups = 10 equally sized groups based on their non-zero median expression. Slope densities are plot for each group.

### Usage

```
generateEvalPlot(
  MedExpr,
  SeqDepth,
  Slopes,
  Name,
  NumExpressionGroups = 10,
  BeforeNorm = TRUE
)
```

### Arguments

MedExpr	non-zero median expression for all genes.
SeqDepth	sequencing depth for each cell/sample.
Slopes	per gene estimates of the count-depth relationship.
Name	name for plot title.
NumExpressionGroups	the number of groups to split the data into, genes are split into equally sized groups based on their non-zero median expression.
BeforeNorm	whether dat have already been normalized.

**Value**

a plot of the un-normalized slope densities.

**Author(s)**

Rhonda Bacher

---

getCounts	<i>getCounts</i>
-----------	------------------

---

**Description**

Convenient helper function to extract the normalized expression matrix from the SummarizedExperiment

**Usage**

```
getCounts(DATA)
```

**Arguments**

DATA	An object of class SummarizedExperiment that contains single-cell expression and metadata
------	---

**Value**

A matrix which contains the count data where genes are in rows and cells are in columns

**Examples**

```
data(ExampleSimSCData)
ExampleData <- SummarizedExperiment::SummarizedExperiment(assays=list("Counts"=ExampleSimSCData))
myData <- getCounts(ExampleData)
```

---

getDens	<i>getDens</i>
---------	----------------

---

**Description**

getDens

**Usage**

```
getDens(ExprGroups, byGroup, RETURN = c("Mode", "Height"))
```

**Arguments**

ExprGroups	expression groups already split.
byGroup	factor (usually slopes) to get density based on ExprGroups.
RETURN	whether to return Mode or Height of density.

**Details**

get density of slopes in different expression groups

**Value**

list, length is equal to NumGroups

---

getSlopes

*Estimate gene specific count-depth relationships*

---

**Description**

This is the gene-specific fitting function, where a median (Tau = .5) quantile regression is fit for each gene. Only genes having at least 10 non-zero expression values are considered.

**Usage**

```
getSlopes(
  Data,
  SeqDepth = 0,
  Tau = 0.5,
  FilterCellNum = 10,
  ditherCounts = FALSE
)
```

**Arguments**

Data	matrix of un-normalized expression counts. Rows are genes and columns are samples.
SeqDepth	vector of sequencing depths estimated as columns sums of un-normalized expression matrix.
Tau	value of quantile for the quantile regression used to estimate gene-specific slopes (default is median, Tau = .5).
FilterCellNum	the number of non-zero expression estimate required to include the genes into the SCnorm fitting (default = 10). The initial
ditherCounts	whether to dither/jitter the counts, may be used for data with many ties, default is FALSE.

**Value**

vector of estimated slopes.

**Author(s)**

Rhonda Bacher

**Examples**

```
data(ExampleSimSCData)
myslopes <- getSlopes(ExampleSimSCData)
```

---

`GetTD`*Fit group regression for specific quantile and degree*

---

**Description**

This is an internal fitting of the group regression. For a single combination of possible tau and d values the group regression is first fit, then predicted values are obtained and regressed against the original sequencing depths. The estimated slope is passed back to the `SCnorm_fit()` function.

**Usage**

```
GetTD(x, InputData)
```

**Arguments**

<code>x</code>	specifies a column of the grid matrix of tau and d.
<code>InputData</code>	contains the expression values, sequencing depths to fit the group regression, and the quantile used in the individual gene regression for grouping.

**Value**

estimated count-depth relationship of predicted values for one value of tau and degree.

**Author(s)**

Rhonda Bacher

---

`normWrapper`*Iteratively fit group regression and evaluate to choose optimal K*

---

**Description**

This function iteratively normalizes using `K` groups and then evaluates whether `K` is sufficient. If the maximum mode received from the `GetK()` function is larger than `.1`, `K` is increased to `K + 1`. Uses params sent from `SCnorm`.

**Usage**

```
normWrapper(  
  Data,  
  SeqDepth = NULL,  
  Slopes = NULL,  
  CondNum = NULL,  
  PrintProgressPlots,  
  PropToUse,  
  Tau,  
  Thresh,  
  ditherCounts  
)
```

**Arguments**

Data	can be a matrix of single-cell expression with cells where rows are genes and columns are samples. Gene names should not be a column in this matrix, but should be assigned to rownames(Data). Data can also be an object of class SummarizedExperiment that contains the single-cell expression matrix and other metadata. The assays slot contains the expression matrix and is named "Counts". This matrix should have one row for each gene and one sample for each column. The colData slot should contain a data.frame with one row per sample and columns that contain metadata for each sample. This data.frame should contain a variable that represents biological condition in the same order as the columns of NormCounts). Additional information about the experiment can be contained in the metadata slot as a list.
SeqDepth	sequencing depth for each cell/sample.
Slopes	per gene estimates of the count-depth relationship.
CondNum	name of group being normalized, just for printing messages.
PrintProgressPlots	whether to automatically produce plot as SCnorm determines the optimal number of groups (default is FALSE, highly suggest using TRUE). Plots will be printed to the current device.
PropToUse	proportion of genes closest to the slope mode used for the group fitting, default is set at .25. This number #' mainly affects speed.
Tau	value of quantile for the quantile regression used to estimate gene-specific slopes (default is median, Tau = .5 ).
Thresh	threshold to use in evaluating the sufficiency of K, default is .1.
ditherCounts	whether to dither/jitter the counts, may be used for data with many ties, default is FALSE.

**Value**

matrix of normalized and scaled expression values for all conditions and the evaluation plots are output for each attempted value of K.

**Author(s)**

Rhonda Bacher

---

plotCountDepth	<i>Evaluate the count-depth relationship before (or after) normalizing the data.</i>
----------------	--

---

**Description**

Quantile regression is used to estimate the dependence of read counts on sequencing depth for every gene. If multiple conditions are provided, a separate plot is provided for each and the filters are applied within each condition separately. The plot can be used to evaluate the extent of the count-depth relationship in the dataset or can be used to evaluate data normalized by alternative methods.

**Usage**

```
plotCountDepth(
  Data,
  NormalizedData = NULL,
  Conditions = NULL,
  Tau = 0.5,
  FilterCellProportion = 0.1,
  FilterExpression = 0,
  NumExpressionGroups = 10,
  NCores = NULL,
  ditherCounts = FALSE
)
```

**Arguments**

Data	can be a matrix of single-cell expression with cells where rows are genes and columns are samples. Gene names should not be a column in this matrix, but should be assigned to <code>rownames(Data)</code> . Data can also be an object of class <code>SummarizedExperiment</code> that contains the single-cell expression matrix and other metadata. The <code>assays</code> slot contains the expression matrix and is named "Counts". This matrix should have one row for each gene and one sample for each column. The <code>colData</code> slot should contain a <code>data.frame</code> with one row per sample and columns that contain metadata for each sample. This <code>data.frame</code> should contain a variable that represents biological condition in the same order as the columns of <code>NormCounts</code> ). Additional information about the experiment can be contained in the <code>metadata</code> slot as a list.
NormalizedData	matrix of normalized expression counts. Rows are genes and columns are samples. Only input this if evaluating already normalized data.
Conditions	vector of condition labels, this should correspond to the columns of the un-normalized expression matrix. If not provided data is assumed to come from same condition/batch.
Tau	value of quantile for the quantile regression used to estimate gene-specific slopes (default is <code>Tau = .5</code> (median)).
FilterCellProportion	the proportion of non-zero expression estimates required to include the genes into the evaluation. Default is <code>.10</code> , and will not go below a proportion which uses less than 10 total cells/samples.
FilterExpression	exclude genes having median of non-zero expression below this threshold from count-depth plots (default = 0).
NumExpressionGroups	the number of groups to split the data into, genes are split into equally sized groups based on their non-zero median expression.
NCores	number of cores to use, default is <code>detectCores() - 1</code> . This will be used to set up a parallel environment using either <code>MulticoreParam</code> (Linux, Mac) or <code>SnowParam</code> (Windows) with <code>NCores</code> using the package <code>BiocParallel</code> .
ditherCounts	whether to dither/jitter the counts, may be used for data with many ties, default is <code>FALSE</code> .

**Value**

returns a data.frame containing each gene's slope (count-depth relationship) and its associated expression group. A plot will be output.

**Author(s)**

Rhonda Bacher

**Examples**

```
data(ExampleSimSCData)
Conditions = rep(c(1,2), each= 90)
#plotCountDepth(Data = ExampleSimSCData, Conditions = Conditions,
  #FilterCellProportion = .1)
```

---

plotWithinFactor      *Evaluate gene-specific factors in the the data.*

---

**Description**

This function can be used to evaluate the extent of gene-specific biases in the data. If a bias exists, the plots provided here will identify whether it affects cells equally or not. Correction for such features may be considered especially if the bias is different between conditions (see SCnorm vignette for details).

**Usage**

```
plotWithinFactor(
  Data,
  withinSample = NULL,
  Conditions = NULL,
  FilterExpression = 0,
  NumExpressionGroups = 4
)
```

**Arguments**

**Data** can be a matrix of single-cell expression with cells where rows are genes and columns are samples. Gene names should not be a column in this matrix, but should be assigned to rownames(Data). Data can also be an object of class SummarizedExperiment that contains the single-cell expression matrix and other metadata. The assays slot contains the expression matrix and is named "Counts". This matrix should have one row for each gene and one sample for each column. The colData slot should contain a data.frame with one row per sample and columns that contain metadata for each sample. This data.frame should contain a variable that represents biological condition in the same order as the columns of NormCounts). Additional information about the experiment can be contained in the metadata slot as a list.

**withinSample** a vector of gene-specific features.

Conditions        vector of condition labels, this should correspond to the columns of the un-normalized expression matrix. If provided the cells will be colored by Condition instead of individually.

FilterExpression        exclude genes having median of non-zero expression below this threshold.

NumExpressionGroups        the number of groups to split the within sample factor into, e.g genes will be split into equally sized groups based on their GC content/Gene length/etc.

**Value**

produces a plot and returns the data the plot is based on.

**Author(s)**

Rhonda Bacher

**Examples**

```
data(ExampleSimSCData)
Conditions = rep(c(1,2), each= 90)
exampleFactor = runif(dim(ExampleSimSCData)[1], 0, 1)
names(exampleFactor) = rownames(ExampleSimSCData)
#plotWithinFactor(Data = ExampleSimSCData,
  #withinSample=exampleFactor, Conditions = Conditions)
```

---

quickReg

*quickReg*

---

**Description**

Perform the single gene regressions using quantile regression.

**Usage**

```
quickReg(x, InputData)
```

**Arguments**

x                    gene to perform the regression on.

InputData            list of data needed for the regression.

**Details**

Perform the single gene regressions using quantile regression.

**Value**

gene slope.

---

redoBox	<i>redoBox</i>
---------	----------------

---

**Description**

redoBox

**Usage**

redoBox(DATA, smallc)

**Arguments**

DATA	data set to.
smallc	what value to ignore, typically is zero.

**Details**

Function to log data and turn zeros to NA to mask/ignore in functions.

**Value**

the dataset has been logged with values below smallc masked.

---

results	<i>results</i>
---------	----------------

---

**Description**

Convenient helper function to extract the results ( normalized data, list of genes filtered out, or scale factors). Results data.frames/matrices are stored in the metadata slot and can also be accessed without the help of this convenience function by calling metadata(DataNorm).

**Usage**

results(DATA, type = c("NormalizedData", "ScaleFactors", "GenesFilteredOut"))

**Arguments**

DATA	An object of class SummarizedExperiment that contains normalized single-cell expression and other metadata, and the output of the SCnorm function.
type	A character variable specifying which output is desired, with possible values "NormalizedData", "ScaleFactors", and "GenesFilteredOut". By default results() will return type="NormalizedData", which is the matrix of normalized counts from SCnorm. By specifying type="ScaleFactors" a matrix of scale factors (only returned if reportSF=TRUE when running SCnorm()) can be obtained. type="GenesFilteredOut" returns a list of genes that were not normalized using SCnorm, these are genes that did not pass the filter criteria.

**Value**

A data.frame containing output as detailed in the description of the type input parameter

**Examples**

```
data(ExampleSimSCData)
Conditions = rep(c(1), each= 90)
#NormData <- SCnorm(Data=ExampleSimSCData, Conditions=Conditions)
#normDataMatrix <- results(NormData)
```

---

scaleNormMultCont      *Scale multiple conditions*

---

**Description**

After conditions are independently normalized with the count-depth effect removed, conditions need to be additionally scaled prior to further analysis. Genes that were normalized in both conditions are split into quartiles based on their un-normalized non-zero medians. Genes in each quartile are scaled to the median fold change of condition specific gene means and overall gene means. This function can be used independently if SCnorm was run across different Conditions separately. However, the input must be as follow: NormData <- list(list(NormData = normalizedDataSet1), list(NormData = normalizedDataSet2)) where normalizedDataSet1 is the normalized matrix obtained using normcounts() on the output of SCnorm().

**Usage**

```
scaleNormMultCont(NormData, OrigData, Genes, useSpikes, useZerosToScale)
```

**Arguments**

NormData	list of matrices of normalized expression counts and scale factors for each condition. Matrix rows are genes and columns are samples.
OrigData	list of matrices of un-normalized expression counts. Matrix rows are genes and columns are samples. Each item in list is a different condition.
Genes	vector of genes that will be used to scale conditions, only want to use genes that were normalized.
useSpikes	whether to use spike-ins to perform between condition scaling (default=FALSE). Assumes spike-in names start with "ERCC-".
useZerosToScale	whether to use zeros when scaling across conditions (default=FALSE).

**Value**

matrix of normalized and scaled expression values for all conditions.

**Author(s)**

Rhonda Bacher

SCnorm

*SCnorm***Description**

Quantile regression is used to estimate the dependence of read counts on sequencing depth for every gene. Genes with similar dependence are then grouped, and a second quantile regression is used to estimate scale factors within each group. Within-group adjustment for sequencing depth is then performed using the estimated scale factors to provide normalized estimates of expression. If multiple conditions are provided, normalization is performed within condition and then normalized estimates are scaled between conditions. If `withinSample=TRUE` then the method from Risso et al. 2011 will be implemented.

**Usage**

```
SCnorm(
  Data = NULL,
  Conditions = NULL,
  PrintProgressPlots = FALSE,
  reportSF = FALSE,
  FilterCellNum = 10,
  FilterExpression = 0,
  Thresh = 0.1,
  K = NULL,
  NCores = NULL,
  ditherCounts = FALSE,
  PropToUse = 0.25,
  Tau = 0.5,
  withinSample = NULL,
  useSpikes = FALSE,
  useZerosToScale = FALSE
)
```

**Arguments**

Data	can be a matrix of single-cell expression with cells where rows are genes and columns are samples. Gene names should not be a column in this matrix, but should be assigned to <code>rownames(Data)</code> . Data can also be an object of class <code>SummarizedExperiment</code> that contains the single-cell expression matrix and other metadata. The <code>assays</code> slot contains the expression matrix and is named "Counts". This matrix should have one row for each gene and one sample for each column. The <code>colData</code> slot should contain a <code>data.frame</code> with one row per sample and columns that contain metadata for each sample. This <code>data.frame</code> should contain a variable that represents biological condition in the same order as the columns of <code>NormCounts</code> ). Additional information about the experiment can be contained in the <code>metadata</code> slot as a list.
Conditions	vector of condition labels, this should correspond to the columns of the expression matrix.
PrintProgressPlots	whether to automatically produce plot as <code>SCnorm</code> determines the optimal number of groups (default is <code>FALSE</code> , highly suggest using <code>TRUE</code> ). Plots will be printed to the current device.

reportSF	whether to provide a matrix of scaling counts in the output (default = FALSE).
FilterCellNum	the number of non-zero expression estimate required to include the genes into the SCnorm fitting (default = 10). The initial grouping fits a quantile regression to each gene, making this value too low gives unstable fits.
FilterExpression	exclude genes having median of non-zero expression from the normalization.
Thresh	threshold to use in evaluating the sufficiency of K, default is .1.
K	the number of groups for normalizing. If left unspecified, an evaluation procedure will determine the optimal value of K (recommended).
NCores	number of cores to use, default is detectCores() - 1. This will be used to set up a parallel environment using either MulticoreParam (Linux, Mac) or SnowParam (Windows) with NCores using the package BiocParallel.
ditherCounts	whether to dither/jitter the counts, may be used for data with many ties, default is FALSE.
PropToUse	proportion of genes closest to the slope mode used for the group fitting, default is set at .25. This number #' mainly affects speed.
Tau	value of quantile for the quantile regression used to estimate gene-specific slopes (default is median, Tau = .5 ).
withinSample	a vector of gene-specific features to correct counts within a sample prior to SCnorm. If NULL(default) then no correction will be performed. Examples of gene-specific features are GC content or gene length.
useSpikes	whether to use spike-ins to perform across condition scaling (default=FALSE). Spike-ins must be stored in the SingleCellExperiment object using altExp() function from SingleCellExperiment. See vignette for example.
useZerosToScale	whether to use zeros when scaling across conditions (default=FALSE).

**Value**

List containing matrix of normalized expression (and optionally a matrix of size factors if reportSF = TRUE ).

**Author(s)**

Rhonda Bacher

**Examples**

```
data(ExampleSimSCData)
Conditions = rep(c(1,2), each= 45)
#DataNorm <- SCnorm(ExampleSimSCData, Conditions,
#FilterCellNum = 10)
#str(DataNorm)
```

SCnormFit

*Fit group quantile regression for K groups***Description**

For each group  $K$ , a quantile regression is fit over all genes (PropToUse) for a grid of possible degree's  $d$  and quantile's  $\tau$ . For each value of  $\tau$  and  $d$ , the predicted expression values are obtained and regressed against the original sequencing depths. The optimal  $\tau$  and  $d$  combination is chosen as that closest to the mode of the gene slopes.

**Usage**

```
SCnormFit(Data, SeqDepth, Slopes, K, PropToUse = 0.25, Tau = 0.5, ditherCounts)
```

**Arguments**

Data	can be a matrix of single-cell expression with cells where rows are genes and columns are samples. Gene names should not be a column in this matrix, but should be assigned to rownames(Data). Data can also be an object of class SummarizedExperiment that contains the single-cell expression matrix and other metadata. The assays slot contains the expression matrix and is named "Counts". This matrix should have one row for each gene and one sample for each column. The colData slot should contain a data.frame with one row per sample and columns that contain metadata for each sample. This data.frame should contain a variable that represents biological condition in the same order as the columns of NormCounts). Additional information about the experiment can be contained in the metadata slot as a list.
SeqDepth	sequencing depth for each cell/sample.
Slopes	per gene estimates of the count-depth relationship.
K	the number of groups for normalizing. If left unspecified, an evaluation procedure will determine the optimal value of $K$ (recommended).
PropToUse	proportion of genes closest to the slope mode used for the group fitting, default is set at .25. This number #' mainly affects speed.
Tau	value of quantile for the quantile regression used to estimate gene-specific slopes (default is median, Tau = .5).
ditherCounts	whether to dither/jitter the counts, may be used for data with many ties, default is FALSE.

**Value**

normalized expression matrix and matrix of scaling factors.

**Author(s)**

Rhonda Bacher

---

`splitGroups`

*splitGroups*

---

**Description**

`splitGroups`

**Usage**

```
splitGroups(DATA, NumGroups = 10)
```

**Arguments**

DATA                vector to be splot.

NumGroups          number of groups

**Details**

helper function to get split a vector into a specified number of groups

**Value**

list, length is equal to NumGroups

# Index

## \* datasets

ExampleSimSCData, 4

correctWithin, 2

evaluateK, 3

ExampleSimSCData, 4

generateEvalPlot, 4

getCounts, 5

getDens, 5

getSlopes, 6

GetTD, 7

normWrapper, 7

plotCountDepth, 8

plotWithinFactor, 10

quickReg, 11

redoBox, 12

results, 12

scaleNormMultCont, 13

SCnorm, 14

SCnormFit, 16

splitGroups, 17